



# **UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA (UNED)**

E.T.S DE INGENIERÍA INFORMÁTICA

DEPARTAMENTO DE INGENIERÍA DE SOFTWARE Y SISTEMAS  
INFORMÁTICOS

## **MÁSTER UNIVERSITARIO DE INVESTIGACIÓN EN INGENIERÍA DE SOFTWARE Y SISTEMAS INFORMÁTICOS**

ITINERARIO EN INGENIERÍA DEL SOFTWARE

TRABAJO FIN DE MÁSTER (31105151)

### **GENERACIÓN AUTOMÁTICA DE MICROSERVICIOS (GAMS)**

AUTOR: LUIS MARÍA CÁMARA ROSSI

DIRECTORA: ELENA RUIZ LARROCHA

CURSO ACADÉMICO 2017-2018: CONVOCATORIA DE SEPTIEMBRE

**MÁSTER UNIVERSITARIO DE INVESTIGACIÓN  
EN INGENIERÍA DE SOFTWARE Y SISTEMAS  
INFORMÁTICOS**

ITINERARIO EN INGENIERÍA DEL SOFTWARE

TRABAJO FIN DE MÁSTER (31105151)

**GENERACIÓN AUTOMÁTICA DE  
MICROSERVICIOS (GAMS)**

TRABAJO ESPECÍFICO PROPUESTO POR EL ALUMNO

AUTOR: LUIS MARÍA CÁMARA ROSSI  
DIRECTORA: ELENA RUIZ LARROCHA

(Página intencionadamente en blanco: segunda hoja, reservada para calificaciones.)

## DECLARACIÓN JURADA DE AUTORÍA DEL TRABAJO CIENTÍFICO, PARA LA DEFENSA DEL TRABAJO FIN DE MÁSTER

Fecha: 11 de septiembre de 2018

Quien suscribe:

**Autor(a):** Luis María Cámara Rossi

**D.N.I.:** 32.817.310-J

Hace constar que es el autor del trabajo:

Generación Automática de MicroServicios (GAMS)

En tal sentido, manifiesto la originalidad de la conceptualización del trabajo, interpretación de datos y la elaboración de las conclusiones, dejando establecido que aquellos aportes intelectuales de otros autores se han referenciado debidamente en el texto de dicho trabajo.

### DECLARACIÓN:

- Garantizo que el trabajo que remito es un documento original y no ha sido publicado, total ni parcialmente, por otros autores en soporte papel ni en formato digital.
- Certifico que he contribuido directamente al contenido intelectual de este manuscrito, a la génesis y análisis de sus datos, por lo cual estoy en condiciones de hacerme públicamente responsable de él.
- No he incurrido en fraude científico, plagio o vicios de autoría; en caso contrario, aceptaré las medidas disciplinarias sancionadoras que correspondan.

Firmado:



Luis M<sup>a</sup> Cámara Rossi  
11/06/2018



IMPRESO TFDM05\_AUTOR  
AUTORIZACIÓN DE PUBLICACIÓN  
CON FINES ACADÉMICOS



Impreso TFDM05\_Autor. Autorización de publicación  
y difusión del TFDM para fines académicos

## Autorización

Autorizo/amos a la Universidad Nacional de Educación a Distancia a difundir y utilizar, con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la memoria de este Trabajo Fin de Máster, como el código, la documentación y/o el prototipo desarrollado.

Firma del/los Autor/es

Luis M<sup>a</sup> Cárdenas Rosal  
11/06/2018  
*[Handwritten signature]*

## RESUMEN DEL TRABAJO

El presente Trabajo Fin de Máster, titulado **Generación Automática de MicroServicios (GAMS)**, se ocupa de la automatización en el desarrollo software de servicios web que siguen la filosofía denominada de “*microservicios*”. El principal objetivo que pretende cubrir este Trabajo es la aceleración en el desarrollo de aplicaciones basadas en este tipo de arquitecturas software.

En los últimos años los sistemas intensivos en software han crecido tanto en número de sistemas como en el volumen de funcionalidades ofrecidas por cada uno de ellos. Sin embargo, las herramientas necesarias para acometer con garantías los desarrollos necesarios por dichos sistemas intensivos en software no han evolucionado al mismo ritmo.

Por ello, como parte de este Trabajo se implementará una plataforma tecnológica orientada a crear y mantener aplicaciones basadas en el paradigma arquitectónico de los microservicios, permitiendo además que dichos servicios sean ejecutados tanto en servidores como en ordenadores personales e incluso en dispositivos embebidos. Esta herramienta hará uso en todo momento de tecnologías que sean estándares *de facto* en Internet, así como de libre distribución.

Los microservicios de la solución podrán desarrollarse sin excesivos conocimientos de programación, utilizando mecanismos al alcance del usuario final, el cual a su vez podrá explotarlos gracias tanto a su interfaz gráfica de usuario (GUI) como a su interfaz entre máquinas (M2M).

## LISTA DE PALABRAS CLAVE

Arquitectura, API, automático, desarrollo, generación, IoT, línea de producto, microservicio, modelo, REST, servicio, SOA, web.

## AGRADECIMIENTOS

A cuantas personas que han formado parte de mi vida, pues de todas ellas he procurado aprender algo y, si soy ahora quien soy, es gracias a esos “pasos en la arena” que han ido marcando mis experiencias.

A mi familia, porque sin ellos no hay comienzo y, como dice el proverbio, “*el comienzo es la mitad del todo*”. Han sido y siguen siendo, sin duda, el apoyo y cariño constantes que nunca faltarán.

A mi mujer, por ejercer de impulso incansable para alcanzar cuantas metas nos vamos proponiendo juntos a lo largo de nuestro camino.

A mis amigos, sin los cuales “esto de la Informática” seguiría siendo un hobby y no la profesión que es ahora. Orgulloso me siento de haber podido estudiar y trabajar a su lado.

A mis profesores y maestros, especialmente a mi Directora, quienes con enorme tesón me enseñaron lo que sé y corrigieron mis equivocaciones. Si ahora he terminado esta Máster, es gracias a ellos.

“*Buscad primero el reino de Dios y su justicia, y todo lo demás se os dará por añadidura*” [San Mateo 6,24].



# ÍNDICE

|   |           |
|---|-----------|
| <b>1.- INTRODUCCIÓN.</b>  | <b>1</b>  |
| 1.1.- Organización de la Memoria del Trabajo.   | 1         |
| 1.2.- Problemática.   | 3         |
| 1.3.- Solución Planteada.   | 4         |
| 1.3.1.- Objetivos a cubrir.   | 5         |
| 1.3.2.- Funcionamiento general.   | 6         |
| 1.3.3.- Metodología seguida y plan de trabajo.  | 7         |
| 1.3.4.- Arquitectura tecnológica.   | 8         |
| 1.4.- Delimitación del Alcance.   | 9         |
| <b>2.- ESTADO DE LA CUESTIÓN.</b>   | <b>10</b> |
| 2.1.- Antecedentes.   | 11        |
| 2.1.1.- Necesidad de automatización en el ejercicio profesional.                                | 12        |
| 2.1.2.- Arquitecturas de Software.  | 13        |
| 2.1.2.1.- Vistas de un sistema desde la perspectiva de la arquitectura.                         | 13        |
| 2.1.2.2.- Técnicas para la selección y definición de arquitecturas.                             | 14        |
| 2.1.2.3.- Implicaciones de la arquitectura.   | 16        |
| 2.1.3.- Líneas de producto.   | 16        |
| 2.1.3.1.- Economías de alcance y economías de escala.   | 17        |
| 2.1.3.2.- Producto ejemplar.  | 18        |
| 2.1.3.3.- Avances.  | 19        |
| 2.1.4.- Ingeniería del Dominio.   | 19        |
| 2.1.4.1.- Ingeniería de Requisitos.   | 19        |
| 2.1.4.2.- Análisis del dominio orientado a características (técnica FODA).                      | 20        |
| 2.1.4.3.- Diseño dirigido por modelos.  | 23        |
| 2.1.5.- Generación automática de código.  | 25        |
| 2.1.5.1.- Desarrollo dirigido por el modelo.  | 25        |
| 2.1.6.- Arquitecturas Orientadas a Servicios.   | 26        |
| 2.2.- Entornos Tecnológicos Disponibles.  | 26        |
| 2.2.1.- PHP   | 27        |
| 2.2.2.- Java.   | 29        |
| 2.2.3.- Microsoft .NET Framework.   | 30        |
| 2.2.4.- Otras tecnologías comúnmente empleadas para el desarrollo en la <i>World Wide Web</i> . | 31        |
| 2.2.4.1.- Orígenes de las aplicaciones web.   | 31        |
| 2.2.4.2.- El lado del cliente.  | 32        |
| 2.2.4.3.- El lado del servidor.   | 33        |
| 2.2.4.4.- Mezclando nuevamente ambos lados.   | 33        |
| 2.2.4.5.- Tendencias arquitectónicas.   | 35        |
| 2.3.- Características Diferenciadoras de este Trabajo.  | 35        |
| <b>3.- PLANTEAMIENTO E HIPÓTESIS.</b>   | <b>37</b> |
| 3.1.- Objetivos a Cubrir.   | 38        |
| 3.2.- Delimitaciones del Alcance.   | 40        |
| 3.3.- Plan de Proyecto.   | 41        |
| 3.3.1.- Participantes.  | 41        |
| 3.3.2.- Etapas.   | 43        |
| 3.3.3.- Productos resultantes.  | 44        |
| <b>4.- RESOLUCIÓN.</b>  | <b>46</b> |
| 4.1.- Metodología de Desarrollo Software Empleada.  | 47        |



|   |            |
|---|------------|
| 4.2.- Análisis Funcional.   | 48         |
| 4.3.- Diseño Técnico.   | 49         |
| 4.4.- Construcción y Pruebas del Producto Software.                           | 50         |
| 4.4.1.- Entorno de desarrollo.  | 51         |
| 4.4.1.1.- MINGW64 (GNU bash 4.4.19)   | 52         |
| 4.4.1.2.- Git for Windows 2.18  | 53         |
| 4.4.1.3.- PHP 7   | 53         |
| 4.4.1.4.- HTML 5  | 53         |
| 4.4.1.5.- CSS 3   | 54         |
| 4.4.1.6.- SASS 3.5.1  | 54         |
| 4.4.1.7.- Compass 1.0.3   | 54         |
| 4.4.1.8.- RUBY 2.2.6  | 55         |
| 4.4.1.9.- JavaScript (ECMAScript 5)   | 56         |
| 4.4.1.10.- SQLite 3   | 57         |
| 4.4.1.11.- SQLite Studio 3.1.1  | 58         |
| 4.4.1.12.- MySQL 5 / MariaDB 10   | 59         |
| 4.4.1.13.- Symfony 3.4.12   | 59         |
| 4.4.2.- Bibliotecas de terceros   | 59         |
| 4.4.3.- Construcción.   | 60         |
| 4.4.4.- Pruebas realizadas.   | 61         |
| 4.5.- Implantación y Puesta en Producción.                                    | 61         |
| 4.5.1.- Instalación.  | 62         |
| 4.5.2.- Configuración inicial (parametrización).                              | 62         |
| <b>5.- LÍNEAS FUTURAS. CONCLUSIONES.</b>                                      | <b>64</b>  |
| 5.1.- Líneas Futuras.   | 64         |
| 5.2.- Conclusiones.   | 65         |
| <b>6.- BIBLIOGRAFÍA Y REFERENCIAS.</b>  | <b>67</b>  |
| <b>7.- LISTADO DE SIGLAS, ABREVIATURAS Y ACRÓNIMOS.</b>                       | <b>113</b> |
| <b>8.- ANEXO A: ANÁLISIS FUNCIONAL.</b>                                       | <b>119</b> |
| 8.1.- Requisitos educidos.  | 119        |
| 8.2.- Diagrama conceptual.  | 121        |
| 8.3.- Diagrama de clases de alto nivel.                                       | 122        |
| 8.4.- Diagrama de casos de uso.   | 125        |
| 8.5.- Casos de uso en formato extendido.                                      | 127        |
| 8.5.1.- Caso de uso: iniciar sesión [CU-1].                                   | 128        |
| 8.5.2.- Caso de uso: gestionar perfil como usuario [CU-2].                    | 129        |
| 8.5.3.- Caso de uso: modificar datos del perfil [CU-2.1].                     | 130        |
| 8.5.4.- Caso de uso: modificar contraseña de acceso [CU-2.2].                 | 131        |
| 8.5.5.- Caso de uso: cerrar sesión [CU-2.3].                                  | 131        |
| 8.5.6.- Caso de uso: gestionar usuarios [CU-3].                               | 132        |
| 8.5.7.- Caso de uso: mostrar usuario existente [CU-3.1].                      | 133        |
| 8.5.8.- Caso de uso: crear nuevo usuario [CU-3.2].                            | 134        |
| 8.5.9.- Caso de uso: modificar usuario existente [CU-3.3].                    | 134        |
| 8.5.10.- Caso de uso: restablecer contraseña de usuario existente [CU-3.3.1]. | 135        |
| 8.5.11.- Caso de uso: eliminar usuario existente [CU-3.4].                    | 135        |
| 8.5.12.- Caso de uso: gestionar aplicaciones [CU-4].                          | 136        |
| 8.5.13.- Caso de uso: crear nueva aplicación [CU-4.1].                        | 137        |
| 8.5.14.- Caso de uso: consultar aplicación existente [CU-4.2].                | 138        |
| 8.5.15.- Caso de uso: gestionar accesos de usuario [CU-4.2.1].                | 138        |
| 8.5.16.- Caso de uso: conceder acceso [CU-4.2.1.1].                           | 139        |



|   |     |
|---|-----|
| 8.5.17.- Caso de uso: bloquear acceso [CU-4.2.1.2].                                     | 139 |
| 8.5.18.- Caso de uso: eliminar acceso [CU-4.2.1.3].                                     | 140 |
| 8.5.19.- Caso de uso: gestionar despliegues [CU-4.2.2].                                 | 140 |
| 8.5.20.- Caso de uso: instalar aplicación en sistema [CU-4.2.2.1].                      | 141 |
| 8.5.21.- Caso de uso: desinstalar aplicación de sistema [CU-4.2.2.2].                   | 141 |
| 8.5.22.- Caso de uso: modificar aplicación existente [CU-4.3].                          | 142 |
| 8.5.23.- Caso de uso: modelar aplicación existente [CU-4.4].                            | 142 |
| 8.5.24.- Caso de uso: gestionar módulos [CU-4.4.1].                                     | 145 |
| 8.5.25.- Caso de uso: consultar módulo existente [CU-4.4.1.1]                           | 145 |
| 8.5.26.- Caso de uso: crear nuevo módulo [CU-4.4.1.2].                                  | 146 |
| 8.5.27.- Caso de uso: modificar módulo existente [CU-4.4.1.3].                          | 146 |
| 8.5.28.- Caso de uso: eliminar módulo existente [CU-4.4.1.4].                           | 147 |
| 8.5.29.- Caso de uso: gestionar tipos de datos [CU-4.4.2].                              | 148 |
| 8.5.30.- Caso de uso: consultar tipo de datos existente [CU-4.4.2.1].                   | 148 |
| 8.5.31.- Caso de uso: crear nuevo tipo de datos [CU-4.4.2.2].                           | 149 |
| 8.5.32.- Caso de uso: modificar tipo de datos existente [CU-4.4.2.3].                   | 149 |
| 8.5.33.- Caso de uso: eliminar tipo de datos existente [CU-4.4.2.4].                    | 150 |
| 8.5.34.- Caso de uso: gestionar políticas de tipo de datos [CU-4.4.3].                  | 151 |
| 8.5.35.- Caso de uso: consultar política existente de tipo de datos [CU-4.4.3.1].       | 151 |
| 8.5.36.- Caso de uso: crear nueva política de tipo de datos [CU-4.4.3.2].               | 152 |
| 8.5.37.- Caso de uso: modificar política existente de tipo de datos [CU-4.4.3.3].       | 152 |
| 8.5.38.- Caso de uso: eliminar política existente de tipo de datos [CU-4.4.3.4].        | 153 |
| 8.5.39.- Caso de uso: gestionar entidades [CU-4.4.4].                                   | 154 |
| 8.5.40.- Caso de uso: consultar entidad existente [CU-4.4.4.1].                         | 154 |
| 8.5.41.- Caso de uso: crear nueva entidad [CU-4.4.4.2].                                 | 155 |
| 8.5.42.- Caso de uso: modificar entidad existente [CU-4.4.4.3].                         | 155 |
| 8.5.43.- Caso de uso: eliminar entidad existente [CU-4.4.4.4].                          | 156 |
| 8.5.44.- Caso de uso: gestionar políticas de entidad [CU-4.4.5].                        | 157 |
| 8.5.45.- Caso de uso: consultar política existente de entidad [CU-4.4.5.1].             | 157 |
| 8.5.46.- Caso de uso: crear nueva política de entidad [CU-4.4.5.2].                     | 158 |
| 8.5.47.- Caso de uso: modificar política existente de entidad [CU-4.4.5.3].             | 158 |
| 8.5.48.- Caso de uso: eliminar política existente de entidad [CU-4.4.5.4].              | 159 |
| 8.5.49.- Caso de uso: gestionar atributos de entidad [CU-4.4.6].                        | 160 |
| 8.5.50.- Caso de uso: consultar atributo existente de entidad [CU-4.4.6.1].             | 160 |
| 8.5.51.- Caso de uso: crear nuevo atributo de entidad [CU-4.4.6.2].                     | 161 |
| 8.5.52.- Caso de uso: modificar atributo existente de entidad [CU-4.4.6.3].             | 161 |
| 8.5.53.- Caso de uso: eliminar atributo existente de entidad [CU-4.4.6.4].              | 162 |
| 8.5.54.- Caso de uso: gestionar políticas de atributo de entidad [CU-4.4.7].            | 162 |
| 8.5.55.- Caso de uso: consultar política existente de atributo de entidad [CU-4.4.7.1]. | 163 |
| 8.5.56.- Caso de uso: crear nueva política de atributo de entidad [CU-4.4.7.2].         | 164 |
| 8.5.57.- Caso de uso: modificar política existente de atributo de entidad [CU-4.4.7.3]. | 164 |
| 8.5.58.- Caso de uso: eliminar política existente de atributo de entidad [CU-4.4.7.4].  | 165 |
| 8.5.59.- Caso de uso: gestionar relaciones [CU-4.4.8].                                  | 166 |
| 8.5.60.- Caso de uso: consultar relación existente [CU-4.4.8.1].                        | 166 |
| 8.5.61.- Caso de uso: crear nueva relación [CU-4.4.8.2].                                | 167 |
| 8.5.62.- Caso de uso: modificar relación existente [CU-4.4.8.3].                        | 167 |
| 8.5.63.- Caso de uso: eliminar relación existente [CU-4.4.8.4].                         | 168 |
| 8.5.64.- Caso de uso: gestionar objetos valor [CU-4.4.9].                               | 169 |
| 8.5.65.- Caso de uso: consultar objeto valor existente [CU-4.4.9.1].                    | 169 |
| 8.5.66.- Caso de uso: crear nuevo objeto valor [CU-4.4.9.2].                            | 170 |
| 8.5.67.- Caso de uso: modificar objeto valor existente [CU-4.4.9.3].                    | 170 |
| 8.5.68.- Caso de uso: eliminar objeto valor existente [CU-4.4.9.4].                     | 171 |
| 8.5.69.- Caso de uso: gestionar políticas de objeto valor [CU-4.4.10].                  | 172 |
| 8.5.70.- Caso de uso: consultar política existente de objeto valor [CU-4.4.10.1].       | 172 |
| 8.5.71.- Caso de uso: crear nueva política de objeto valor [CU-4.4.10.2].               | 173 |



|   |     |
|---|-----|
| 8.5.72.- Caso de uso: modificar política existente de objeto valor [CU-4.4.10.3].             | 173 |
| 8.5.73.- Caso de uso: eliminar política existente de objeto valor [CU-4.4.10.4].              | 174 |
| 8.5.74.- Caso de uso: gestionar atributos de objeto valor [CU-4.4.11].                        | 175 |
| 8.5.75.- Caso de uso: consultar atributo existente de objeto valor [CU-4.4.11.1].             | 175 |
| 8.5.76.- Caso de uso: crear nuevo atributo de objeto valor [CU-4.4.11.2].                     | 176 |
| 8.5.77.- Caso de uso: modificar atributo existente de objeto valor [CU-4.4.11.3].             | 176 |
| 8.5.78.- Caso de uso: eliminar atributo existente de objeto valor [CU-4.4.11.4].              | 177 |
| 8.5.79.- Caso de uso: gestionar políticas de atributo de objeto valor [CU-4.4.12].            | 178 |
| 8.5.80.- Caso de uso: consultar política existente de atributo de objeto valor [CU-4.4.12.1]. | 179 |
| 8.5.81.- Caso de uso: crear nueva política de atributo de objeto valor [CU-4.4.12.2].         | 179 |
| 8.5.82.- Caso de uso: modificar política existente de atributo de objeto valor [CU-4.4.12.3]. | 180 |
| 8.5.83.- Caso de uso: eliminar política existente de atributo de objeto valor [CU-4.4.12.4].  | 180 |
| 8.5.84.- Caso de uso: gestionar servicios [CU-4.4.13].  | 181 |
| 8.5.85.- Caso de uso: consultar servicio existente [CU-4.4.13.1].                             | 182 |
| 8.5.86.- Caso de uso: crear nuevo servicio [CU-4.4.13.2].                                     | 182 |
| 8.5.87.- Caso de uso: modificar servicio existente [CU-4.4.13.3].                             | 183 |
| 8.5.88.- Caso de uso: eliminar servicio existente [CU-4.4.13.4].                              | 183 |
| 8.5.89.- Caso de uso: gestionar procesos por lotes [CU-4.4.14].                               | 184 |
| 8.5.90.- Caso de uso: consultar proceso por lotes existente [CU-4.4.14.1].                    | 185 |
| 8.5.91.- Caso de uso: crear nuevo proceso por lotes [CU-4.4.14.2].                            | 185 |
| 8.5.92.- Caso de uso: modificar proceso por lotes existente [CU-4.4.14.3].                    | 186 |
| 8.5.93.- Caso de uso: eliminar proceso por lotes existente [CU-4.4.14.4].                     | 187 |
| 8.5.94.- Caso de uso: eliminar aplicación existente [CU-4.5].                                 | 187 |
| 8.5.95.- Caso de uso: gestionar idiomas y traducciones [CU-5].                                | 188 |
| 8.5.96.- Caso de uso: crear nuevo idioma [CU-5.1].  | 189 |
| 8.5.97.- Caso de uso: consultar idioma existente [CU-5.2].                                    | 190 |
| 8.5.98.- Caso de uso: consultar traducciones [CU-5.3].  | 190 |
| 8.5.99.- Caso de uso: crear nueva traducción [CU-5.3.1].                                      | 191 |
| 8.5.100.- Caso de uso: modificar traducción [CU-5.3.2].                                       | 191 |
| 8.5.101.- Caso de uso: eliminar traducción [CU-5.3.3].  | 192 |
| 8.5.102.- Caso de uso: modificar idioma existente [CU-5.4].                                   | 192 |
| 8.5.103.- Caso de uso: eliminar idioma existente [CU-5.5].                                    | 193 |
| 8.5.104.- Caso de uso: gestión de sistemas y despliegues [CU-5].                              | 193 |
| 8.5.105.- Caso de uso: creación de un nuevo sistema [CU-6.1].                                 | 194 |
| 8.5.106.- Caso de uso: consulta de un sistema existente [CU-6.2].                             | 195 |
| 8.5.107.- Caso de uso: modificación de un sistema existente [CU-6.3].                         | 195 |
| 8.5.108.- Caso de uso: eliminación de un sistema existente [CU-6.4].                          | 196 |
| 8.5.109.- Caso de uso: gestionar información [CU-7].  | 196 |
| 8.5.110.- Caso de uso: listar registro existente de entidad [CU-7.1].                         | 198 |
| 8.5.111.- Caso de uso: consultar registro existente de entidad [CU-7.2].                      | 198 |
| 8.5.112.- Caso de uso: crear nuevo registro de entidad [CU-7.3].                              | 199 |
| 8.5.113.- Caso de uso: modificar registro existente de entidad [CU-7.4].                      | 199 |
| 8.5.114.- Caso de uso: eliminar registro existente de entidad [CU-7.5].                       | 200 |

## **9.- ANEXO B: DISEÑO TÉCNICO.**

**201**

|   |     |
|---|-----|
| 9.1.- Arquitectura tecnológica de la solución.                          | 201 |
| 9.2.- Documentación de código fuente.                                   | 208 |
| 9.3.- Modelo de datos.  | 210 |
| 9.3.1.- Diagrama de clases.   | 211 |
| 9.3.2.- Diagramas de bases de datos.                                    | 217 |
| 9.3.2.1.- Diagrama entidad/relación de la aplicación de administración. | 218 |
| 9.3.2.2.- Diagrama relacional de la aplicación de administración.       | 219 |
| 9.4.- Modelo de procesos.   | 220 |
| 9.4.1.- Diagrama de navegación.   | 220 |
| 9.4.2.- Catálogo de procesos.   | 228 |



|  |     |
|--|-----|
| 9.5.- Diseño de la interfaz de usuario.                | 240 |
| 9.5.1.- Usabilidad.                                    | 240 |
| 9.5.2.- Diseño web adaptativo (responsive web design). | 241 |

## **10.- ANEXO C: MANUAL DE USUARIO.** **242**

|   |     |
|---|-----|
| 10.1.- Requisitos del sistema.  | 244 |
| 10.1.1.- Requisitos generales del sistema.  | 244 |
| 10.2.- Proceso de instalación.  | 245 |
| 10.2.1.- Instalación de referencia en la plataforma PC, basada en Intel y Windows.                | 245 |
| 10.2.1.1.- Instalación del sistema operativo.   | 246 |
| 10.2.1.2.- Instalación del servidor web.  | 247 |
| 10.2.1.3.- Instalación del sistema gestor de bases de datos.                                      | 247 |
| 10.2.1.4.- Instalación manual de los controladores de acceso a la base de datos.                  | 255 |
| 10.2.1.5.- Creación del inicio de sesión en el sistema gestor de bases de datos.                  | 258 |
| 10.2.1.6.- Instalación de PHP para Windows.   | 262 |
| 10.2.1.7.- Creación de la estructura de directorios.  | 264 |
| 10.2.1.8.- Publicación en el servidor Web.  | 265 |
| 10.2.1.9.- Creación de un nuevo sitio Web.  | 266 |
| 10.2.1.10.- Instalación de la base de datos.  | 268 |
| 10.2.1.11.- Pruebas de funcionamiento.  | 268 |
| 10.3.- Elementos generales de la interfaz de usuario.   | 269 |
| 10.3.1.- Acceso web.  | 269 |
| 10.3.2.- Diseño web adaptativo ( <i>responsive</i> ).   | 269 |
| 10.3.3.- Cabecera: barra superior de navegación.  | 271 |
| 10.3.3.1.- Menú de aplicaciones.  | 271 |
| 10.3.3.2.- Menú de administración.  | 272 |
| 10.3.3.3.- Menú de usuario.   | 273 |
| 10.3.3.4.- Menú de enlaces de la UNED.  | 273 |
| 10.3.3.5.- Control de búsqueda global.  | 274 |
| 10.3.4.- Pie de página: barra inferior de información.  | 274 |
| 10.3.5.- Contenidos en múltiples idiomas.   | 275 |
| 10.4.- Pantalla de inicio.  | 275 |
| 10.5.- Pantalla de información acerca de la solución.   | 276 |
| 10.6.- Pantalla de autenticación.   | 277 |
| 10.6.1.- Introducir credenciales.   | 278 |
| 10.6.2.- Restablecer contraseña.  | 279 |
| 10.7.- Panel de control (pantalla principal).   | 282 |
| 10.8.- Pantalla de administración de la solución.   | 283 |
| 10.9.- Pantalla de administración de sistemas y despliegues.                                      | 284 |
| 10.10.- Pantalla de administración para la consulta de un sistema existente.                      | 286 |
| 10.11.- Pantalla de administración para la modificación de un sistema existente.                  | 287 |
| 10.12.- Pantalla de administración de usuarios y accesos.   | 288 |
| 10.13.- Pantalla de administración para la consulta de un usuario existente.                      | 289 |
| 10.14.- Pantalla de administración para la creación de un nuevo usuario.                          | 291 |
| 10.15.- Pantalla de administración para la modificación de un usuario existente y sus accesos.    | 292 |
| 10.16.- Pantalla de administración de aplicaciones generadas para consumidores.                   | 293 |
| 10.17.- Pantalla de administración para el modelado de una aplicación generada para consumidores. | 295 |
| 10.18.- Pantalla de administración de idiomas y traducciones.                                     | 296 |
| 10.19.- Pantalla de Administración de Consulta de un Idioma.                                      | 298 |
| 10.20.- Pantalla de Administración para la Creación de un Nuevo Idioma.                           | 299 |
| 10.21.- Pantalla de Administración de Modificación de un Idioma.                                  | 300 |

## ÍNDICE DE FIGURAS

|  |     |
|--|-----|
| Fig. 2.1.- Árbol de características en un análisis FODA (fuente: [85]).                                | 22  |
| Fig. 3.1.- Estimación inicial; planificación del esfuerzo.   | 44  |
| Fig. 4.1.- Ciclo de vida del "Proceso Unificado de Rational" (RUP).                                    | 48  |
| Fig. 4.2.- Consola de mandatos del entorno MINGW64 (GNU bash 4.4.19).                                  | 53  |
| Fig. 4.3.- Logotipo de Git for Windows.  | 53  |
| Fig. 4.4.- Logotipo de PHP.  | 53  |
| Fig. 4.5.- Logotipo de HTML5.  | 53  |
| Fig. 4.6.- Logotipo de CSS3.   | 54  |
| Fig. 4.7.- Logotipo de SASS.   | 54  |
| Fig. 4.8.- Logotipo de COMPASS.  | 54  |
| Fig. 4.9.- Línea de mandatos para la instalación de SASS y Compass desde RUBY GEMS.                    | 55  |
| Fig. 4.10.- Logotipo de RUBY.  | 55  |
| Fig. 4.11.- Instalación del entorno Ruby para Windows.   | 56  |
| Fig. 4.12.- Logotipo de SQLite.  | 57  |
| Fig. 4.13.- Logotipo de SQLite Studio.   | 58  |
| Fig. 4.14.- Base de datos del modelo de GAMS, abierta con SQLite Studio.                               | 58  |
| Fig. 8.1.- Diagrama de requisitos.   | 120 |
| Fig. 8.2.- Diagrama conceptual.  | 121 |
| Fig. 8.3.- Diagrama de clases de alto nivel.   | 123 |
| Fig. 8.4.- Actores participantes en la solución GAMS.  | 125 |
| Fig. 8.5.- Vista de casos de uso primarios.  | 126 |
| Fig. 8.6.- Casos de uso secundarios de "gestionar perfil como usuario".                                | 130 |
| Fig. 8.7.- Casos de uso secundarios de "gestionar usuarios".   | 133 |
| Fig. 8.8.- Casos de uso secundarios de "gestionar aplicaciones".                                       | 137 |
| Fig. 8.9.- Diagrama de casos de uso secundarios de "modelar aplicación existente".                     | 144 |
| Fig. 8.10.- Casos de uso secundarios de "gestionar módulos".   | 145 |
| Fig. 8.11.- Casos de uso secundarios de "gestionar tipos de datos".                                    | 148 |
| Fig. 8.12.- Casos de uso secundarios de "gestionar políticas de tipo de datos".                        | 151 |
| Fig. 8.13.- Casos de uso secundarios de "gestionar entidades".   | 154 |
| Fig. 8.14.- Casos de uso secundarios de "gestionar políticas de entidad".                              | 157 |
| Fig. 8.15.- Casos de uso secundarios de "gestionar atributos de entidad".                              | 160 |
| Fig. 8.16.- Casos de uso secundarios de "gestionar políticas de atributo de entidad".                  | 163 |
| Fig. 8.17.- Casos de uso secundarios de "gestionar relaciones".  | 166 |
| Fig. 8.18.- Casos de uso secundarios de "gestionar objetos valor".                                     | 169 |
| Fig. 8.19.- Casos de uso secundarios de "gestionar políticas de objeto valor".                         | 172 |
| Fig. 8.20.- Casos de uso secundarios de "gestionar atributos de objeto valor".                         | 175 |
| Fig. 8.21.- Casos de uso secundarios de "gestionar políticas de atributo de objeto valor".             | 178 |
| Fig. 8.22.- Casos de uso secundarios de "gestionar servicios".   | 182 |
| Fig. 8.23.- Casos de uso secundarios de "gestionar procesos por lotes".                                | 185 |
| Fig. 8.24.- Diagrama de casos de uso secundarios de "gestionar idiomas y traducciones".                | 189 |
| Fig. 8.25.- Diagrama de casos de uso secundarios de "gestión de sistemas y despliegues".               | 194 |
| Fig. 8.26.- Casos de uso secundarios de "gestionar información".                                       | 197 |
| Fig. 9.1.- Estilo arquitectónico de la solución.   | 204 |
| Fig. 9.2.- Diagrama detallado de arquitectura.   | 207 |
| Fig. 9.3.- Diagrama de despliegue.   | 208 |
| Fig. 9.4.- Documentación de código fuente PHPDOC; ejemplo de uso en la clase "Application Controller". | 209 |
| Fig. 9.5.- Documentación del código fuente de la solución GAMS, generada mediante la herramienta SAMI. | 210 |
| Fig. 9.6.- Estructura de directorio del código fuente del paquete AppBundle de la solución GAMS.       | 212 |

|   |     |
|---|-----|
| Fig. 9.7.- Diagrama de clases: vista detallada de un extracto de clases del subdirectorio “AppBundle/Controller”.     | 214 |
| Fig. 9.8.- Diagrama de clases: vista reducida de las clases del subdirectorio “AppBundle/Entity”.                     | 215 |
| Fig. 9.9.- Diagrama de clases: vista detallada de la jerarquía de clases que modelan los atributos del dominio.       | 216 |
| Fig. 9.10.- Diagrama entidad/relación de la aplicación de administración.   | 218 |
| Fig. 9.11.- Diagrama relacional de la aplicación de administración.   | 219 |
| Fig. 9.12.- Diagrama de navegación a través de la interfaz gráfica de usuario, basada en tecnologías web.             | 223 |
| Fig. 9.13.- Diagrama de secuencia general para las operaciones con la interfaz de usuario.                            | 239 |
| Fig. 10.1.- Programa cargador web de Microsoft SQL Server 2017 edición Express.                                       | 248 |
| Fig. 10.2.- Selección de la opción a descargar e instalar de Microsoft SQL Server 2017 edición Express.               | 249 |
| Fig. 10.3.- Instalador de Microsoft SQL Server 2017 edición Express.  | 249 |
| Fig. 10.4.- Características a instalar de Microsoft SQL Server 2017 edición Express.                                  | 250 |
| Fig. 10.5.- Nombre de la instalación del servidor Microsoft SQL Server 2017 Express.                                  | 251 |
| Fig. 10.6.- Modo mixto de autenticación para Microsoft SQL Server 2017 edición Express.                               | 252 |
| Fig. 10.7.- Resultado satisfactorio de la instalación de Microsoft SQL Server 2017 edición Express.                   | 253 |
| Fig. 10.8.- Instalación de Microsoft SQL Server 2017 Management Studio.   | 254 |
| Fig. 10.9.- Creación de un DSN para verificar la instalación del controlador ODBC v17 para Microsoft SQL Server 2017. | 256 |
| Fig. 10.10.- DSN de sistema configurado para el acceso a Microsoft SQL Server 2017.                                   | 257 |
| Fig. 10.11.- Creación del inicio de sesión en Microsoft SQL Server (paso 1 de 4).                                     | 259 |
| Fig. 10.12.- Creación del inicio de sesión en Microsoft SQL Server (paso 2 de 4).                                     | 260 |
| Fig. 10.13.- Creación del inicio de sesión en Microsoft SQL Server (paso 3 de 4).                                     | 260 |
| Fig. 10.14.- Creación del inicio de sesión en Microsoft SQL Server (paso 4 de 4).                                     | 261 |
| Fig. 10.15.- Inicio de sesión creado para GAMS en Microsoft SQL Server 2017.  | 261 |
| Fig. 10.16.- Instalación de PHP 7 desde Microsoft Web Platform Installer.   | 262 |
| Fig. 10.17.- Consola de administración (MMC) de "Administrador de Internet Information Services (IIS)".               | 265 |
| Fig. 10.18.- Creación de un nuevo Sitio Web para publicar la aplicación GCVN.   | 266 |
| Fig. 10.19.- Configuración HTTP del nuevo sitio web.  | 267 |
| Fig. 10.20.- Configuración S/HTTP del sitio web, indicando el certificado SSL/TLS.                                    | 267 |
| Fig. 10.21.- Ejemplo de visualización web adaptada a teléfonos móviles de con baja resolución.                        | 270 |
| Fig. 10.22.- Ejemplo de visualización web adaptada a una tableta con conexión inalámbrica.                            | 270 |
| Fig. 10.23.- Cabecera, barra inicial.   | 271 |
| Fig. 10.24.- Cabecera, barra completamente extendida.   | 271 |
| Fig. 10.25.- Cabecera, barra comprimida.  | 271 |
| Fig. 10.26.- Menú de aplicaciones, extendido.   | 271 |
| Fig. 10.27.- Menú de administración, extendido.   | 272 |
| Fig. 10.28.- Menú asociado al perfil del usuario, extendido.  | 273 |
| Fig. 10.29.- Menú de la UNED, extendido.  | 273 |
| Fig. 10.30.- Control de búsqueda global.  | 274 |
| Fig. 10.31.- Pie de página inferior, completamente extendido.   | 274 |
| Fig. 10.32.- Pie de página inferior, parcialmente contraído.  | 274 |
| Fig. 10.33.- Pie de página inferior, completamente contraído.   | 275 |
| Fig. 10.34.- Pantalla de inicio.  | 275 |
| Fig. 10.35.- Pantalla de información acerca de la solución.   | 277 |
| Fig. 10.36.- Pantalla de autenticación: introducir credenciales.  | 278 |
| Fig. 10.37.- Pantalla de autenticación: recordar contraseña.  | 279 |
| Fig. 10.38.- Pantalla de autenticación: envío de correo para restablecer contraseña.                                  | 280 |
| Fig. 10.39.- Correo electrónico para restablecimiento de contraseñas.   | 280 |



|  |     |
|--|-----|
| Fig. 10.40.- Pantalla de restablecimiento de contraseña, vista desde un teléfono móvil.                          | 281 |
| Fig. 10.41.- Pantalla principal, denominada panel de control.  | 282 |
| Fig. 10.42.- Pantalla de administración general de la solución.  | 283 |
| Fig. 10.43.- Pantalla de administración de sistemas y despliegues.   | 284 |
| Fig. 10.44.- Menú desplegable de acciones a realizar sobre un sistema registrado en la solución GAMS.            | 284 |
| Fig. 10.45.- Menú desplegable para el inicio de una aplicación, desde un sistema registrado en la solución GAMS. | 285 |
| Fig. 10.46.- Pantalla de visualización de un sistema existente.  | 286 |
| Fig. 10.47.- Pantalla de modificación de un sistema existente.   | 287 |
| Fig. 10.48.- Pantalla de administración de usuarios y accesos.   | 288 |
| Fig. 10.49.- Pantalla de administración para la consulta de un usuario.  | 289 |
| Fig. 10.50.- Pantalla de administración para la creación de un nuevo usuario.                                    | 291 |
| Fig. 10.51.- Pantalla de administración de modificación de un usuario y sus accesos (primera parte).             | 292 |
| Fig. 10.52.- Pantalla de administración de modificación de un usuario y sus accesos (segunda parte).             | 292 |
| Fig. 10.53.- Pantalla de administración de aplicaciones generadas para consumidores.                             | 293 |
| Fig. 10.54.- Opciones para ejecutar una aplicación en función del sistema donde se haya desplegado.              | 294 |
| Fig. 10.55.- Acciones para modificar o modelar una aplicación generada para consumidores.                        | 294 |
| Fig. 10.56.- Pantalla de administración para el modelado de una aplicación, tabla de propiedades.                | 295 |
| Fig. 10.57.- Pantalla de administración para el modelado de una aplicación, árbol de elementos.                  | 295 |
| Fig. 10.58.- Pantalla de administración de idiomas y traducciones.   | 296 |
| Fig. 10.59.- Pantalla de administración de consulta de un idioma.  | 298 |
| Fig. 10.60.- Pantalla de administración para la creación de un nuevo idioma.                                     | 299 |
| Fig. 10.61.- Pantalla de administración de modificación de idioma.   | 300 |

## ÍNDICE DE TABLAS

|   |     |
|---|-----|
| Tabla 9.1.- Rutas de acceso a cada una de las acciones de los diferentes controladores. | 228 |
| Tabla 9.2.- Relación entre escenarios de casos de uso y procesos.                       | 237 |



# 1.- INTRODUCCIÓN.

*Todo lo que una persona puede imaginar, otras  
podrán hacerlo realidad.*

Julio Verne (1828-1905)

El presente Trabajo Fin de Máster, titulado **Generación Automática de MicroServicios (GAMS)**, se ocupa de la automatización en el desarrollo software de servicios web que siguen la filosofía denominada de “*microservicios*” [1].

Uno de los principales objetivos que se pretende cubrir es la aceleración en el desarrollo de aplicaciones basadas en este tipo de arquitecturas software.

Comenzaremos esta introducción abordando los siguientes puntos:

- Organización de la Memoria del Trabajo.
- Problemática.
- Solución planteada.
- Delimitación del alcance.

## 1.1.- Organización de la Memoria del Trabajo.

La organización de la presente Memoria del Trabajo Fin de Máster es la siguiente:

- **Introducción**, incluyendo una exposición de las necesidades y problemática que motivan la creación del presente Proyecto.
- **Estado de la cuestión**, donde se investigan las soluciones existentes actualmente (incluyendo proyectos de software libre [2], productos comerciales, iniciativas institucionales, así como estándares internacionales) para abordar y solucionar las necesidades planteadas.



- **Planteamiento e hipótesis**, que expone los objetivos, delimita el alcance del Trabajo, así como refleja el plan de proyecto establecido.
- **Resolución**, introducción al material utilizado para la realización del apartado software del Proyecto. Repasa los productos obtenidos, ligados a la Ingeniería del Software, que serán profundamente detallados en los respectivos anexos (el Anexo A está dedicado al Análisis Funcional, el Anexo B está dedicado al Diseño Técnico y el Anexo C constituye el Manual de Usuario).
- **Líneas futuras y conclusiones**, proponiendo futuras ampliaciones de la funcionalidad desarrollada, así como presentando las conclusiones finales acerca del presente Proyecto.
- **Bibliografía y referencias**, listado de toda la bibliografía y material de referencia consultado durante la realización del Proyecto.
- **Anexo A: productos del Análisis Funcional**, donde se presenta el material ligado a la Ingeniería del Software referente a la fase de definición del proyecto (diagramas y esquemas, entre otros, con una exposición detallada de los mismos).
- **Anexo B: productos del Diseño Técnico**, donde se presenta el material ligado a la Ingeniería del Software referente a la fase de diseño del proyecto (diagramas y esquemas, entre otros, con una exposición detallada de los mismos).
- **Anexo C: Manual de Usuario**, que contiene las pautas de instalación y utilización del programa, así como capturas de todas y cada una de las pantallas del mismo, ofreciendo una visión exhaustiva del funcionamiento de la solución construida.

## 1.2.- Problemática.

En los últimos años, los sistemas intensivos en software han crecido tanto en número de sistemas como en el volumen de funcionalidades ofrecidas por cada uno de ellos [1].

Sin embargo, las herramientas precisas para acometer con garantías los desarrollos necesarios por dichos sistemas intensivos en software no han evolucionado al mismo ritmo.

Los avances realizados recientemente están orientados a gestionar la ejecución de las aplicaciones resultantes (ya sean tecnologías como Docker [3] y Kubernetes [4] para la gestión de contenedores [5], o Vagrant [6] y Puppet [7] más asociadas inicialmente hacia la virtualización [8]), controlándolas desde los entornos de desarrollo y pruebas, así como trabajando en replicar y configurar de manera adecuada los entornos de producción.

Precisamente de la coordinación entre el trabajo de programación en los entornos de desarrollo y pruebas, y el trabajo de operaciones de los entornos de producción, aparecen nuevas propuestas metodológicas y de “buenas prácticas” englobadas en lo que comúnmente se denomina “DevOps” [9].

Sin embargo, aunque todas estas tecnologías abogan por técnicas para ejecutar sistemas intensivos en software, ninguna pone el foco en cómo abordar el diseño y construcción de dichos sistemas.



### 1.3.- Solución Planteada.

El principal objetivo del presente Trabajo Fin de Máster es la creación de una plataforma tecnológica que permita crear y mantener aplicaciones basadas en el paradigma arquitectónico de los microservicios, permitiendo además que dichos servicios sean ejecutados tanto en servidores como en ordenadores personales e incluso en dispositivos embebidos.

Para ello, se planteará el desarrollo de una herramienta que construya automáticamente aplicaciones, eliminando la necesidad de programar dichas aplicaciones, al menos en el sentido tradicional de programación. Para ello, se abordarán técnicas de diseño dirigido por modelos [10] como de programación por parte del usuario final [11].

Esta herramienta hará uso en todo momento de tecnologías que sean estándares *de facto* en Internet, como son HTML (*HyperText Mark-Up Language*, véase [12] y [13]) para la presentación de la información, CSS (*Cascading Style Sheets* [13]) para establecer estilos visuales en dicha presentación, JavaScript [14] para dotar de actividad la interfaz de usuario, JSON (*JavaScript Object Notation* [15]) y XML (*Extensible Mark-Up Language* [16]) como formatos interoperables de información y REST (*Representational State Transfer* [17]) como principio arquitectónico para construir los microservicios y que se interrelacionen entre sí.

Desde la perspectiva de la infraestructura del servidor, se hará uso de la tecnología PHP [18], servidores Apache [19] e IIS [20], bases de datos Microsoft SQL Server [21], MySQL [22]/MariaDB [23] y SQLite [24], así como sistemas operativos Microsoft Windows [25] y Linux [26].

### 1.3.1.- Objetivos a cubrir.

Los objetivos a cubrir por este Proyecto de **Generación Automática de MicroServicios** son los siguientes:

- Disponer de una solución que facilite la creación y mantenimiento de aplicaciones orientadas a servicios, basadas en una arquitectura con el estilo propio de los microservicios.
- Los microservicios deben poder desarrollarse sin excesivos conocimientos de programación, utilizando mecanismos al alcance del usuario final (véase [11]).
- Los microservicios deben poder explotarse por parte del usuario final, dotando a la solución de una interfaz gráfica de usuario (GUI, *Graphical User Interface* [27]) con capacidades de interfaz entre máquinas (M2M, *Machine-To-Machine* [28]).
- Debe hacerse uso de cuantos estándares existan, tanto técnicos como de documentación, en lo que al tratamiento y transmisión de la información se refiera.

Las principales motivaciones que impulsan la creación de este Proyecto son las siguientes:

- Ventajas de disponer de una herramienta que facilite el diseño y desarrollo de servicios web, ahorrando la mayor parte del tiempo dedicado a la programación tradicionalmente manual de los mismos.
- Los servicios web resultantes deben estar basados en un estilo arquitectónico propio de los microservicios.
- La solución final debe poder ejecutarse en plataformas muy diversas, en cuanto a capacidad de cómputo.

### **1.3.2.- Funcionamiento general.**

El sistema de **Generación Automática de MicroServicios**, como producto software resultante, pretende ser una herramienta intuitiva, a la vez que potente, para la definición (modelado) y operación (uso) de aplicaciones y servicios web. Gracias precisamente a estar basada en interfaces web, podrá ser utilizada con total independencia del puesto de trabajo en el que se ubiquen los usuarios.

La labor inicial de implantación del sistema por parte de los administradores responsables del mismo conlleva, además, la configuración y parametrización, lo que se traduce en la creación de las aplicaciones generadas para los consumidores que definirán, a modo de meta-esquema, el contenido de la información que pueda gestionarse.

El sistema software se basa en la gestión de 14 entidades principales, de las cuales presentamos ahora un esbozo y sobre las que profundizaremos más detalladamente en el capítulo 4 (Resolución):

- **Sistema:** representación de un sistema con capacidad para ejecutar la plataforma GAMS.
- **Usuario:** credenciales de acceso e información de un usuario, definiendo además si éste se encuentra habilitado o bloqueado, así como su rol dentro del sistema (como administrador o como consumidor).
- **Aplicación:** representación de una aplicación a ejecutar por la plataforma GAMS.
- **Módulo:** subconjunto de elementos en los que se divide una aplicación.
- **Tipo de Datos:** permite definir el dominio de la información contenida en el valor de un atributo; es decir, delimita el conjunto de valores que puede tomar.
- **Entidad:** concepto modelado con identidad propia.



- **Objeto Valor:** concepto modelado de interés por su representación, su valor, dado que carece de identidad propiamente dicha.
- **Atributo:** característica asociada a una entidad u objeto valor, con capacidad para albergar un valor según un tipo de datos.
- **Relación:** modela la posible asociación existente, dos a dos, entre entidades y objetos valor.
- **Operación:** algoritmo ligado a una entidad.
- **Evento:** algoritmo ligado a una entidad, objeto valor o atributo, que se ejecuta en un determinado momento de su ciclo de vida.
- **Política:** regla de validación que deben cumplir determinados elementos de dominio (como los tipos de datos, las entidades, los objeto valor y los atributos), para que su información (o estado) pueda ser considerado como válido.
- **Servicio:** representación de un algoritmo independiente.
- **Proceso por lotes:** planificación de la ejecución automática de un servicio.

### **1.3.3.- Metodología seguida y plan de trabajo.**

Debido a que desde el primer momento se ha decidido aplicar un plan de trabajo incremental, que permita ir obteniendo subproductos funcionales a medida que se avanza en el desarrollo del proyecto, se ha optado por una metodología reconocida y suficientemente flexible que permitiese el esquema deseado.

Por ello, la elección ha sido el Proceso Unificado de Rational (más conocido por sus siglas inglesas RUP, *Rational Unified Process*), haciendo uso del *Unified Modeling Language* (UML), descritos en [29], [30] y [31]. El actual responsable de ambos es el *Object Management Group* (OMG) [32], quien ha desarrollado la metodología de Arquitectura Dirigida por Modelos (MDA, *Model Driven Architecture*) [33] que por su complejidad no se utiliza en el presente Proyecto.

Más adelante (véase apartado 4.1) se comentarán estos detalles con mayor profundidad.

#### **1.3.4.- Arquitectura tecnológica.**

Como no podía ser de otra manera, la tecnología juega un papel fundamental en el desarrollo de este Proyecto.

La aplicación software **Generación Automática de MicroServicios** ha sido concebida utilizando modernos Entornos de Desarrollo Integrados (IDE, *Integrated Development Environment*) [34] así como aplicando técnicas de generación automática de código [35].

La plataforma de desarrollo escogida ha sido PHP 7 [18], HTML 5 [12], CSS 3 [13] y JavaScript [14], utilizando el entorno JetBrains PhpStorm [36], los sistemas gestores de bases de datos MariaDB [23] y SQLite [24], todo ello funcionando sobre Microsoft Windows Server 2012 R2 [37] e Internet Information Services (IIS) 8.5 [38].

Sin embargo, en aras de lograr que la solución final pueda ejecutarse en plataformas muy diversas y de propósito más específico, se han utilizado asimismo un sistema NAS [39] basado en la plataforma Synology DSM [40] y un dispositivo reducido como es Raspberry Pi [41] (con el objetivo de simular el Internet de las Cosas).

La solución resultante estará orientada hacia la definición y ejecución de servicios [42], los cuales a su vez estarán basados en lo que se denominan microservicios, debido al estilo con el que éstos serán desarrollados.

## 1.4.- Delimitación del Alcance.

Es evidente que un mismo sistema no puede abarcar todas las posibilidades imaginables teniendo al mismo tiempo una gran sencillez de manejo. Por ello se ha optado por crear un sistema potente, flexible y eficaz pero siempre teniendo muy presente que debe ser sencillo de manejar, para que pueda tener una utilidad verdaderamente real.

Si bien se han tomado muchas decisiones de compromiso (véase apartado 3.2, donde se explican en profundidad el alcance y las limitaciones del sistema), resumimos a continuación las que, a nuestro juicio, son más relevantes:

- Sólo los usuarios con el rol de administrador del sistema podrán modelar aplicaciones. Este mecanismo está pensado para evitar que se produzca cambios no deseados en el modelo de una aplicación generada por parte de cualquier usuario consumidor; una posible mejora consiste en la definición de permisos individuales (*fine-grained permissions*) sobre la capacidad de modelado.
- Se permitirá para las aplicaciones definir su modelo de datos, no implementando ningún mecanismo para el modelo de procesos.
- Los lenguajes de las restricciones se basan en expresiones JavaScript y SQL que se incrustan directamente en la interfaz de usuario y la base de datos.

## 2.- ESTADO DE LA CUESTIÓN.

*Las situaciones difíciles se hacen más difíciles si se excluyen los pequeños pasos que pueden ayudar.*

Hugh Prather (1938-2010)

La complejidad de los sistemas intensivos en software aumenta constantemente hacia niveles sin precedente. Por otro lado, el número de dispositivos conectados entre sí y a través de Internet crece superando las previsiones más optimistas. Sin embargo, el número de profesionales dedicados a las Tecnologías de la Información y de las Comunicaciones no sigue el mismo ritmo. Esto hace cada vez más evidente la necesidad de aumentar en varios órdenes de magnitud la efectividad del trabajo a realizar.

A lo largo de las últimas décadas, se han propuesto numerosas metodologías para el diseño y desarrollo de sistemas basados en software. Cada nueva versión de los entornos de desarrollo integrados y herramientas de diseño asistido que se lanza al mercado realiza nuevas promesas en términos de mejora de productividad para sus usuarios.

Pero lo cierto es que para la construcción de aplicaciones se sigue necesitando personal técnico altamente cualificado, no observándose que los tiempos de desarrollo realmente disminuyan. En definitiva, la productividad es similar o incluso decrece [43] ante el aumento de la complejidad.

Además, la llegada del denominado “Internet de las Cosas” [44] está suponiendo un incremento considerable tanto en el número de proyectos como en la inversión económica que se realiza en los mismos [45]. En los últimos años este tipo de dispositivos ha saltado de la domótica a los procesos industriales, multiplicando sus apariciones y aplicaciones en diferentes contextos de la vida diaria [46].



¿Y cómo se interconectan y relacionan todos estos dispositivos y sistemas? A través de servicios distribuidos. Desde el comienzo de la computación distribuida, la especialización de los diferentes sistemas ha sido una constante, así como la evolución de los diferentes protocolos de comunicaciones. En estos momentos, la tendencia se dirige hacia servicios cada vez más especializados, incluso a costa de reducir drásticamente su tamaño; son los denominados “microservicios”.

Trataremos el estado de la cuestión en el que actualmente se encuentra la generación automática de soluciones basadas en microservicios, a través de los siguientes apartados:

- Principales antecedentes.
- Entornos tecnológicos disponibles.
- Características diferenciadoras de este Proyecto.

## **2.1.- Antecedentes.**

Se presentan a continuación las diferentes disquisiciones del estado de la ciencia de aquellas cuestiones sobre las que se asientan las necesidades que promueven la creación de un proyecto como este de Generación Automática de MicroServicios:

- Necesidad de automatización en el ejercicio profesional.
- Arquitecturas de software.
- Líneas de producto.
- Ingeniería del Dominio.
- Generación automática de código.

### **2.1.1.- Necesidad de automatización en el ejercicio profesional.**

Las Tecnologías de la Información y de las Comunicaciones (TIC), permiten realizar de manera automática tareas que anteriormente venían realizándose “a mano” por personal más o menos cualificado. Entre los principales avances está precisamente el tratamiento automatizado de grandes volúmenes de información [47].

Desde que se acuñara el término “Ingeniería del Software” (véanse [48] y [49]) a finales de la década de 1960, tanto el número de sistemas intensivos en software [50] como la complejidad de los mismos [51] se han visto incrementados de manera exponencial. Sin embargo, tanto el número de profesionales dedicados al desarrollo de dichos sistemas [52] como la calidad general de los mismos no ha seguido el mismo crecimiento.

Hoy en día la industria basa la construcción del desarrollo de software en un enfoque artesanal, centrado en la adquisición de conocimiento y experiencia por parte de los profesionales del sector a lo largo de su carrera profesional hasta llegar a un cierto grado de pericia y destreza (denominado comúnmente en el mundo empresarial mediante el anglicismo “know-how” [53]).

Sin embargo, el número de profesionales con un grado elevado de pericia es muy reducido ([54] y [55]), añadiendo a ello la dificultad, o clara imposibilidad, por parte de aquellos profesionales más noveles o menos duchos en aplicar las técnicas que utilizan los profesionales altamente eficientes [56].

Aunque la Ingeniería del Software es hoy la base fundamental de dichos conocimientos [57], gracias a la incorporación de nuevas ramas de saber como la Ingeniería de Dominio [58] y, sobre todo, a la incorporación de un nuevo repertorio de técnicas al alcance de los profesionales [59], empezamos a vislumbrar un futuro donde la productividad vendrá determinada por la utilización de herramientas que automaticen labores que al día de hoy siguen realizándose manualmente [60].

Una vez entendida la necesidad de automatizar el trabajo de desarrollo, se describirán las cualidades del sistema (o familia de sistemas) a construir a través de la arquitectura del mismo, de manera que podamos establecer pautas, guías y asunciones a la hora de construirlo, anticipando así las ventajas e inconvenientes de dichas decisiones.

### **2.1.2.- Arquitecturas de Software.**

Comenzaremos con varias definiciones formales ofrecidas por ISO para referirse a las arquitecturas de software [48] como *“el conjunto fundamental de conceptos o propiedades de un sistema en su entorno, incorporados sus elementos, relaciones y los principios de su diseño y evolución”*, refiriéndose a un sistema [49] como *“la combinación de elementos que interactúan entre ellos, organizados para lograr uno o más propósitos [...] creados por el hombre para ofrecer productos o servicios dentro de un entorno definido.”* y definiendo un servicio [61] como *“la ejecución de actividades, trabajo o responsabilidades”*.

#### 2.1.2.1.- Vistas de un sistema desde la perspectiva de la arquitectura.

Por tanto, podemos definir la arquitectura de un sistema como el conjunto de estructuras que lo definen. Desde un punto de vista de implementación, podemos ver que estas estructuras dividen al sistema en módulos que a su vez se descomponen en servicios compuestos asimismo por clases, procedimientos y demás artefactos.

Desde el punto de vista de la dinámica del sistema, estas estructuras definen la manera en que se relacionan los diferentes componentes entre sí, además de permitir relacionar el sistema con terceros sistemas en aquellos puntos definidos.

Asimismo, existe un tercer conjunto de estructuras, esta vez físicas, que permiten describir y establecer los entornos de ejecución de las estructuras lógicas definidas en las componentes estáticas y dinámicas del sistema. Desde esta perspectiva, es el hardware el que define las propiedades del diseño. Cuando además la solución pueda ejecutarse en plataformas diferentes, esta perspectiva física deberá cuidarse especialmente para procurar cumplir con todos los requisitos expresados en la definición del sistema.

#### 2.1.2.2.- Técnicas para la selección y definición de arquitecturas.

En [62] podemos encontrar recomendaciones acerca de qué debe incorporar una “buena” arquitectura, que resumimos a continuación. Desde el punto de vista del proceso, las recomendaciones dadas son las siguientes:

- Para lograr integridad conceptual y consistencia técnica, una arquitectura debería ser producto de una persona o de un reducido conjunto de personas.
- El equipo de arquitectura debe priorizar los atributos de la misma en base a una lista ordenada de requisitos de calidad bien definidos, informando de los compromisos adquiridos.
- Se debe documentar la arquitectura desde diferentes puntos de vista.
- Se debe evaluar la arquitectura para determinar su capacidad de entregar sistemas que cumplan con los requisitos dentro de los límites marcados por los atributos de calidad que se hayan definido.
- Toda arquitectura moderna debe permitir tanto una construcción incremental de sistemas (hacerlo crecer poco a poco, permitiendo incorporar funcionalidad a medida que transcurre el tiempo), como una construcción en paralelo (facilitando la colaboración de varios miembros o equipos que trabajen sobre diferentes aspectos del sistema a crear).

Desde la perspectiva de las estructuras en las que se descomponga una arquitectura, las recomendaciones dadas son las siguientes:



- La arquitectura debe definir módulos conocidos cuya funcionalidad sea asignada en base a los principios de ocultación de la información [63] y separación de responsabilidades [64].
- Los atributos de calidad deben lograrse mediante el uso de patrones y tácticas conocidas, específicas para cada uno de ellos.
- Ninguna arquitectura debería depender de una versión específica de un producto ni herramienta comerciales; de lo contrario, debe abstraerse para permitir una rápida y fácil evolución si éste cambiase.
- Los módulos productores de información deberían ser diferentes a los módulos que consuman dicha información (un ejemplo de arquitectura dirigida por este principio es la denominada arquitectura CQRS, *Command and Query Responsibility Segregation* [65]).
- No se debe esperar una correspondencia uno a uno entre los módulos (desde una perspectiva estática, los artefactos de código que construyen una funcionalidad) y los componentes (desde una perspectiva dinámica, el conjunto de procesos que ejecutan dicha funcionalidad). La concurrencia es precisamente la medida de dicha correspondencia.
- Los procesos ejecutables deben articularse de manera que no se exija una afinidad (dependencia) con la infraestructura que vaya a ejecutarlos.
- El número de posibles interacciones debe tanto sistematizarse como minimizarse; de esta manera, se logra: una mayor cohesión al realizar las mismas tareas de la misma forma a lo largo de todo el sistema, mejorar la comprensión por parte del equipo de desarrollo, minimizar el tiempo necesario para la construcción y posterior modificación, logrando con ello un incremento en la confianza global en la solución.

- La arquitectura debe ofrecer conjuntos bien definidos y documentados en todos aquellos aspectos que requieran la gestión de recursos en los que exista competencia por su uso, como puedan ser los accesos a la red o el tiempo de procesador, ofreciendo con ello guías de implementación y uso de las mejores prácticas existentes.

### 2.1.2.3.- Implicaciones de la arquitectura.

La importancia de la arquitectura radica en que servirá para definir el conjunto de estructuras sobre el que construir un sistema. Por ello, se hace imprescindible conocer los atributos de calidad y los requerimientos de dicho sistema, para escoger así la arquitectura que mejor encaje con dichas necesidades y objetivos.

Cuando se trate de un conjunto de sistemas intensivos en software, la arquitectura cobra una mayor dimensión, una mayor importancia. De hecho, una de las primeras tareas es obtener una clara visión de dicho conjunto de sistemas, en lo que se asemejan entre sí y en qué se diferencian. Hablaremos en estos casos de familias o líneas de producto.

### **2.1.3.- Líneas de producto.**

Una línea de productos software [66] es *“un conjunto de sistemas intensivos en software que comparten un conjunto gestionado de características comunes, las cuales satisfacen las necesidades específicas de un segmento particular de mercado o de una determinada misión, y por ello son desarrolladas partiendo de un conjunto común de artefactos mediante una guía prescriptora”*.

A finales de la década de 1960, los sistemas intensivos en software comienzan a ser objeto de estudio desde puntos de vista formales. Un momento histórico se produce cuando se celebra las primeras “Conferencias sobre Ingeniería del Software” bajo el auspicio de la OTAN, que reunió durante varios días de los años 1968 [67] y 1969 [68] a personalidades dedicadas al mundo de la computación venidas desde distintas ramas del saber, como las Matemáticas, la Ingeniería o la Educación. En dichas conferencias ya se denota la importancia de ofrecer a los programadores herramientas capaces de automatizar parte de su trabajo, denominando a las mismas “factorías de software”.

A mediados de la década de 1970 se comienzan a realizar estudios sobre las similitudes (*commonalities*, en terminología inglesa) y divergencias (*variabilities*) en conjuntos de programas. A estos conjuntos se los denominó familias de programas [69].

Por aquel entonces, ya se intuían los beneficios de identificar sistemas en los cuales sus propiedades comunes fuesen mayoritarias de manera que, aplazando el estudio de sus diferencias, se pudiesen lograr ventajas competitivas en términos de reducción de esfuerzo así como reducción de la complejidad global de la familia. Estas reducciones, en definitiva, son las que permiten la disminución de costes económicos, humanos y materiales, así como la disminución de los plazos necesarios para lanzar un producto al mercado.

#### 2.1.3.1.- Economías de alcance y economías de escala.

Cuando se compara el proceso de desarrollo de los sistemas intensivos en software, se observa que son los esfuerzos de desarrollo inicial (abarcando el análisis, diseño, construcción y pruebas) los que suponen la práctica totalidad de los costes; el coste de replicación es prácticamente nulo. Diremos en este caso que nos encontramos ante un modelo de economía de alcance [70].

Por el contrario, en la industria tradicional, la fase de producción, donde se fabrica el producto, es sin duda la más costosa. Diremos en este caso que nos encontramos ante un modelo de economía de escala [70].

Mientras que los modelos de economía de escala se centran en producir muchas unidades iguales para reducir los costes, los modelos de economía de alcance se centran en elaborar conjuntamente varios productos aprovechando las similitudes existentes entre los mismos, es decir, aprovechando al máximo las similitudes y controlando las posibles variaciones.

#### 2.1.3.2.- *Producto ejemplar.*

Tras el estudio de las líneas de producto queda claro que, en el caso de los sistemas intensivos en software, el principal objetivo es reducir el esfuerzo del proceso de desarrollo inicial y, ya construido este primer producto, posteriormente realizar las variaciones mínimas para ir obteniendo el resto de productos de la familia.

Es por ello que dicho primer producto cobra una importancia radical, tanto por la selección del mismo (en caso de disponer de diferentes opciones), como por la manera en la que sea desarrollado; es decir, debidamente descompuesto entre lo que se entenderá como factores comunes de la familia y las posibles variaciones (o divergencias) de este primer producto respecto del resto.

A este primer producto, que podrá ser utilizado como base (o plantilla) a partir de la cual construir el resto de la línea de productos, se le denominará “ejemplar” o “arquetipo”.

Cabe en este punto mencionar la existencia de técnicas específicas, como por ejemplo EDD [71], para la variación del ejemplar a la hora de crear el resto de productos de la familia.

### 2.1.3.3.- Avances.

A raíz del enfoque propuesto por el desarrollo de familias de productos software, surgen nuevas disciplinas. Por un lado, aquéllas orientadas a formalizar las metodologías de análisis y diseño, culminando en la Ingeniería de Dominio [58]. Por otro, aquéllas orientadas a aplicar nuevas metodologías de desarrollo, así como aplicando técnicas de programación e incorporando nuevas herramientas en los procesos de desarrollo software, destacando la programación generativa [72] y los lenguajes específicos de dominio (*domain-specific language*, DSL) [73].

Los siguientes dos apartados explicarán estos avances con mayor profundidad.

### **2.1.4.- Ingeniería del Dominio.**

Ya en la década del 2000, los conceptos de variabilidad y familias de productos son ampliados gracias a un estudio más formal del dominio del problema, separado del dominio de la solución.

Surge así como disciplina independiente la Ingeniería de Dominio, la cual engloba técnicas orientadas a estudiar el problema desde sus orígenes a través de la Ingeniería de Requisitos y su posterior modelado a nivel conceptual a través del diseño dirigido por modelos; de la conjunción de ambos se benefician las familias de productos añadiendo, como explicamos anteriormente, metodologías y técnicas de desarrollo específicas.

#### 2.1.4.1.- Ingeniería de Requisitos.

La Ingeniería de Requisitos [74], reconocida como una de las prácticas habituales en Ingeniería del Software, tiene por objetivo “*recomendar aproximaciones para la especificación de requisitos de software que produzcan una documentación completa y sin ambigüedad*”.

Esta disciplina consta de numerosas técnicas, tanto para abordar a los interlocutores (ya sean, por ejemplo, usuarios, clientes, expertos en la material o terceros consultores) [75], como para documentar tanto textual [76] como gráficamente [77] los requisitos educidos.

En cuanto a su plasmación más concreta, el estándar UML [78] orientado a la creación de diagramas visuales (normalmente realizados en las fases de análisis y diseño de un proyecto software) que sirvan de base para la posterior construcción, incluye el diagrama de casos de uso [79] como su principal herramienta para esta fase de un proyecto.

En los últimos años y dado el auge de las metodologías ágiles [80] (principalmente Scrum [81]), las técnicas aplicadas para la educación de requisitos incluyen épicas e historias de usuario [82], mediante las cuales se describe el comportamiento del sistema a medida que los usuarios interaccionan con él.

#### 2.1.4.2.- *Análisis del dominio orientado a características (técnica FODA).*

Una vez recogidos los requisitos, tanto funcionales como no funcionales, la siguiente fase en el desarrollo de un sistema intensivo en software siguiendo un método tradicional como es el “desarrollo en cascada” (*waterfall development lifecycle* [83]) es el denominado “análisis funcional” (véase el anexo 8, el cual representa el análisis funcional del presente Trabajo).

Por su importancia, así como por el uso tan extendido ligado a las familias de productos, destacamos entre las diferentes técnicas que ofrece la Ingeniería de Dominio, aquella denominada “análisis del dominio orientado a sus características” [84] (más conocida por sus siglas en inglés, FODA, *Feature-Oriented Domain Analysis*).

El análisis FODA ofrece soporte para la definición y clasificación, a niveles tanto arquitectónico como funcional, de los denominados “productos de dominio”.

Dichos productos se obtienen a partir del análisis de dominio, aplicando una sucesión de “refinamientos” y obteniendo, en definitiva, una familia de productos sobre la que trabajar. Es decir, cada uno de estos productos se distingue por determinados “factores” (elementos comunes y variaciones existentes en la familia), los cuales son modelizados mediante el siguiente conjunto de operaciones:

- Agregación: operación por la que se abstraen conjuntos de factores en un único factor de nivel superior y, por tanto, pasan a ser ignorados individualmente a partir de dicho nivel.
- Descomposición: operación inversa a la agregación, mediante la cual un factor superior es dividido (descompuesto) en diferentes factores de nivel inferior, cada uno de ellos representando elementos comunes o determinadas variaciones.
- Generalización: operación mediante la cual aquellos factores comunes a la familia son sustituidos por un nuevo factor de nivel superior.
- Especialización: operación inversa a la generalización, donde un factor común de nivel superior es sustituido por uno o varios factores de nivel inferior que refinan (particularizan en mayor detalle) al anterior.
- Parametrización: operación por la que un factor, que representa a un componente parametrizable de la familia, es adaptado para un determinado producto, a través de la personalización de determinados parámetros variables que forman parte de dicho factor.

Podemos de esta manera definir un producto de la familia como el conjunto de factores, de entre todos los disponibles, que son utilizados para construir dicho producto.

Gráficamente, FODA permite representar la familia mediante un “árbol de características” y al producto como un subconjunto determinado de factores. La fig. 2.1 (extraída de [85]) muestra un ejemplo de la notación gráfica utilizada por FODA.

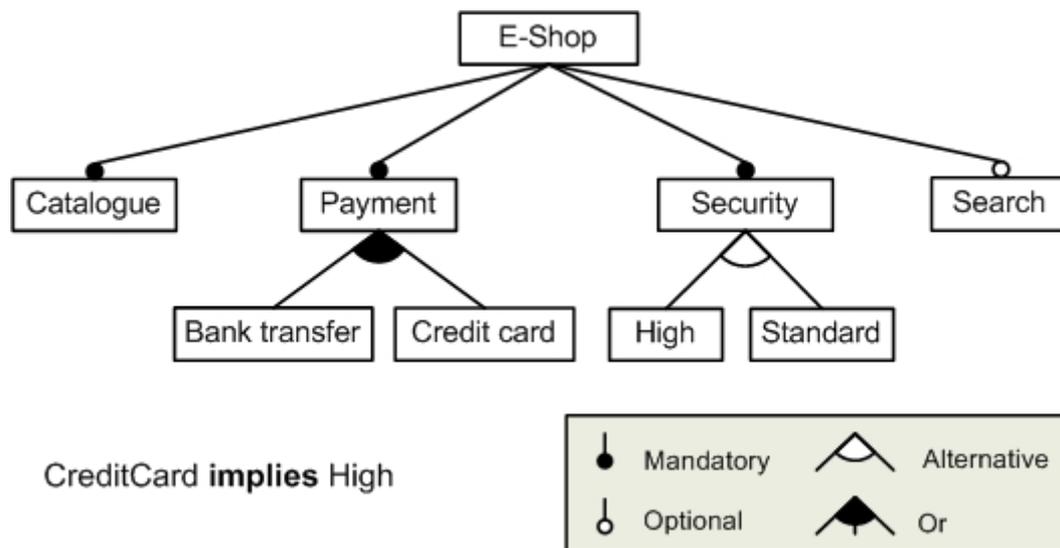


Fig. 2.1.- Árbol de características en un análisis FODA (fuente: [85]).

Obsérvese que mediante el análisis FODA es posible determinar, para cada factor característico:

- La obligatoriedad (u opcionalidad) de que todos los productos cuenten con dicho factor.
- La posibilidad de que, al contar un producto con un determinado factor, dicho producto deba escoger (contener) obligatoriamente entre una (y sólo una) de las diferentes alternativas (factores de nivel inferior) disponibles (obtenidos ya sea mediante descomposición o mediante especialización).
- La posibilidad de que, al contar un producto con un determinado factor, dicho producto pueda contener a varios (incluso todos o ninguno, si fuesen opcionales) de sus factores de nivel inferior (nuevamente fruto de descomposiciones o especializaciones).

Por último, tal y como se representa en la fig. 2.1, es posible añadir al árbol de características determinadas “restricciones”, normalmente expresadas mediante lógica de predicados; por ejemplo, que determinados factores obliguen a incluir (o no) a otros.

### 2.1.4.3.- Diseño dirigido por modelos.

El testigo de la educación de requisitos y análisis funcional se recoge en la fase de diseño. Mientras que los conceptos objeto de estudio durante el análisis comprenden el universo del discurso en los términos en los que el usuario interpreta la información (es decir, el dominio del problema), durante la fase de diseño los términos empleados se corresponden al dominio de la solución.

El diseño dirigido por modelos (*model-driven design*, MDD) es una técnica que debe englobarse dentro de una disciplina mayor, la Ingeniería Dirigida por Modelos (*model-driven engineering*, MDE) [10], que se caracteriza por:

- Uso de modelos para representar los principales aspectos (o aspectos de especial relevancia) de un sistema intensivo en software (o una familia de ellos).
- Representación de los modelos mediante un lenguaje específico de dominio (DSL, *domain specific language* [73]).
- Dichos DSL se definen separando su notación (cómo se dibujan sus diagramas o cómo se escriben sus textos) de su sintaxis abstracta (esta última, representada mediante un “meta-modelo”).
- La automatización lograda se basa en el uso de transformaciones, partiendo del modelo original, obteniendo un conjunto de artefactos resultantes (normalmente basados en texto, que pueden ser ficheros de código fuente en un lenguaje de programación o archivos de configuración en formato XML [16], JSON [15] o YAML [86]).

Uno de los principales hitos del MDD fue la publicación de la iniciativa de arquitecturas dirigidas por modelos (*model-driven architecture*, MDA) [33] auspiciada por la misma organización que desarrolla, entre otros, el estándar UML.

MDA se basa a su vez en la especificación “Meta-Object Facility” (MOF) [87], la cual define en primer lugar un “meta-modelo” sobre el que construir lenguajes específicos de dominio, entre los que se incluyen el propio UML [29] o MDA, entre otros. Así pues, gracias a este “modelo de modelos” o “meta-modelo”, es posible definir de una manera sistemática lenguajes específicos para un dominio concreto de soluciones.

Los modelos pueden contribuir con un papel fundamental en el desarrollo de sistemas intensivos en software:

- Ayudando a razonar sobre dichos sistemas antes de que deban ser construidos; por ejemplo, en aras de una validación previa, gracias a mecanismos de simulación.
- Facilitando la comunicación entre los diferentes participantes de un proyecto, como usuarios, desarrolladores o expertos del dominio, por mencionar algunos.
- Como mecanismo de documentación, a cualquiera de los posibles niveles: arquitectónico, analítico, de diseño e incluso de desarrollo.
- Como artefacto fuente (origen) del que, mediante transformaciones, obtener artefactos de un nivel conceptual menor (como código fuente en algún lenguaje de programación); en este caso, el beneficio se deriva de la automatización del proceso de obtención de dichos productos (como la reducción de tiempos de desarrollo o la sistematización de los mismos).

Por tanto, parece cada vez más evidente la relación, gracias a los modelos, que puede establecerse entre los productos de la fase de diseño y los productos de la fase de desarrollo. Trataremos a continuación de técnicas específicas que sirven de puente entre los productos de ambas fases.

### **2.1.5.- Generación automática de código.**

Una vez abordadas las fases de educación de requisitos, análisis funcional y diseño técnico, el proyecto o producto debe disponer de toda la documentación (y objetivos claros) necesarios para abordar la fase de construcción y pruebas del mismo.

Es en esta fase de construcción en la que se escribirá el código fuente del sistema a desarrollar, debiendo validarse su correcto funcionamiento en base a las especificaciones y expectativas de todos los implicados.

#### *2.1.5.1.- Desarrollo dirigido por el modelo.*

Tras estudiar cuestiones como la Ingeniería de Dominio y las Líneas de Producto en los apartados anteriores, el paso lógico es integrar dichos modelos, junto con los factores comunes y las características de variabilidad detectadas en el sistema a construir.

El desarrollo dirigido por modelos (model-driven development, MDD) es la disciplina encargada del estudio de las diferentes técnicas de desarrollo software, en la que la fase de construcción se ve dirigida, de manera total o parcialmente automática, por los modelos diseñados en la fase anterior.

Dichos modelos suelen construirse con lenguajes específicos de dominio y estar basados en líneas de producto.

Entre el conjunto de técnicas, destacan las herramientas de programación automática y programación generativa [72].

### **2.1.6.- Arquitecturas Orientadas a Servicios.**

La década de 1990 se caracterizó por la popularización de la *World Wide Web* [88] y el uso de protocolos basados en la pila TCP/IP (véase [88] para conocer los principales estándares) y al paradigma de invocación de procedimientos remotos (*Remote Procedure Call*, RPC; véanse [90], [91] y [92]).

Ante la disparidad de los protocolos propietarios, la década del 2000 promovió la interoperabilidad entre diferentes sistemas y fabricantes, principalmente a través de los estándares HTTP (*HyperText Transfer Protocol* [93]), SOAP (*Simple Object Access Protocol* [94]) y WSDL (*Web Services Description Language* [95]), dando así forma a lo que hoy conocemos como Arquitecturas Orientadas a Servicios (*Service Oriented Architectures*, SOA [96]).

Ante la complejidad que alcanzaron dichos estándares, así como para mejorar el rendimiento que dicho tipo de arquitecturas eran capaces de ofrecer, surge un nuevo estilo denominado REST (*Representational State Transfer* [17]), el cual hace uso de mecanismos más modernos y de estándares abiertos, no controlados por ningún fabricante en concreto. REST se caracteriza por utilizar los verbos de HTTP para obtener y modificación la información (denominada “recursos”) en lugar de utilizar protocolos y mensajes adicionales, como ocurre con SOAP y WSDL.

## **2.2.- Entornos Tecnológicos Disponibles.**

Una vez analizados, a nivel funcional, los elementos relacionados con los requisitos de la solución a desarrollar, nos centraremos en las herramientas tecnológicas que existen en la actualidad, gracias a las cuales es posible construir el sistema informático que cubra los requerimientos solicitados.



Partiendo del objetivo de construir un sistema informático accesible a través de Internet, las tecnologías Web se presentan como la solución más plausible para abordar proyectos de esta índole.

Comentaremos a continuación varias tecnologías que pueden ser las de uso más extendido en la actualidad gracias a la versatilidad, fácil manejo y potencia que ofrecen sus herramientas:

- PHP, desarrollada mediante código abierto de libre disposición, es la tecnología escogida para el desarrollo de la solución GAMS del presente Trabajo Fin de Máster.
- Java, desarrollada originalmente por Sun Microsystems, hoy propiedad de Oracle.
- .NET Framework, desarrollada por Microsoft.
- Otras tecnologías comúnmente empleadas para el desarrollo en la *World Wide Web*.

### **2.2.1.- PHP**

El origen de PHP (*PHP Hypertext Preprocessor*, anteriormente denominado *Personal Home Page Tools* [18]) se remonta al año 1994 [97]; creado originalmente por Rasmus Lerdorf como una herramienta de *scripting* para crear páginas HTML dinámicas [98] en “el lado del servidor” que facilitó enormemente el desarrollo frente a las técnicas anteriormente existentes basadas en CGI (*Common Gateway Interface* [100]) y SSI (*Server-Side Includes* [100]) existentes, siendo muy similar a las tecnologías ASP (*Microsoft’s ActiveX Server Pages* [101]) y JSP (*Java Server Pages* [102]) disponibles en entornos profesionales.



Desde hace tiempo el desarrollo de esta plataforma ha sido gestionado por un equipo de profesionales [103], los cuales han mantenido el espíritu original, ofreciendo una solución de software de libre distribución y basado en código abierto [104].

Hoy en día, PHP es, con diferencia, la plataforma más utilizada para el desarrollo de sitios y aplicaciones Web (véanse [105] y [106]).

La solución GAMS desarrollada como parte del presente Trabajo Fin de Máster utiliza la tecnología PHP en su versión 7, pues de la misma cabe destacar varias características que la convierten en idónea para la tarea:

- PHP puede ejecutarse en prácticamente cualquier dispositivo hardware, por limitado que resulte, estando disponible en prácticamente cualquier sistema operativo.
- PHP se ofrece como una solución de código abierto y libre distribución.
- PHP está orientado al desarrollo de aplicaciones y servicios Web dinámicos, facilitando la creación de HTML y el acceso a bases de datos.
- Aunque PHP en sí no puede considerarse un estándar, sí permite emplear y ofrecer todos los estándares *de facto* propios de la Web.
- El lenguaje de programación PHP, a partir de su versión 5, es un lenguaje dinámico débilmente tipado que permite aplicar la mayoría de las técnicas de programación orientada a objetos.
- El código fuente de PHP esencialmente se interpreta, por lo que basta con crear o modificar un fichero de código fuente para desplegar nueva funcionalidad, sin necesidad de ejecutar un ciclo de compilación, ensamblaje y despliegue en el servidor.

### **2.2.2.- Java.**

El entorno Java [107] nació de la mano de Sun Microsystems [108], siendo hoy propiedad de Oracle [109], con el objetivo de crear un sistema capaz de comunicarse con todos los elementos eléctricos y electrónicos de un hogar. Sin embargo, cuando la industria de la televisión por cable abandonó esa idea, fue el navegador NetScape Navigator quien acogió a Java y su capacidad de crear ricas interfaces para la World Wide Web.

Rápidamente, el ecosistema Java fue creciendo hasta convertirse en el sistema de desarrollo actualmente más utilizado por las grandes multinacionales, gracias a su capacidad de ejecutarse en múltiples plataformas de fabricantes diferentes (IBM AIX [110], HP UX [111], Sun Solaris [112], Red Hat Enterprise Linux [112], Oracle Linux [114] y Microsoft Windows [25], entre otros).

El entorno actual está basado en una máquina virtual que interpreta el lenguaje máquina en el sistema operativo sobre el que se ejecutan las aplicaciones [102]. Dicho entorno de ejecución es conocido como *Java Runtime Environment (JRE)*, cuyo entorno de desarrollo es el *Java Development Kit (JDK)*.

Mediante Java es posible crear aplicaciones para interfaces gráficas de usuario (GUI) [27], aplicaciones y servicios Web [115], así como aplicaciones para dispositivos móviles, basadas en la plataforma *Java 2 Micro Edition (J2ME)* [116].

Existen numerosos entornos de desarrollo para la plataforma Java, si bien los más extendidos están basados en código abierto (como el caso de ECLIPSE [117]) o en código de libre distribución (como el caso de NetBeans [118]). Existen sin embargo algunos entornos propietarios caracterizados por la gran cantidad de herramientas que los complementan y que los hacen atractivos a los desarrolladores aún a pesar de su coste económico, siendo el más conocido IntelliJ IDEA [119] de la empresa JetBrains [120].

En abril de 2009 [121] se hizo pública la adquisición de la empresa Sun Microsystems por parte del gigante informático Oracle [109], pasando la plataforma Java a formar parte de la familia de productos de esta segunda compañía.

### **2.2.3.- Microsoft .NET Framework.**

La plataforma Microsoft .NET Framework [122] es el pilar sobre el que construir aplicaciones que permitan una rica experiencia por parte de los usuarios, garanticen comunicaciones sencillas y seguras, a la par que dispongan de herramientas para modelar los procesos de negocio de las empresas. Es la plataforma desarrollada por el mayor fabricante de software del mundo y está perfectamente integrada en el conjunto de sistemas operativos más extendido del planeta: Microsoft Windows.

Gracias a un modelo de programación coherente y muy completo, así como a sus innumerables API's [123], la plataforma .NET Framework ayuda a construir las aplicaciones que permitan realizar cuantas tareas sean necesarias, permitiendo escoger de manera indiferente entre varios lenguajes de programación, múltiples servicios y dispositivos electrónicos.

La plataforma .NET permite a los desarrolladores y profesionales de sistemas contar con una tecnología de desarrollo software robusta y segura, añadiendo a sus soluciones herramientas de gestión, así como actualizaciones que facilitan la construcción, depuración e implantación de software altamente confiable. Todo esto, sin olvidar su integración con las herramientas estándares del sistema operativo.

Esta plataforma admite multitud de lenguajes de programación gracias al *Common Language Runtime* (CLR) [124], módulo que provee el soporte necesario para lenguajes estáticos clásicos como Visual Basic, C# ó C++, así como el advenimiento del novedoso *Dynamic Language Runtime* (DLR) [125] y los lenguajes dinámicos como Managed JScript, IronRuby o IronPython.

.NET Framework ofrece soluciones pioneras que permiten un rápido desarrollo de aplicaciones y resulta en un incremento de la productividad. ASP.NET es la familia de tecnologías que permiten a los desarrolladores Web crear desde páginas personales hasta extensísimos portales empresariales compuestos por complejas aplicaciones dinámicas. Gracias al uso de diferentes API, como AJAX (Asynchronous JavaScript and XML) [126], MVC (Model-View-Controller) [127] o WebAPI [128], se hace posible desarrollar rápidamente experiencias basadas en Web mucho más eficientes, interactivas y altamente personalizadas, las cuales son capaces de funcionar en los navegadores más extendidos.

La herramienta más popular para el desarrollo de aplicaciones basadas en .NET es sin duda Microsoft Visual Studio [129]; esta herramienta es un completo entorno de desarrollo integrado (IDE) que permite escribir y depurar código, mantener y acceder a bases de datos, ejecutar aplicaciones e incluso gestionar el ciclo de vida de las mismas.

#### **2.2.4.- Otras tecnologías comúnmente empleadas para el desarrollo en la *World Wide Web*.**

##### 2.2.4.1.- *Orígenes de las aplicaciones web.*

Originalmente [88] la *World Wide Web* se concibió para publicar documentos estáticos y compartirlos a través de las redes universitarias, científicas y militares de la por aquel entonces incipiente Internet y sus protocolos TCP/IP [88] y HTTP [93]. Una de sus grandes novedades fue la posibilidad de navegar entre diferentes documentos gracias a los enlaces de hipertexto incrustados como parte del contenido, quedando definida así la primera versión de HTML [12] (pueden conocerse más detalles de dicha historia en [130]).

Si bien los documentos HTML iniciales eran elaborados por personas, rápidamente se demostró que dicho contenido estático no permitía explotar todas las posibilidades que la Web ofrecía. Por ello se concibieron mecanismos para ofrecer contenidos de manera dinámica [98], es decir, contenidos generados por el servidor en función del usuario que se conectaba o la información a la que accedía. Nacían así las primeras aplicaciones Web.

Las aplicaciones Web modernas cuentan con arquitecturas basadas en múltiples capas [131] y, dependiendo del tamaño del equipo de desarrollo, permiten una clara especialización entre la interfaz gráfica de usuario [27], la interfaz entre máquinas [28], la lógica de negocio y el acceso a datos.

#### 2.2.4.2.- El lado del cliente.

Así se han ido creando herramientas de desarrollo y entornos de programación específicamente diseñados para optimizar la construcción de aplicaciones y servicios basados en Web, los cuales hacen uso de tecnologías “del lado del cliente” como páginas HTML [12], código JavaScript [14], hojas de estilo CSS [13] y esquemas de datos basados en JSON [15] o XML [16], así como plantillas XSLT [99] (*extensible style sheet language transformations*).

Debido a la cada vez mayor exigencia por parte de los usuarios, la complejidad técnica en el lado del cliente aumenta a un ritmo vertiginoso. En los últimos años, el número de *frameworks* de desarrollo (basados en JavaScript, *lingua franca* para animar aplicaciones web) no deja de aumentar. Una búsqueda rápida por las principales comparativas entre *frameworks* eleva la cifra por encima de la veintena, y únicamente considerando los más populares.

Aunque muchas son las opciones, no queremos dejar de mencionar JQuery (véanse [132] y [133]), pues hace ya más de una década que supuso un cambio revolucionario en la manera de afrontar el desarrollo de aplicaciones web interactivas.

Igualmente, en cuando a las hojas de estilo en cascada se refiere, la incorporación de *frameworks* como Bootstrap [134] o Foundation [135] supuso igualmente un cambio revolucionario en el diseño de sitios y aplicaciones web, sobre todo a partir de la incorporación de las tables y teléfonos móviles como principales dispositivos de acceso a la web. Esta tendencia se afianzó en lo que hoy se conoce como “diseño web adaptativo” (*responsive web design*) [136]. En el apartado 10.3.2 puede el lector consultar cómo se aplica esta técnica a la solución GAMS desarrollada en el presente Trabajo.

#### 2.2.4.3.- El lado del servidor.

Para “el lado del servidor”, se han desarrollado herramientas, lenguajes y entornos específicos, entre los que cabe destacar, además de los mencionados PHP [18], JSP [102], ASP [101], ASP.NET [122], otros asimismo utilizados como:

- Node.js para el uso de JavaScript en el lado del servidor (véanse [137] y [138]).
- Perl (véanse [139], [140] y [141]).
- Python (véanse [142] y [143] para conocer el lenguaje; véase además [144] acerca de las diferentes opciones para desarrollar aplicaciones web, sobresaliendo Django [145] como una de las más utilizadas por su destacado sistema de plantillas para construir interfaces web, facilitando también la creación de APIs basados en REST [146]).
- Ruby on Rails (que unifica un moderno lenguaje de programación dinámico como es Ruby [147], con un *framework* de desarrollo ágil como es Rails [148]).

#### 2.2.4.4.- Mezclando nuevamente ambos lados.

En los últimos años, además, están apareciendo múltiples *frameworks* que vuelven a mezclar las tecnologías “del lado del cliente” con las “del lado del servidor”, prometiendo una mayor interactividad de las aplicaciones para con el usuario. Destacan en este sentido Angular [149] de Google [150] y React [151] de Facebook [152].

Mencionar igualmente el auge de soluciones en “el lado del cliente” que sin embargo requieren herramientas en “el lado del servidor”. Esto se debe a la cada vez mayor complejidad de estos *frameworks*. Para ello, muchas soluciones hacen uso de lenguajes web de más alto nivel y una serie de procesos encargados de transformar (técnicamente no se puede hablar de compilación) los recursos fuente (utilizados durante las etapas de diseño y desarrollo) en recursos objeto (que serán publicados en el servidor web) a los que efectivamente accederán los clientes. Ejemplos de este tipo de *frameworks* son: VueJS [153], EmberJS [154] y ExtJS [155].

En cuanto a los lenguajes web de más alto nivel, cabe distinguir dos grandes grupos:

- Aquellos que producen JavaScript, ofreciendo un lenguaje con características más robustas (como la verificación estática de tipos de datos, encapsulación o modularidad, por mencionar algunas). Destacar en esta categoría a Dart [156], CoffeeScript [157] y TypeScript [158].
- Aquellos que producen CSS, los cuales enriquecen tanto sintáctica como semánticamente las hojas de estilo, facilitando además el trabajo con las mismas. En esta categoría encontramos a SASS [159] y LESS [160].

Destacar, entre otras muchas, las tecnologías utilizadas para la ejecución de los mencionados procesos de transformación: Gulp [161] (para la ejecución de tareas definidas por el programador) y Webpack [162] (para empaquetar múltiples ficheros, por ejemplo, generando un único archivo JavaScript minimizado, a partir de varios módulos dependientes entre sí). En [163] pueden encontrarse diversas técnicas para aprovechar conjuntamente algunas de estas herramientas.

#### 2.2.4.5.- Tendencias arquitectónicas.

Por último, mencionar dos estilos arquitectónicos de software que comienzan a aparecer en el mercado, aunque dado que en estos momentos están enfocados a la creación de interfaces gráficas de usuario y no a la comunicación entre servicios, no se profundizará en ellos, remitiendo al lector a la correspondiente bibliografía:

- Aplicaciones de página única: (SPA, *single page application* [164]), en las que la aplicación consta de una única página HTML cuyo contenido es completamente construido y alterado haciendo uso de JavaScript, quedando el lenguaje HTML relegado a una mera plantilla.
- Aplicaciones web progresivas (PWA, *progressive web applications* [165]): en las que, de manera coordinada con el explorador web, éstas adquieren capacidades propias de las aplicaciones nativas y acceso a recursos locales que una aplicación web tradicional no tiene permitido.

### 2.3.- Características Diferenciadoras de este Trabajo.

La solución de **Generación Automática de MicroServicios** (GAMS) pretende cubrir el hueco actualmente existente de programas ofreciendo las siguientes ventajas fundamentales:

- GAMS puede ejecutarse en prácticamente cualquier dispositivo hardware, por limitado que resulte, así como estar disponible en prácticamente cualquier ordenador y servidor, con independencia de su sistema operativo.
- GAMS está totalmente orientado a la construcción por parte del usuario final de aplicaciones y servicios Web, automatizando la creación de la infraestructura tecnológica, así como el acceso a bases de datos y la intercomunicación entre servicios.



- GAMS permite emplear y ofrecer todos los estándares *de facto* propios de la Web.

Gracias a estos aspectos, todo ello unido a la posibilidad de que varias personas trabajen simultáneamente desde ubicaciones diferentes, por ser una aplicación Web multiusuario, el sistema de **Generación Automática de MicroServicios** es una herramienta útil desde el primer momento.

### 3.- PLANTEAMIENTO E HIPÓTESIS.

*No hay enigmas. Si un problema puede plantearse, también puede resolverse.*

Ludwig Wittgenstein (1889-1951)

Uno de los objetivos de todo Trabajo Fin de Máster es la realización de un proyecto académico que demuestre que el alumno autor del mismo, a través de un desarrollo original, ha obtenido los conocimientos y competencias necesarias para el ulterior ejercicio profesional, incluyendo la actividad investigadora.

Precisamente con este objetivo se ha planteado el desarrollo de este Trabajo Fin de Máster, partiendo de una supuesta relación comercial entre una compañía de software actuando como proveedor (representada por el alumno) y una organización actuando como cliente (representada por la directora).

Con esta ficción se van a representar las fases habituales en el ciclo de vida de un proyecto de construcción software, desde la concepción y planificación del mismo hasta la implantación final de la solución.

Para el desarrollo del Proyecto **Generación Automática de MicroServicios**, como parte del planteamiento y de las hipótesis a considerar, trataremos los siguientes temas:

- Objetivos a cubrir.
- Delimitaciones del alcance.
- Plan de proyecto.

### 3.1.- Objetivos a Cubrir.

El principal objetivo de la solución de **Generación Automática de MicroServicios** es la automatización en la creación y mantenimiento de sistemas intensivos en software, basados en arquitecturas orientada a servicios, que sigan la filosofía de los microservicios.

La solución se construye teniendo presentes tanto las enormes divergencias existentes en las capacidades de cómputo de los diferentes dispositivos de uso cotidiano (como son dispositivos embarcados, propios del *Internet de las Cosas* [44], los dispositivos móviles como teléfonos y tabletas, los ordenadores personales y los servidores de alto rendimiento -HPC, *High Performance Computing* [166]-), como la excesiva complejidad de las arquitecturas de servicios distribuidos, sin olvidar las limitaciones de los actuales entornos de desarrollo que podemos encontrar en el mercado (incluyendo proyectos de software libre [2], productos propietarios, iniciativas institucionales, así como estándares internacionales). Por todas estas razones, se pretende ofrecer las siguientes ventajas:

- GAMS puede ejecutarse en prácticamente cualquier dispositivo hardware, por limitado que resulte, así como estar disponible en prácticamente cualquier ordenador y servidor, con independencia de su sistema operativo.
- GAMS está totalmente orientado a la construcción por parte del usuario final de aplicaciones y servicios Web, automatizando la creación de la infraestructura tecnológica, así como el acceso a bases de datos y la intercomunicación entre servicios.
- GAMS permite emplear y ofrecer todos los estándares *de facto* propios de la Web.

Además, como parte de los requerimientos inicialmente planteados, se pretende la construcción de un sistema software flexible y ampliable, que permita aumentar su vigencia dando cobertura a los nuevos requisitos que surjan con el paso del tiempo, sin necesidad de realizar mantenimientos evolutivos sobre el mismo; es decir, sin necesidad de efectuar cambios que requieran la programación de nuevos módulos o adiciones en los ya existentes.

Los objetivos a cubrir por la solución de **Generación Automática de MicroServicios** son los siguientes:

- Disponer de una solución que facilite la creación y mantenimiento de aplicaciones orientadas a servicios, basadas en una arquitectura con el estilo propio de los microservicios.
- Los microservicios deben poder desarrollarse sin excesivos conocimientos de programación, utilizando mecanismos al alcance del usuario final (véase [11]).
- Los microservicios deben poder explotarse por parte del usuario final, dotando a la solución de una interfaz gráfica de usuario (GUI, *Graphical User Interface* [27]) con capacidades similares a la interfaz entre máquinas (M2M, *Machine-To-Machine* [28]).
- Debe hacerse uso de cuantos estándares existan, tanto técnicos como de documentación, en lo que al tratamiento y transmisión de la información se refiera.

Las principales motivaciones que impulsan la creación de este Proyecto son las siguientes:

- Ventajas de disponer de una herramienta que facilite la creación y mantenimiento de microservicios, ahorrando la mayor parte del tiempo dedicado al desarrollo “manual” tradicional.

- Los servicios web resultantes deben estar basados en un estilo arquitectónico propio de los microservicios.
- La solución final debe poder ejecutarse en plataformas muy diversas, en cuanto a capacidad de cómputo.

Para ello, se planteará un acceso universal vía Web, basado en estándares *de facto* de la Web, tanto a nivel de interfaz gráfica de usuario (HTML [12], CSS [13] y JavaScript [14]) como a nivel de interfaz entre servicios (JSON [15] y REST [17]).

### **3.2.- Delimitaciones del Alcance.**

Como es evidente que un mismo sistema no puede abarcar todas las posibilidades imaginables teniendo al mismo tiempo una gran sencillez de manejo, se ha optado por crear un sistema potente, flexible y eficaz, teniendo siempre muy presente que debe ser sencillo de manejar, de cara a ofrecer una utilidad verdaderamente real.

Si bien se han tomado muchas decisiones de compromiso, indicamos a continuación las que, a nuestro juicio, son más relevantes:

- Sólo los usuarios con el rol de administrador del sistema podrán modelar aplicaciones. Este mecanismo está pensado para evitar que se produzca cambios no deseados en el modelo de una aplicación generada por parte de cualquier usuario consumidor; una posible mejora consiste en la definición de permisos individuales (*fine-grained permissions*) sobre la capacidad de modelado.
- Se permitirá para las aplicaciones definir su modelo de datos, no implementando ningún mecanismo para el modelo de procesos.
- Los lenguajes de las restricciones se basan en expresiones JavaScript y SQL que se incrustan directamente en la interfaz de usuario y la base de datos.

### 3.3.- Plan de Proyecto.

Como parte de todo desarrollo informático, estamos definiendo en el presente Trabajo un plan de proyecto con el objetivo de establecer los mecanismos de gestión que permitan obtener un producto software de calidad, controlado, en el que queden claramente identificados todos los participantes del mismo, así como los productos obtenidos en cada una de las diferentes etapas del desarrollo.

Para ello, se comentarán a continuación los principales puntos de los que consta este Plan:

- **Participantes:** se describen a continuación los roles que participarán en el proyecto, así como las personas encargadas de asumir los compromisos de dichos roles.
- **Etapas:** se enumeran las etapas de las que constará el ciclo de vida que será seguido para el desarrollo del proyecto.
- **Productos resultantes:** se describirán todos los productos a obtener como parte del desarrollo software.

#### 3.3.1.- Participantes.

En la realización del sistema GAMS participan los siguientes roles:

- **Cliente:** es la persona que encarga la realización del proyecto y se responsabiliza de que cumpla los requisitos propios y de los usuarios.
- **Usuario:** es la persona que utilizará el sistema resultante. Cuando existan diferentes tipos de usuarios, estos accederán al sistema utilizando diferentes roles (en este Proyecto se han definido dos roles: administrador y consumidor).

- **Jefe de proyecto:** es la persona responsable de que el proyecto se lleve a cabo en los términos inicialmente previstos, tanto en tiempo como en costes. A su vez, es el responsable de tomar las medidas correctoras necesarias cuando se produzcan desviaciones en la planificación.
- **Analista:** es la persona responsable de educir los requisitos funcionales y no funcionales de la aplicación a través de su trabajo junto al cliente y al usuario. Realiza el diseño técnico de la aplicación y se comunicará con el jefe de proyecto y el equipo de programadores durante la fase de construcción del software. Una vez construido el sistema, forma a los usuarios en el manejo de la aplicación y realizará las debidas pruebas funcionales junto a éstos.
- **Programador:** es la persona encargada de codificar en un programa de ordenador la solución informática deseada, haciendo para ello uso del análisis funcional y el diseño técnico previamente especificados. Está igualmente encargado de realizar (parte de) las pruebas definidas para validar la corrección del sistema.

La matriz de responsabilidad nos muestra, para cada una de las personas que participan de algún modo durante el ciclo de vida del sistema de **Generación Automática de MicroServicios**, el rol que desempeñan:

| Participante<br>(relación con el Proyecto) | Rol     |         |                  |          |             |
|--|---------|---------|------------------|----------|-------------|
|  | Cliente | Usuario | Jefe de proyecto | Analista | Programador |
| Elena Ruiz Larrocha (directora)            | X       | X       |                  |          |             |
| Luis María Cámara Rossi (autor)            |         |         | X                | X        | X           |

### 3.3.2.- Etapas.

Para el desarrollo del Proyecto, se ha decidido seguir la metodología basada en el Proceso Unificado de Desarrollo (RUP, *Rational Unified Process*) [30], definido por la empresa Rational y convertido hoy en uno de los principales referentes en el desarrollo orientado a objetos, como es el que nos ocupa.

En esta metodología, se propugna un desarrollo incremental mediante el cual la aplicación será enfocada de manera global pero construida mediante sucesivas iteraciones que irán refinando la solución. Una de las frases más conocidas que ha acuñado esta sistemática es: “*analiza un poco, diseña un poco, programa un poco*”.

Las etapas de las que consta el desarrollo son:

- **Educción de requisitos:** en la cual se obtendrán los requisitos funcionales (qué hacer) y no funcionales (cómo y con qué hacerlo).
- **Análisis funcional:** en la que se analizarán los requisitos de cara a obtener los casos de uso que permitan describir el funcionamiento a desarrollar. Se basa en el dominio del problema, es decir, en términos inteligibles para el usuario.
- **Diseño técnico:** etapa donde se modelizará el dominio de la solución en términos informáticos; es decir, se planteará un sistema que dé cobertura a los requisitos del usuario.
- **Construcción y pruebas:** en la que se codificará la solución planteada y se verificará que cumple con el análisis funcional y el diseño técnico.
- **Implantación:** etapa en la cual el sistema resultante será instalado y se realizará una configuración inicial (parametrización) del mismo.

Se presenta a continuación la estimación inicial del esfuerzo, medido en jornadas, necesario para elaborar el presente Proyecto:

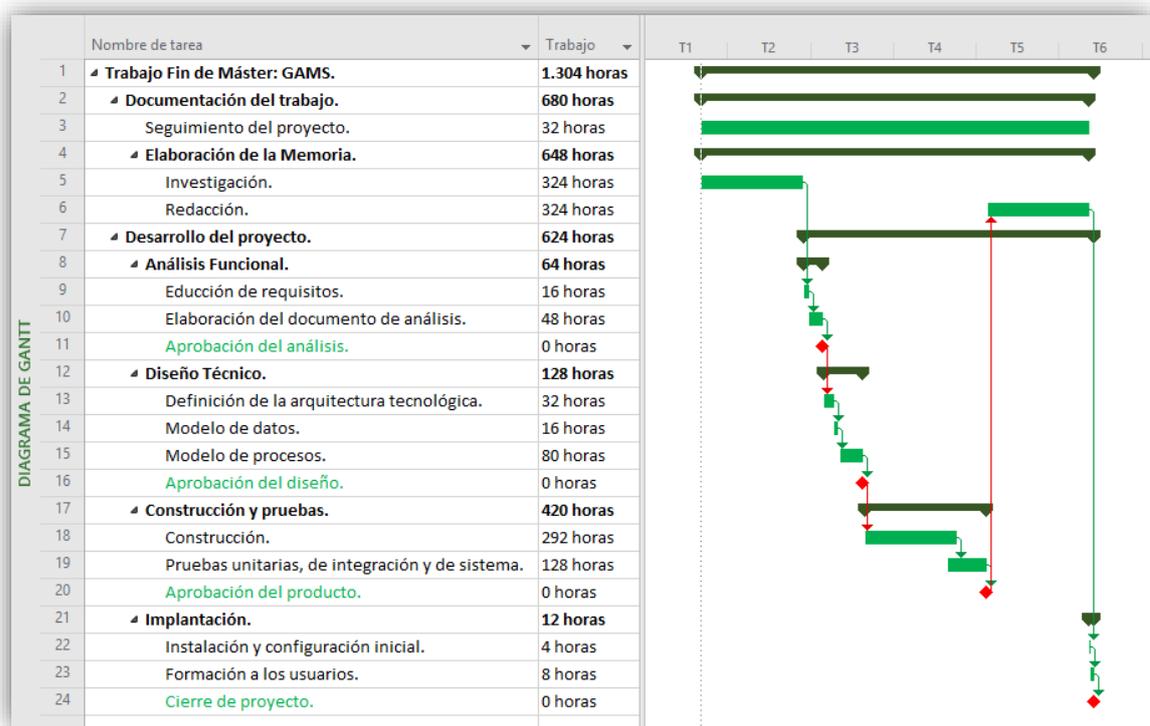


Fig. 3.1.- Estimación inicial; planificación del esfuerzo.

Como puede observarse, se ha estimado un total de 1.304 horas de trabajo. Aparecen reflejados en verde los hitos de finalización de cada una de las sucesivas fases en las que se descompone el proyecto.

La tarea de documentación debe entenderse desde la doble perspectiva de, por un lado, gestionar el seguimiento de la realización del Trabajo Fin de Máster y, por otro, elaborar la presente documentación. Por esta razón su duración viene determinada por el ciclo de vida completo del desarrollo del proyecto.

### 3.3.3.- Productos resultantes.

Como parte del desarrollo del sistema informático, se espera obtener de cada una de las etapas una serie de productos que, finalmente, no sólo permitan implantar la solución final sino documentar el proceso seguido para construir dicha solución.

El objetivo de esta documentación es permitir reproducir dichos pasos a la par que facilitar los mantenimientos que puedan derivarse de la implantación de la solución, los cuales suelen clasificarse en correctivos (que corrigen deficiencias detectadas en el normal funcionamiento del sistema) o evolutivos (que añaden nueva funcionalidad).

Los productos resultantes del proyecto son:

- **Análisis funcional:** contiene los requisitos del usuario, así como una descripción basada en el dominio del problema de la solución a desarrollar. Para una fácil comprensión, se realizará un análisis funcional basado en casos de uso, mediante los cuales se explican los posibles procesos que puede llevar a cabo un usuario para lograr una tarea particular.
- **Diseño técnico:** contiene la modelización de la solución en términos puramente informáticos. Consta de un modelo de datos, donde se describe cómo se realizará la persistencia de la información tratada mediante la aplicación, así como de un modelo de procesos, en el que se describen las operaciones asociadas a los casos de uso.
- **Producto software:** es el sistema resultante de la construcción del proyecto.
- **Manuales y documentación de usuario:** debido a las posibilidades que ofrecen las aplicaciones Web, en este proyecto la documentación irá embebida en la propia aplicación.

## 4.- RESOLUCIÓN.

*El verdadero progreso es el que pone la tecnología al alcance de todos.*

Henry Ford (1844-1929)

Se presenta a continuación un resumen de los productos obtenidos durante el desarrollo de la solución software construida para dar cobertura a los requisitos comentados en el presente Proyecto.

Con el objetivo de sacar fuera del cuerpo principal de la Memoria material voluminoso, de origen eminentemente técnico, de cara a no romper el hilo conductor del presente documento (sin dejar incompleto el material resultante), se ha dispuesto el Anexo A para la documentación exhaustiva relacionada con el Análisis Funcional (apartado 4.2), el Anexo B para la documentación relacionada con el Diseño Técnico (apartado 4.3), así como el Anexo C para el Manual de Usuario.

Se aborda a continuación la resolución del software haciendo especial hincapié en aspectos de Ingeniería del Software que deberían formar parte de cualquier desarrollo profesional. Así, siguiendo los productos comentados anteriormente (véase apartado 3.3.3), trataremos:

- **Metodología de desarrollo software**, empleada para la construcción de la solución informática.
- **Análisis Funcional**, donde se exponen los requisitos cubiertos, así como la problemática que debe abordar la solución informática.
- **Diseño Técnico**, donde se exponen los modelos en términos informáticos, la información que maneja el software (modelo de datos) así como los procedimientos que intervienen en dicha manipulación (modelo de procesos).
- **Construcción**, detallando la arquitectura software implantada y las pruebas realizadas sobre el sistema desarrollado.

- **Implantación y puesta en producción**, indicando la configuración (parametrización) realizada inicialmente con el objetivo de permitir utilizar el sistema nada más instalarlo.

#### **4.1.- Metodología de Desarrollo Software Empleada.**

Como se ha comentado anteriormente (véanse apartados 1.3.3 y 3.3.2), se utiliza el ciclo de vida basado en el Proceso Unificado de Rational (RUP) [30].

Gracias al modelo iterativo que se describe en dicha metodología, es posible abordar el proyecto completo mediante refinamientos sucesivos para permitir desarrollar el producto resultante sin necesidad de abordarlo en su totalidad sino por módulos que, posteriormente, serán integrados.

En cada una de estas iteraciones se realizará un ciclo de vida completo del módulo, es decir, se analizará, diseñará y codificará la funcionalidad de dicho módulo. Una vez que todos los módulos han sido desarrollados, se aborda una última fase de integración, con el objetivo de unir dichos módulos para obtener el resultado final completo.

La fig. 4.1 presenta un gráfico que muestra el esfuerzo medio en cada una de las fases de las que constan las sucesivas iteraciones. Este gráfico fue realizado por el equipo de Rational que definió el Proceso Unificado y estudió sus beneficios [167]:

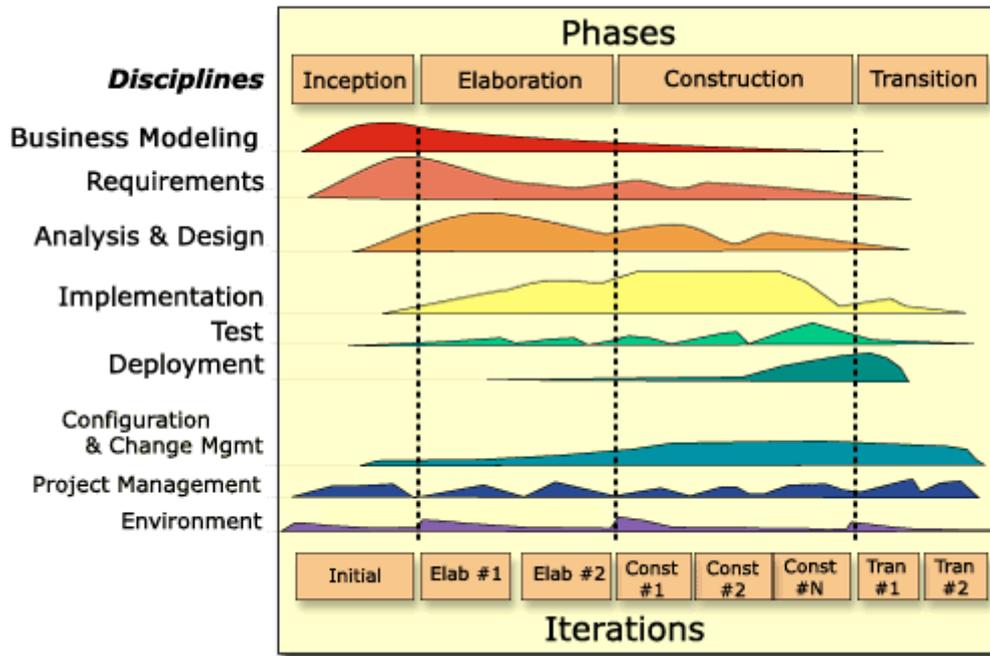


Fig. 4.1.- Ciclo de vida del "Proceso Unificado de Rational" (RUP).

## 4.2.- Análisis Funcional.

El objetivo del Análisis Funcional es predecir el comportamiento del sistema sin necesidad de modelar ni conocer los detalles técnicos de su implementación concreta.

Durante la fase de Análisis Funcional, se establecen una serie de reuniones tanto con el cliente como con el usuario de cara a determinar los requisitos de la solución a elaborar. Asimismo, se realizan una serie de casos de uso para determinar el comportamiento del sistema desde un punto de vista funcional, es decir, desde el punto de vista del usuario del sistema.

Con el objetivo de sacar fuera del cuerpo principal de la Memoria todo el material voluminoso de origen eminentemente técnico, se ha dispuesto el Anexo A para la documentación exhaustiva relacionada con el Análisis Funcional. Este Anexo A cubre en profundidad todos los productos introducidos en el presente punto, a saber:

- **Requisitos educidos:** los requisitos son el conjunto de necesidades que debe cubrir el sistema y por las cuales se aborda la construcción del mismo. El proceso de educación de requisitos consiste en la gestión de los mismos (obtención, clasificación, validación y seguimiento a lo largo del proyecto).
- **Diagrama conceptual:** mediante el diagrama conceptual se representan de manera gráfica y concisa todas las entidades (conceptos) que intervienen en el sistema, así como las relaciones existentes entre cada una de ellas.
- **Diagrama de casos de uso:** es el mecanismo para representar gráficamente el conjunto de funcionalidades que ofrece el sistema a desarrollar; refleja igualmente los actores (usuarios del sistema, externos al mismo, ya sean humanos u otros programas) que intervienen en cada unidad funcional (caso de uso).
- **Casos de uso extendidos:** donde se describen uno a uno los diferentes casos de uso que componen la funcionalidad ofrecida por el sistema.

### 4.3.- Diseño Técnico.

El objetivo del Diseño Técnico es modelar el sistema informático, fijando los detalles técnicos de su implementación concreta. Mientras que el Análisis Funcional describe el “qué” de la cuestión en el dominio del problema, el Diseño Técnico describe el “cómo” de la misma en el dominio de la solución.

Durante la fase de Diseño Técnico, los analistas definen la arquitectura tecnológica en la que se basará la solución a desarrollar (sistemas operativos, gestores de bases de datos, entornos y *frameworks* de desarrollo, entre otros) así como los detalles concretos de cómo abordar la construcción de dicha solución (interfaces de usuario, interfaces con otros sistemas, lógica de negocio y acceso a datos).

Ahora, los casos de uso se modelan mediante procesos que implementan la funcionalidad concreta y mediante datos asociados a dichos procesos que fluyen a lo largo del programa.

Con el objetivo de sacar fuera del cuerpo principal de la Memoria todo el material voluminoso de origen eminentemente técnico, se ha dispuesto el Anexo B para la documentación exhaustiva relacionada con el Diseño Técnico. Este Anexo B cubre en profundidad todos los productos introducidos en el presente punto, a saber:

- **Arquitectura tecnológica de la solución:** define el conjunto de tecnologías que darán soporte al sistema informático a desarrollar; entre otros, se establece el sistema operativo sobre el que funcionará la solución, el gestor de base de datos que albergará la información, así como los entornos de desarrollo.
- **Modelo de datos:** establece cómo se organizará la información en los diferentes almacenes de datos (por ejemplo, bases de datos y ficheros).
- **Modelo de procesos:** cataloga todas las interfaces y operaciones con las que cuenta la solución para ofrecer la funcionalidad requerida.

#### **4.4.- Construcción y Pruebas del Producto Software.**

El objetivo de la fase de Construcción y Pruebas es desarrollar el sistema software que servirá de solución al problema planteado. Para ello se hará uso de los productos obtenidos en las fases previas de Análisis Funcional y Diseño Técnico.

Para garantizar la correcta realización de la solución, así como que los requisitos han sido cubiertos, se realizarán una serie de pruebas tanto técnicas (denominadas “pruebas unitarias”, “pruebas de integración” y “pruebas de sistema”) como funcionales (denominadas “pruebas de usuario”).

Gracias al “modelo de datos”, definido durante el Diseño Técnico, se construyen las bases de datos (o ficheros de datos) que albergarán la información de la aplicación, así como las estructuras de información que fluirán por el código del programa. Con el “modelo de procesos” se establece la navegación a través de la interfaz, así como las operaciones concretas que serán ejecutadas durante el funcionamiento de la aplicación resultante. Todo ello junto constituirá el sistema finalmente desarrollado.

Describiremos brevemente los procedimientos seguidos en los siguientes apartados:

- **Entorno de desarrollo:** donde se enumerarán las herramientas, *frameworks* y entornos de desarrollo integrados (IDE [34]) utilizados para la construcción de la solución.
- **Bibliotecas de terceros:** componentes de software [168] desarrollados por terceras personas o compañías, incorporados a la solución de cara a reutilizar la funcionalidad que ofrecen, sin necesidad de volverla a desarrollar.
- **Construcción:** donde se explicarán los pasos seguidos para el desarrollo de la solución informática.
- **Pruebas realizadas:** donde se detallarán las pruebas realizadas para garantizar el correcto funcionamiento del sistema, así como la validez del mismo.

#### **4.4.1.- Entorno de desarrollo.**

La solución planteada en el presente Trabajo Final de Máster supone un verdadero reto profesional debido a la miríada de tecnologías utilizadas y combinadas entre sí, todas ellas punteras en sus respectivos campos de aplicación.

Por ello, se enumeran a continuación todas las herramientas, utilidades, *frameworks* de desarrollo, así como los diferentes entornos de desarrollo integrados (IDE) que se han utilizado para la construcción de la solución final, describiendo brevemente para cada una de ellas la finalidad con la que han sido utilizadas durante las etapas de construcción y pruebas.

Debido al elevado número de tecnologías involucradas, en la medida de lo posible, se ha optado por las versiones más estables y compatibles de cada una de ellas, de manera que la solución final pueda ser considerada lo suficientemente robusta como para representar a un sistema comercial operando en entornos de producción de ámbito empresarial.

- Lenguajes de programación: HTML 5, CSS 3, JavaScript 1.5, PHP 7.1, SASS, Ruby 2.2.6.
- Bases de datos: SQLite 3, MySQL 5/MariaDB 10, Microsoft SQL Server 2014.
- Frameworks de desarrollo: Symfony 3, Composer, Twig 2.
- Herramientas y entornos de desarrollo integrados: MINGW64 (GNU bash 4.4.19), Git for Windows 2.18, Compass 1.0.1, phpDocumentor 2

Se describen a continuación, brevemente, todas y cada una de ellas.

#### *4.4.1.1.- MINGW64 (GNU bash 4.4.19)*

Entorno de herramientas propias de sistemas operativos UNIX [169] y LINUX, capaces de ser ejecutadas desde entornos Microsoft Windows. Necesario para la ejecución de Git en entornos Windows.

```
Administrator@VW2012R2D2 MINGW64 /
$ git --version
git version 2.18.0.windows.1

Administrator@VW2012R2D2 MINGW64 /
$ help
GNU bash, version 4.4.19(2)-release (x86_64-pc-msys)
These shell commands are defined internally. Type `help' to see this list.
Type `help name' to find out more about the function `name'.
Use `info bash' to find out more about the shell in general.
Use `man -k' or `info' to find out more about commands not in this list.
```

Fig. 4.2.- Consola de mandatos del entorno MINGW64 (GNU bash 4.4.19).

#### 4.4.1.2.- Git for Windows 2.18

Git para Windows [170] ofrece de manera nativa en el sistema operativo Windows un conjunto de herramientas que hacen disponible el sistema de gestión de código fuente Git: emulador de la interfaz de mandatos (*shell*) BASH, herramienta gráfica que permite comparar visualmente versiones de código fuente y ejecutar los mandatos propios de Git, así como una integración con la herramienta Explorador de Windows.



Fig. 4.3.- Logotipo de Git for Windows.

#### 4.4.1.3.- PHP 7

Para el desarrollo de la solución GAMS, se ha utilizado la versión 7 del lenguaje de programación PHP. En el apartado 2.2.1 (página 27), se analiza con mayor profundidad esta tecnología, muy popular para el desarrollo web, tanto de sitios como de aplicaciones y servicios.



Fig. 4.4.- Logotipo de PHP.

#### 4.4.1.4.- HTML 5

HTML 5 (*HyperText Mark-Up Language*, versión 5, véanse [12] y [171]) es la especificación actual del lenguaje de etiquetas utilizado para construir páginas web.



Fig. 4.5.- Logotipo de HTML5.

#### 4.4.1.5.- CSS 3

CSS 3 (*Cascading Style Sheets*, versión 3, véase en [13] la especificación más reciente y en [172] el conjunto de módulos de los que consta actualmente) es el lenguaje para definir cómo deberán representarse los elementos HTML, por ejemplo, visualmente en pantalla o impresos en papel.



Fig. 4.6.-  
Logotipo de  
CSS3.

#### 4.4.1.6.- SASS 3.5.1

El lenguaje SASS [159] se utiliza para definir hojas de estilo en cascada avanzadas, capaces de ser interpretadas por múltiples navegadores, de manera más fácil y sencilla que utilizando directamente CSSv3.



Fig. 4.7.-  
Logotipo de  
SASS.

#### 4.4.1.7.- Compass 1.0.3

COMPASS [173] es una herramienta de línea de mandatos basada en el lenguaje RUBY que automatiza la traducción de hojas de estilo escritas en lenguaje SASS, ampliando además dicho lenguaje con multitud de funciones de tipo *macro*, produciendo ficheros con sintaxis CSS interpretables por los principales navegadores web.



Fig. 4.8.- Logotipo de COMPASS.

```
Administrator: Start Command Prompt with Ruby
C:\Users\Administrator>gem install compass
Fetching: sass-3.4.25.gem (100%)
Successfully installed sass-3.4.25
Fetching: multi_json-1.12.1.gem (100%)
Successfully installed multi_json-1.12.1
Fetching: compass-core-1.0.3.gem (100%)
Successfully installed compass-core-1.0.3
Fetching: compass-import-once-1.0.5.gem (100%)
Successfully installed compass-import-once-1.0.5
Fetching: chunky_png-1.3.8.gem (100%)
Successfully installed chunky_png-1.3.8
Fetching: rb-fsevent-0.10.2.gem (100%)
Successfully installed rb-fsevent-0.10.2
Fetching: ffi-1.9.18-x64-mingw32.gem (100%)
Successfully installed ffi-1.9.18-x64-mingw32
Fetching: rb-inotify-0.9.10.gem (100%)
Successfully installed rb-inotify-0.9.10
Fetching: compass-1.0.3.gem (100%)
  Compass is charityware. If you love it, please donate on our behalf at http://umdf.org/compass Thanks!
Successfully installed compass-1.0.3
Parsing documentation for sass-3.4.25
Installing ri documentation for sass-3.4.25
Parsing documentation for multi_json-1.12.1
Installing ri documentation for multi_json-1.12.1
Parsing documentation for compass-core-1.0.3
Installing ri documentation for compass-core-1.0.3
Parsing documentation for compass-import-once-1.0.5
Installing ri documentation for compass-import-once-1.0.5
Parsing documentation for chunky_png-1.3.8
Installing ri documentation for chunky_png-1.3.8
Parsing documentation for rb-fsevent-0.10.2
Installing ri documentation for rb-fsevent-0.10.2
Parsing documentation for ffi-1.9.18-x64-mingw32
Installing ri documentation for ffi-1.9.18-x64-mingw32
Parsing documentation for rb-inotify-0.9.10
Installing ri documentation for rb-inotify-0.9.10
Parsing documentation for compass-1.0.3
Installing ri documentation for compass-1.0.3
Done installing documentation for sass, multi_json, compass-core, compass-import-once, chunky_png, rb-fsevent, ffi, rb-inotify, compass after 18 seconds
9 gems installed

C:\Users\Administrator>sass -v
Sass 3.4.25 (Selective Steve)

C:\Users\Administrator>
```

Fig. 4.9.- Línea de mandatos para la instalación de SASS y Compass desde RUBY GEMS.

#### 4.4.1.8.- RUBY 2.2.6

El entorno de programación Ruby [147] se ha utilizado como parte del entorno de desarrollo por ser necesario para ejecutar la herramienta COMPASS asociada al lenguaje SASS para definir hojas de estilo avanzadas.



Fig. 4.10.- Logotipo de RUBY.

De entre las opciones disponibles, se ha optado por la versión 2.2.6 de RubyInstaller [174] para Windows.



Fig. 4.11.- Instalación del entorno Ruby para Windows.

#### 4.4.1.9.- *JavaScript (ECMAScript 5)*

El lenguaje de programación JavaScript es el principal lenguaje de programación utilizado para desarrollar páginas web dinámicas. Es un lenguaje de *script* interpretado, dinámico, con orientación a objetos y que permite funciones de orden superior [175].

Aunque inicialmente se desarrolló como parte del navegador Netscape [176], actualmente todos los navegadores modernos implementan al menos la versión 5 del estándar ECMAScript, que fue redactada y publicada en estándar internacional ECMA-262 [14] por la Asociación Europea de Fabricantes de Computadores (*European Computer Manufacturers Association*, ECMA [177]), la cual se encarga además de mantener varias especificaciones relacionadas, como son:

- ECMA-402 [178]: API de internalización, que describe los objetos que ofrecen el soporte para la adaptación lingüística y cultural de aplicaciones basadas en ECMAScript.
- ECMA-404 [179]: sintaxis para el formato JSON [15] de intercambio de datos, aprobado además como estándar ISO/IEC 21778 [180] e inicialmente definido a través de varias RFC (4627, 7158, 7159 y 8259; véase [181]) del IETF [182].
- ECMA-414 [183]: documento resumen que muestra la relación entre los estándares anteriormente mencionados.

#### 4.4.1.10.- *SQLite 3*

SQLite [184] es una biblioteca compartida (típicamente, un fichero \*.DLL en Windows o un fichero con formato SO en UNIX/LINUX) que implementa un motor de bases de datos que se ejecuta dentro del proceso del ejecutable principal que la invoque (típicamente, un programa \*.EXE en Windows o un ELF en UNIX/LINUX).



Fig. 4.12.- Logotipo de SQLite.

SQLite se ha convertido en el sistema gestor de bases de datos de mayor uso a nivel mundial [185], principalmente gracias a su utilización en dispositivos embarcados y de limitada capacidad, como teléfonos móviles y tabletas.

SQLite se distingue particularmente de otras soluciones debido a:

- Es de dominio público [186] y se distribuye como código fuente de libre distribución [2].
- Cada versión es cuidadosamente verificada, logrando una cobertura de código [187] del 100%, lo que le ha labrado reputación en el sector como una de las soluciones más robustas existentes.

#### 4.4.1.11.- *SQLite Studio 3.1.1*

Para la gestión de las bases de datos creadas mediante SQLite, en el entorno de programación se hace uso de SQLite Studio [188], la cual es una sencilla pero potente herramienta gráfica orientada tanto al modelado de datos como al manejo de los mismos, facilitando las tareas de importación y exportación de información y permitiendo trabajar simultáneamente en un entorno gráfico (Windows, MacOS o LINUX) como orientado a texto (SQL).



Fig. 4.13.- Logotipo de SQLite Studio.

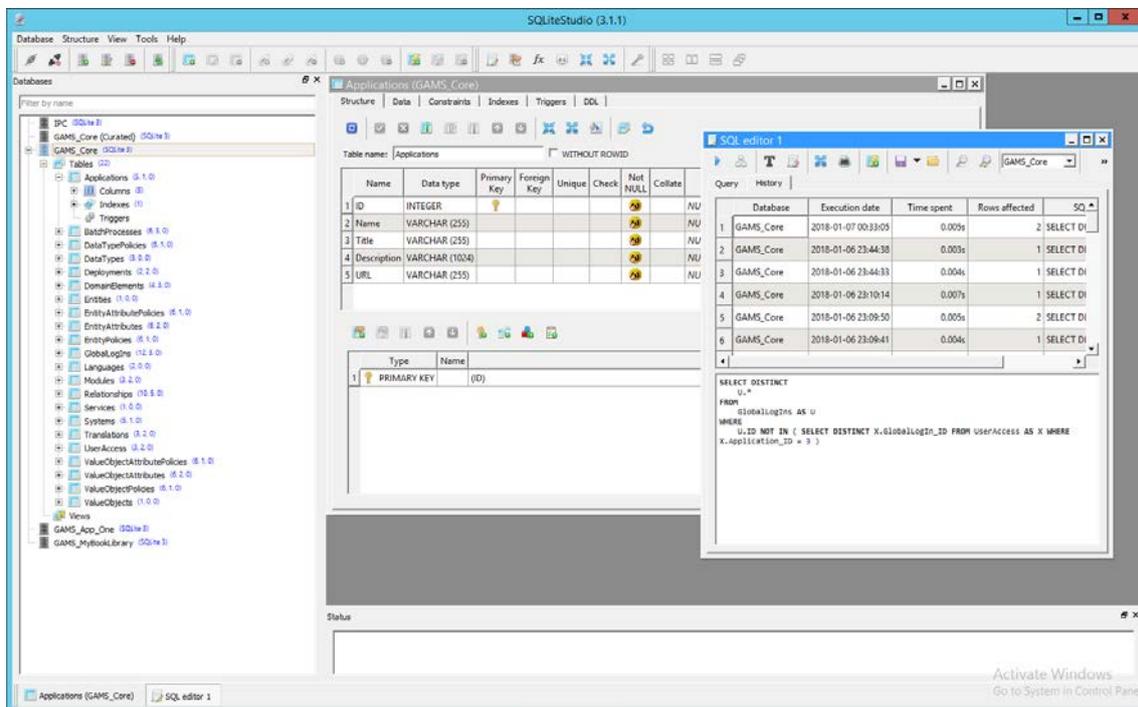


Fig. 4.14.- Base de datos del modelo de GAMS, abierta con SQLite Studio.

Si bien esta herramienta anteriormente se ofrecía como de libre distribución, recientemente ha sido publicada en GitHub [189], por lo que ahora también es posible obtener el código fuente completo de la misma.

#### 4.4.1.12.- MySQL 5 / MariaDB 10

Sistema gestor de bases de datos. MySQL [22] se popularizó hace más de una década hasta su adquisición por parte de Oracle. En ese momento, y ante las dudas que causó el modelo de licenciamiento de su nuevo propietario, parte del equipo original de desarrollo creó a partir del código fuente de MySQL un nuevo sistema gestor, totalmente compatible, al que denominó MariaDB [23].

#### 4.4.1.13.- Symfony 3.4.12

Symfony es el marco de desarrollo software elegido para la construcción de la solución. En [<https://symfony.com/doc/3.4/index.html>] se puede obtener documentación completa acerca de este marco de trabajo. Cabe destacar además [[https://symfony.com/pdf/Symfony\\_book\\_master.pdf](https://symfony.com/pdf/Symfony_book_master.pdf)] por ser una completa guía para entender las diferentes decisiones de diseño, la arquitectura propuesta, así como para comenzar a trabajar con Symfony de una manera productiva.

#### **4.4.2.- Bibliotecas de terceros**

Se enumeran a continuación aquellas bibliotecas de terceros que por su utilidad han sido incorporadas a la solución construida. Todos estos componentes de software [168], siguiendo con los objetivos del presente Trabajo Fin de Máster (véase el apartado 1.3.1 en la página 5) son ofrecidos libremente por terceras personas o compañías.

- Composer 1.6.5 [190]: gestor de dependencias para aplicaciones PHP.
- Twig 2: biblioteca PHP orientada a la creación de plantillas HTML.
- Doctrine ORM 2.5: para el acceso a base de datos utilizando técnicas de programación a objetos.
- FOSRestBundle 2.3: para facilitar la publicación de servicios web siguiendo el patrón REST.

- FOSUserBundle 2.0: para facilitar la gestión de usuarios en las aplicaciones Symfony.
- Swift Mailer Bundle 3.0: facilita el envío de correos electrónicos y es utilizado por FOSUserBundle para el recordatorio de contraseñas.
- SAMI: herramienta de generación automática de documentación, a partir de los comentarios de código en sintaxis PHPDOC (véase [191]).

#### **4.4.3.- Construcción.**

La fase de construcción se lleva a cabo siguiendo las mejores prácticas de la industria y recomendaciones por parte de los diferentes fabricantes y desarrolladores de las herramientas descritas en los puntos anteriores 4.4.1 y 4.4.2.

En cuanto al diseño y construcción de la base de datos, siguiendo el modelo entidad/relación, así como el modelo relacional, se ha desarrollado según se especificó en el Diseño Técnico. Nos remitimos a [192] y [193] donde pueden encontrarse las técnicas más comunes para el diseño y desarrollo de bases de datos.

La interfaz de usuario ha seguido las directrices de usabilidad [194] y eficiencia [98], permitiendo no sólo un desarrollo ágil sino, especialmente, que el usuario se sienta cómodo e identifique rápidamente todas las capacidades del sistema, reduciendo todo lo posible la curva de aprendizaje y aumentando su eficacia en el uso de la aplicación desarrollada.

El Anexo C contiene el Manual de Usuario donde puede observarse el estado final de toda la funcionalidad ofrecida por la aplicación, así como una explicación de las diferentes opciones que se presentan al usuario de la misma.

#### **4.4.4.- Pruebas realizadas.**

Las pruebas realizadas han sido guiadas por las recomendaciones asentadas a través de la Ingeniería del Software [195] y los criterios establecidos por los principales modelos de Gestión de Calidad Software [196], como puedan ser CMMI [197] o MÉTRICA [198].

Para la realización de pruebas automáticas, se han desarrollado como parte de la solución pruebas automatizadas unitarias y de sistema. Gracias al entorno de desarrollo, es posible probar de manera automática y exhaustiva todas las operaciones realizadas por las diferentes capas lógicas (lógica de negocio y acceso a datos).

Para complementar estas pruebas automáticas, así como para comprobar la corrección de la capa de interfaz Web de usuario, se han utilizado procedimientos manuales de verificación, basados en un plan de pruebas. Este plan de pruebas permite determinar la calidad de todos los procesos catalogados durante la fase de Diseño Técnico (véase apartado 9.4.2), indicando para cada proceso si la tarea (manual) se ha realizado exitosamente o no.

Una vez que todas las tareas asociadas a los procesos tuvieron éxito, se consideró concluida la etapa de construcción.

### **4.5.- Implantación y Puesta en Producción.**

Una vez concluidas las labores de construcción del sistema software, se procede a la creación de los procesos de instalación del mismo, así como a una configuración inicial (comúnmente denominada “parametrización”).

#### **4.5.1.- Instalación.**

Para la instalación del sistema software resultante, tarea que resulta compleja de acometer, se ha procedido a la creación de un manual específico para cada plataforma de referencia, el cual puede consultarse en el apartado 10.2.

En este punto es importante resaltar, especialmente para la migración de la aplicación de una máquina a otra (por ejemplo, durante operaciones de mantenimiento), que la base de datos deberá tratarse aparte (generalmente, de manera manual), pues el procedimiento de instalación descrito contempla únicamente una fase inicial de puesta en producción.

Mencionar que, en cualquier caso, es necesario disponer de los requisitos de la arquitectura para que la aplicación funcione correctamente. Véase el apartado 9.1 para más información.

#### **4.5.2.- Configuración inicial (parametrización).**

La configuración inicial (parametrización de la aplicación) constará de los siguientes elementos:

- Un usuario con el rol de administrador, “Administrador”, que permitirá mantener inicialmente la aplicación y con el que, por ejemplo, se puede realizar la gestión de las aplicaciones basadas en microservicios.
- Un usuario con el rol de consumidor, “UsuarioPrueba”, al que se permitirá acceder a las aplicaciones de prueba, con el objetivo de facilitarle la gestión de la información sin permitir que realice cambios sobre los esquemas de datos.
- Una aplicación sencilla de prueba, “Agenda Personal”, que consta de los principales elementos (tipos de datos, entidades, atributos y políticas).



- Una aplicación de prueba más compleja, “Gestión Bibliotecaria”, que constará de todos los elementos posibles (módulos, tipos de datos, entidades, objetos valor, políticas y servicios).

La organización inicial que se presenta a continuación corresponde al estado en que se dejará la base de datos del sistema GAMS y de sus dos aplicaciones iniciales de prueba, en lo referente al modelado de dichas aplicaciones.

Como principal ventaja del sistema de **Generación Automática de MicroServicios**, es que dicha organización inicial podrá ser alterada en cualquier momento por parte de los administradores del sistema mediante los mantenimientos oportunos, sin necesidad de cambiar el código de la solución ni de cada una de las aplicaciones, evitando realizar costosos esfuerzos.



## 5.- LÍNEAS FUTURAS. CONCLUSIONES.

*La vida debe ser un continuo aprendizaje.*  
Gustave Flaubert (1821-1880)

Una vez desarrollado e implantado el sistema de **Generación Automática de MicroServicios** que hemos presentado a lo largo de la presente Memoria, nos queda indicar futuros pasos de actuación que pueden realizarse en base al estado de la cuestión resultante. Los hemos dividido en dos apartados:

- Líneas futuras: en las cuales se describen nuevas actuaciones en base al resultado obtenido del sistema desarrollado.
- Conclusiones: en las cuales se describe de manera general el estado de la cuestión contemplando ya el programa creado en este Trabajo.

### 5.1.- Líneas Futuras.

Si bien el sistema ha sido concebido desde el primer momento como suficientemente flexible, también es cierto que existen numerosas decisiones de compromiso que se han considerado a la hora de abordar la construcción del mismo (véase apartado 3.2).

Así pues, la primera línea de actuación sería ampliar el actual sistema de **Generación Automática de MicroServicios** para ir incorporando todas aquellas cuestiones que se consideren de interés. Mencionamos a continuación algunas especialmente relevantes:

- Ampliar el actual mecanismo de seguridad, de manera que los usuarios administradores puedan restringir el acceso a la información, por ejemplo, a nivel de elemento de dominio (entidades, objetos valor y atributos) o incluso de registro. Técnicas como el “control de acceso basado en roles” [199] podrían ser interesantes para esta línea de trabajo.
- Ampliar el modelo de dominio de la aplicación de administración (que sirve de “meta modelo” para las aplicaciones generadas para consumidores) de cara al desarrollo de flujos de trabajo, que den cobertura a la ejecución de los servicios, métodos, eventos y disparadores actualmente implantados.
  - En tecnología PHP, destacar actualmente los proyectos Workflower [200], PVM [201] y Omni-Workflow [202] para la definición y ejecución de flujos de trabajo; destacar que el tercero de ellos hace a su vez uso de la biblioteca BPMN.IO [203] para ofrecer a los usuarios una herramienta de diagrama, basada en tecnologías web, para el diseño de procesos de negocio mediante las notaciones BPMN [204], DMN [205] y CMMN [206].
- Construir un sistema de expresiones de alto nivel que permite al usuario final definir restricciones y condiciones, sin necesidad de conocer lenguajes de programación específicos como JavaScript o SQL.
- Facilitar la importación de entidades de diferentes microservicios, de manera que se puedan aplicar patrones de diseño como “entidades remotas” y “mapa de contextos”.

## 5.2.- Conclusiones.

Se ha presentado en este Proyecto una solución versátil para desarrollar aplicaciones basadas en la arquitectura de los microservicios, tanto en sus aspectos de modelado de la información como en sus aspectos de gestión de dicha información.



Desde el comienzo, se ha pretendido emular (hasta donde es posible) el desarrollo de un sistema software como si se tratase de un entorno profesional, diferenciando el equipo de clientes y usuarios del equipo de desarrollo, así como obteniendo los productos característicos de toda solución comercial (documentación, análisis funcional, diseño técnico y productos ejecutables).

Queda así cubierta la doble intención de todo Trabajo Fin de Máster: servir de último paso previo a la obtención del Título, así como realizar una investigación de interés para la comunidad desarrollando un sistema productivo.

Se confía haber construido una solución que cubra el hueco encontrado en el estado actual de la cuestión, ofreciendo a los usuarios una aplicación capaz de modelar aplicaciones basadas en microservicios, tan necesaria en ámbitos académicos y profesionales, a la par que posibilita la gestión de la información modelada sin requerir grandes esfuerzos de cara a presentarla visualmente.

Igualmente, desde el planteamiento de aportar una solución flexible a la par que tecnológicamente moderna, se ha logrado obtener un producto cuya vida útil se prolongará a lo largo de los próximos años sin necesidad de requerir costosos mantenimientos (tanto en términos económicos como de esfuerzo) para conservar su sentido práctico.

## 6.- BIBLIOGRAFÍA Y REFERENCIAS.

*La lectura es como el alimento; el provecho no está en proporción de lo que se come, sino de lo que se digiere.*

Jaime Luciano Balmes (1810-1848)

Se enumeran a continuación todas las referencias (bibliográficas y Web) utilizadas durante la elaboración del presente Trabajo Fin de Máster, ordenadas según su aparición en el presente documento:

- [1] FANO, Andrew; GERSHMAN, Anatole.  
*The future of business services in the age of ubiquitous computing.*  
Communications of the ACM, volume 45, número 12, diciembre de 2002,  
páginas 83-87.  
DOI: 10.1145/585597.585620
- [2] FREE SOFTWARE FOUNDATION (FSF).  
*Qué es el software libre.*  
[En línea] <https://www.gnu.org/philosophy/free-sw.html>  
[Consultado: 2 de mayo de 2017]
- [3] DOCKER, INC.  
*Docker for microservices: accelerate shipping code while unlocking developer innovation.*  
[En línea] <https://www.docker.com/solutions/microservices>  
[Consultado: 5 de septiembre de 2018]
- [4] KUBERNETES.  
What is Kubernetes?  
[En línea] <https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/>  
[Consultado: 5 de septiembre de 2018]

- [5] DOCKER, INC.  
*What is a container?*  
[En línea] <https://www.docker.com/resources/what-container>  
[Consultado: 5 de septiembre de 2018]
- [6] HASHICORP.  
*Introduction to Vagrant.*  
[En línea] <https://www.vagrantup.com/intro/index.html>  
[Consultado: 5 de septiembre de 2018]
- [7] PUPPET, INC.  
*Overview of Puppet's architecture.*  
[En línea] <https://puppet.com/docs/puppet/5.5/architecture.html>  
[Consultado: 5 de septiembre de 2018]
- [8] VMWARE, INC.  
*Understanding full virtualization, paravirtualization, and hardware assist.*  
[En línea]  
[https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/VMware\\_paravirtualization.pdf](https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/VMware_paravirtualization.pdf)  
[Consultado: 5 de septiembre de 2018]
- [9] EBERT, Christof; GALLARDO, Gorka; HERNANTES, Josune;  
SERRANO, Nicolas.  
*DevOps.*  
IEEE Software, volumen 33, número 3, mayo-junio de 2016, páginas 94-100.  
DOI: 10.1109/MS.2016.68
- [10] GARCIA MOLINA, Jesús; *et al.*  
*Desarrollo de Software Dirigido por Modelos: Conceptos, Métodos y Herramientas.*  
RA-MA Editorial, primera edición, 2013, 583 páginas.  
ISBN-13: 978-84-9964-215-4

- [11] LIEBERMAN, Henry; PATERNÒ, Fabio; WULF, Volker (editores).  
*End User Development*.  
Springer, Human–Computer Interaction Series, volumen 9, 2006, 492 páginas.  
ISSN: 1571-5035  
ISBN-13: 978-1-4020-5386-3
- [12] WORLD WIDE WEB CONSORTIUM (W3C).  
*Hypertext Markup Language (HTML) 5.2 Recommendation*.  
[En línea] <https://www.w3.org/TR/html5/>  
[Consultado: 20 de mayo de 2018]
- [13] WORLD WIDE WEB CONSORTIUM (W3C).  
*Cascading Style Sheets (CSS) Snapshot 2017*.  
[En línea] <http://www.w3.org/TR/CSS/>  
[Consultado: 2 de mayo de 2018]
- [14] EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION (ECMA)  
*Standard ECMA-262: ECMAScript 2017 Language Specification, 8<sup>th</sup> Edition (June 2017)*.  
[En línea] <https://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>  
[Consultado: 20 de mayo de 2018]
- [15] CROCKFORD, Douglas.  
*Introducing JSON*.  
[En línea] <https://www.json.org/json-es.html>  
[Consultado: 10 de septiembre de 2018]
- [16] WORLD WIDE WEB CONSORTIUM (W3C).  
*Extensible Mark-Up Language (XML) 1.1*.  
[En línea] <http://www.w3.org/TR/xml11>  
[Consultado: 16 de septiembre de 2016]

- [17] FIELDING, Roy Thomas.  
*Representational State Transfer (REST)*.  
Irvine: Universidad de California, Architectural Styles and the Design of Network-based Software Architectures, capítulo 5, 2000, páginas 75-105.  
[En línea] [https://www.ics.uci.edu/~fielding/pubs/dissertation/rest\\_arch\\_style.htm](https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm)  
[En línea] [https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding\\_dissertation.pdf](https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf)  
[Consultado: 10 de septiembre de 2018]
- [18] THE PHP GROUP  
*¿Qué es PHP?*  
[En línea] <http://php.net/manual/es/intro-what-is.php>  
[Consultado: 8 de septiembre de 2018]
- [19] THE APACHE SOFTWARE FOUNDATION.  
*About the Apache HTTP Server Project*.  
[En línea] [https://httpd.apache.org/ABOUT\\_APACHE.html](https://httpd.apache.org/ABOUT_APACHE.html)  
[Consultado: 10 de septiembre de 2018]
- [20] MICROSOFT CORPORATION  
*Introduction to IIS Architectures*.  
[En línea] <https://docs.microsoft.com/en-us/iis/get-started/introduction-to-iis/introduction-to-iis-architecture>  
[Consultado: 10 de septiembre de 2018]
- [21] MICROSOFT CORPORATION  
*Documentación de SQL Server*.  
[En línea] <https://docs.microsoft.com/es-es/sql/sql-server/sql-server-technical-documentation?view=sql-server-2017>  
[Consultado: 10 de septiembre de 2018]



- [22] ORACLE CORPORATION.  
*Getting Started with MySQL.*  
[En línea] <https://dev.mysql.com/doc/mysql-getting-started/en/>  
[Consultado: 10 de septiembre de 2018]
- [23] MARIADB FOUNDATION.  
*About MariaDB.*  
[En línea] <https://mariadb.org/about/>  
[Consultado: 10 de septiembre de 2018]
- [24] HIPP, Richard.  
*About SQLite.*  
[En línea] <https://www.sqlite.org/about.html>  
[Consultado: 10 de septiembre de 2018]
- [25] MICROSOFT CORPORATION.  
*Sitio web de Microsoft Windows.*  
[En línea] <https://www.microsoft.com/es-es/windows>  
[Consultado: 10 de septiembre de 2018]
- [26] TORVALDS, Linus.  
*LINUX's History.*  
[En línea] <https://www.cs.cmu.edu/~awb/linux.history.html>  
[Consultado: 10 de septiembre de 2018]
- [27] FOWLER, Martin.  
*GUI Architectures: Model View Controller.*  
[En línea] <https://martinfowler.com/eaDev/uiArchs.html#ModelViewController>  
[Consultado: 22 de julio de 2018]

- [28] TELEFÓNICA MOVISTAR.  
Conectividad Gestionada M2M.  
[En línea] <http://www.movistar.es/grandes-empresas/soluciones/fichas/machine-to-machine-m2m/>  
[Consultado: 10 de septiembre de 2018]
- [29] BOOCH, Grady; JACOBSON, Ivar; RUMBAUGH, James.  
*El lenguaje unificado de modelado.*  
Madrid: Pearson Addison Wesley, 1999. 432 páginas.  
ISBN-10: 84-7829-028-1
- [30] BOOCH, Grady; JACOBSON, Ivar; RUMBAUGH, James.  
*El proceso unificado de desarrollo software.*  
Madrid: Pearson Addison Wesley, 2000. 438 páginas.  
ISBN-10: 84-7829-036-2
- [31] BOOCH, Grady; JACOBSON, Ivar; RUMBAUGH, James.  
*El lenguaje unificado de modelado. Manual de referencia.*  
Madrid: Pearson Addison Wesley, 2000. 526 páginas.  
ISBN-10: 84-7829-037-0
- [32] OBJECT MANAGEMENT GROUP (OMG).  
[En línea] <http://www.omg.org/>  
[Consultado: 22 de septiembre de 2016]
- [33] OBJECT MANAGEMENT GROUP (OMG).  
*Model-Driven Architecture (MDA) Guide 2.0.*  
[En línea] <https://www.omg.org/cgi-bin/doc?ormsc/14-06-01.pdf>  
[Consultado: 1º de agosto de 2018]

- [34] COMPUTER ASSOCIATES VERACODE.  
*What is an integrated development environment (IDE)?*  
[En línea] <https://www.veracode.com/security/integrated-development-environments>  
[Consultado: 10 de septiembre de 2018]
- [35] BOCK, Jason; HAZZARD, Kevin.  
*Metaprogramming in .NET.*  
Manning Publications, 2013, 334 páginas.  
ISBN-13: 978-1-617290-26-6
- [36] JETBRAINS, S.R.O.  
*PhpStorm: lightning-smart PHP IDE.*  
[En línea] <https://www.jetbrains.com/phpstorm/>  
[Consultado: 8 de septiembre de 2018]
- [37] MICROSOFT CORPORATION.  
*Microsoft Windows Server 2012 R2 and Windows Server 2012.*  
[En línea] [https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-R2-and-2012/hh801901\(v=ws.11\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-R2-and-2012/hh801901(v=ws.11))  
[Consultado: 8 de septiembre de 2018]
- [38] MICROSOFT CORPORATION.  
*Installing IIS 8.5 on Windows Server 2012 R2.*  
[En línea] <https://docs.microsoft.com/en-us/iis/install/installing-iis-85/installing-iis-85-on-windows-server-2012-r2>  
[Consultado: 8 de septiembre de 2018]
- [39] YANG, Jinfeng.  
*Network Attached Storage (NAS).*  
[En línea] <https://pdfs.semanticscholar.org/presentation/62a8/be9d2ebc59fc9ed610e4b7ebe10196b51fa6.pdf>  
[Consultado: 10 de septiembre de 2018]

- [40] SYNOLOGY, INC.  
*Synology DiskStation Manager.*  
[En línea] <https://www.synology.com/es-es/dsm>  
[Consultado: 10 de septiembre de 2018]
- [41] RASPBERRY PI FOUNDATION.  
*Getting started with the Raspberry Pi.*  
[En línea] <https://projects.raspberrypi.org/en/projects/raspberry-pi-getting-started>  
[Consultado: 10 de septiembre de 2018]
- [42] GARBAJOSA SOPEÑA, Juan; MORENO NAVARRO, Juan José;  
SORIANO CAMINO, Francisco Javier.  
*Tecnologías Software Orientadas a Servicios.*  
Madrid: Fundación madri+d para el Conocimiento, 2009. 221 páginas.  
Colección de Informes de Vigilancia Tecnológica madri+d.  
ISBN-13: 978-84-612-6834-4
- [43] MCKINSEY GLOBAL INSTITUTE.  
*Solving the productivity puzzle: the role of demand and the promise of digitalization.*  
McKinsey and Company, 1ª edición, febrero de 2018.  
[En línea]  
[https://www.mckinsey.com/~/\\_/media/McKinsey/Global%20Themes/Meeting%20societys%20expectations/Solving%20the%20productivity%20puzzle/MGI-Solving-the-Productivity-Puzzle-Report-February-22-2018.ashx](https://www.mckinsey.com/~/_/media/McKinsey/Global%20Themes/Meeting%20societys%20expectations/Solving%20the%20productivity%20puzzle/MGI-Solving-the-Productivity-Puzzle-Report-February-22-2018.ashx)  
[Consultado: 13 de mayo de 2018]
- [44] CIRUELA, Sergio; DELGADO, Miguel; MARÍN, Nicolás.  
*Comunicádonos con las cosas.*  
Ciencia Cognitiva, volumen 3, número 3, 2009, páginas 68-71.  
ISSN: 1988-7884

- [45] MACGILLIVRAY, Carrie; TORCHIA, Marcus.  
*Internet of Things: Market Spend & Trend Outlook for 2018 and Beyond.*  
IDC, 1ª edición, febrero de 2018.  
[En línea] <https://www.idc.com/getdoc.jsp?containerId=US43599418>  
[Consultado: 13 de mayo de 2018]
- [46] GARTNER.  
*Gartner Says 6.4 Billion Connected "Things" Will Be in Use in 2016, Up 30 Percent From 2015.*  
[En línea] <https://www.gartner.com/newsroom/id/3165317>  
[Consultado: 13 de mayo de 2018]
- [47] BAUER, Friedrich.  
*Origins and Foundations of Computing: In Cooperation with Heinz Nixdorf MuseumsForum.*  
Springer, primera edición, 2009, 142 páginas.  
ISBN-13: 978-3-642-029912
- [48] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO).  
*ISO/IEC/IEEE 42010:2011(E): Systems and software engineering – Architecture description.*  
1ª edición, 2011.
- [49] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO).  
*ISO/IEC/IEEE 15288:2015(E): Systems and software engineering – System life cycle processes.*  
1ª edición, 2015.

- [50] NATIONAL SCIENCE BOARD.  
*Science and Engineering Indicators 2014, chapter 6: Industry, Technology and the Global Marketplace.*  
Arlington VA: National Science Foundation (NSB 14-01), 2014.  
[En línea] <http://www.nsf.gov/statistics/seind14/content/chapter-6/chapter-6.pdf>  
[Consultado: 30 de abril de 2017]
- [51] NEAMTIU, Iulian; SUH, Steve.  
*Studying software evolution for taming software complexity.*  
Australian Software Engineering Conference 2010, abril de 2010.  
[En línea] <http://www.cs.ucr.edu/~neamtiu/pubs/suh10aswec.pdf>  
[Consultado: 30 de abril de 2017]
- [52] BÉZIVIN, Jean.  
*The imposible equation.*  
[En línea] <http://modelseverywhere.wordpress.com/2010/10/31/the-impossible-equation/>  
[Consultado: 30 de abril de 2017]
- [53] VARIOS AUTORES.  
*“Know-how”. Collins English Dictionary - Complete & Unabridged 10th Edition.*  
HarperCollins Publishers.  
[En línea] <http://www.collinsdictionary.com/dictionary/english/know-how>  
[Consultado: 30 de abril de 2017]
- [54] HORNE, Andrew.  
*The IT Skills Gap.*  
CIO Network (Forbes), 2 de enero de 2012  
[En línea] <https://www.forbes.com/sites/ciocentral/2012/01/02/the-it-skills-gap/print/>  
[Consultado: 30 de abril de 2017]

- [55] NIELSEN, Tod.  
*2012: Time To Adopt Platform As A Service.*  
CIO Network (Forbes), 2 de enero de 2012.  
[En línea] <https://www.forbes.com/sites/ciocentral/2012/01/02/2012-time-to-adopt-platfrom-as-a-service/print/>  
[Consultado: 30 de abril de 2017]
- [56] BROOKS, Frederick.  
*The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition.*  
2ª edición. Addison-Wesley Professional, 1995. 336 páginas.  
ISBN-13: 978-0201835953
- [57] PRESSMAN, Roger.  
*Ingeniería de Software. Un enfoque práctico.*  
McGraw-Hill Interamericana de España, séptima edición, 1º de marzo de 2010.  
ISBN-10: 6071503140
- [58] REINHARTZ-BERGER, Iris; STURM, Arnon; CLARK, Tony;  
COHEN, Sholom; BETTIN, Jorn. (editores)  
*Domain Engineering: Product Lines, Languages, and Conceptual Models.*  
Springer-Verlag, primera edición, 2013, 404 páginas.  
ISBN-13: 978-3642366536
- [59] CUEVAS, Gonzalo; et al.  
*Gestión del Proceso Software.*  
Madrid: Editorial Centro de Estudios Ramón Areces, 2002. 447 páginas.  
ISBN-10: 84-8004-546-9

- [60] COOK, Steve; GREENFIELD, Jack; KENT, Stuart; SHORT, Keith.  
*Software Factories: assembling applications with patterns, models, frameworks, and tools.*  
1ª edición. Wiley, 2004. 696 páginas.  
ISBN-13: 978-0471202844
- [61] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO).  
*ISO/IEC/IEEE 12207:2017(E): Systems and software engineering – Software life cycle processes.*  
1ª edición, 2015.
- [62] BASS, Len; CLEMENTS, Paul; KAZMAN, Rick.  
*Software Architecture in Practice, Third Edition.*  
Addison-Wesley Professional, 2012, 640 páginas.  
ISBN-13: 978-0321815736
- [63] PARNAS, David Lorge.  
*On the criteria to be used in decomposing systems into modules.*  
Nueva York: Magazine Communications of the ACM, volumen 15, número 12, diciembre de 1972, páginas 1053-1058.  
DOI: 10.1145/361598.361623
- [64] TARR, Peri; OSSHER, Harold; HARRISON, William; SUTTON, Stanley.  
*N degrees of separation: multi-dimensional separation of concerns.*  
Los Angeles: Proceedings of the 1999 International Conference on Software Engineering, mayo de 1999, páginas 107-119.  
ISSN: 0270-5257.

- [65] DEBSKI, Andrzej; SZCZEPANIK, Bartłomiej; MALAWSKI, Maciej; SPAHR, Stefan; MUTHIG, Dirk.  
*A Scalable, Reactive Architecture for Cloud Applications.*  
IEEE Software, volumen 35, número 2, marzo/abril de 2018, páginas 62-71.  
ISSN: 0740-7459.  
DOI: 10.1109/MS.2017.265095722
- [66] CLEMENTS, Paul; NORTHROP, Linda.  
*Software Product Lines: Practices and Patterns.*  
Addison-Wesley Professional, tercera edición, agosto de 2001, 608 páginas.  
ISBN-10: 0-201-70332-7.
- [67] NAUR, Peter; RANDELL, Brian. (editores).  
*Software Engineering: report of a conference sponsored by the NATO Science Committee, Garmisch, Germany, 7-11 Oct. 1968.*  
Bruselas: División Científica de la O.T.A.N., 1969, 231 páginas.  
[En línea] <http://homepages.cs.ncl.ac.uk/brian.randell/NATO/>  
[Consultado: 30 de abril de 2017]
- [68] NAUR, Peter; RANDELL, Brian. (editores).  
*Software Engineering: report of a conference sponsored by the NATO Science Committee, Garmisch, Germany, 7-11 Oct. 1968.*  
Bruselas: División Científica de la O.T.A.N., 1969, 231 páginas.  
[En línea] <http://homepages.cs.ncl.ac.uk/brian.randell/NATO/>  
[Consultado: 30 de abril de 2017]
- [69] PARNAS, David Lorge.  
*On the Design and Development of Program Families.*  
IEEE Transactions on Software Engineering, volumen SE-2, número 1, marzo de 1976; páginas 1-9.  
ISSN: 0098-5589.  
DOI: 10.1109/TSE.1976.233797

- [70] KEAT, Paul; YOUNG, Philip.  
*Economía de empresa.*  
Prentice Hall, cuarta edición, octubre de 2004, 784 páginas.  
ISBN-13: 978-9702604419
- [71] HERADIO, Rubén.  
*Metodología de desarrollo de software basada en el paradigma generativo: realización mediante la transformación de ejemplares.*  
UNED, Tesis Doctoral, 2007, 234 páginas.  
[En línea] <http://e-spacio.uned.es/fez/view.php?pid=tesisuned:IngInf-Rheradio>  
[Consultado: 1º de agosto de 2018]
- [72] CZARNECKI, Krzysztof; EISENECKER, Ulrich.  
*Generative programming.*  
Addison Wesley, primera edición, 2000, 832 páginas.  
ISBN-13: 978-0-201-30977-5
- [73] FOWLER, Martin; PARSONS, Rebecca.  
*Domain-Specific Languages.*  
Addison-Wesley Professional, primera edición, 2010, 640 páginas.  
ISBN-13: 978-0321712943
- [74] IEEE COMPUTER SOCIETY.  
*IEEE Std. 830-1998: IEEE Recommended Practice for Software Requirements Specifications.*  
Nueva York: IEEE Computer Society, octubre de 1998, 39 páginas.  
ISBN-10: 0-7381-0332-2
- [75] WIEGERS, Karl.  
*Software Requirements, 2nd edition.*  
Microsoft Press, 2003, 516 páginas.  
ISBN-13: 978-0-7356-1879-4

- [76] WIEGERS, Karl.  
*More About Software Requirements.*  
Microsoft Press, 2006, primera edición, 2006, 201 páginas.  
ISBN-13: 978-0-7356-2267-8
- [77] BEATTY, Joy; CHEN, Anthony.  
*Visual Models for Software Requirements.*  
Microsoft Press, primera edición, 2012, 442 páginas.  
ISBN-13: 978-0-7356-6772-3
- [78] OBJECT MANAGEMENT GROUP.  
*About the Unified Modeling Language (UML) Specification, version 2.5.1.*  
[En línea] <https://www.omg.org/spec/UML/>  
[Consultado: 1º de agosto de 2018]
- [79] COCKBURN, Alistair.  
*Writing Effective Use Cases.*  
Addison-Wesley, primera edición, octubre de 2000, 304 páginas.  
ISBN-13: 978-0201702255.
- [80] Varios autores.  
*Manifiesto ágil.*  
[En línea] <http://agilemanifesto.org/iso/es/manifiesto.html>  
[Consultado: 1º de agosto de 2018]
- [81] SCHWABER, Ken.  
*Agile Software Development with SCRUM.*  
Pearson, primera edición, octubre de 2001, 176 páginas.  
ISBN-13: 978-0130676344

- [82] COHN, Mike.  
*User Stories Applied: For Agile Software Development*.  
Addison-Wesley Professional, primera edición, marzo de 2004, 304 páginas.  
ISBN-13: 978-0321205681
- [83] ROYCE, Winston.  
*Managing the development of large software systems: concepts and techniques*.  
IEEE Computer Society, Western Electric Show and Convention Technical  
Papers, 1970, páginas 1-9.
- [84] KANG, Kyo; et al.  
*Feature-Oriented Domain Analysis (FODA) Feasibility Study*.  
Software Engineering Institute, primera edición, 1990, 161 páginas.  
[En línea] [https://resources.sei.cmu.edu/asset\\_files/TechnicalReport/1990\\_005\\_001\\_15872.pdf](https://resources.sei.cmu.edu/asset_files/TechnicalReport/1990_005_001_15872.pdf)  
[Consultado: 1º de agosto de 2018]
- [85] WIKIPEDIA.  
*Feature model*.  
[En línea] <http://upload.wikimedia.org/wikipedia/commons/d/d6/E-shopFM.jpg>  
[En línea] [http://en.wikipedia.org/wiki/Feature\\_model](http://en.wikipedia.org/wiki/Feature_model)  
[Consultado: 15 de septiembre de 2016]
- [86] BEN-KIKI, Oren; EVANS, Clark; DOT NET, Ingy.  
*YAML Ain't Markup Language (YAML™) Version 1.2*.  
[En línea] <http://yaml.org/spec/1.2/spec.html>  
[Consultado: 10 de septiembre de 2018]
- [87] OBJECT MANAGEMENT GROUP (OMG).  
*Meta Object Facility (MOF) Specification, version 2.5.1*.  
[En línea] <https://www.omg.org/spec/MOF/2.5.1/PDF>  
[Consultado: 1º de agosto de 2018]

- [88] WORLD WIDE WEB FOUNDATION.  
*History of the Web.*  
[En línea] <https://webfoundation.org/about/vision/history-of-the-web/>  
[Consultado: 9 de septiembre de 2018]
- [89] ASSOCIATION MANAGEMENT SOLUTIONS, LLC (AMS).  
*Official Internet Protocol Standards.*  
[En línea] <https://www.rfc-editor.org/standards>  
[Consultado: 13 de mayo de 2018]
- [90] BIRREL, Andrew; NELSON, Bruce Jay.  
*Implementing remote procedure calls.*  
Nueva York: ACM Transactions on Computer Systems (TOCS), volumen 2,  
número 1, febrero de 1984, páginas 39-59.  
DOI: 10.1145/2080.357392
- [91] SUN MICROSYSTEMS, INC.  
*RPC: Remote Procedure Call Protocol Specification.*  
[En línea] <https://tools.ietf.org/html/rfc1050>  
[Consultado: 2 de junio de 2018]
- [92] SUN MICROSYSTEMS, INC.  
*RPC: Remote Procedure Call Protocol Specification Version 2.*  
[En línea] <https://tools.ietf.org/html/rfc5531>  
[Consultado: 2 de junio de 2018]
- [93] INTERNET ENGINEERING TASK FORCE (IETF),  
*Hypertext Transfer Protocol (HTTP/1.1).*  
[En línea] <https://tools.ietf.org/html/rfc2616>  
[Consultado: 10 de septiembre de 2018]

- [94] WORLD WIDE WEB CONSORTIUM (W3C).  
*SOAP Version 1.2.*  
[En línea] <https://www.w3.org/TR/soap/>  
[Consultado: 7 de septiembre de 2017]
- [95] WORLD WIDE WEB CONSORTIUM (W3C).  
*Web Services Description Language (WSDL). Version 2.0, Part 1: Core Language.*  
[En línea] <https://www.w3.org/TR/wsdl/>  
[Consultado: 7 de septiembre de 2017]
- [96] ERL, Thomas.  
Service-oriented architecture: a field guide to integrating XML and Web services.  
Prentice Hall, primera edición, 2004, 536 páginas.  
ISBN-10: 0131428985
- [97] THE PHP GROUP.  
*Historia de PHP.*  
[En línea] <http://php.net/manual/es/history.php.php>  
[Consultado: 10 de enero 2017]
- [98] GULBRANSEN, David; RAWLINGS, Kenrick.  
*Edición Especial DHTML: HTML Dinámico.*  
Madrid: Prentice Hall Iberia, 1998. 580 páginas.  
ISBN-10: 84-8322-069-5
- [99] WORLD WIDE WEB CONSORTIUM (W3C).  
*Extensible StyleSheet Language Transformations (XSLT) 1.0.*  
[En línea] <http://www.w3.org/TR/xslt>  
[Consultado: 11 de enero de 2017]

- [100] LUJÁN-MORA, Sergio.  
*Programación de servidores web con CGI, SSI e IDC.*  
Alicante: Editorial Club Universitario, 2001. 105 páginas.  
ISBN-13: 978-84-8454-136-3.  
[En línea] <http://rua.ua.es/dspace/handle/10045/16997>  
[En línea] [https://rua.ua.es/dspace/bitstream/10045/16997/1/sergio\\_lujan-programacion\\_de\\_servidores\\_web\\_cgi\\_ssi\\_idc.pdf](https://rua.ua.es/dspace/bitstream/10045/16997/1/sergio_lujan-programacion_de_servidores_web_cgi_ssi_idc.pdf)  
[Consultado: 12 de enero de 2017]
- [101] MICROSOFT CORPORATION.  
*Scenario: Build a Classic ASP Website on IIS.*  
[En línea] <https://docs.microsoft.com/en-us/iis/application-frameworks/running-classic-asp-applications-on-iis-7-and-iis-8/scenario-build-a-classic-asp-website-on-iis>  
[Consultado: 10 de septiembre de 2018]
- [102] SUN MICROSYSTEMS.  
*Java Server Pages (JSP).*  
[En línea] <http://java.sun.com/products/jsp/download/index.html>  
[Consultado: 11 de enero de 2017]
- [103] THE PHP GROUP.  
*PHP Credits.*  
[En línea] <http://php.net/credits.php>  
[Consultado: 12 de enero de 2017]
- [104] THE PHP GROUP.  
*PHP Licensing.*  
[En línea] <http://php.net/license/index.php>  
[Consultado: 12 de enero de 2017]

- [105] Q-SUCCESS.  
*W3TECHS: usage statistics and market share of server-side programming languages for websites (September 2018).*  
[En línea] [https://w3techs.com/technologies/overview/programming\\_language/all](https://w3techs.com/technologies/overview/programming_language/all)  
[Consultado: 9 de septiembre de 2018]
- [106] BUILTWITH PTY LTD.  
*Framework usage distribution in the top 1 million sites: distribution for websites using framework technologies.*  
[En línea] <https://trends.builtwith.com/framework>  
[Consultado: 9 de septiembre de 2018]
- [107] SUN MICROSYSTEMS.  
*Java.*  
[En línea] <http://java.sun.com/>  
[Consultado: 16 de septiembre de 2016]
- [108] SUN MICROSYSTEMS.  
*Sitio web.*  
[En línea] <http://www.sun.es/>  
[Consultado: 16 de septiembre de 2016]
- [109] ORACLE CORPORATION.  
*Portal Web para España.*  
[En línea] <http://www.oracle.com/es/>  
[Consultado: 23 de abril de 2017]
- [110] INTERNATIONAL BUSINESS MACHINE (IBM).  
*AIX: UNIX for IBM Power Systems.*  
[En línea] <https://www.ibm.com/es-es/it-infrastructure/power/os/aix>  
[Consultado: 10 de septiembre de 2018]

- [111] HEWLETT PACKARD ENTERPRISE (HPE).  
*HPE Integrity con HP-UX.*  
[En línea] <https://www.hpe.com/es/es/servers/hp-ux.html>  
[Consultado: 10 de septiembre de 2018]
- [112] ORACLE CORPORATION.  
*Oracle Solaris 11.*  
[En línea] <https://www.oracle.com/es/solaris/solaris11/>  
[Consultado: 10 de septiembre de 2018]
- [113] RED HAT, INC.  
*Red Hat Enterprise Linux.*  
[En línea] <https://www.redhat.com/es/technologies/linux-platforms/enterprise-linux>  
[Consultado: 10 de septiembre de 2018]
- [114] ORACLE CORPORATION.  
*Oracle Linux.*  
[En línea] <http://www.oracle.com/technetwork/server-storage/linux/overview/index.html>  
[Consultado: 10 de septiembre de 2018]
- [115] ORACLE CORPORATION.  
Java Standard Edition.  
[En línea] <http://www.oracle.com/technetwork/java/javase/overview/index.html>  
[Consultado: 10 de septiembre de 2018]
- [116] SUN MICROSYSTEMS.  
Java 2 Micro Edition (J2ME).  
[En línea] <http://java.sun.com/javame/>  
[Consultado: 11 de enero de 2017]



- [117] *ECLIPSE IDE WebSite.*  
[En línea] <http://www.eclipse.org/>  
[Consultado: 11 de enero de 2017]
- [118] NETBEANS  
*NetBeans IDE WebSite.*  
[En línea] <http://www.netbeans.org/>  
[Consultado: 11 de enero de 2017]
- [119] JETBRAINS.  
*IntelliJ IDEA.*  
[En línea] <http://www.JetBrains.com/idea/>  
[Consultado: 11 de enero de 2017]
- [120] JETBRAINS.  
*Company information.*  
[En línea] <http://www.jetbrains.com/company/>  
[Consultado: 1º de mayo de 2018]
- [121] ORACLE.  
*Oracle Buys Sun.*  
[En línea] <http://www.oracle.com/us/corporate/press/018363>  
[Consultado: 23 de abril de 2017]
- [122] MICROSOFT CORPORATION.  
*What is .NET?*  
[En línea] <https://www.microsoft.com/net/learn/what-is-dotnet>  
[Consultado: 9 de septiembre de 2018]
- [123] MICROSOFT CORPORATION  
*.NET API Browser.*  
[En línea] <https://docs.microsoft.com/en-us/dotnet/api/>  
[Consultado: 9 de septiembre de 2018]



- [124] MICROSOFT CORPORATION.  
*Introducción a Common Language Runtime (CLR).*  
[En línea] <https://docs.microsoft.com/es-es/dotnet/standard/clr>  
[Consultado: 9 de septiembre de 2018]
- [125] MICROSOFT CORPORATION.  
*Información general acerca de Dynamic Language Runtime (DLR).*  
[En línea] <https://docs.microsoft.com/es-es/dotnet/framework/reflection-and-codedom/dynamic-language-runtime-overview>  
[Consultado: 9 de septiembre de 2018]
- [126] KHOSRAVI, Shahram.  
*ASP.NET AJAX Programmer's Reference with ASP.NET 2.0 or ASP.NET 3.5.*  
Indianápolis: Wiley Publishing, 2007. 1.522 páginas.  
Wrox Programmer To Programmer Series.  
ISBN-13: 978-0-470-10998-4
- [127] MICROSOFT CORPORATION.  
*Learn About ASP.NET MVC (Model-View-Controller).*  
[En línea] <https://www.asp.net/mvc>  
[Consultado: 9 de septiembre de 2018]
- [128] FREEMAN, Adam.  
*Expert ASP.NET Web API 2 for MVC Developers.*  
Apress, 2014, 688 páginas.  
ISBN-13: 978-1484200865
- [129] MICROSOFT CORPORATION.  
*Microsoft Visual Studio WebSite.*  
[En línea] <http://msdn.microsoft.com/vstudio/>  
[Consultado: 11 de enero de 2017]

- [130] RAGGET, Dave; *et al.*  
Raggett on HTML 4 (second edition).  
Addison-Wesley Professional, 2ª edición, 1997, 464 páginas.  
ISBN-10: 0201178052  
[En línea] <https://www.w3.org/People/Raggett/book4/ch02.html>  
[Consultado: 9 de septiembre de 2018]
- [131] MICROSOFT PATTERNS AND PRACTICES TEAM.  
*Microsoft Application Architecture Guide (Second Edition)*.  
Microsoft Press, 2009, 496 páginas.  
ISBN-13: 978-0735627109  
[En línea] <https://archive.codeplex.com/?p=apparchguide>  
[En línea] [https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff650706\(v=pandp.10\)](https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff650706(v=pandp.10))  
[Consultado: 9 de septiembre de 2018]
- [132] NODE+JS INTERACTIVE.  
*About JQuery*.  
[En línea] <https://learn.jquery.com/about-jquery/>  
[Consultado: 9 de septiembre de 2018]
- [133] AUBRY, Christophe; VAN LANCKER, Luc.  
*jQuery. El framework JavaScript para sitios dinámicos e interactivos. 3ª edición*.  
Ediciones ENI, 2017, 490 páginas.  
ISBN-13: 978-2409010064
- [134] TWITTER  
*About Bootstrap 4.1*.  
[En línea] <https://getbootstrap.com/docs/4.1/about/overview/>  
[Consultado: 9 de septiembre de 2018]

- [135] ZURB.  
*About Foundation.*  
[En línea] <https://foundation.zurb.com/showcase/about.html>.  
[Consultado: 9 de septiembre de 2018]
- [136] LEPAGE, Pete.  
*Aspectos básicos del diseño web adaptable.*  
[En línea] <https://developers.google.com/web/fundamentals/design-and-ux/responsive/?hl=es>  
[Consultado: 10 de septiembre de 2018]
- [137] NODE.JS FOUNDATION.  
*Acerca de Node.js.*  
[En línea] <https://nodejs.org/es/about/>  
[Consultado: 9 de septiembre de 2018]
- [138] DOGLIO, Fernando.  
*Pro REST API Development with Node.js..*  
Apress, 2015, 200 páginas.  
ISBN-13: 978-1-4842-0918-9
- [139] CHRISTIANSEN, Tom; D'FOY, Brian; ORWANT, Jon; WALL, Larry.  
*Programming Perl, 4th Edition.*  
O'Reilly Media, 2012, 1.176 páginas.  
ISBN-13: 978-0596004927
- [140] HANSEN, Ask; LAPWORTH, Leo; SPIER, Robert.  
*Application Programming with Perl.*  
[En línea] <https://www.perl.org/app.html>  
[Consultado: 9 de septiembre de 2018]



- [141] CARDIM, Eden; DIMENT, Kieren; KURI, Jay; ROBINSON, Jess; TROUT, Matt..  
*The Definitive Guide to Catalyst: Writing Extensible, Scalable and Maintainable Perl-Based Web Applications.*  
Apress, 2009, 362 páginas.  
ISBN-13: 978-1430223658
- [142] LUTZ, Mark.  
*Learning Python, 5th Edition.*  
O'Reilly Media, 2013, 1.648 páginas.  
ISBN-13: 978-1449355739
- [143] LUTZ, Mark.  
*Programming Python, 4th Edition.*  
O'Reilly Media, 2011, 1.632 páginas.  
ISBN-13: 978-0596158101
- [144] THE PYTHON SOFTWARE FOUNDATION.  
*Web Frameworks for Python.*  
[En línea] <https://wiki.python.org/moin/WebFrameworks>  
[Consultado: 9 de septiembre de 2018]
- [145] DJANGO SOFTWARE FOUNDATION.  
*Django de un vistazo.*  
[En línea] <https://docs.djangoproject.com/es/2.1/intro/overview/>  
[Consultado: 9 de septiembre de 2018]
- [146] VINCENT, William.  
*REST APIs with Django: Build powerful web APIs with Python and Django.*  
Publicación independiente, 2018, 193 páginas.  
ISBN-13: 978-1983029981

- [147] MATSUMOTO, Yukihiro.  
*Acerca de Ruby.*  
[En línea] <https://www.ruby-lang.org/es/about/>  
[Consultado: 9 de septiembre de 2018]
- [148] HEINEMEIER, David; *et al.*  
*Getting Started with Rails: What is Rails?*  
[En línea] [https://guides.rubyonrails.org/getting\\_started.html#what-is-rails-questionmark](https://guides.rubyonrails.org/getting_started.html#what-is-rails-questionmark)  
[Consultado: 9 de septiembre de 2018]
- [149] GOOGLE, INC.  
*What is Angular?*  
[En línea] <https://angular.io/docs>  
[Consultado: 9 de septiembre de 2018]
- [150] GOOGLE, INC.  
*About Google.*  
[En línea] [https://www.google.es/intl/en\\_es/about/](https://www.google.es/intl/en_es/about/)  
[Consultado: 9 de septiembre de 2018]
- [151] FACEBOOK.  
*What is React?*  
[En línea] <https://reactjs.org/tutorial/tutorial.html#what-is-react>  
[Consultado: 9 de septiembre de 2018]
- [152] FACEBOOK.  
*Sitio web de Facebook Open Source.*  
[En línea] <https://opensource.fb.com/>  
[Consultado: 9 de septiembre de 2018]

- [153] VUE.  
*Guide v2: Introduction: What is Vue.js?*  
[En línea] <https://vuejs.org/v2/guide/#What-is-Vue-js>  
[Consultado: 9 de septiembre de 2018]
- [154] TILDE, INC.  
*What is Ember?*  
[En línea] [https://guides.emberjs.com/release/#toc\\_what-is-ember](https://guides.emberjs.com/release/#toc_what-is-ember)  
[Consultado: 9 de septiembre de 2018]
- [155] SENCHA, INC.  
*Getting Started with Ext JS Open Tooling.*  
[En línea] [https://docs.sencha.com/extjs/6.6.0/guides/getting\\_started/open\\_tooling.html](https://docs.sencha.com/extjs/6.6.0/guides/getting_started/open_tooling.html)  
[Consultado: 9 de septiembre de 2018]
- [156] GOOGLE, LLC.  
*Dart language programming.*  
[En línea] <https://www.dartlang.org/>  
[Consultado: 10 de septiembre de 2018]
- [157] YOUNG, Ian.  
*CoffeeScript Application Development.*  
Packt Publishing, 2013, 258 páginas.  
ISBN-13: 978-1782162667
- [158] LUCCO, Steve.  
*TypeScript: Application-scale JavaScript.*  
[En línea] [https://www.microsoft.com/en-us/research/wp-content/uploads/2013/01/steve-lucco\\_modernprogramming.pdf](https://www.microsoft.com/en-us/research/wp-content/uploads/2013/01/steve-lucco_modernprogramming.pdf)  
[Consultado: 9 de septiembre de 2018]

- [159] SASS  
*Sass Basics: Guide.*  
[En línea] <http://sass-lang.com/guide>  
[Consultado: 10 de agosto de 2017]
- [160] THE CORE LESS TEAM.  
*LESS Overview.*  
[En línea] <http://lesscss.org/#overview>  
[Consultado: 10 de septiembre de 2018]
- [161] THE GULP TEAM.  
*Gulp website.*  
[En línea] <https://gulpjs.com/>  
[Consultado: 10 de septiembre de 2018]
- [162] JS FOUNDATION.  
*Webpack concepts.*  
[En línea] <https://webpack.js.org/concepts/>  
[Consultado: 10 de septiembre de 2018]
- [163] BAUMGARTNER, Stefan.  
*Front-End Tooling with Gulp, Bower, and Yeoman. 1st Edition.*  
Manning Publications, 2016, 240 páginas.  
ISBN-13: 978-1617292743
- [164] MIKOWSKI, Michael; POWELL, Josh.  
*Single Web Page Applications: JavaScript ent-to-end. 1st Edition.*  
Manning Publications, 2013, 432 páginas.  
ISBN-13: 978-1617290756

- [165] ATER, Tal.  
*Building Progressive Web Apps.*  
O'Reilly Media, 2017, 288 páginas.  
ISBN-13: 978-1491961650
- [166] NATIONAL INSTITUTE FOR COMPUTATIONAL SCIENCES (NICS).  
*What is HPC?*  
[En línea] <https://www.nics.tennessee.edu/computing-resources/what-is-hpc>  
[Consultado: 9 de septiembre de 2018]
- [167] INTERNATIONAL BUSINESS MACHINE (IBM).  
*IBM Rational Unified Process (RUP) Lifecycle diagram.*  
[En línea]  
<http://www.ibm.com/developerworks/rational/library/apr07/santos/fig1.gif>  
[Consultado: 10 de enero de 2017]
- [168] BUTLER, Greg; JARZABEK, Stan (editores).  
*Generative and Component-Based Software Engineering.*  
Springer, Generative and component based software engineering: second international symposium; revised papers / GCSE 2000, Erfurt, Germany, del 9 al 12 de septiembre de 2000. Lecture Notes in Computer Science, volumen 2177.  
ISBN-10: 3-540-42578-0
- [169] KERNINGHAN, Brian; PIKE, Robert.  
*The Unix Programming Environment.*  
1ª edición. Prentice-Hall, 1983, 357 páginas.  
ISBN-13: 978-0139376818
- [170] GITHUB  
*Sitio web de Git for Windows.*  
[En línea] <https://gitforwindows.org/>  
[Consultado: 20 de mayo de 2018]

- [171] WEB HYPERTEXT APPLICATION TECHNOLOGY WORKING GROUP (WHATWG).  
*HTML Living Standard.*  
[En línea] <https://html.spec.whatwg.org/>  
[Consultado: 20 de mayo de 2018]
- [172] WORLD WIDE WEB CONSORTIUM (W3C).  
*CSS specifications, current work and how to participate.*  
[En línea] <https://www.w3.org/Style/CSS/current-work>  
[Consultado: 20 de mayo de 2018]
- [173] COMPASS.  
*Sitio web.*  
[En línea] <http://compass-style.org/>  
[Consultado: 10 de agosto de 2017]
- [174] RUBY INSTALLER FOR WINDOWS  
*Área de descargas.*  
[En línea] <https://rubyinstaller.org/downloads/>  
[Consultado: 10 de agosto de 2017]
- [175] NORMARK, Kurt.  
*Introduction to higher-order functions.*  
[En línea] [http://people.cs.aau.dk/~normark/prog3-03/html/notes/higher-order-fu\\_themes-intr-section.html](http://people.cs.aau.dk/~normark/prog3-03/html/notes/higher-order-fu_themes-intr-section.html)  
[Consultado: 20 de mayo de 2018]
- [176] CHAPMAN, Stephen.  
*A brief History of JavaScript: Learn About the Early History of the Java Programming Language.*  
[En línea] <https://www.thoughtco.com/a-brief-history-of-javascript-2037675>  
[Consultado: 9 de septiembre de 2018]

- [177] EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION (ECMA)  
*History of ECMA.*  
[En línea] <https://www.ecma-international.org/memento/history.htm>  
[Consultado: 20 de mayo de 2018]
- [178] EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION (ECMA)  
*Standard ECMA-402: ECMAScript 2017 Internationalization API Specification, 4<sup>th</sup> Edition (June 2017).*  
[En línea] <https://www.ecma-international.org/publications/files/ECMA-ST/ECMA-402.pdf>  
[Consultado: 20 de mayo de 2018]
- [179] EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION (ECMA)  
*Standard ECMA-404: The JSON Data Interchange Syntax, 2<sup>nd</sup> Edition (December 2017).*  
[En línea] <https://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>  
[Consultado: 20 de mayo de 2018]
- [180] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO).  
*ISO/IEC 21778:2017 Information technology -- The JSON data interchange syntax.*  
[En línea] <https://www.iso.org/standard/71616.html>  
[En línea] [http://standards.iso.org/ittf/PubliclyAvailableStandards/c071616\\_ISO\\_IEC\\_21778\\_2017.zip](http://standards.iso.org/ittf/PubliclyAvailableStandards/c071616_ISO_IEC_21778_2017.zip)  
[Consultado: 10 de septiembre de 2018]

- [181] INTERNET ENGINEERING TASK FORCE (IETF).  
*The JavaScript Object Notation (JSON) Data Interchange Format.*  
[En línea] <https://tools.ietf.org/html/rfc8259>  
[En línea] <https://tools.ietf.org/html/rfc7159>  
[En línea] <https://tools.ietf.org/html/rfc7158>  
[En línea] <https://tools.ietf.org/html/rfc4627>  
[Consultado: 10 de septiembre de 2018]
- [182] INTERNET ENGINEERING TASK FORCE (IETF)  
*Who we are.*  
[En línea] <http://ietf.org/about/who/>  
[Consultado: 10 de septiembre de 2018]
- [183] EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION (ECMA).  
*Standard ECMA-414: ECMAScript Specification Suite, 3<sup>rd</sup> Edition*  
*(December 2017).*  
[En línea] <https://www.ecma-international.org/publications/files/ECMA-ST/ECMA-414.pdf>  
[Consultado: 20 de mayo de 2018]
- [184] SQLITE CONSORTIUM  
*About SQLite.*  
[En línea] <https://www.sqlite.org/about.html>  
[Consultado: 20 de mayo de 2018]
- [185] SQLITE.  
*SQLite: most widely deployed and use database engine.*  
[En línea] <https://www.sqlite.org/mostdeployed.html>  
[Consultado: 20 de mayo de 2018]



[186] OPEN SOURCE INITIATIVE

*The Open Source Definition (Annotated).*

[En línea] <https://opensource.org/osd-annotated>

[Consultado: 20 de mayo de 2018]

[187] MICROSOFT CORPORATION.

*Using code coverage to determine how much code is being tested.*

[En línea] <https://msdn.microsoft.com/en-us/library/dd537628.aspx>

[Consultado: 20 de mayo de 2018]

[188] SALAWA, Pawel.

*Página web de SQLite Studio.*

[En línea] <https://sqlitestudio.pl/>

[Consultado: 20 de mayo de 2018]

[189] SALAWA, Pawel.

*Wiki de SQLite Studio en GitHub.*

[En línea] <https://github.com/pawelsalawa/sqlitestudio/wiki/SQLiteStudio>

[Consultado: 20 de mayo de 2018]

[190] ADERMANN, Nils; et al.

*Sitio web de COMPOSER.*

[En línea] <https://getcomposer.org/>

[Consultado: 12 de agosto de 2017]

[191] BEDNARCZYK, Marieke; et al.

*Sitio web de PHP DOCUMENTOR.*

[En línea] <https://phpdoc.org/>

[Consultado: 12 de agosto de 2017]

- [192] MIGUEL, Adoración de; PIATTINI, Mario.  
*Fundamentos y modelos de bases de datos.*  
Madrid: Editoria RA-MA, 1999. 2ª edición, 515 páginas.  
ISBN-10: 84-7897-361-3
- [193] MARCOS, Esperanza; MIGUEL, Adoración de; PIATTINI, Mario.  
*Diseño de bases de datos relacionales.*  
Madrid: Editoria RA-MA, 1999. 1ª edición, 549 páginas.  
ISBN-10: 84-7897-385-0
- [194] NIELSEN, Jakob  
*Usabilidad: diseño de sitios Web.*  
Madrid: Prentice Hall PTR, 2001. 416 páginas.  
ISBN-10: 84-205-3008-5
- [195] PRESSMAN, Roger.  
*Ingeniería del Software. Un enfoque práctico.*  
Aravaca: McGraw-Hill, 2002. 5ª edición, 601 páginas.  
ISBN-10: 84-481-3214-9
- [196] MINGUET, Jesús M<sup>a</sup>; HERNÁNDEZ, Juan F<sup>co</sup>.  
*La calidad del software y su medida.*  
Madrid: Editorial Centro de Estudios Ramón Areces. 263 páginas.  
ISBN-10: 84-8004-611-2
- [197] CMMI INSTITUTE  
*Capability Maturity Model Integration (CMMI) Development, version 2.0*  
[En línea] <https://cmmiinstitute.com/products/cmmi/dev>  
[Consultado: 20 de mayo de 2018]

- [198] CONSEJO SUPERIOR DE INFORMÁTICA, MINISTERIO DE HACIENDA Y FUNCIÓN PÚBLICA, GOBIERNO DE ESPAÑA.  
*Metodología MÉTRICA Versión 3.*  
[En línea]  
[https://administracionelectronica.gob.es/pae\\_Home/pae\\_Documentacion/pae\\_Metodolog/pae\\_Metrica\\_v3.html](https://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/pae_Metrica_v3.html)  
[Consultado: 20 de mayo de 2018]
- [199] SANDHU, Ravi; COYNE, Edward, FEINSTEIN, Hal; YOUMAN, Charles.  
*Role-Based Access Control Models.*  
IEEE Computer, volumen 29, número 2, febrero de 1996, páginas 38-47.  
ISSN: 0018-9162  
DOI: 10.1109/2.485845
- [200] GOTO, Hidenori; KUBO, Atsuhiko.  
*Workflower: A BPMN 2.0 workflow engine for PHP (GitHub site).*  
[En línea] <https://github.com/phpmentors-jp/workflower>  
[Consultado: 9 de septiembre de 2018]
- [201] FORMA-PRO.  
*Process Virtual Machine (PVM).*  
[En línea] <https://pvm.forma-pro.com/>  
[Consultado: 9 de septiembre de 2018]
- [202] OMNISPHERE INFORMATION SYSTEMS, CORP.  
*Omni-Workflow: a Workflow Engine for the Web: Workflow and Case Management for WordPress.*  
[En línea] <https://omniworkflow.com/>  
[Consultado: 9 de septiembre de 2018]



- [203] CAMUNDA SERVICES, GMBH.  
*Web-based tooling for BPMN, DMN and CMMN.*  
[En línea] <http://bpmn.io/>  
[Consultado: 10 de septiembre de 2018]
- [204] OBJECT MANAGEMENT GROUP (OMG).  
*About the Business Process Model and Notation Specification (version 2.0).*  
[En línea] <https://www.omg.org/spec/BPMN/2.0/About-BPMN/>  
[Consultado: 10 de septiembre de 2018]
- [205] OBJECT MANAGEMENT GROUP (OMG).  
*About the Decision Model and Notation Specification (version 1.2 beta).*  
[En línea] <https://www.omg.org/spec/DMN/About-DMN/>  
[Consultado: 10 de septiembre de 2018]
- [206] OBJECT MANAGEMENT GROUP (OMG).  
*About the Case Management Model and Notation Specification (version 1.1).*  
[En línea] <https://www.omg.org/spec/CMMN/About-CMMN/>  
[Consultado: 10 de septiembre de 2018]
- [207] IEEE COMPUTER SOCIETY.  
*IEEE STD-610: computer dictionary. Compilation of IEEE standard computer glossaries.*  
1990.
- [208] TOSHIBA CORPORATION.  
*Hoja de producto de TOSHIBA Qosmio X70-B-10T.*  
[En línea] <https://www.toshiba.es/laptops/qosmio/qosmio-x70-b/qosmio-x70-b-10t>  
[Consultado: 14 de julio de 2018]



- [209] SYNOLOGY, INCORPORATED.  
*Hoja de producto de Synology DiskStation DS415+.*  
[En línea] [https://global.download.synology.com/download/Document/DataSheet/DiskStation/15-year/DS415+/Synology\\_DS415+\\_Data\\_Sheet\\_esn.pdf](https://global.download.synology.com/download/Document/DataSheet/DiskStation/15-year/DS415+/Synology_DS415+_Data_Sheet_esn.pdf)  
[Consultado: 14 de julio de 2018]
- [210] SYNOLOGY, INCORPORATED.  
*Paquetes DSM complementarios: Web Station.*  
[En línea] <https://www.synology.com/es-es/dsm/packages/WebStation>  
[Consultado: 14 de julio de 2018]
- [211] SYNOLOGY, INCORPORATED.  
DiskStation Manager, Artículos de ayuda, Web Station.  
[En línea] [https://www.synology.com/es-es/knowledgebase/DSM/help/WebStation/application\\_webserv\\_desc](https://www.synology.com/es-es/knowledgebase/DSM/help/WebStation/application_webserv_desc)  
[Consultado: 14 de julio de 2018]
- [212] SYNOLOGY, INCORPORATED.  
*Paquetes DSM complementarios: Apache HTTP Server 2.2.*  
[En línea] [https://www.synology.com/es-es/dsm/packages/Apache2\\_2](https://www.synology.com/es-es/dsm/packages/Apache2_2)  
[Consultado: 14 de julio de 2018]
- [213] SYNOLOGY, INCORPORATED.  
*Paquetes DSM complementarios: PHP 7.0.30-0026.*  
[En línea] [https://www.synology.com/es-es/dsm/packages/PHP7\\_0](https://www.synology.com/es-es/dsm/packages/PHP7_0)  
[Consultado: 14 de julio de 2018]
- [214] SYNOLOGY, INCORPORATED.  
*Paquetes DSM complementarios: MariaDB 10.*  
[En línea] <https://www.synology.com/es-es/dsm/packages/MariaDB10>  
[Consultado: 14 de julio de 2018]



- [215] SYNOLOGY, INCORPORATED.  
*DiskStation Manager, Artículos de ayuda, MaríaDB 10 5.*  
[En línea] <https://www.synology.com/es-es/knowledgebase/DSM/help/MariaDB10/mariadb>  
[Consultado: 14 de julio de 2018]
- [216] SYNOLOGY, INCORPORATED.  
*Paquetes DSM complementarios: phpMyAdmin 4.7.8-0175.*  
[En línea] <https://www.synology.com/es-es/dsm/packages/phpMyAdmin>  
[Consultado: 14 de julio de 2018]
- [217] REENSKAUG, Trygve.  
*MVC. XEROX PARC 1978-79.*  
[En línea] <http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html>  
[Consultado: 22 de julio de 2018]
- [218] REENSKAUG, Trygve.  
*The Model-View-Controoler (MVC). Its past and present.*  
Conferencia JavaZone, Oslo, 18 y 19 de septiembre de 2003.  
[En línea] [http://heim.ifi.uio.no/~trygver/2003/javazone-jaoo/MVC\\_pattern.pdf](http://heim.ifi.uio.no/~trygver/2003/javazone-jaoo/MVC_pattern.pdf)  
[Consultado: 22 de julio de 2018]
- [219] FOWLER, Martin.  
*Patterns of Enterprise Application Architecture.*  
Addison-Wesley Professional, primera edición, noviembre de 2002, 560 páginas.  
ISBN-13: 978-0321127426
- [220] SENSIOLABS.  
*Symfony 3.4 Documentation: Databases and the Doctrine ORM.*  
[En línea] <https://symfony.com/doc/3.4/doctrine.html>  
[Consultado: 22 de julio de 2018]

- [221] SENSIOLABS.  
*Symfony 3.4 Documentation: Routing.*  
[En línea] <https://symfony.com/doc/3.4/routing.html>  
[Consultado: 22 de julio de 2018]
- [222] SENSIOLABS.  
*Symfony 3.4 Documentation: Controller.*  
[En línea] <https://symfony.com/doc/3.4/controller.html>  
[Consultado: 22 de julio de 2018]
- [223] SENSIOLABS.  
*Symfony 3.4 Documentation: Creating and Using Templates.*  
[En línea] <https://symfony.com/doc/3.4/templating.html>  
[Consultado: 22 de julio de 2018]
- [224] POTENCIER, Fabien.  
*Sami: an API documentation generator.*  
[En línea] <https://github.com/FriendsOfPHP/sami>  
[Consultado: 10 de septiembre de 2018]
- [225] FONTICONS, INC.  
*Font Awesome 4.7, the iconic font and CSS toolkit.*  
[En línea] <https://fontawesome.com/v4.7.0/>  
[Consultado: 10 de septiembre de 2018]
- [226] THE PHP GROUP.  
*Manual de PHP: instalación y configuración.*  
[En línea] <http://php.net/manual/es/install.php>  
[Consultado: 23 de julio de 2018]



- [227] THE PHP GROUP.  
*Manual de PHP: objetos de datos de PHP.*  
[En línea] <http://php.net/manual/es/book.pdo.php>  
[Consultado: 23 de julio de 2018]
- [228] THE PHP GROUP.  
*Manual de PHP: extensiones de bases de datos específicas del proveedor.*  
[En línea] <http://php.net/manual/es/refs.database.vendors.php>  
[Consultado: 23 de julio de 2018]
- [229] STENBERG, Daniel.  
*cURL: command line tool and library for transferring data with URLs.*  
[En línea] <https://www.gitbook.com/download/pdf/book/bagder/everything-curl>  
[En línea] <https://curl.haxx.se/download.html>  
[Consultado: 10 de septiembre de 2018]
- [230] LIN, Joyce.  
*First 5 things to try if you're new to Postman.*  
[En línea] <http://blog.getpostman.com/2018/04/11/first-5-things-to-try-if-youre-new-to-postman/>  
[Consultado: 10 de septiembre de 2018]
- [231] MICROSOFT CORPORATION.  
*Instalar e implementar Windows Server 2012 R2 y Windows Server 2012.*  
[En línea] [https://msdn.microsoft.com/es-es/library/hh831620\(v=ws.11\).aspx](https://msdn.microsoft.com/es-es/library/hh831620(v=ws.11).aspx)  
[Consultado: 10 de septiembre de 2018]
- [232] MICROSOFT CORPORATION.  
*Installing IIS 8.5 on Windows Server 2012 R2.*  
[En línea] <https://docs.microsoft.com/en-us/iis/install/installing-iis-85/installing-iis-85-on-windows-server-2012-r2>  
[Consultado: 10 de septiembre de 2018]

- [233] NGINX, INC.  
*NGINX: connect, secure and scale microservices.*  
[En línea] <https://www.nginx.com/solutions/microservices/>  
[Consultado: 10 de septiembre de 2018]
- [234] MICROSOFT CORPORATION.  
*Descargas de SQL Server.*  
[En línea] <https://www.microsoft.com/es-es/sql-server/sql-server-downloads>  
[Consultado: 10 de septiembre de 2018]
- [235] MICROSOFT CORPORATION  
*SQL Server para desarrolladores.*  
[En línea] <https://www.microsoft.com/es-es/sql-server/application-development>  
[Consultado: 10 de septiembre de 2018]
- [236] MICROSOFT CORPORATION  
*Herramientas de desarrollo de SQL.*  
[En línea] <https://www.microsoft.com/es-es/sql-server/developer-tools>  
[Consultado: 10 de septiembre de 2018]
- [237] MICROSOFT CORPORATION.  
*Create PHP apps using SQL Server on Windows.*  
[En línea] <https://www.microsoft.com/en-us/sql-server/developer-get-started/php/windows/>  
[Consultado: 10 de septiembre de 2018]
- [238] SENSIOLABS.  
*Symfony 3.4: Databases and the Doctrine ORM.*  
[En línea] <https://symfony.com/doc/3.4/doctrine.html>  
[Consultado: 10 de septiembre de 2018]



- [239] DOCTRINE TEAM.  
*Doctrine DBAL v2.7 Documentation: Platforms.*  
[En línea] <https://www.doctrine-project.org/projects/doctrine-dbal/en/2.7/reference/platforms.html>  
[Consultado: 10 de septiembre de 2018]
- [240] MICROSOFT CORPORATION.  
*Download Microsoft SQL Server 2017 Express from Official Microsoft Download Center.*  
[En línea] <https://www.microsoft.com/en-us/download/details.aspx?id=55994>  
[Consultado: 10 de septiembre de 2018]
- [241] MICROSOFT CORPORATION.  
*Instalar SQL Server desde el Asistente para la instalación (programa de instalación).*  
[En línea] <https://docs.microsoft.com/es-es/sql/database-engine/install-windows/install-sql-server-from-the-installation-wizard-setup?view=sql-server-2017>  
[Consultado: 10 de septiembre de 2018]
- [242] MICROSOFT CORPORATION.  
*Descarga de SQL Server Management Studio (SSMS).*  
[En línea] <https://docs.microsoft.com/es-es/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-2017>  
[Consultado: 10 de septiembre de 2018]
- [243] MICROSOFT CORPORATION.  
*Microsoft ODBC Driver 17 for SQL Server - Windows, Linux y macOS.*  
[En línea] <https://www.microsoft.com/es-ES/download/details.aspx?id=56567>  
[Consultado: 10 de septiembre de 2018]

- [244] MICROSOFT CORPORATION.  
*Microsoft Drivers 5.3 for PHP for SQL Server (Download Center).*  
[En línea] <https://www.microsoft.com/en-us/download/details.aspx?id=57163>  
[Consultado: 10 de septiembre de 2018]
- [245] MICROSOFT CORPORATION  
*Microsoft Drivers for PHP for SQL Server (GitHub site).*  
[En línea] <https://github.com/Microsoft/msphpsql>  
[Consultado: 10 de septiembre de 2018]
- [246] MICROSOFT CORPORATION.  
*Microsoft Drivers for PHP for SQL Server (Documentation).*  
[En línea] <https://docs.microsoft.com/en-us/sql/connect/php/microsoft-php-driver-for-sql-server?view=sql-server-2017>  
[Consultado: 10 de septiembre de 2018]
- [247] MICROSOFT CORPORATION.  
*PDO\_SQLSRV Driver Reference (Documentation).*  
[En línea] <https://docs.microsoft.com/en-us/sql/connect/php/pdo-sqlsrv-driver-reference?view=sql-server-2017>  
[Consultado: 10 de septiembre de 2018]
- [248] MICROSOFT CORPORATION.  
*Loading the Microsoft Drivers for PHP for SQL Server.*  
[En línea] <https://docs.microsoft.com/en-us/sql/connect/php/loading-the-php-sql-driver?view=sql-server-2017>  
[Consultado: 10 de septiembre de 2018]
- [249] MICROSOFT CORPORATION.  
*Microsoft SQL Server 2017: roles de nivel de servidor.*  
[En línea] <https://docs.microsoft.com/es-es/sql/relational-databases/security/authentication-access/server-level-roles?view=sql-server-2017>  
[Consultado: 28 de julio de 2018]



- [250] MICROSOFT CORPORATION.  
*Web Platform Installer: Download WebPI 5.0.*  
[En línea] <https://www.iis.net/downloads/microsoft/web-platform-installer>  
[Consultado: 10 de septiembre de 2018]
- [251] MICROSOFT CORPORATION.  
*Using the Microsoft Web Platform Installer.*  
[En línea] <https://docs.microsoft.com/en-us/iis/install/web-platform-installer/using-the-microsoft-web-platform-installer>  
[Consultado: 10 de septiembre de 2018]
- [252] THE PHP GROUP.  
*PHP For Windows: Binaries and sources Releases.*  
[En línea] <https://windows.php.net/download/>  
[Consultado: 10 de septiembre de 2018]
- [253] THE PHP GROUP.  
*Instalación manual de PHP en Windows.*  
[En línea] <http://php.net/manual/es/install.windows.manual.php>  
[Consultado: 10 de septiembre de 2018]
- [254] THE PHP GROUP.  
*Manejador de Procesos FastCGI (FPM).*  
[En línea] <http://php.net/manual/es/install.fpm.php>  
[Consultado: 10 de septiembre de 2018]
- [255] MOZILLA CONTRIBUTORS.  
*Responsive Design Mode.*  
[En línea] [https://developer.mozilla.org/en-US/docs/Tools/Responsive\\_Design\\_Mode](https://developer.mozilla.org/en-US/docs/Tools/Responsive_Design_Mode)  
[Consultado: 10 de septiembre de 2018]



[256] MOZILLA CONTRIBUTORS.

*Edición de Desarrollador.*

[En línea] [https://developer.mozilla.org/es/docs/Mozilla/Firefox/](https://developer.mozilla.org/es/docs/Mozilla/Firefox/Edici%C3%B3n_de_Desarrollador)

[Edici%C3%B3n de Desarrollador](https://developer.mozilla.org/es/docs/Mozilla/Firefox/Edici%C3%B3n_de_Desarrollador)

[Consultado: 10 de septiembre de 2018]

## 7.- LISTADO DE SIGLAS, ABREVIATURAS Y ACRÓNIMOS.

*Un pensamiento original vale mil citas insignificantes.*  
Diógenes (412 a.C. - 323 a.C.)

Se enumeran a continuación todas las siglas, abreviaturas y acrónimos utilizados durante la elaboración del presente Trabajo Fin de Máster, ordenados alfabéticamente.

- **API**
  - *Application Programming Interface*, interfaz de programación de aplicaciones.
- **ASP y ASP.NET**
  - *ActiveX Server Pages*, tecnología de páginas web de servidor de Microsoft, basadas inicialmente en COM/ActiveX y posteriormente en .NET Framework.
- **BPEL**
  - *Business Process Execution Language*, lenguaje de ejecución de procesos de negocio.
- **BPM**
  - *Business Process Management*, sistema de gestión (modelado y ejecución) de procesos de negocio.
- **BPMN**
  - *Business Process Modeling Notation*, notación para el modelado de procesos de negocio.



- **BRMS**
  - *Business Rule Management System*, sistema de gestión (modelado y ejecución) de reglas de negocio.
- **CSS**
  - *Cascading Style Sheet*, hoja de estilos en cascada.
- **CU**
  - Caso de uso.
- **DBMS**
  - *DataBase Management System*, sistema gestor de bases de datos.
- **DMN**
  - *Decision Model and Notation*, estándar que define la notación y el modelado de decisiones.
- **DSM**
  - *Synology DiskStation Manager*, sistema operativo de Synology para sus dispositivos NAS basado en LINUX.
- **FTP**
  - *File Transfer Protocol*, protocolo de transferencia de archivos.
- **FTPS**
  - *File Transfer Protocolo Secure*, protocolo seguro basado en SSH de transferencia de archivos.
- **GIT**
  - Sistema de control de versiones.
- **HATEOAS**
  - *Hypermedia As The Engine Of Application State*, utilización de hipermédios como motor del estado de una aplicación.



- **HTTP**
  - *HyperText Transfer Protocol*, protocolo de transferencia de hipertexto.
- **HTTPS**
  - Véase S/HTTP.
- **IDE**
  - *Integrated Development Environment*, entorno de desarrollo integrado.
- **IIS**
  - *Microsoft Internet Information Server*, servidor de internet de Microsoft.
- **IoT**
  - *Internet of Things*, el internet de las cosas.
- **JSON**
  - *JavaScript Object Notation*, notación de objetos de JavaScript, en formato de texto.
- **JSONP**
  - *JSON with Padding*, técnica de comunicación HTTP mediante JavaScript que permite realizar llamadas AJAX a dominios diferentes del original.
- **JSP**
  - *Java Server Pages*, tecnología de páginas de servidor basadas en Java de Sun Microsystems (hoy, de Oracle).
- **MOF**
  - *Meta-Object Facility*, marco de trabajo para el modelado de metadatos.
- **MVC**
  - *Model-View-Controller*, estilo arquitectónico de aplicaciones basado en la organización de componentes en tipos: Modelo-Vista-Controlador.



- **OAS**
  - *OpenAPI Specification*, especificación de API abierto.
- **PC**
  - *Personal Computer*, computador personal.
- **PHP**
  - Originalmente, *Personal HyperText Processor*, procesador de hipertexto personal.
  - Actualmente, *PHP: HyperText Processor*.
- **PWA**
  - *Progressive Web Application*: estilo arquitectónico de aplicaciones web progresivas.
- **RDBMS**
  - *Relational DataBase Management System*, sistema gestor de bases de datos relacionales.
- **REST**
  - *REpresentational State Transfer*, transferencia de estado mediante representación.
- **RUP**
  - *Rational Unified Process*, proceso unificado de Rational (metodología de desarrollo software).
- **SFTP**
  - *Secure File Transfer Protocol*, protocolo seguro de transferencia de archivos, basado en SSL/TLS.



- **S/HTTP**
  - *Secure Hyper Text Transfer Protocol*, protocolo seguro de transferencia de hipertexto.
- **SOA**
  - *Service Oriented Architecture*, arquitectura orientada a servicios.
- **SOAP**
  - *Simple Object Access Protocol*, protocolo simple de acceso a objetos.
- **SPA**
  - *Single Page Application*, estilo arquitectónico de aplicación web de página única.
- **SQL**
  - *Structured Query Language*, lenguaje de consulta y manipulación de datos, utilizado con sistemas gestores de bases de datos relacionales (véase RDBMS).
- **SSL**
  - *Secure Socket Layer*, capa de seguridad a nivel de *socket*.
- **TLS**
  - *Transport Layer Security*, seguridad a nivel de la capa de transporte.
- **UML**
  - *Unified Modeling Language*, lenguaje unificado de modelado.
- **URI**
  - *Uniform Resource Identifier*, identificador uniforme de recurso.
- **URL**
  - *Uniform Resource Locator*, localizador uniforme de recurso.



- **WBS**
  - *Work Break-Down Structure*, estructura jerárquica de descomposición del trabajo.
- **WSDL**
  - *Web Service Description Language*, lenguaje de descripción de servicios web.
- **XML**
  - *eXtensible Mark-up Language*, lenguaje extensible basado en etiquetas.
- **XP**
  - *Extremme Programming*, programación extrema (metodología de desarrollo software).
- **XSL**
  - *XML Stylesheet Language*, lenguaje de hojas de estilo basadas en XML.
- **XSLT**
  - *XML Stylesheet Language Transformations*, transformaciones para el lenguaje de hojas de estilo basadas en XML.

## 8.- ANEXO A: ANÁLISIS FUNCIONAL.

*El que no sabe lo que busca no entiende lo que encuentra.*

Claude Bernard (1813-1878)

Se presentan a continuación los diferentes productos obtenidos durante la fase de Análisis Funcional.

El objetivo del presente Anexo A es sacar fuera del cuerpo principal de la Memoria material voluminoso, de naturaleza eminentemente técnica, de cara a no romper el hilo conductor del presente documento sin, evidentemente, dejar incompleto el material resultante.

Este anexo A cubre en profundidad todos los productos introducidos en el punto 4.2 de Análisis Funcional:

- Requisitos educidos.
- Diagrama conceptual.
- Diagrama de casos de uso.
- Casos de uso extendidos.

### 8.1.- Requisitos educidos.

Atendiendo a la definición de requisito establecida por IEEE (véanse [207] y [59]), es la “*condición o capacidad necesitada por un usuario para resolver un problema o alcanzar un objetivo*”; desde el punto de vista de otro sistema son la “*condición o capacidad que debe cumplir o poseer un sistema o un componente del mismo para satisfacer un contrato, un estándar, una especificación u otro documento impuesto de manera formal*”.

Mediante el proceso de educación realizado se establecieron los siguientes requisitos a cubrir:

- Requisitos funcionales:
  - [R-1]: los usuarios podrán modelar sus propias aplicaciones, sin necesidad de conocimientos de programación.
  - [R-2]: generar aplicaciones basadas en microservicios a partir de los modelos definidos por los usuarios.
  - [R-3]: posibilitar la explotación de la información gestionada por las aplicaciones generadas.
  - [R-4]: posibilitar la comunicación entre sistemas, a través de microservicios ofrecidos por las aplicaciones.
- Requisitos no funcionales:
  - [R-5]: se utilizarán cuantos estándares existentes sea posible, frente a mecanismos y tecnologías propietarias.
  - [R-6]: se posibilitará el uso de entornos de ejecución de diferente naturaleza y capacidad, como computadores personales y dispositivos embarcables.

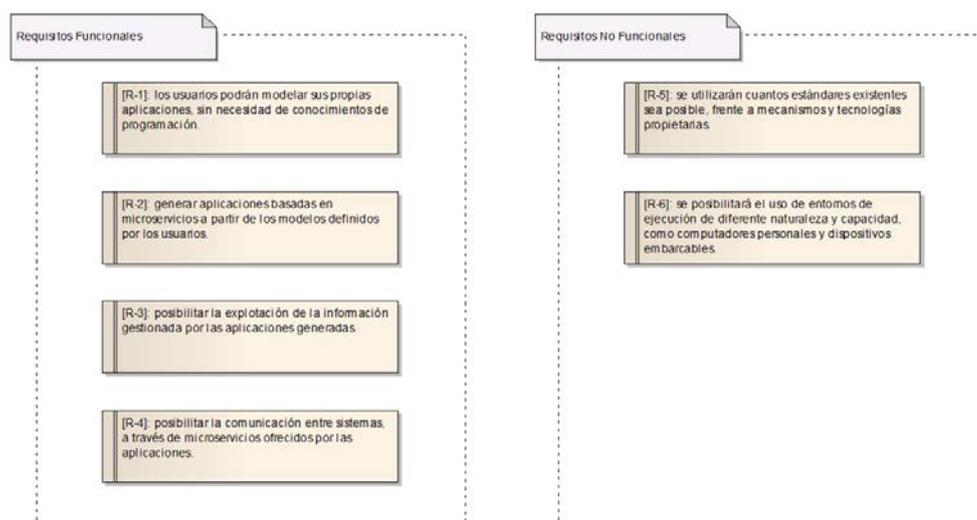


Fig. 8.1.- Diagrama de requisitos.

## 8.2.- Diagrama conceptual.

Mediante el diagrama conceptual, obtenemos una visión de alto nivel (muy general) de las principales entidades (conceptos) que participan en el sistema y que serán debidamente modeladas en la etapa de Diseño Técnico.

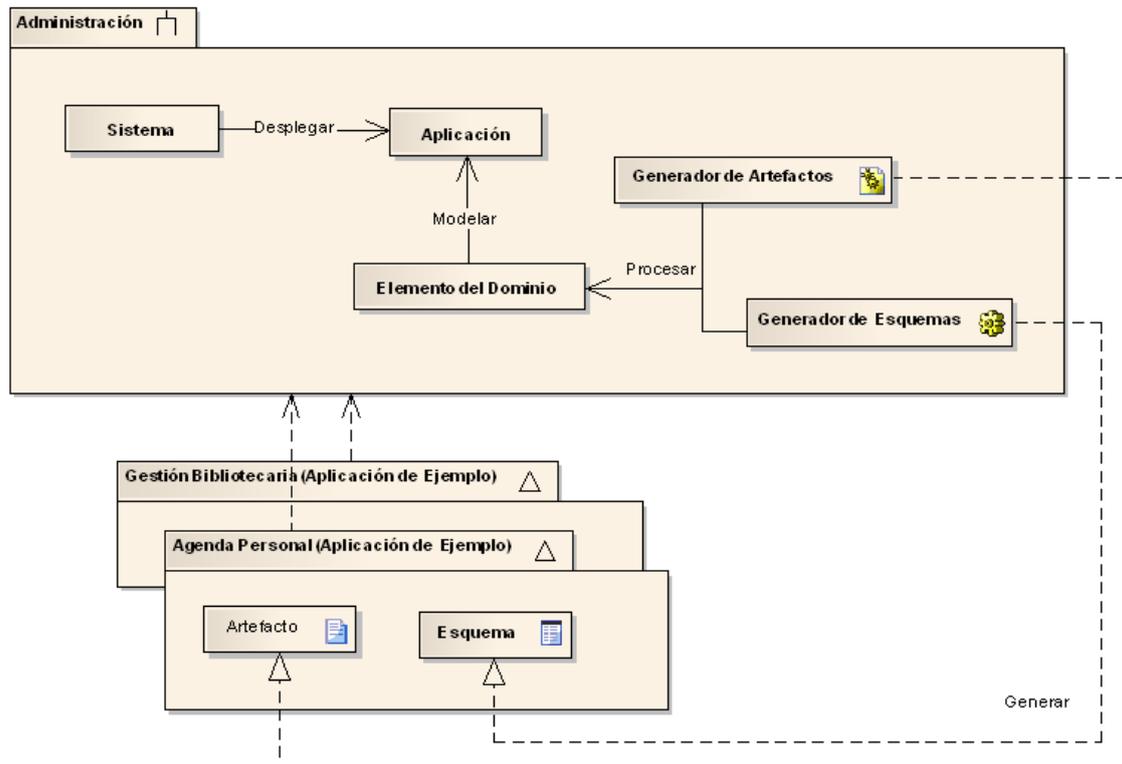


Fig. 8.2.- Diagrama conceptual.

Como puede apreciarse en el diagrama, la solución constará de:

- **Sistema**, que representa un sistema gestionado como parte de la solución, con capacidad para ejecutar una o más aplicaciones que hayan sido previamente desplegadas en el mismo.
- **Aplicación**, que representa una aplicación modelada por el usuario administrador y utilizada por los usuarios consumidores, capaz de gestionar la información a través de los microservicios automáticamente generados.

- **Elemento del dominio**, representación de los diferentes elementos que pueden utilizar los usuarios administradores para modelar aplicaciones generadas para consumidores; en el siguiente apartado se realizará una breve descripción de cada uno de ellos.
- **Generador de artefactos**, mediante el cual se irán construyendo los diferentes artefactos de los que harán uso las aplicaciones generadas por los usuarios administradores; para ello se utilizarán técnicas de desarrollo dirigido por modelos y programación generativa automática.
- **Generador de esquemas**, mediante el cual se irán construyendo los esquemas de bases de datos de los que harán uso las aplicaciones generadas por los usuarios administradores; para ello se utilizarán técnicas de desarrollo dirigido por modelos y programación generativa automática.

Cabe distinguir en la solución de **Generación Automática de MicroServicios** dos tipos de aplicaciones:

- **Aplicación de administración**: mediante esta aplicación los usuarios administradores modelarán las diferentes aplicaciones generadas para consumidores; la aplicación de administración es única en el sistema.
- **Aplicaciones generadas para consumidores**: son aquellas aplicaciones que han sido construidas por los usuarios administradores haciendo uso de la aplicación de administración; cada aplicación generada puede ser desplegada en uno o varios sistemas.

### 8.3.- Diagrama de clases de alto nivel.

Debido al elevado nivel de abstracción del diagrama conceptual, presentamos a continuación un diagrama de clases de alto nivel, orientado a clarificar de manera funcional los diferentes conceptos que componen el sistema.

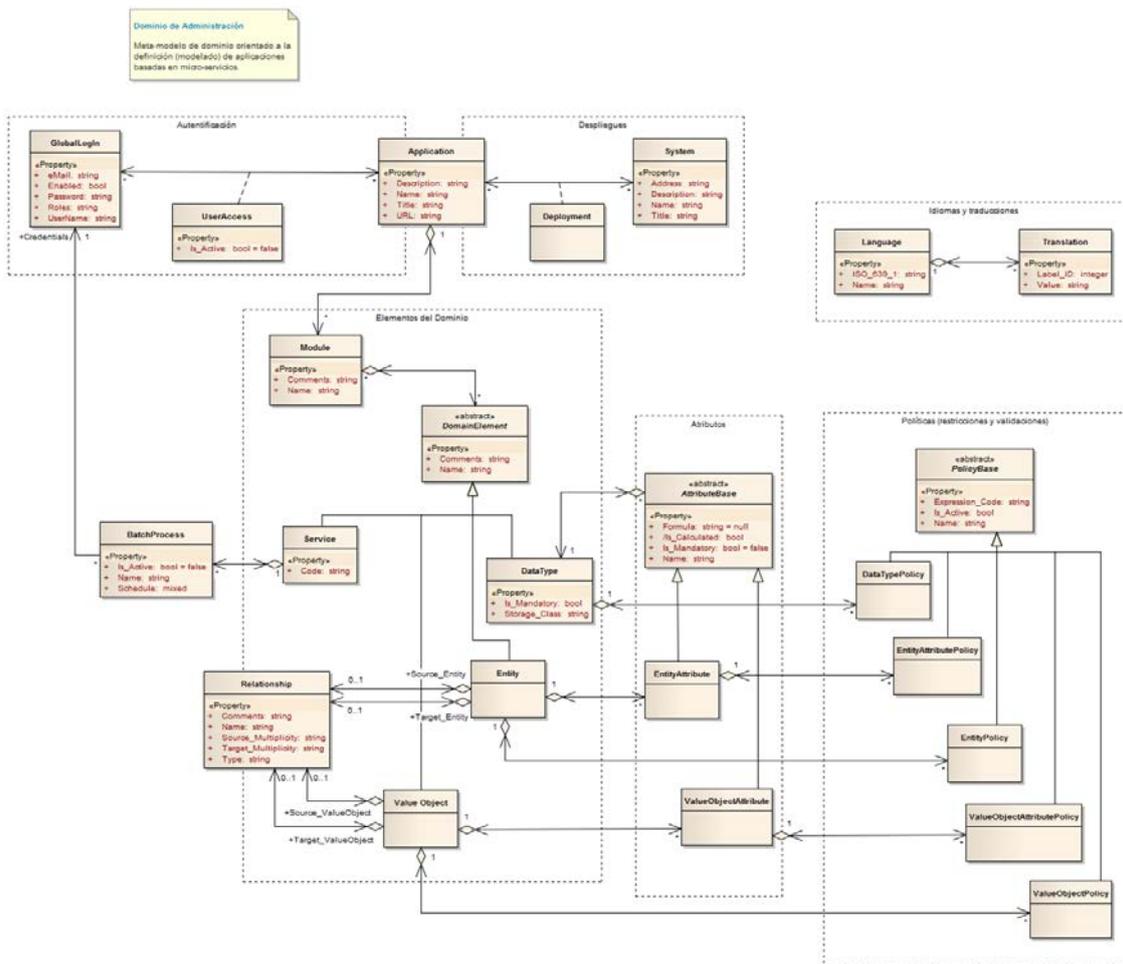


Fig. 8.3.- Diagrama de clases de alto nivel.

Téngase en cuenta que los elementos de dominio de la aplicación de administración deben ser considerados como meta-elementos de las aplicaciones generadas para los consumidores.

Como puede apreciarse en el diagrama anterior, los elementos del dominio ofrecidos por la solución para modelar aplicaciones son los siguientes:

- **Usuario:** credenciales de acceso e información de un usuario, definiendo además si éste se encuentra habilitado o bloqueado, así como su rol dentro del sistema (como administrador o como consumidor).



- **Sistema:** representación de un sistema con capacidad para ejecutar tanto la plataforma GAMS como las aplicaciones generadas para consumidores.
- **Aplicación:** representación de una aplicación generada para consumidores, la cual será ejecutada por la plataforma GAMS.
- **Módulo:** subconjunto de elementos en los que se divide una aplicación.
- **Tipo de Datos:** permite definir el dominio de la información contenida en el valor de un atributo; es decir, delimita el conjunto de valores que puede tomar.
- **Entidad:** concepto modelado con identidad propia.
- **Objeto Valor:** concepto modelado de interés por su representación, su valor, dado que carece de identidad propiamente dicha.
- **Atributo:** característica asociada a una entidad u objeto valor, con capacidad para albergar un valor según un tipo de datos.
- **Relación:** modela la posible asociación existente, dos a dos, entre entidades y objetos valor.
- **Operación:** algoritmo ligado a una entidad.
- **Evento:** algoritmo ligado a una entidad, objeto valor o atributo, que se ejecuta en un determinado momento de su ciclo de vida.
- **Política:** regla de validación que debe cumplir determinados elementos de dominio (como los tipos de datos, las entidades, los objeto valor y los atributos), para que su información (o estado) pueda ser considerado como válido.
- **Servicio:** representación de un algoritmo independiente.
- **Proceso por lotes:** planificación de la ejecución automática de un servicio.

## 8.4.- Diagrama de casos de uso.

Mediante el diagrama de casos de uso, modelamos la funcionalidad del sistema según la percepción de los usuarios del mismo, a quienes denominaremos *actores*.

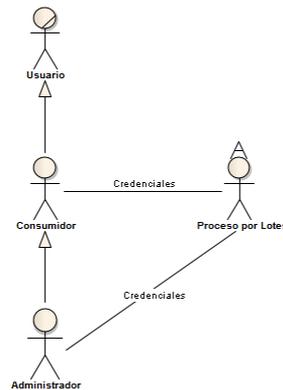


Fig. 8.4.- Actores participantes en la solución GAMS.

La solución de **Generación Automática de MicroServicios** contempla dos actores humanos, los cuales se corresponden a usuarios con el rol consumidor y con el rol administrador, respectivamente; asimismo, contempla que el *sistema* pueda ejecutar de manera automática *servicios*, a través de los denominados *procesos por lotes* (quienes actúan como *agentes*), los cuales personifican la identidad del usuario que haya sido configurado para tal efecto.

El objetivo perseguido con el diagrama presentado a continuación es enumerar tanto a los actores como a los casos de uso, permitiendo identificar qué actores participan en cada caso de uso.

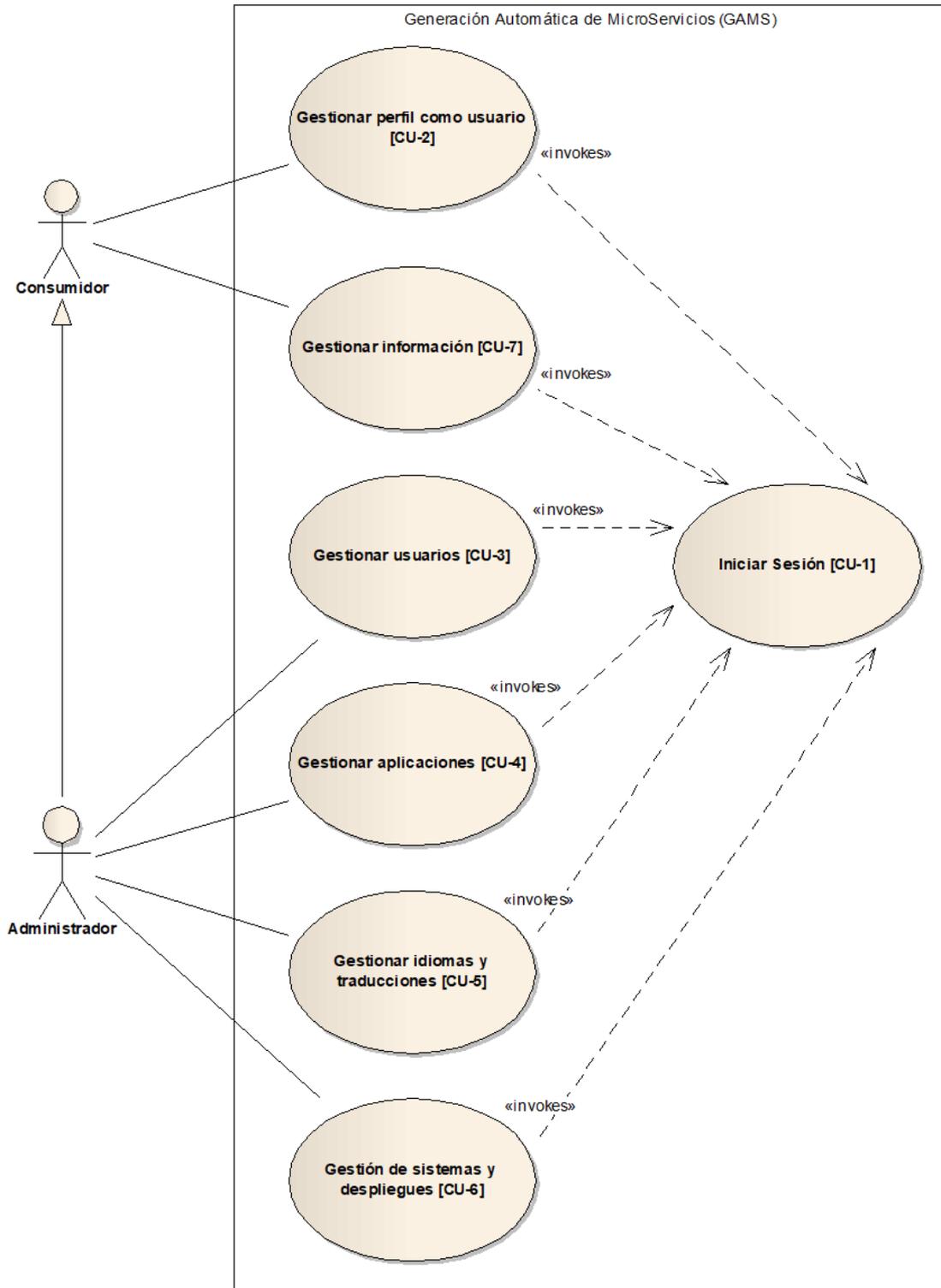


Fig. 8.5.- Vista de casos de uso primarios.

En la vista de casos de uso aparecen representados únicamente los denominados *casos de uso primarios*, es decir, aquellos especialmente relevantes que forman el conjunto principal de acciones realizadas por los usuarios. Además, existen los *casos de uso secundarios*, que son aquéllos que pueden realizarse como parte de algún escenario alternativo de un caso de uso primario.

Obsérvese cómo en fig. 8.5 se muestran dos actores, los cuales representan los roles con los que pueden actuar los usuarios de la solución:

- **Consumidor:** es el usuario que hace uso de las aplicaciones generadas, gestionando la información de las mismas a través de la interfaz gráfica de usuario o a través de los microservicios asociados.
- **Administrador:** es aquel usuario que, siendo también consumidor (y, por tanto, pudiendo realizar las mismas tareas que éste), tiene potestad para utilizar la aplicación de administración, modelando con ella nuevas aplicaciones generadas y desplegándolas en los diferentes sistemas gestionados.

## 8.5.- Casos de uso en formato extendido.

Los casos de uso son unidades de funcionalidad visibles externamente proporcionadas por el sistema (o uno de sus módulos), representadas mediante una secuencia de intercambio de mensajes entre el sistema y los diferentes actores que interactúan con él.

Un caso de uso consta de un *curso típico de eventos*, entendido como el comportamiento normal (o más probable) y puede constar de *escenarios alternativos*, que son posibles variaciones y excepciones del *curso típico* que pueden producirse a lo largo de éste.

Si bien se ha omitido en las descripciones de los casos de uso, por brevedad y concisión, es importante indicar que todos los casos de uso (a excepción de, evidentemente, el de “iniciar sesión”) requieren que el usuario: 1) se haya registrado previamente en la solución y, 2) se haya autenticado antes de poder trabajar con cualquier aplicación. Esto significa que el caso de uso [CU-1] es el primero que todo usuario realizará con el sistema.

Asimismo, en áreas de una mayor brevedad y concisión, no se consideran como escenarios alternativos aquellos casos en los que, introduciendo el usuario información incorrecta o inadecuada, el sistema le informa de este hecho y le permite corregir el dato erróneo antes de continuar. Esto se ha considerado así porque, en la práctica, el sistema se limita simplemente a validar los datos, sin realizar en la práctica ninguna acción adicional.

Se describen a continuación todos los casos de uso, tanto primarios como secundarios, de los que consta el sistema; el código que identifica a cada caso de uso está ordenado jerárquicamente y no de manera lineal.

#### **8.5.1.- Caso de uso: iniciar sesión [CU-1].**

**Tipo:** primario.

**Actor:** consumidor.

**Propósito:** permitir a un usuario autenticarse en el sistema y obtener las autorizaciones pertinentes.

**Referencias:** requisito [R-3].

**Descripción:** el usuario dispondrá de la opción para identificarse; si las credenciales son correctas, el sistema le identificará como administrador o como persona. Desde ese momento, podrá utilizar libremente la aplicación sin necesidad de volver a validarse.

**Curso típico de eventos:**

- 1) El usuario conoce y recuerda sus credenciales (nombre de usuario y contraseña). Las introduce en el control correspondiente.
- 2) El sistema valida dichas credenciales y autoriza, si son correctas, al usuario. Le otorga el rol correspondiente (administrador o persona). Le dirige a la página principal de la aplicación.

#### **Escenario alternativo 1:**

- 1) El usuario ha olvidado la contraseña. Introduce en el control correspondiente únicamente su correo electrónico.
- 2) El sistema le envía por correo electrónico una nueva contraseña con la que realizar la autenticación.
- 3) El usuario, haciendo uso de un enlace único aleatorio recibido como parte del correo electrónico, establece una nueva contraseña con la que realizar la autenticación.
- 4) El sistema le redirige a la página de inicio, para proceder con la autenticación.

#### **Escenario alternativo 2:**

- 1) El usuario introduce incorrectamente sus credenciales.
- 2) El sistema le indica la condición de error y le permite volver a iniciar el proceso de autenticación.

### **8.5.2.- Caso de uso: gestionar perfil como usuario [CU-2].**

**Tipo:** primario.

**Actor:** consumidor.

**Propósito:** permitir a un usuario gestionar los datos de su propio perfil en el sistema.

**Referencias:** requisito [R-3]; casos de uso [CU-2.1], [CU-2.2] y [CU-2.3].

**Descripción:** el usuario podrá gestionar su propio perfil en el sistema, modificando su nombre, correo electrónico o contraseña de acceso, así como cerrar la sesión una vez finalizado su trabajo en el sistema.

**Curso típico de eventos:**

Este caso de uso primario es el nexo de partida de varios casos de uso secundarios y permite al usuario acceder a estos otros casos, por lo que no existe como tal un único escenario. Por ello, nos remitimos a los casos de uso secundarios del apartado referencias para más información. Se presenta a continuación una ampliación de la vista de casos de uso representando estos nexos:

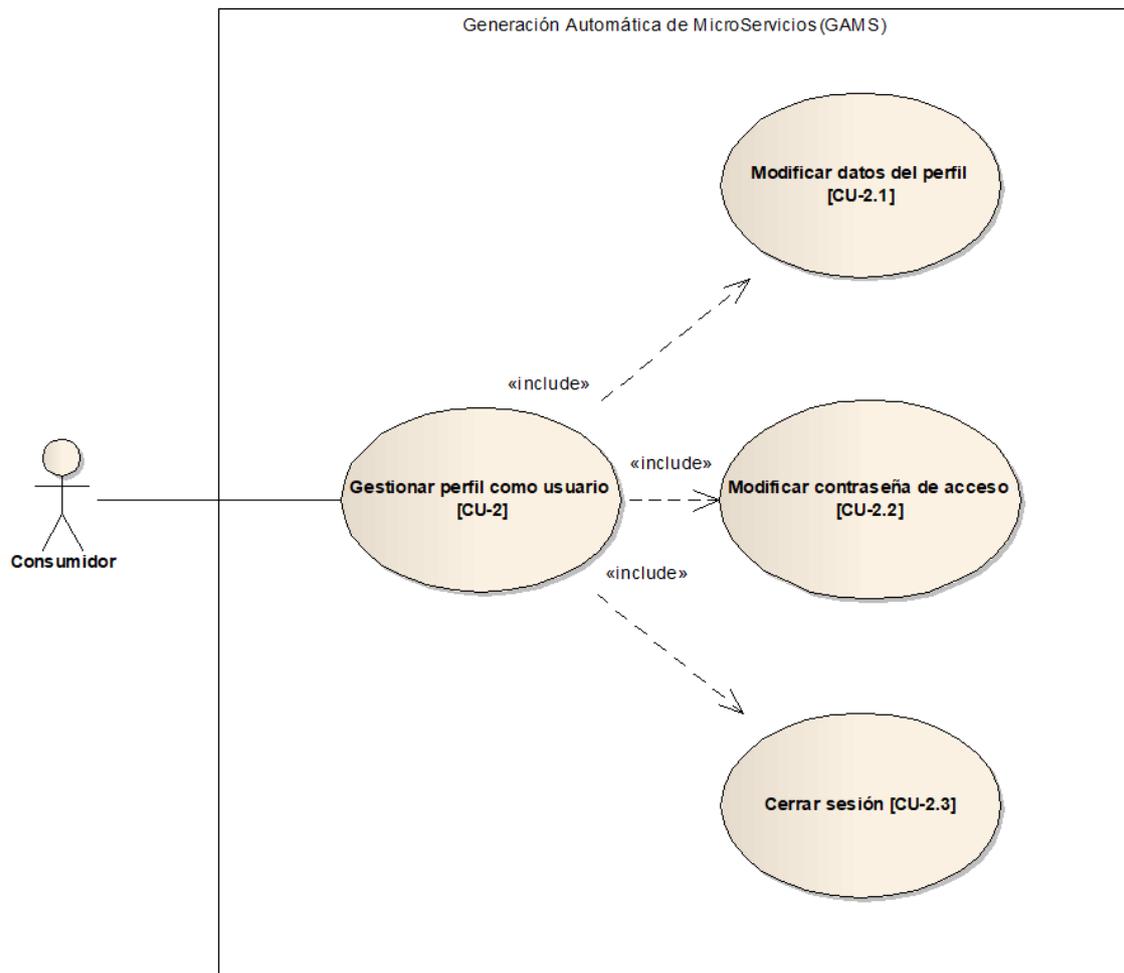


Fig. 8.6.- Casos de uso secundarios de "gestionar perfil como usuario".

### 8.5.3.- Caso de uso: modificar datos del perfil [CU-2.1].

**Tipo:** secundario.

**Actor:** consumidor.

**Propósito:** permitir a un usuario modificar los datos de su propio perfil en el sistema.

**Referencias:** requisito [R-3], caso de uso [CU-2].

**Descripción:** el usuario dispondrá de la opción de modificar tanto su nombre de usuario como su correo electrónico, con los que se identifica en el sistema.

**Curso típico de eventos:**

1) El usuario establece los nuevos valores para los datos “nombre” y “correo electrónico”, los cuales forman parte de su perfil.

2) El sistema verifica que los nuevos valores sean válidos, así como únicos en el sistema; de ser así, actualiza el perfil del usuario y muestra una confirmación por la actualización satisfactoria.

#### **8.5.4.- Caso de uso: modificar contraseña de acceso [CU-2.2].**

**Tipo:** secundario.

**Actor:** consumidor.

**Propósito:** permitir a un usuario modificar su contraseña de acceso al sistema.

**Referencias:** requisito [R-3], caso de uso [CU-2].

**Descripción:** el usuario dispondrá de la opción de modificar su contraseña de acceso al sistema.

**Curso típico de eventos:**

1) El usuario establece su nueva contraseña de acceso al sistema, indicando igualmente la actual.

2) El sistema verifica que la contraseña actual es válida, como mecanismo de seguridad, y establece la nueva contraseña indicada para el acceso.

#### **8.5.5.- Caso de uso: cerrar sesión [CU-2.3].**

**Tipo:** primario.

**Actor:** consumidor.

**Propósito:** permitir a un usuario finalizar su trabajo (sesión actual) en el sistema.

**Referencias:** requisito [R-3], caso de uso [CU-2].

**Descripción:** el usuario dispondrá de la opción para finalizar su sesión y, con ello, dejar de trabajar con el sistema.

**Curso típico de eventos:**

- 1) El usuario solicita, a través de la opción correspondiente de menú, cerrar su sesión actual de trabajo.
- 2) El sistema cierra la sesión y le muestra la “pantalla de inicio”.

**8.5.6.- Caso de uso: gestionar usuarios [CU-3].**

**Tipo:** primario.

**Actor:** administrador.

**Propósito:** permitir a un administrador crear nuevos usuarios, modificar las propiedades y rol de los usuarios existentes, así como eliminar todos los datos del sistema de terceros usuarios.

**Referencias:** requisito [R-3].

**Descripción:** el administrador dispondrá de la opción de eliminar todos los datos de cualquier otro usuario (es decir, no podrá eliminar sus propios datos), establecer el rol asociado (administrador o consumidor), así como admitir o bloquear el acceso tanto al sistema como a cada una de las aplicaciones. Si se opta por eliminar los datos, éstos no podrán recuperarse; el usuario tendrá que ser registrado de nuevo para volver a utilizar el sistema.

**Curso típico de eventos:**

Este caso de uso primario es el nexo de partida de varios casos de uso secundarios y permite al usuario acceder a estos otros casos, por lo que no existe como tal un único escenario. Por ello, nos remitimos a los casos de uso secundarios del apartado referencias para más información. Se presenta a continuación una ampliación de la vista de casos de uso representando estos nexos:

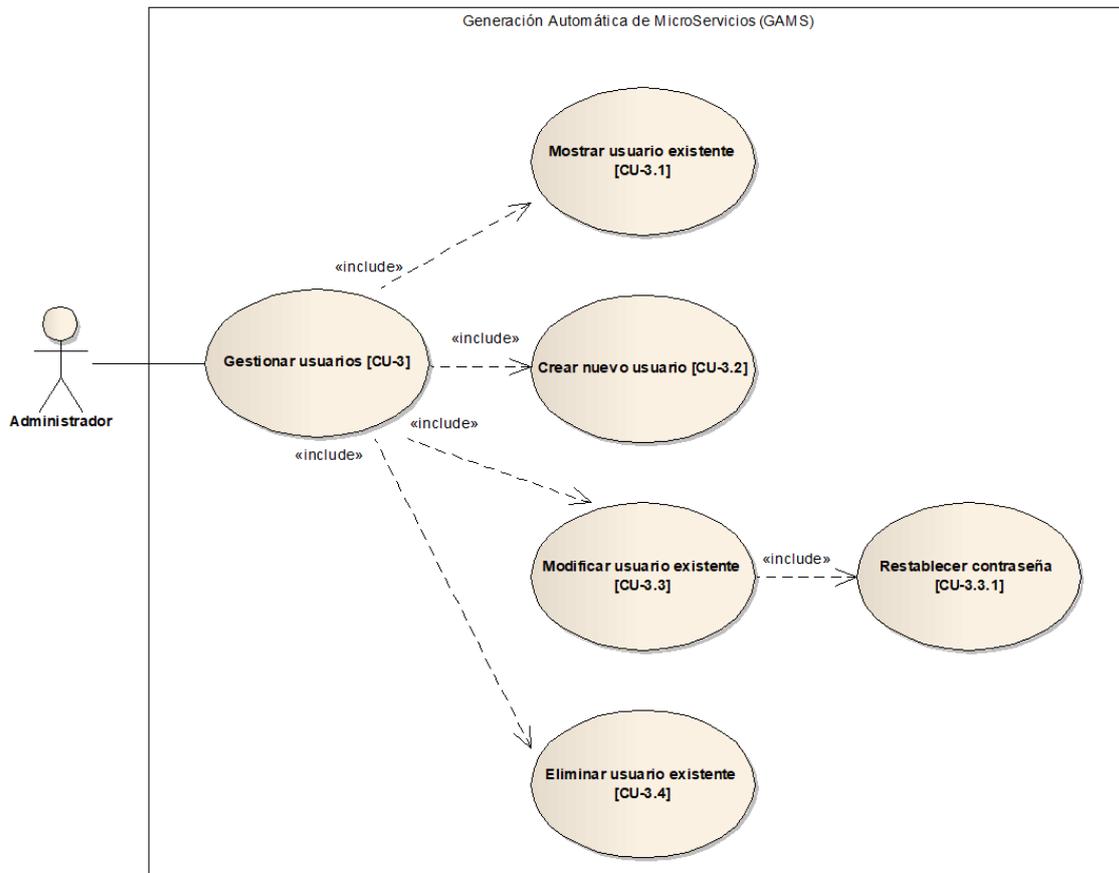


Fig. 8.7.- Casos de uso secundarios de "gestionar usuarios".

### 8.5.7.- Caso de uso: mostrar usuario existente [CU-3.1].

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador consultar la información del usuario seleccionado.

**Referencias:** requisito [R-3], caso de uso [CU-3].

**Descripción:** el administrador dispondrá de la opción de consultar la información y accesos de cualquier usuario, tanto administrador como consumidor, registrado en la solución.

**Curso típico de eventos:**

- 1) El administrador selecciona a uno de los usuarios registrados en la solución.

2) El sistema muestra toda la información relacionada con dicho usuario, tanto sus propiedades como los accesos a las diferentes aplicaciones.

#### **8.5.8.- Caso de uso: crear nuevo usuario [CU-3.2].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador crear (registrar) a un nuevo usuario en la solución.

**Referencias:** requisito [R-3], caso de uso [CU-3].

**Descripción:** el administrador seleccionará “crear nuevo usuario” y establecerá los atributos del mismo (nombre, correo electrónico, activo y rol). Tanto nombre de usuario como el correo electrónico deben ser únicos.

**Curso típico de eventos:**

1) El administrador selecciona “crear nuevo usuario” y establece los atributos del mismo.

2) El sistema verifica la información especificada y, en caso de ser correcta, crea al nuevo usuario.

#### **8.5.9.- Caso de uso: modificar usuario existente [CU-3.3].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador modificar la información asociada a un usuario previamente registrado en la solución.

**Referencias:** requisito [R-3], caso de uso [CU-3].

**Descripción:** el administrador seleccionará “modificar usuario existente” y establecerá nuevos valores a determinados atributos del mismo (nombre, correo electrónico, activo y rol). Tanto el nombre de usuario como el correo electrónico deben seguir siendo únicos.

**Curso típico de eventos:**

1) El administrador selecciona “modificar usuario existente” y establece los atributos del mismo.

2) El sistema verifica la información especificada y, en caso de ser correcta, modificará los datos del usuario especificado.

#### **8.5.10.- Caso de uso: restablecer contraseña de usuario existente [CU-3.3.1].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador restablecer la contraseña de un usuario previamente registrado en la solución.

**Referencias:** requisito [R-3], caso de uso [CU-3.3].

**Descripción:** el administrador seleccionará “restablecer contraseña de usuario existente” y establecerá la nueva contraseña para dicho usuario.

**Curso típico de eventos:**

1) El administrador selecciona “restablecer contraseña de usuario existente” y establece la misma.

2) El sistema modifica la contraseña del usuario especificado.

#### **8.5.11.- Caso de uso: eliminar usuario existente [CU-3.4].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador eliminar toda la información de un usuario previamente registrado en la solución.

**Referencias:** requisito [R-3], caso de uso [CU-3].

**Descripción:** el administrador podrá eliminar cualquier usuario existente, menos a sí mismo; de esta manera se garantiza que siempre exista en el sistema al menos un usuario con el rol administrador.

**Curso típico de eventos:**

1) El administrador selecciona “eliminar usuario existente” y especifica el usuario que deberá ser eliminado.

2) El sistema elimina el registro del usuario y todos los posibles accesos a las diferentes aplicaciones.

#### **8.5.12.- Caso de uso: gestionar aplicaciones [CU-4].**

**Tipo:** primario.

**Actor:** administrador.

**Propósito:** permitir a un administrador gestionar las aplicaciones generadas para consumidores.

**Referencias:** requisitos [R-1] y [R-2]; casos de uso [CU-4.1], [CU-4.2], [CU-4.3], [CU-4.4] y [CU-4.5].

**Descripción:** el administrador dispondrá de las opciones de crear nuevas aplicaciones, así como consultar, modificar, modelar y eliminar las aplicaciones generadas previamente. Si se opta por eliminar una aplicación, tanto sus metadatos como sus datos no podrán recuperarse. Asimismo, se podrán gestionar los accesos de usuario y los despliegues de cada una de las aplicaciones.

**Curso típico de eventos:**

Este caso de uso primario es el nexo de partida de varios casos de uso secundarios y permite al usuario acceder a estos otros casos, por lo que no existe como tal un único escenario. Por ello, nos remitimos a los casos de uso secundarios del apartado referencias para más información. Se presenta a continuación una ampliación de la vista de casos de uso representando estos nexos:

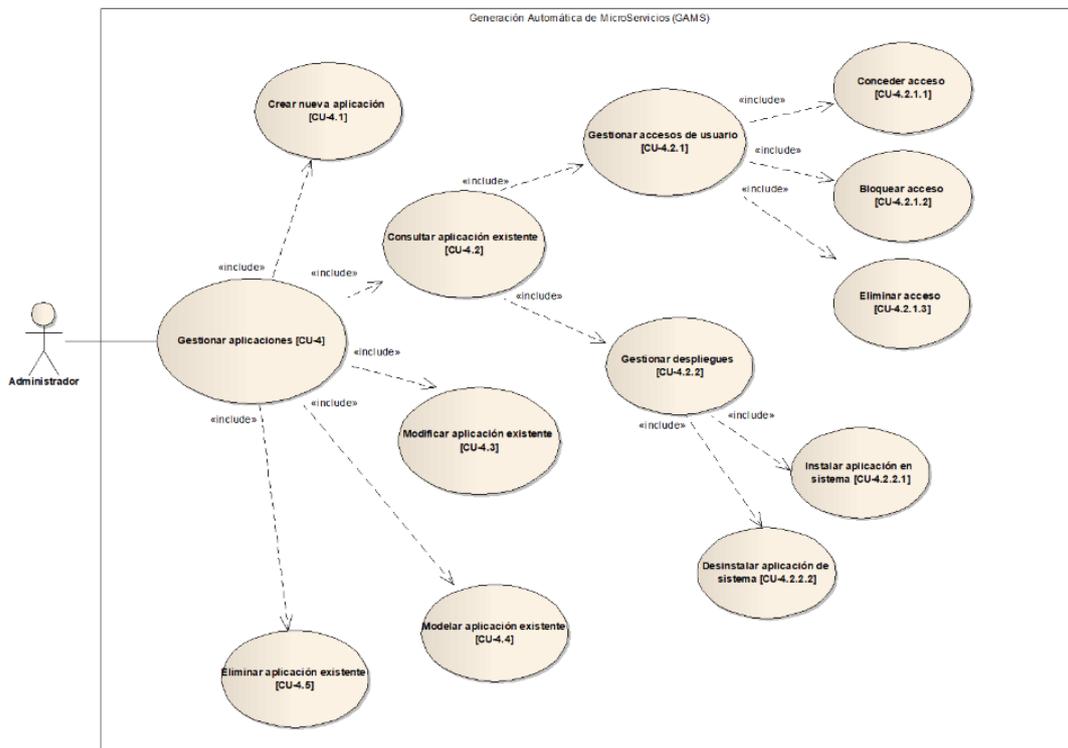


Fig. 8.8.- Casos de uso secundarios de "gestionar aplicaciones".

### 8.5.13.- Caso de uso: crear nueva aplicación [CU-4.1].

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador crear una nueva aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4].

**Descripción:** el administrador seleccionará “crear nueva aplicación” y establecerá los atributos iniciales de la misma (nombre, título y URL). Todos estos atributos deben ser únicos entre las aplicaciones, de manera que los usuarios no se confundan entre dos aplicaciones diferentes.

**Curso típico de eventos:**

1) El administrador selecciona “crear nueva aplicación” y establece los atributos de la misma.

2) El sistema verifica la información especificada y, en caso de ser correcta, crea la nueva aplicación.

#### **8.5.14.- Caso de uso: consultar aplicación existente [CU-4.2].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador consultar la información de la aplicación seleccionada.

**Referencias:** requisitos [R-1], [R-2] y [R-6], casos de uso [CU-4], [CU-4.2.1] y [CU-4.2.2].

**Descripción:** el administrador dispondrá de la opción de consultar la información de la aplicación especificada, incluyendo los accesos de cualquier usuario y los sistemas en los que se encuentre desplegada.

**Curso típico de eventos:**

- 1) El administrador selecciona una de las aplicaciones generadas.
- 2) El sistema muestra toda la información relacionada con dicha aplicación, tanto sus atributos, como los usuarios que tengan algún acceso concedido y los diferentes sistemas en los que se encuentre desplegada.

#### **8.5.15.- Caso de uso: gestionar accesos de usuario [CU-4.2.1].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador gestionar los diferentes accesos de usuario a la aplicación especificada.

**Referencias:** requisitos [R-1], [R-2] y [R-3], casos de uso [CU-4.2], [CU-4.2.1.1], [CU-4.2.1.2.] y [CU-4.2.1.3].

**Descripción:** el administrador dispondrá de la opción de gestionar los diferentes accesos a la aplicación generada por parte de cualquier usuario.

**Curso típico de eventos:**

- 1) El administrador selecciona la opción “gestionar accesos de usuario” para una de las aplicaciones generadas.

2) El sistema muestra la información de los diferentes accesos a la aplicación, así como una relación de todos los usuarios que todavía no tienen acceso.

#### **8.5.16.- Caso de uso: conceder acceso [CU-4.2.1.1].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador conceder un acceso de usuario en una aplicación.

**Referencias:** requisitos [R-1], [R-2] y [R-3], caso de uso [CU-4.2].

**Descripción:** el administrador dispondrá de la opción de conceder un acceso al usuario indicado en la aplicación especificada; dicho acceso podrá estar inicialmente habilitado (operativo) o deshabilitado (bloqueado); el usuario no tendrá, previamente, ningún acceso sobre la aplicación.

##### **Curso típico de eventos:**

1) El administrador selecciona la opción “conceder acceso habilitado” para un usuario concreto sobre una aplicación especificada.

2) La solución concede acceso al usuario en la aplicación especificada; dicho acceso estará habilitado desde dicho momento.

##### **Escenario alternativo 1:**

1) El administrador selecciona la opción “conceder acceso deshabilitado” para un usuario concreto sobre una aplicación especificada.

2) La solución concede acceso al usuario en la aplicación especificada pero dicho acceso estará inicialmente bloqueado.

#### **8.5.17.- Caso de uso: bloquear acceso [CU-4.2.1.2].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador bloquear un acceso de usuario en una aplicación.

**Referencias:** requisitos [R-1], [R-2] y [R-3], caso de uso [CU-4.2].

**Descripción:** el administrador dispondrá de la opción de bloquear un acceso al usuario indicado en la aplicación especificada; dicho acceso debe estar previamente habilitado (operativo).

**Curso típico de eventos:**

- 1) El administrador selecciona la opción “bloquear acceso” para un usuario concreto sobre una aplicación especificada.
- 2) La solución bloquea el acceso al usuario en la aplicación especificada.

**8.5.18.- Caso de uso: eliminar acceso [CU-4.2.1.3].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador eliminar el acceso de un usuario en una aplicación.

**Referencias:** requisitos [R-1], [R-2] y [R-3], caso de uso [CU-4.2].

**Descripción:** el administrador dispondrá de la opción de eliminar un acceso previamente otorgado al usuario indicado en la aplicación especificada.

**Curso típico de eventos:**

- 1) El administrador selecciona la opción “eliminar acceso” para un usuario concreto sobre una aplicación especificada.
- 2) La solución elimina el acceso al usuario en la aplicación especificada; asimismo, eliminará todos los procesos por lotes que dependan de dichas credenciales.

**8.5.19.- Caso de uso: gestionar despliegues [CU-4.2.2].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador gestionar los diferentes despliegues de la aplicación indicada a lo largo de los diferentes sistemas registrados.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6], casos de uso [CU-4.2], [CU-4.2.2.1], y [CU-4.2.2.2].

**Descripción:** el administrador dispondrá de la opción de gestionar los diferentes despliegues de una misma aplicación generada en cualquiera de los sistemas registrados en la solución.

**Curso típico de eventos:**

1) El administrador selecciona la opción “gestionar despliegues” para una de las aplicaciones generadas.

2) El sistema muestra la información de los diferentes sistemas en los que la aplicación ha sido desplegada y en qué condiciones, así como una relación de todos los sistemas en los que todavía no lo ha sido.

**8.5.20.- Caso de uso: instalar aplicación en sistema [CU-4.2.2.1].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador desplegar (instalar) una aplicación generada en un sistema registrado.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.2.2].

**Descripción:** el administrador dispondrá de la opción de desplegar (instalar) la aplicación especificada en el sistema indicado; la aplicación no deberá encontrarse previamente instalada en dicho sistema.

**Curso típico de eventos:**

1) El administrador selecciona la opción “instalar” para la aplicación especificada sobre el sistema indicado.

2) La solución instalará la aplicación especificada en el sistema indicado.

**8.5.21.- Caso de uso: desinstalar aplicación de sistema [CU-4.2.2.2].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador desinstalar una aplicación generada en un sistema registrado.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.2.2].

**Descripción:** el administrador dispondrá de la opción de desinstalar la aplicación especificada del sistema indicado; la aplicación deberá encontrarse previamente instalada en dicho sistema.

**Curso típico de eventos:**

- 1) El administrador selecciona la opción “desinstalar” para la aplicación especificada en el sistema indicado.
- 2) La solución desinstalará la aplicación especificada del sistema indicado.

**8.5.22.- Caso de uso: modificar aplicación existente [CU-4.3].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador modificar la información asociada a una aplicación generada a través de la solución.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4].

**Descripción:** el administrador seleccionará “modificar aplicación existente” y establecerá nuevos valores a determinados atributos del mismo (nombre, título y URL). Todos estos atributos deben ser únicos entre las aplicaciones, de manera que los usuarios no se confundan entre dos aplicaciones diferentes.

**Curso típico de eventos:**

- 1) El administrador selecciona “modificar aplicación existente” y establece los nuevos valores para los atributos de la misma.
- 2) El sistema verifica la información especificada y, en caso de ser correcta, actualiza la información de aplicación especificada.

**8.5.23.- Caso de uso: modelar aplicación existente [CU-4.4].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador modelar el dominio una aplicación generada a través de la solución.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; casos de uso [CU-4], [CU-4.4.1], [CU-4.4.1.1], [CU-4.4.1.2], [CU-4.4.1.3], [CU-4.4.1.4], [CU-4.4.2], [CU-4.4.2.1], [CU-4.4.2.2], [CU-4.4.2.3], [CU-4.4.2.4], [CU-4.4.3], [CU-4.4.3.1], [CU-4.4.3.2], [CU-4.4.3.3], [CU-4.4.3.4], [CU-4.4.4], [CU-4.4.4.1], [CU-4.4.4.2], [CU-4.4.4.3], [CU-4.4.4.4], [CU-4.4.5], [CU-4.4.5.1], [CU-4.4.5.2], [CU-4.4.5.3], [CU-4.4.5.4], [CU-4.4.6], [CU-4.4.6.1], [CU-4.4.6.2], [CU-4.4.6.4], [CU-4.4.7], [CU-4.4.7.1], [CU-4.4.7.2], [CU-4.4.7.3], [CU-4.4.7.4], [CU-4.4.8], [CU-4.4.8.1], [CU-4.4.8.2], [CU-4.4.8.3], [CU-4.4.8.4], [CU-4.4.9], [CU-4.4.9.1], [CU-4.4.9.2], [CU-4.4.9.3], [CU-4.4.9.4], [CU-4.4.10], [CU-4.4.10.1], [CU-4.4.10.2], [CU-4.4.10.3], [CU-4.4.10.4], [CU-4.4.11], [CU-4.4.11.1], [CU-4.4.11.2], [CU-4.4.11.3], [CU-4.4.11.4], [CU-4.4.12], [CU-4.4.12.1], [CU-4.4.12.2], [CU-4.4.12.3], [CU-4.4.12.4], [CU-4.4.13], [CU-4.4.13.1], [CU-4.4.13.2], [CU-4.4.13.3], [CU-4.4.13.4], [CU-4.4.14], [CU-4.4.14.1], [CU-4.4.14.2], [CU-4.4.14.3] y [CU-4.4.14.4].

**Descripción:** el administrador seleccionará “modelar aplicación existente” y diseñará el modelo de dominio de la aplicación especificada.

**Curso típico de eventos:**

Este caso de uso primario es el nexo de partida de varios casos de uso secundarios y permite al usuario acceder a estos otros casos, por lo que no existe como tal un único escenario. Por ello, nos remitimos a los casos de uso secundarios del apartado referencias para más información. Se presenta a continuación una ampliación de la vista de casos de uso representando estos nexos:

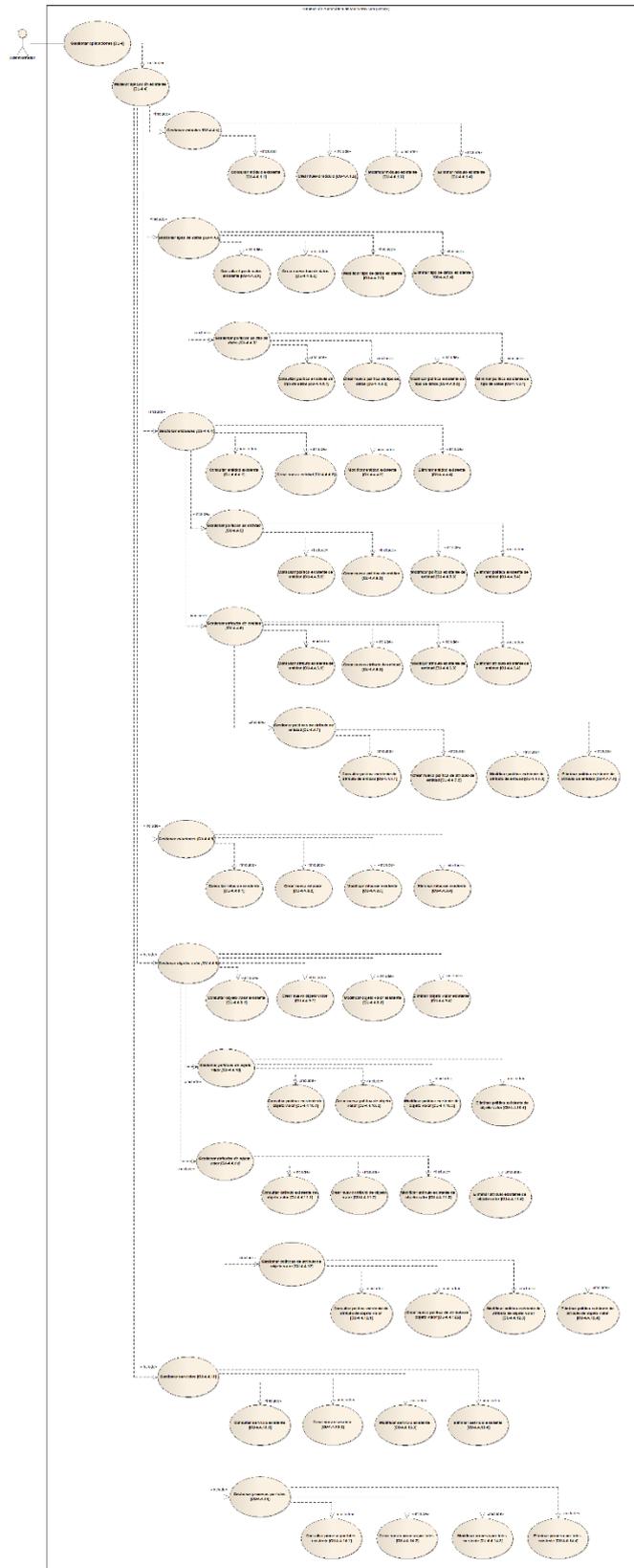


Fig. 8.9.- Diagrama de casos de uso secundarios de “modelar aplicación existente”.

#### 8.5.24.- Caso de uso: gestionar módulos [CU-4.4.1].

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador gestionar los módulos utilizados como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; casos de uso [CU-4.4], [CU-4.4.1.1], [CU-4.4.1.2], [CU-4.4.1.3] y [CU-4.4.1.4].

**Descripción:** el administrador dispondrá de la opción de crear nuevos módulos, así como consultar, modificar y eliminar los módulos modelados previamente. Si se opta por eliminar un módulo, todos los elementos contenidos en el mismo se eliminarán y no podrán recuperarse, así como los datos de usuario asociados a dichos elementos.

**Curso típico de eventos:**

Este caso de uso secundario es el nexo de partida de varios casos de uso secundarios y permite al usuario acceder a estos otros casos, por lo que no existe como tal un único escenario. Por ello, nos remitimos a los casos de uso secundarios del apartado referencias para más información.

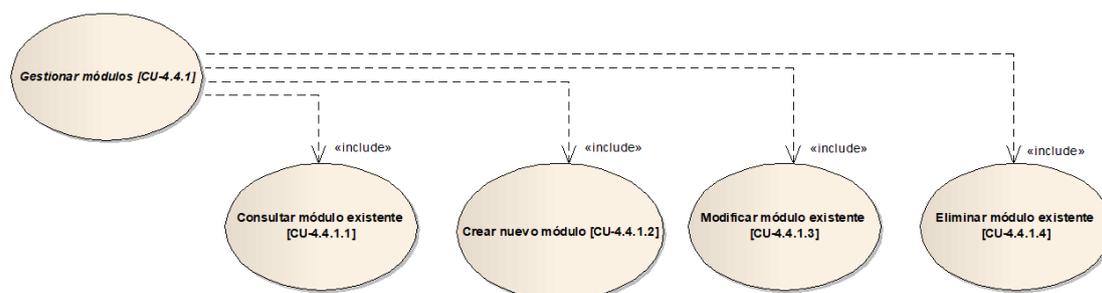


Fig. 8.10.- Casos de uso secundarios de "gestionar módulos".

#### 8.5.25.- Caso de uso: consultar módulo existente [CU-4.4.1.1]

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador consultar la información del módulo seleccionado.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.1].

**Descripción:** el administrador dispondrá de la opción de consultar la información del módulo especificado.

**Curso típico de eventos:**

1) El administrador selecciona uno de los módulos que forman parte del dominio de la aplicación generada que está siendo modelada.

2) El sistema muestra toda la información relacionada con dicho módulo.

#### **8.5.26.- Caso de uso: crear nuevo módulo [CU-4.4.1.2].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador crear un nuevo módulo, como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.1].

**Descripción:** el administrador seleccionará “crear nuevo módulo” y establecerá los atributos iniciales del mismo (nombre). No se permite que dos módulos de la misma aplicación tengan el mismo nombre, de manera que los usuarios no puedan confundirlos.

**Curso típico de eventos:**

1) El administrador selecciona “crear nuevo módulo” y establece los atributos del mismo.

2) El sistema verifica la información especificada y, en caso de ser correcta, crea el nuevo módulo.

#### **8.5.27.- Caso de uso: modificar módulo existente [CU-4.4.1.3].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador modificar un módulo existente, como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.1].

**Descripción:** el administrador seleccionará “modificar módulo existente” y actualizará las propiedades que considere del mismo (solamente es modificable el atributo “nombre”). No se permite que dos módulos de la misma aplicación tengan el mismo nombre, de manera que los usuarios no puedan confundirlos.

**Curso típico de eventos:**

1) El administrador selecciona “modificar módulo existente” y establece los nuevos valores para las propiedades del módulo seleccionado.

2) El sistema verifica la información especificada y, en caso de ser correcta, actualiza las propiedades del módulo seleccionado con los valores modificados.

**8.5.28.- Caso de uso: eliminar módulo existente [CU-4.4.1.4].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador eliminar un módulo existente, como parte del modelado del dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.1].

**Descripción:** el administrador podrá eliminar cualquier módulo existente, eliminando consigo todos y cada uno de los diferentes elementos contenidos en dicho módulo, como puedan ser entidades, atributos o políticas, entre otros. Si se opta por eliminar un módulo, tanto el esquema (metadatos) como la información (datos de usuario) asociados a dicho módulo (así como de sus elementos contenidos) se borrarán y no podrán recuperarse.

**Curso típico de eventos:**

1) El administrador selecciona “eliminar módulo existente” y especifica el módulo a eliminar.

2) El sistema elimina toda la información del modelo de dominio, así como de los datos de usuario, tanto del módulo seleccionado como de todos y cada uno de los diferentes elementos contenidos en el mismo.

### 8.5.29.- Caso de uso: gestionar tipos de datos [CU-4.4.2].

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador gestionar los tipos de datos utilizados como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; casos de uso [CU-4.4], [CU-4.4.2.1], [CU-4.4.2.2], [CU-4.4.2.3], [CU-4.4.2.4] y [CU-4.4.3].

**Descripción:** el administrador dispondrá de la opción de crear nuevos tipos de datos, así como consultar, modificar y eliminar los tipos de datos modelados previamente. No se permitirá eliminar un tipo de datos mientras se hayan modelado atributos que dependan del mismo; si se opta por eliminar un tipo de datos, todas sus políticas serán igualmente eliminadas; una vez que un tipo de datos haya sido eliminado, ni éste ni sus políticas podrán recuperarse.

#### **Curso típico de eventos:**

Este caso de uso secundario es el nexo de partida de varios casos de uso secundarios y permite al usuario acceder a estos otros casos, por lo que no existe como tal un único escenario. Por ello, nos remitimos a los casos de uso secundarios del apartado referencias para más información.

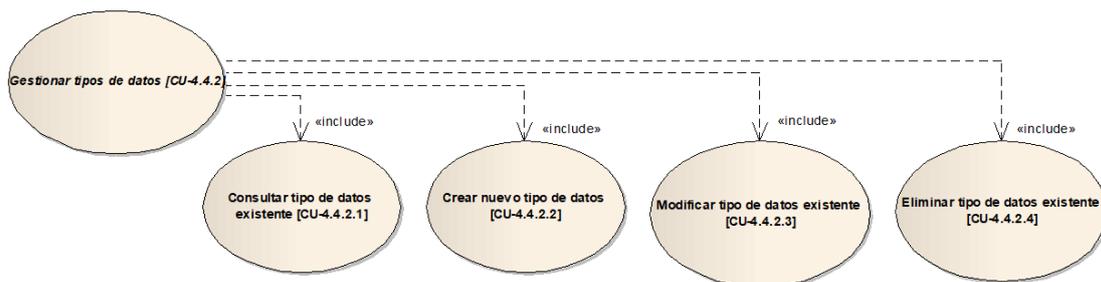


Fig. 8.11.- Casos de uso secundarios de “gestionar tipos de datos”.

### 8.5.30.- Caso de uso: consultar tipo de datos existente [CU-4.4.2.1].

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador consultar la información del tipo de datos seleccionado.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.2].

**Descripción:** el administrador dispondrá de la opción de consultar la información del tipo de datos especificado.

**Curso típico de eventos:**

- 1) El administrador selecciona uno de los tipos de datos que forman parte del dominio de la aplicación generada que está siendo modelada.
- 2) El sistema muestra toda la información relacionada con dicho tipo de datos.

#### **8.5.31.- Caso de uso: crear nuevo tipo de datos [CU-4.4.2.2].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador crear un nuevo tipo de datos, como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.2].

**Descripción:** el administrador seleccionará “crear nuevo tipo de datos” y establecerá los atributos iniciales del mismo (nombre, almacenamiento y obligatoriedad). No se permite que dos tipos de datos pertenecientes al mismo módulo tengan el mismo nombre, impidiendo así que los usuarios puedan confundirlos.

**Curso típico de eventos:**

- 1) El administrador selecciona “crear nuevo tipo de datos” y establece los atributos del mismo.
- 2) El sistema verifica la información especificada y, en caso de ser correcta, crea el nuevo tipo de datos.

#### **8.5.32.- Caso de uso: modificar tipo de datos existente [CU-4.4.2.3].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador modificar un tipo de datos existente, como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.2].

**Descripción:** el administrador seleccionará “modificar tipo de datos existente” y actualizará las propiedades que considere del mismo (nombre y obligatoriedad, el almacenamiento no podrá alterarse). No se permite que dos tipos de datos pertenecientes al mismo módulo tengan el mismo nombre, impidiendo así que los usuarios puedan confundirlos.

**Curso típico de eventos:**

1) El administrador selecciona “modificar tipo de datos existente” y establece los nuevos valores para las propiedades del tipo de datos seleccionado.

2) El sistema verifica la información especificada y, en caso de ser correcta, actualiza las propiedades del tipo de datos seleccionado con los valores modificados.

**8.5.33.- Caso de uso: eliminar tipo de datos existente [CU-4.4.2.4].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador eliminar un tipo de datos existente, como parte del modelado del dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.2].

**Descripción:** el administrador podrá eliminar cualquier tipo de datos existente que no esté siendo referenciado (utilizado) por ningún atributo del modelo, ya sea un atributo de entidad o un atributo de objeto valor.

**Curso típico de eventos:**

1) El administrador selecciona “eliminar tipo de datos existente” y especifica el tipo de datos a eliminar.

2) El sistema verifica que el tipo de datos no esté siendo referenciado por ningún atributo del modelo y elimina toda la información de dicho tipo de datos en el modelo de dominio.

### 8.5.34.- Caso de uso: gestionar políticas de tipo de datos [CU-4.4.3].

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador gestionar las políticas de un tipo de datos, utilizadas como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; casos de uso [CU-4.4], [CU-4.4.2], [CU-4.4.3.1], [CU-4.4.3.2], [CU-4.4.3.3] y [CU-4.4.3.4].

**Descripción:** el administrador dispondrá de la opción de crear nuevas políticas de tipos de datos, así como consultar, modificar y eliminar las políticas modeladas previamente. Si se opta por eliminar una política de tipo de datos, ésta no podrá recuperarse.

**Curso típico de eventos:**

Este caso de uso secundario es el nexo de partida de varios casos de uso secundarios y permite al usuario acceder a estos otros casos, por lo que no existe como tal un único escenario. Por ello, nos remitimos a los casos de uso secundarios del apartado referencias para más información.

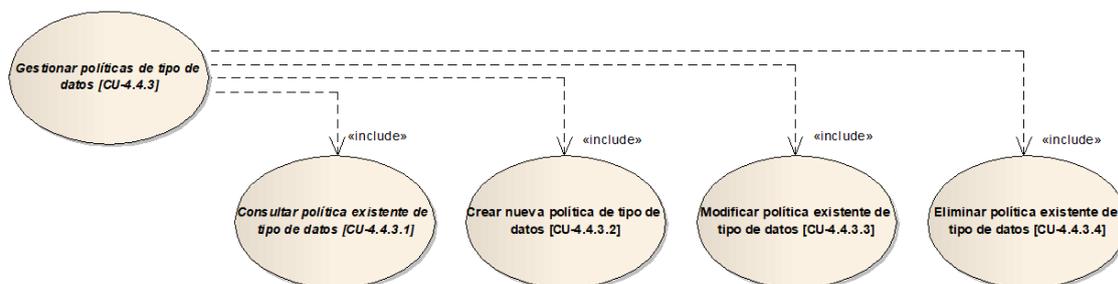


Fig. 8.12.- Casos de uso secundarios de “gestionar políticas de tipo de datos”.

### 8.5.35.- Caso de uso: consultar política existente de tipo de datos [CU-4.4.3.1].

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador consultar la información de la política de tipo de datos seleccionada.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.3].

**Descripción:** el administrador dispondrá de la opción de consultar la información de la política de tipo de datos especificada.

**Curso típico de eventos:**

1) El administrador selecciona una de las políticas de tipo de datos que forman parte del dominio de la aplicación generada que está siendo modelada.

2) El sistema muestra toda la información relacionada con dicha política de tipo de datos.

#### **8.5.36.- Caso de uso: crear nueva política de tipo de datos [CU-4.4.3.2].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador crear una nueva política de tipo de datos, como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.3].

**Descripción:** el administrador seleccionará “crear nueva política de tipo de datos” y establecerá los atributos iniciales de la misma (nombre, está activa y código de validación). No se permite que dos políticas de tipos de datos pertenecientes al mismo tipo de datos tengan el mismo nombre, impidiendo así que los usuarios puedan confundirlas.

**Curso típico de eventos:**

1) El administrador selecciona “crear nueva política de tipo de datos” y establece los atributos de la misma.

2) El sistema verifica la información especificada y, en caso de ser correcta, crea la nueva política de tipo de datos.

#### **8.5.37.- Caso de uso: modificar política existente de tipo de datos [CU-4.4.3.3].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador modificar una política de tipo de datos existente, como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.3].

**Descripción:** el administrador seleccionará “modificar política de tipo de datos existente” y actualizará las propiedades que considere de la misma (nombre, está activa y código de validación). No se permite que dos políticas del mismo tipo de datos tengan el mismo nombre, de manera que los usuarios no puedan confundirlas.

**Curso típico de eventos:**

1) El administrador selecciona “modificar política de tipo de datos existente” y establece los nuevos valores para las propiedades de la política seleccionada.

2) El sistema verifica la información especificada y, en caso de ser correcta, actualiza las propiedades de la política de tipos de datos seleccionada con los valores modificados.

#### **8.5.38.- Caso de uso: eliminar política existente de tipo de datos [CU-4.4.3.4].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador eliminar una política de tipo de datos existente, como parte del modelado del dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.3].

**Descripción:** el administrador podrá eliminar cualquier política de tipo de datos existente.

**Curso típico de eventos:**

1) El administrador selecciona “eliminar política de tipo de datos existente” y especifica la política de tipo de datos a eliminar.

2) El sistema elimina toda la información de dicha política de tipo de datos en el modelo de dominio.

### 8.5.39.- Caso de uso: gestionar entidades [CU-4.4.4].

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador gestionar las entidades utilizadas como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; casos de uso [CU-4.4], [CU-4.4.4.1], [CU-4.4.4.2], [CU-4.4.4.3], [CU-4.4.4.4], [CU-4.4.5] y [CU-4.4.6].

**Descripción:** el administrador dispondrá de la opción de crear nuevas entidades, así como consultar, modificar y eliminar las entidades modeladas previamente. Si se opta por eliminar una entidad, todos los elementos contenidos en la misma (políticas, atributos y relaciones) se eliminarán y no podrán recuperarse, así como los datos de usuario asociados a dichos elementos.

#### **Curso típico de eventos:**

Este caso de uso secundario es el nexo de partida de varios casos de uso secundarios y permite al usuario acceder a estos otros casos, por lo que no existe como tal un único escenario. Por ello, nos remitimos a los casos de uso secundarios del apartado referencias para más información.

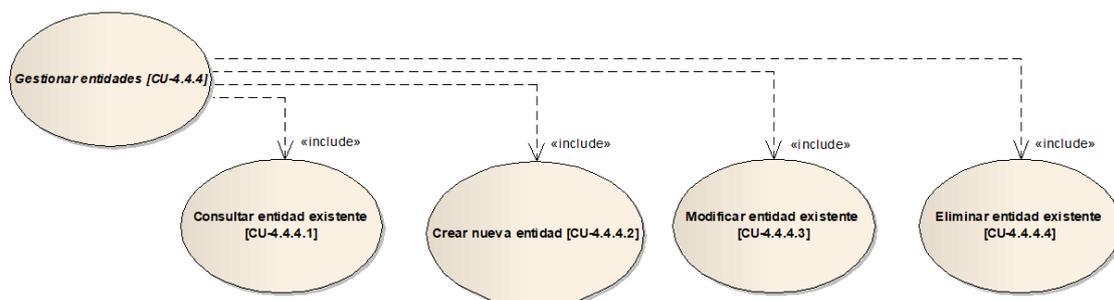


Fig. 8.13.- Casos de uso secundarios de “gestionar entidades”.

### 8.5.40.- Caso de uso: consultar entidad existente [CU-4.4.4.1].

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador consultar la información de la entidad seleccionada.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.4].

**Descripción:** el administrador dispondrá de la opción de consultar la información de la entidad especificada.

**Curso típico de eventos:**

1) El administrador selecciona una de las entidades que forman parte del dominio de la aplicación generada que está siendo modelada.

2) El sistema muestra toda la información relacionada con dicha entidad.

#### **8.5.41.- Caso de uso: crear nueva entidad [CU-4.4.2].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador crear una nueva entidad, como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.4].

**Descripción:** el administrador seleccionará “crear nueva entidad” y establecerá los atributos iniciales de la misma (solamente “nombre”). No se permite que dos entidades pertenecientes al mismo módulo tengan el mismo nombre, impidiendo así que los usuarios puedan confundirlas.

**Curso típico de eventos:**

1) El administrador selecciona “crear nueva entidad” y establece los atributos de la misma.

2) El sistema verifica la información especificada y, en caso de ser correcta, crea la nueva entidad.

#### **8.5.42.- Caso de uso: modificar entidad existente [CU-4.4.3].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador modificar una entidad existente, como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.4].

**Descripción:** el administrador seleccionará “modificar entidad existente” y actualizará las propiedades que considere de la misma (solamente “nombre”). No se permite que dos entidades pertenecientes al mismo módulo tengan el mismo nombre, de manera que los usuarios no puedan confundirlas.

**Curso típico de eventos:**

1) El administrador selecciona “modificar entidad existente” y establece los nuevos valores para las propiedades de la entidad seleccionada.

2) El sistema verifica la información especificada y, en caso de ser correcta, actualiza las propiedades de la entidad seleccionada con los valores modificados.

**8.5.43.- Caso de uso: eliminar entidad existente [CU-4.4.4.4].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador eliminar una entidad existente, como parte del modelado del dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.4].

**Descripción:** el administrador podrá eliminar cualquier entidad existente, eliminando consigo a todos y cada uno de los diferentes elementos contenidos en dicha entidad, como son: atributos, políticas y relaciones. Si se opta por eliminar una entidad, tanto el esquema (metadatos) como la información (datos de usuario) asociados a dicha entidad (y de sus elementos contenidos) se borrarán y no podrán recuperarse.

**Curso típico de eventos:**

1) El administrador selecciona “eliminar entidad existente” y especifica la entidad a eliminar.

2) El sistema elimina toda la información del modelo de dominio, así como de los datos de usuario, tanto de la entidad seleccionada como de todos y cada uno de los diferentes elementos contenidos en la misma.

#### 8.5.44.- Caso de uso: gestionar políticas de entidad [CU-4.4.5].

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador gestionar las políticas de entidad utilizadas como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; casos de uso [CU-4.4], [CU-4.4.5.1], [CU-4.4.5.2], [CU-4.4.5.3] y [CU-4.4.5.4].

**Descripción:** el administrador dispondrá de la opción de crear nuevas políticas de entidad, así como consultar, modificar y eliminar las políticas de entidad modeladas previamente. Si se opta por eliminar una política de entidad, todas sus propiedades se eliminarán y no podrán recuperarse.

##### **Curso típico de eventos:**

Este caso de uso secundario es el nexo de partida de varios casos de uso secundarios y permite al usuario acceder a estos otros casos, por lo que no existe como tal un único escenario. Por ello, nos remitimos a los casos de uso secundarios del apartado referencias para más información.

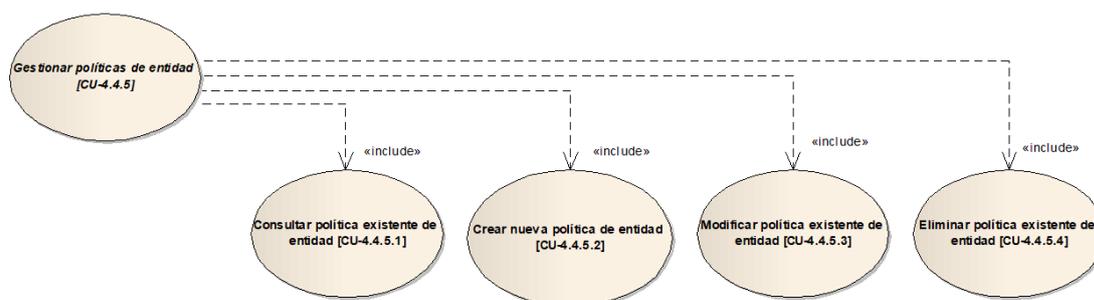


Fig. 8.14.- Casos de uso secundarios de “gestionar políticas de entidad”.

#### 8.5.45.- Caso de uso: consultar política existente de entidad [CU-4.4.5.1].

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador consultar la información de la política de entidad seleccionada.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.5].

**Descripción:** el administrador dispondrá de la opción de consultar la información de la política de entidad especificada.

**Curso típico de eventos:**

1) El administrador selecciona una de las políticas de entidad que forman parte del dominio de la aplicación generada que está siendo modelada.

2) El sistema muestra toda la información relacionada con dicha política de entidad.

#### **8.5.46.- Caso de uso: crear nueva política de entidad [CU-4.4.5.2].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador crear una nueva política de entidad, como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.5].

**Descripción:** el administrador seleccionará “crear nueva política de entidad” y establecerá los atributos iniciales de la misma (nombre, está activa y código de validación). No se permite que dos políticas de entidad pertenecientes a la misma entidad tengan el mismo nombre, impidiendo así que los usuarios puedan confundirlas.

**Curso típico de eventos:**

1) El administrador selecciona “crear nueva política de entidad” y establece los atributos de la misma.

2) El sistema verifica la información especificada y, en caso de ser correcta, crea la nueva política de entidad.

#### **8.5.47.- Caso de uso: modificar política existente de entidad [CU-4.4.5.3].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador modificar una política de entidad existente, como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.5].

**Descripción:** el administrador seleccionará “modificar política de entidad existente” y actualizará las propiedades que considere de la misma (nombre, está activa y código de validación). No se permite que dos políticas de entidad pertenecientes a la misma entidad tengan el mismo nombre, de manera que los usuarios no puedan confundirlas.

**Curso típico de eventos:**

1) El administrador selecciona “modificar política de entidad existente” y establece los nuevos valores para las propiedades de la política seleccionada.

2) El sistema verifica la información especificada y, en caso de ser correcta, actualiza las propiedades de la política de entidad seleccionada con los valores modificados.

**8.5.48.- Caso de uso: eliminar política existente de entidad [CU-4.4.5.4].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador eliminar una política de entidad existente, como parte del modelado del dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.5].

**Descripción:** el administrador podrá eliminar cualquier política de entidad existente.

**Curso típico de eventos:**

1) El administrador selecciona “eliminar política de entidad existente” y especifica la política de entidad a eliminar.

2) El sistema elimina toda la información de dicha política de entidad en el modelo de dominio.

### 8.5.49.- Caso de uso: gestionar atributos de entidad [CU-4.4.6].

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador gestionar los atributos de entidad utilizados como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; casos de uso [CU-4.4], [CU-4.4.6.1], [CU-4.4.6.2], [CU-4.4.6.3] y [CU-4.4.6.4].

**Descripción:** el administrador dispondrá de la opción de crear nuevos atributos de entidad, así como consultar, modificar y eliminar los atributos de entidad modelados previamente. Si se opta por eliminar un atributo de entidad, tanto sus propiedades como los datos del usuario asociados a dicho atributo se eliminarán y no podrán recuperarse.

#### **Curso típico de eventos:**

Este caso de uso secundario es el nexo de partida de varios casos de uso secundarios y permite al usuario acceder a estos otros casos, por lo que no existe como tal un único escenario. Por ello, nos remitimos a los casos de uso secundarios del apartado referencias para más información.

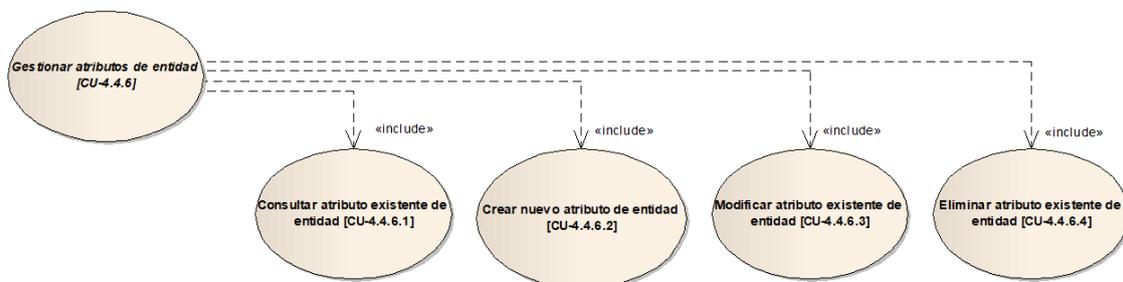


Fig. 8.15.- Casos de uso secundarios de “gestionar atributos de entidad”.

### 8.5.50.- Caso de uso: consultar atributo existente de entidad [CU-4.4.6.1].

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador consultar la información del atributo de entidad seleccionado.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.6].

**Descripción:** el administrador dispondrá de la opción de consultar la información del atributo de entidad especificado.

**Curso típico de eventos:**

1) El administrador selecciona uno de los atributos de entidad que forman parte del dominio de la aplicación generada que está siendo modelada.

2) El sistema muestra toda la información relacionada con dicho atributo de entidad.

#### **8.5.51.- Caso de uso: crear nuevo atributo de entidad [CU-4.4.6.2].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador crear un nuevo atributo de entidad, como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.6].

**Descripción:** el administrador seleccionará “crear nuevo atributo de entidad” y establecerá los, valga la redundancia, atributos iniciales del mismo (nombre, tipo de datos, obligatoriedad y código de la fórmula en caso de atributos calculados). No se permite que dos atributos de entidad pertenecientes a la misma entidad tengan el mismo nombre, impidiendo así que los usuarios puedan confundirlos.

**Curso típico de eventos:**

1) El administrador selecciona “crear nuevo atributo de entidad” y establece las propiedades del mismo.

2) El sistema verifica la información especificada y, en caso de ser correcta, crea el nuevo atributo de entidad.

#### **8.5.52.- Caso de uso: modificar atributo existente de entidad [CU-4.4.6.3].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador modificar un atributo existente de entidad, como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.6].

**Descripción:** el administrador seleccionará “modificar atributo existente de entidad” y actualizará las propiedades que considere del mismo (solamente “nombre”). No se permite que dos atributos pertenecientes a la misma entidad tengan el mismo nombre, impidiendo así que los usuarios puedan confundirlos.

**Curso típico de eventos:**

1) El administrador selecciona “modificar atributo existente de entidad” y establece los nuevos valores para las propiedades del atributo de entidad seleccionado.

2) El sistema verifica la información especificada y, en caso de ser correcta, actualiza las propiedades del atributo de entidad seleccionado con los valores modificados.

#### **8.5.53.- Caso de uso: eliminar atributo existente de entidad [CU-4.4.6.4].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador eliminar un atributo existente de entidad, como parte del modelado del dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.6].

**Descripción:** el administrador podrá eliminar cualquier atributo existente de entidad.

**Curso típico de eventos:**

1) El administrador selecciona “eliminar atributo existente de entidad” y especifica el atributo de entidad a eliminar.

2) El sistema elimina toda la información de dicho atributo de entidad, tanto en el modelo de dominio (metadatos) como a nivel de información (datos de usuario).

#### **8.5.54.- Caso de uso: gestionar políticas de atributo de entidad [CU-4.4.7].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador gestionar las políticas de atributo de entidad utilizadas como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; casos de uso [CU-4.4], [CU-4.4.7.1], [CU-4.4.7.2], [CU-4.4.7.3] y [CU-4.4.7.4].

**Descripción:** el administrador dispondrá de la opción de crear nuevas políticas de atributo de entidad, así como consultar, modificar y eliminar las políticas de atributo de entidad modeladas previamente. Si se opta por eliminar una política de atributo de entidad, todas sus propiedades se eliminarán y no podrán recuperarse.

#### Curso típico de eventos:

Este caso de uso secundario es el nexo de partida de varios casos de uso secundarios y permite al usuario acceder a estos otros casos, por lo que no existe como tal un único escenario. Por ello, nos remitimos a los casos de uso secundarios del apartado referencias para más información.

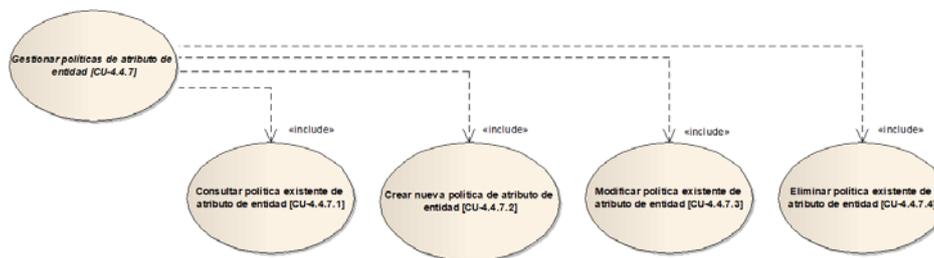


Fig. 8.16.- Casos de uso secundarios de “gestionar políticas de atributo de entidad”.

#### 8.5.55.- Caso de uso: consultar política existente de atributo de entidad [CU-4.4.7.1].

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador consultar la información de la política de atributo de entidad seleccionada.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.7].



**Descripción:** el administrador dispondrá de la opción de consultar la información de la política de atributo de entidad especificada.

**Curso típico de eventos:**

- 1) El administrador selecciona una de las políticas de atributo de entidad que forman parte del dominio de la aplicación generada que está siendo modelada.
- 2) El sistema muestra toda la información relacionada con dicha política de atributo de entidad.

**8.5.56.- Caso de uso: crear nueva política de atributo de entidad [CU-4.4.7.2].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador crear una nueva política de atributo de entidad, como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.7].

**Descripción:** el administrador seleccionará “crear nueva política de atributo de entidad” y establecerá los atributos iniciales de la misma (nombre, está activa y código de validación). No se permite que dos políticas de atributo de entidad pertenecientes al mismo atributo de entidad tengan el mismo nombre, impidiendo así que los usuarios puedan confundirlas.

**Curso típico de eventos:**

- 1) El administrador selecciona “crear nueva política de atributo de entidad” y establece los atributos de la misma.
- 2) El sistema verifica la información especificada y, en caso de ser correcta, crea la nueva política de atributo de entidad.

**8.5.57.- Caso de uso: modificar política existente de atributo de entidad [CU-4.4.7.3].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador modificar una política existente de atributo, como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.7].

**Descripción:** el administrador seleccionará “modificar política existente de atributo” y actualizará las propiedades que considere de la misma (nombre, está activa y código de validación). No se permite que dos políticas de atributo pertenecientes al mismo atributo tengan el mismo nombre, de manera que los usuarios no puedan confundirlas.

**Curso típico de eventos:**

1) El administrador selecciona “modificar política existente de atributo” y establece los nuevos valores para las propiedades de la política seleccionada.

2) El sistema verifica la información especificada y, en caso de ser correcta, actualiza las propiedades de la política de atributo seleccionada con los valores modificados.

**8.5.58.- Caso de uso: eliminar política existente de atributo de entidad [CU-4.4.7.4].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador eliminar una política de atributo existente de entidad, como parte del modelado del dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.7].

**Descripción:** el administrador podrá eliminar cualquier política existente de atributo de entidad.

**Curso típico de eventos:**

1) El administrador selecciona “eliminar política existente de atributo de entidad” y especifica la política de atributo de entidad a eliminar.

2) El sistema elimina toda la información de dicha política de atributo de entidad en el modelo de dominio.

### 8.5.59.- Caso de uso: gestionar relaciones [CU-4.4.8].

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador gestionar las relaciones (asociaciones binarias entre entidades y objetos valor), utilizadas como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; casos de uso [CU-4.4], [CU-4.4.8.1], [CU-4.4.8.2], [CU-4.4.8.3] y [CU-4.4.8.4].

**Descripción:** el administrador dispondrá de la opción de crear nuevas relaciones, así como consultar, modificar y eliminar las relaciones modeladas previamente. Si se opta por eliminar una relación, todas sus propiedades se eliminarán y no podrán recuperarse.

#### **Curso típico de eventos:**

Este caso de uso secundario es el nexo de partida de varios casos de uso secundarios y permite al usuario acceder a estos otros casos, por lo que no existe como tal un único escenario. Por ello, nos remitimos a los casos de uso secundarios del apartado referencias para más información.

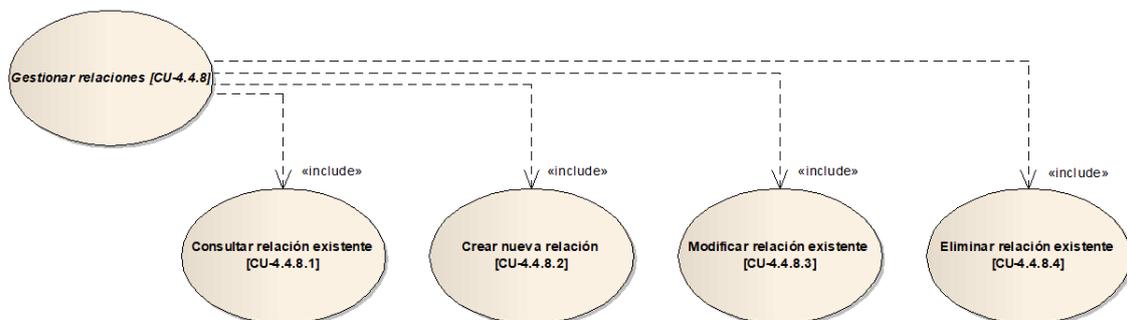


Fig. 8.17.- Casos de uso secundarios de “gestionar relaciones”.

### 8.5.60.- Caso de uso: consultar relación existente [CU-4.4.8.1].

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador consultar la información de la relación seleccionada.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.8].

**Descripción:** el administrador dispondrá de la opción de consultar la información de la relación especificada.

**Curso típico de eventos:**

1) El administrador selecciona una de las relaciones que forman parte del dominio de la aplicación generada que está siendo modelada.

2) El sistema muestra toda la información asociada con dicha relación.

#### **8.5.61.- Caso de uso: crear nueva relación [CU-4.4.8.2].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador crear una nueva relación (asociación binaria entre entidades y objetos valor), como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.8].

**Descripción:** el administrador seleccionará “crear nueva relación” y establecerá los atributos iniciales de la misma (nombre, comentarios, tipología, entidad y objeto valor de origen, multiplicidad del origen, entidad y objeto valor de destino, multiplicidad del destino). No se permite que dos relaciones pertenecientes al mismo módulo tengan el mismo nombre, impidiendo así que los usuarios puedan confundirlas.

**Curso típico de eventos:**

1) El administrador selecciona “crear nueva relación” y establece los atributos de la misma.

2) El sistema verifica la información especificada y, en caso de ser correcta, crea la nueva relación.

#### **8.5.62.- Caso de uso: modificar relación existente [CU-4.4.8.3].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador modificar una relación existente, como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.8].

**Descripción:** el administrador seleccionará “modificar relación existente” y actualizará las propiedades que considere de la misma (solamente pueden modificarse el nombre y los comentarios). No se permite que dos relaciones pertenecientes al mismo módulo tengan el mismo nombre, impidiendo así que los usuarios puedan confundirlas.

**Curso típico de eventos:**

1) El administrador selecciona “modificar relación existente” y establece los nuevos valores para las propiedades de la relación seleccionada.

2) El sistema verifica la información especificada y, en caso de ser correcta, actualiza las propiedades de la relación seleccionada con los valores modificados.

#### **8.5.63.- Caso de uso: eliminar relación existente [CU-4.4.8.4].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador eliminar una relación existente, como parte del modelado del dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.8].

**Descripción:** el administrador podrá eliminar cualquier relación existente. Si se opta por eliminar una relación, tanto el esquema (metadatos) como la información (datos de usuario) asociados a dicha relación se borrarán y no podrán recuperarse.

**Curso típico de eventos:**

1) El administrador selecciona “eliminar relación existente” y especifica la relación a eliminar.

2) El sistema elimina toda la información del modelo de dominio, así como de los datos de usuario, asociada a la relación seleccionada.

### 8.5.64.- Caso de uso: gestionar objetos valor [CU-4.4.9].

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador gestionar los objetos valor utilizados como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; casos de uso [CU-4.4], [CU-4.4.9.1], [CU-4.4.9.2], [CU-4.4.9.3], [CU-4.4.9.4], [CU-4.4.10] y [CU-4.4.11].

**Descripción:** el administrador dispondrá de la opción de crear nuevos objetos valor, así como consultar, modificar y eliminar los objetos valor modelados previamente. Si se opta por eliminar un objeto valor, todos los elementos contenidos en el mismo (políticas, atributos y relaciones) se eliminarán y no podrán recuperarse, así como los datos de usuario asociados a dichos elementos.

#### **Curso típico de eventos:**

Este caso de uso secundario es el nexo de partida de varios casos de uso secundarios y permite al usuario acceder a estos otros casos, por lo que no existe como tal un único escenario. Por ello, nos remitimos a los casos de uso secundarios del apartado referencias para más información.

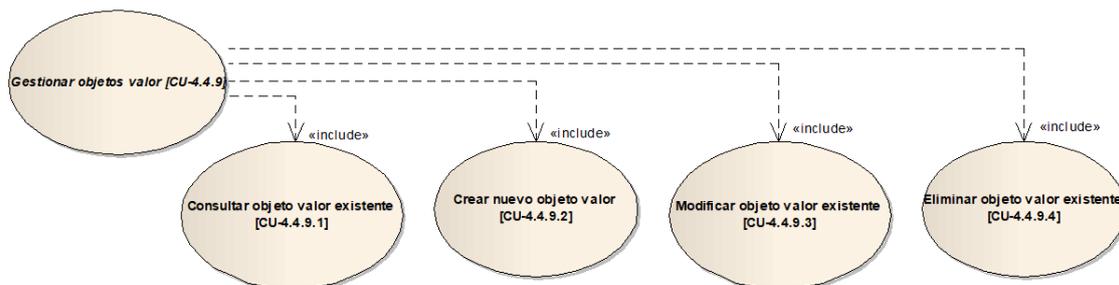


Fig. 8.18.- Casos de uso secundarios de “gestionar objetos valor”.

### 8.5.65.- Caso de uso: consultar objeto valor existente [CU-4.4.9.1].

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador consultar la información del objeto valor seleccionado.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.9].

**Descripción:** el administrador dispondrá de la opción de consultar la información del objeto valor especificado.

**Curso típico de eventos:**

- 1) El administrador selecciona uno de los objetos valor que forman parte del dominio de la aplicación generada que está siendo modelada.
- 2) El sistema muestra toda la información relacionada con dicho objeto valor.

#### **8.5.66.- Caso de uso: crear nuevo objeto valor [CU-4.4.9.2].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador crear un nuevo objeto valor, como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.9].

**Descripción:** el administrador seleccionará “crear nuevo objeto valor” y establecerá los atributos iniciales del mismo (solamente “nombre”). No se permite que dos objetos valor pertenecientes al mismo módulo tengan el mismo nombre, de manera que los usuarios no puedan confundirlos.

**Curso típico de eventos:**

- 1) El administrador selecciona “crear nuevo objeto valor” y establece los atributos del mismo.
- 2) El sistema verifica la información especificada y, en caso de ser correcta, crea el nuevo objeto valor.

#### **8.5.67.- Caso de uso: modificar objeto valor existente [CU-4.4.9.3].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador modificar un objeto valor existente, como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.9].

**Descripción:** el administrador seleccionará “modificar objeto valor existente” y actualizará las propiedades que considere del mismo (solamente “nombre”). No se permite que dos objetos valor pertenecientes al mismo módulo tengan el mismo nombre, impidiendo así que los usuarios puedan confundirlos.

**Curso típico de eventos:**

1) El administrador selecciona “modificar objeto valor existente” y establece los nuevos valores para las propiedades del objeto valor seleccionado.

2) El sistema verifica la información especificada y, en caso de ser correcta, actualiza las propiedades del objeto valor seleccionado con los valores modificados.

**8.5.68.- Caso de uso: eliminar objeto valor existente [CU-4.4.9.4].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador eliminar un objeto valor existente, como parte del modelado del dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.9].

**Descripción:** el administrador podrá eliminar cualquier objeto valor existente, eliminando consigo a todos y cada uno de los diferentes elementos contenidos en dicho objeto valor, como son: atributos, políticas y relaciones. Si se opta por eliminar un objeto valor, tanto el esquema (metadatos) como la información (datos de usuario) asociados a dicho objeto valor (y de sus elementos contenidos) se borrarán y no podrán recuperarse.

**Curso típico de eventos:**

1) El administrador selecciona “eliminar objeto valor existente” y especifica el objeto valor a eliminar.

2) El sistema elimina toda la información del modelo de dominio, así como de los datos de usuario, tanto del objeto valor seleccionado como de todos y cada uno de los diferentes elementos contenidos en el mismo.

### 8.5.69.- Caso de uso: gestionar políticas de objeto valor [CU-4.4.10].

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador gestionar las políticas de objeto valor utilizadas como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; casos de uso [CU-4.4], [CU-4.4.10.1], [CU-4.4.10.2], [CU-4.4.10.3] y [CU-4.4.10.4].

**Descripción:** el administrador dispondrá de la opción de crear nuevas políticas de objeto valor, así como consultar, modificar y eliminar las políticas de objeto valor modeladas previamente. Si se opta por eliminar una política de objeto valor, todas sus propiedades se eliminarán y no podrán recuperarse.

**Curso típico de eventos:**

Este caso de uso secundario es el nexo de partida de varios casos de uso secundarios y permite al usuario acceder a estos otros casos, por lo que no existe como tal un único escenario. Por ello, nos remitimos a los casos de uso secundarios del apartado referencias para más información.

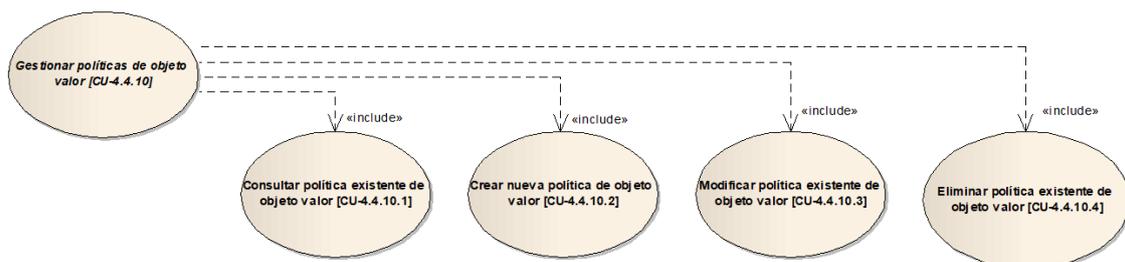


Fig. 8.19.- Casos de uso secundarios de “gestionar políticas de objeto valor”.

### 8.5.70.- Caso de uso: consultar política existente de objeto valor [CU-4.4.10.1].

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador consultar la información de la política de objeto valor seleccionada.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.10].

**Descripción:** el administrador dispondrá de la opción de consultar la información de la política de objeto valor especificada.

**Curso típico de eventos:**

1) El administrador selecciona una de las políticas de objeto valor que forman parte del dominio de la aplicación generada que está siendo modelada.

2) El sistema muestra toda la información relacionada con dicha política de objeto valor.

#### **8.5.71.- Caso de uso: crear nueva política de objeto valor [CU-4.4.10.2].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador crear una nueva política de objeto valor, como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.7].

**Descripción:** el administrador seleccionará “crear nueva política de objeto valor” y establecerá los atributos iniciales de la misma (nombre, está activa y código de validación). No se permite que dos políticas de objeto valor pertenecientes al mismo objeto valor tengan el mismo nombre, impidiendo así que los usuarios puedan confundirlas.

**Curso típico de eventos:**

1) El administrador selecciona “crear nueva política de objeto valor” y establece los atributos de la misma.

2) El sistema verifica la información especificada y, en caso de ser correcta, crea la nueva política de objeto valor.

#### **8.5.72.- Caso de uso: modificar política existente de objeto valor [CU-4.4.10.3].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador modificar una política existente de objeto valor, como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.10].

**Descripción:** el administrador seleccionará “modificar política existente de objeto valor” y actualizará las propiedades que considere de la misma (nombre, está activa y código de validación). No se permite que dos políticas pertenecientes al mismo objeto valor tengan el mismo nombre, de manera que los usuarios no puedan confundirlas.

**Curso típico de eventos:**

1) El administrador selecciona “modificar política existente de objeto valor” y establece los nuevos valores para las propiedades de la política seleccionada.

2) El sistema verifica la información especificada y, en caso de ser correcta, actualiza las propiedades de la política de objeto valor seleccionada con los valores modificados.

#### **8.5.73.- Caso de uso: eliminar política existente de objeto valor [CU-4.4.10.4].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador eliminar una política existente de objeto valor, como parte del modelado del dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.10].

**Descripción:** el administrador podrá eliminar cualquier política existente de objeto valor.

**Curso típico de eventos:**

1) El administrador selecciona “eliminar política existente de objeto valor” y especifica la política de objeto valor a eliminar.

2) El sistema elimina toda la información de dicha política de objeto valor en el modelo de dominio.

### 8.5.74.- Caso de uso: gestionar atributos de objeto valor [CU-4.4.11].

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador gestionar los atributos de objeto valor utilizados como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; casos de uso [CU-4.4], [CU-4.4.11.1], [CU-4.4.11.2], [CU-4.4.11.3] y [CU-4.4.11.4].

**Descripción:** el administrador dispondrá de la opción de crear nuevos atributos de objeto valor, así como consultar, modificar y eliminar los atributos de objeto valor modelados previamente. Si se opta por eliminar un atributo de objeto valor, todas sus propiedades, así como los datos de usuario asociados, se eliminarán y no podrán recuperarse.

#### **Curso típico de eventos:**

Este caso de uso secundario es el nexo de partida de varios casos de uso secundarios y permite al usuario acceder a estos otros casos, por lo que no existe como tal un único escenario. Por ello, nos remitimos a los casos de uso secundarios del apartado referencias para más información.

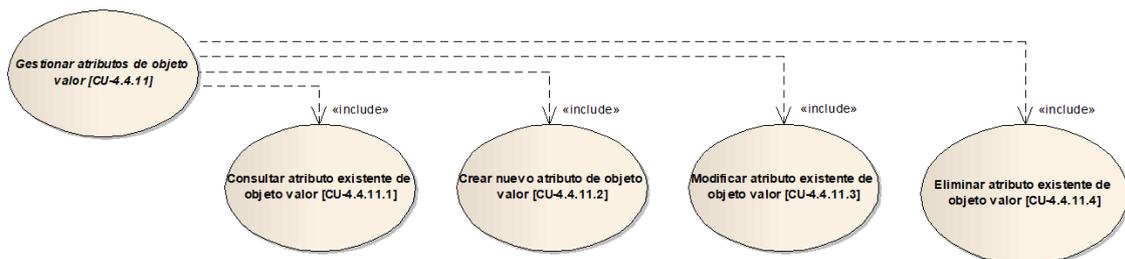


Fig. 8.20.- Casos de uso secundarios de “gestionar atributos de objeto valor”.

### 8.5.75.- Caso de uso: consultar atributo existente de objeto valor [CU-4.4.11.1].

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador consultar la información del atributo de objeto valor seleccionado.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.11].

**Descripción:** el administrador dispondrá de la opción de consultar la información del atributo de objeto valor especificado.

**Curso típico de eventos:**

1) El administrador selecciona uno de los atributos de objeto valor que forman parte del dominio de la aplicación generada que está siendo modelada.

2) El sistema muestra toda la información relacionada con dicho atributo de objeto valor.

**8.5.76.- Caso de uso: crear nuevo atributo de objeto valor [CU-4.4.11.2].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador crear un nuevo atributo de objeto valor, como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.11].

**Descripción:** el administrador seleccionará “crear nuevo atributo de objeto valor” y establecerá los, valga la redundancia, atributos iniciales del mismo (nombre, tipo de datos, obligatoriedad y código de la fórmula en caso de atributos calculados). No se permite que dos atributos de objeto valor pertenecientes al mismo objeto valor tengan el mismo nombre, impidiendo así que los usuarios puedan confundirlos.

**Curso típico de eventos:**

1) El administrador selecciona “crear nuevo atributo de objeto valor” y establece las propiedades del mismo.

2) El sistema verifica la información especificada y, en caso de ser correcta, crea el nuevo atributo de objeto valor.

**8.5.77.- Caso de uso: modificar atributo existente de objeto valor [CU-4.4.11.3].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador modificar un atributo existente de objeto valor, como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.11].

**Descripción:** el administrador seleccionará “modificar atributo existente de objeto valor” y actualizará las propiedades que considere del mismo (solamente “nombre”). No se permite que dos atributos pertenecientes al mismo objeto valor tengan el mismo nombre, impidiendo así que los usuarios puedan confundirlos.

**Curso típico de eventos:**

1) El administrador selecciona “modificar atributo existente de objeto valor” y establece los nuevos valores para las propiedades del atributo de objeto valor seleccionado.

2) El sistema verifica la información especificada y, en caso de ser correcta, actualiza las propiedades del atributo de objeto valor seleccionado con los valores modificados.

#### **8.5.78.- Caso de uso: eliminar atributo existente de objeto valor [CU-4.4.11.4].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador eliminar un atributo existente de objeto valor, como parte del modelado del dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.11].

**Descripción:** el administrador podrá eliminar cualquier atributo existente de objeto valor.

**Curso típico de eventos:**

1) El administrador selecciona “eliminar atributo existente de objeto valor” y especifica el atributo de objeto valor a eliminar.

2) El sistema elimina toda la información de dicho atributo de objeto valor, tanto en el modelo de dominio (metadatos) como a nivel de información (datos de usuario).

### 8.5.79.- Caso de uso: gestionar políticas de atributo de objeto valor [CU-4.4.12].

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador gestionar las políticas de atributo de objeto valor utilizadas como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; casos de uso [CU-4.4], [CU-4.4.12.1], [CU-4.4.12.2], [CU-4.4.12.3] y [CU-4.4.12.4].

**Descripción:** el administrador dispondrá de la opción de crear nuevas políticas de atributo de objeto valor, así como consultar, modificar y eliminar las políticas de atributo de objeto valor modeladas previamente. Si se opta por eliminar una política de atributo de objeto valor, todas sus propiedades se eliminarán y no podrán recuperarse.

#### **Curso típico de eventos:**

Este caso de uso secundario es el nexo de partida de varios casos de uso secundarios y permite al usuario acceder a estos otros casos, por lo que no existe como tal un único escenario. Por ello, nos remitimos a los casos de uso secundarios del apartado referencias para más información.

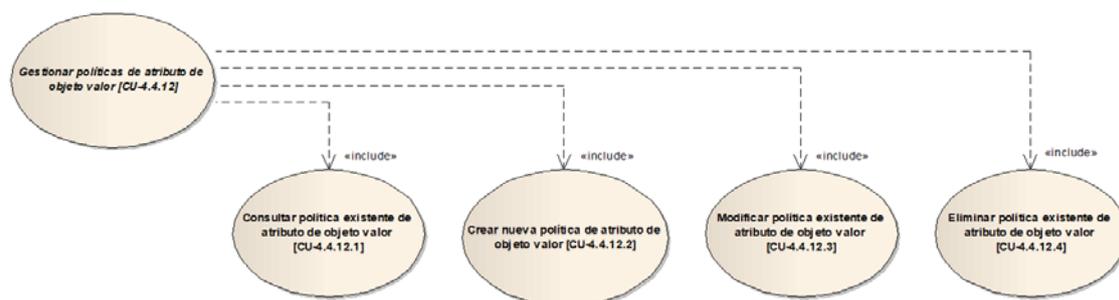


Fig. 8.21.- Casos de uso secundarios de “gestionar políticas de atributo de objeto valor”.

### **8.5.80.- Caso de uso: consultar política existente de atributo de objeto valor**

**[CU-4.4.12.1].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador consultar la información de la política de atributo de objeto valor seleccionada.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.12].

**Descripción:** el administrador dispondrá de la opción de consultar la información de la política de atributo de objeto valor especificada.

**Curso típico de eventos:**

- 1) El administrador selecciona una de las políticas de atributo de objeto valor que forman parte del dominio de la aplicación generada que está siendo modelada.
- 2) El sistema muestra toda la información relacionada con dicha política de atributo de objeto valor.

### **8.5.81.- Caso de uso: crear nueva política de atributo de objeto valor [CU-4.4.12.2].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador crear una nueva política de atributo de objeto valor, como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.12].

**Descripción:** el administrador seleccionará “crear nueva política de atributo de objeto valor” y establecerá los atributos iniciales de la misma (nombre, está activa y código de validación). No se permite que dos políticas de atributo de objeto valor pertenecientes al mismo atributo de objeto valor tengan el mismo nombre, impidiendo así que los usuarios puedan confundirlas.

**Curso típico de eventos:**

- 1) El administrador selecciona “crear nueva política de atributo de objeto valor” y establece los atributos de la misma.

2) El sistema verifica la información especificada y, en caso de ser correcta, crea la nueva política de atributo de objeto valor.

**8.5.82.- Caso de uso: modificar política existente de atributo de objeto valor [CU-4.4.12.3].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador modificar una política existente de atributo de objeto valor, como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.12].

**Descripción:** el administrador seleccionará “modificar política existente de atributo de objeto valor” y actualizará las propiedades que considere de la misma (nombre, está activa y código de validación). No se permite que dos políticas pertenecientes al mismo atributo de objeto valor tengan el mismo nombre, de manera que los usuarios no puedan confundirlas.

**Curso típico de eventos:**

1) El administrador selecciona “modificar política existente de atributo de objeto valor” y establece los nuevos valores para las propiedades de la política seleccionada.

2) El sistema verifica la información especificada y, en caso de ser correcta, actualiza las propiedades de la política de atributo de objeto valor seleccionada con los valores modificados.

**8.5.83.- Caso de uso: eliminar política existente de atributo de objeto valor [CU-4.4.12.4].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador eliminar una política existente de atributo de objeto valor, como parte del modelado del dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.12].

**Descripción:** el administrador podrá eliminar cualquier política existente de atributo de objeto valor.

**Curso típico de eventos:**

1) El administrador selecciona “eliminar política existente de atributo de objeto valor” y especifica la política de atributo de objeto valor a eliminar.

2) El sistema elimina toda la información de dicha política de atributo de objeto valor en el modelo de dominio.

**8.5.84.- Caso de uso: gestionar servicios [CU-4.4.13].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador gestionar los servicios utilizados como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; casos de uso [CU-4.4], [CU-4.4.13.1], [CU-4.4.13.2], [CU-4.4.13.3], [CU-4.4.13.4], y [CU-4.4.14].

**Descripción:** el administrador dispondrá de la opción de crear nuevos servicios, así como consultar, modificar y eliminar los servicios modelados previamente. Si se opta por eliminar un servicio, todos los elementos contenidos en el mismo (procesos por lotes) igualmente se eliminarán y no podrán recuperarse.

**Curso típico de eventos:**

Este caso de uso secundario es el nexo de partida de varios casos de uso secundarios y permite al usuario acceder a estos otros casos, por lo que no existe como tal un único escenario. Por ello, nos remitimos a los casos de uso secundarios del apartado referencias para más información.

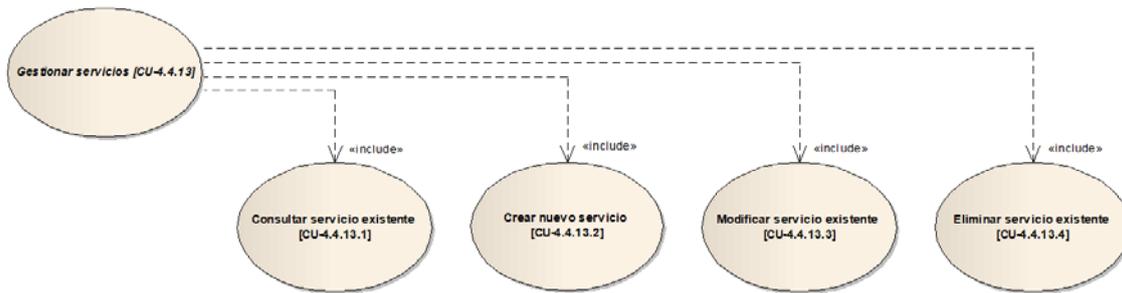


Fig. 8.22.- Casos de uso secundarios de “gestionar servicios”.

#### 8.5.85.- Caso de uso: consultar servicio existente [CU-4.4.13.1].

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador consultar la información del servicio seleccionado.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.13].

**Descripción:** el administrador dispondrá de la opción de consultar la información del servicio especificado.

**Curso típico de eventos:**

- 1) El administrador selecciona uno de los servicios que forman parte del dominio de la aplicación generada que está siendo modelada.
- 2) El sistema muestra toda la información relacionada con dicho servicio.

#### 8.5.86.- Caso de uso: crear nuevo servicio [CU-4.4.13.2].

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador crear un nuevo servicio, como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.13].

**Descripción:** el administrador seleccionará “crear nuevo servicio” y establecerá los atributos iniciales del mismo (solamente “nombre”). No se permite que dos servicios pertenecientes al mismo módulo tengan el mismo nombre, de manera que los usuarios no puedan confundirlos.

**Curso típico de eventos:**

- 1) El administrador selecciona “crear nuevo servicio” y establece los atributos del mismo.
- 2) El sistema verifica la información especificada y, en caso de ser correcta, crea el nuevo servicio.

**8.5.87.- Caso de uso: modificar servicio existente [CU-4.4.13.3].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador modificar un servicio existente, como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.13].

**Descripción:** el administrador seleccionará “modificar servicio existente” y actualizará las propiedades que considere del mismo (nombre). No se permite que dos servicios pertenecientes al mismo módulo tengan el mismo nombre, impidiendo así que los usuarios puedan confundirlos.

**Curso típico de eventos:**

- 1) El administrador selecciona “modificar servicio existente” y establece los nuevos valores para las propiedades del servicio seleccionado.
- 2) El sistema verifica la información especificada y, en caso de ser correcta, actualiza las propiedades del servicio seleccionado con los valores modificados.

**8.5.88.- Caso de uso: eliminar servicio existente [CU-4.4.13.4].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador eliminar un servicio existente, como parte del modelado del dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.13].

**Descripción:** el administrador podrá eliminar cualquier servicio existente, eliminando consigo a todos y cada uno de los diferentes elementos contenidos en dicho servicio, como son los procesos por lotes. Si se opta por eliminar un servicio, la información asociada a dicho servicio (y a sus elementos contenidos) se borrará y no podrá recuperarse.

**Curso típico de eventos:**

1) El administrador selecciona “eliminar servicio existente” y especifica el servicio a eliminar.

2) El sistema elimina toda la información del modelo de dominio tanto del servicio seleccionado como de todos y cada uno de los diferentes elementos contenidos en el mismo.

**8.5.89.- Caso de uso: gestionar procesos por lotes [CU-4.4.14].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador gestionar los procesos por lotes utilizados como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; casos de uso [CU-4.4.13], [CU-4.4.14.1], [CU-4.4.14.2], [CU-4.4.14.3] y [CU-4.4.14.4].

**Descripción:** el administrador dispondrá de la opción de crear nuevos procesos por lotes, así como consultar, modificar y eliminar los procesos por lotes modelados previamente. Si se opta por eliminar un proceso por lote, la información del mismo se borrará y no podrá recuperarse.

**Curso típico de eventos:**

Este caso de uso secundario es el nexo de partida de varios casos de uso secundarios y permite al usuario acceder a estos otros casos, por lo que no existe como tal un único escenario. Por ello, nos remitimos a los casos de uso secundarios del apartado referencias para más información.

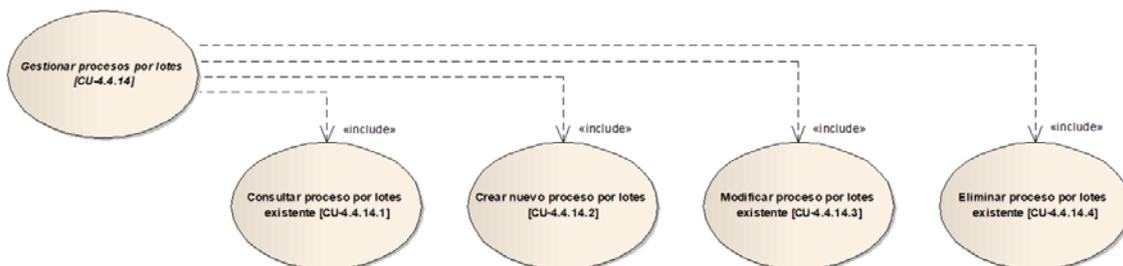


Fig. 8.23.- Casos de uso secundarios de “gestionar procesos por lotes”.

#### 8.5.90.- Caso de uso: consultar proceso por lotes existente [CU-4.4.14.1].

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador consultar la información del proceso por lotes seleccionado.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.14].

**Descripción:** el administrador dispondrá de la opción de consultar la información del proceso por lotes especificado.

**Curso típico de eventos:**

- 1) El administrador selecciona uno de los procesos por lotes que forman parte del dominio de la aplicación generada que está siendo modelada.
- 2) El sistema muestra toda la información relacionada con dicho proceso por lotes.

#### 8.5.91.- Caso de uso: crear nuevo proceso por lotes [CU-4.4.14.2].

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador crear un nuevo proceso por lotes, como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.14].

**Descripción:** el administrador seleccionará “crear nuevo proceso por lotes” y establecerá las propiedades iniciales del mismo (nombre, servicio, credenciales de ejecución y programación temporal). No se permite que dos procesos por lotes pertenecientes al mismo servicio tengan el mismo nombre, impidiendo así que los usuarios puedan confundirlos.

**Curso típico de eventos:**

1) El administrador selecciona “crear nuevo proceso por lotes” y establece las propiedades del mismo.

2) El sistema verifica la información especificada y, en caso de ser correcta, crea el nuevo proceso por lotes y programa su ejecución automática.

**8.5.92.- Caso de uso: modificar proceso por lotes existente [CU-4.4.14.3].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador modificar un proceso por lotes existente, como parte del modelo de dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.14].

**Descripción:** el administrador seleccionará “modificar proceso por lotes existente” y actualizará las propiedades que considere del mismo (nombre, credenciales de ejecución y programación temporal). No se permite que dos procesos por lotes pertenecientes al mismo servicio tengan el mismo nombre, impidiendo así que los usuarios puedan confundirlos.

**Curso típico de eventos:**

1) El administrador selecciona “modificar proceso por lotes existente” y establece los nuevos valores para las propiedades del proceso por lotes seleccionado.

2) El sistema verifica la información especificada y, en caso de ser correcta, actualiza las propiedades del proceso por lotes seleccionado con los valores modificados.

#### **8.5.93.- Caso de uso: eliminar proceso por lotes existente [CU-4.4.14.4].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador eliminar un proceso por lotes existente, como parte del modelado del dominio de una aplicación generada para consumidores.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4.4.14].

**Descripción:** el administrador podrá eliminar cualquier proceso por lotes existente.

**Curso típico de eventos:**

1) El administrador selecciona “eliminar proceso por lotes existente” y especifica el proceso por lotes a eliminar.

2) El sistema elimina toda la información de dicho proceso por lotes, incluyendo la programación de ejecución automática.

#### **8.5.94.- Caso de uso: eliminar aplicación existente [CU-4.5].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador eliminar toda la información, tanto a nivel de metadatos como de datos, de una aplicación generada a través de la solución.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-4].

**Descripción:** el administrador podrá eliminar cualquier aplicación generada existente en la solución.

**Curso típico de eventos:**

1) El administrador selecciona “eliminar aplicación existente” y especifica qué aplicación deberá ser eliminada.

2) El sistema desinstala la aplicación seleccionada de todos los sistemas registrados en los que se encuentre instalada; elimina asimismo toda la información asociada, incluyendo: metadatos de modelado, datos introducidos por los consumidores y accesos de usuario. Los datos eliminados no podrán recuperarse.



### **8.5.95.- Caso de uso: gestionar idiomas y traducciones [CU-5].**

**Tipo:** primario.

**Actor:** administrador.

**Propósito:** permitir a un administrador gestionar los idiomas que podrán utilizarse en las aplicaciones generadas como parte de la solución.

**Referencias:** requisitos [R-1] y [R-3]; casos de uso [CU-5.1], [CU-5.2], [CU-5.3], [CU-5.4] y [CU-5.5].

**Descripción:** el administrador dispondrá de la opción de crear nuevos idiomas, así como consultar, modificar y eliminar los idiomas registrados previamente. Si se opta por eliminar un idioma, los datos del idioma y de todas las traducciones asociadas al mismo se borrarán y no podrán recuperarse.

**Curso típico de eventos:**

Este caso de uso primario es el nexo de partida de varios casos de uso secundarios y permite al usuario acceder a estos otros casos, por lo que no existe como tal un único escenario. Por ello, nos remitimos a los casos de uso secundarios del apartado referencias para más información. Se presenta a continuación una ampliación de la vista de casos de uso representando estos nexos:

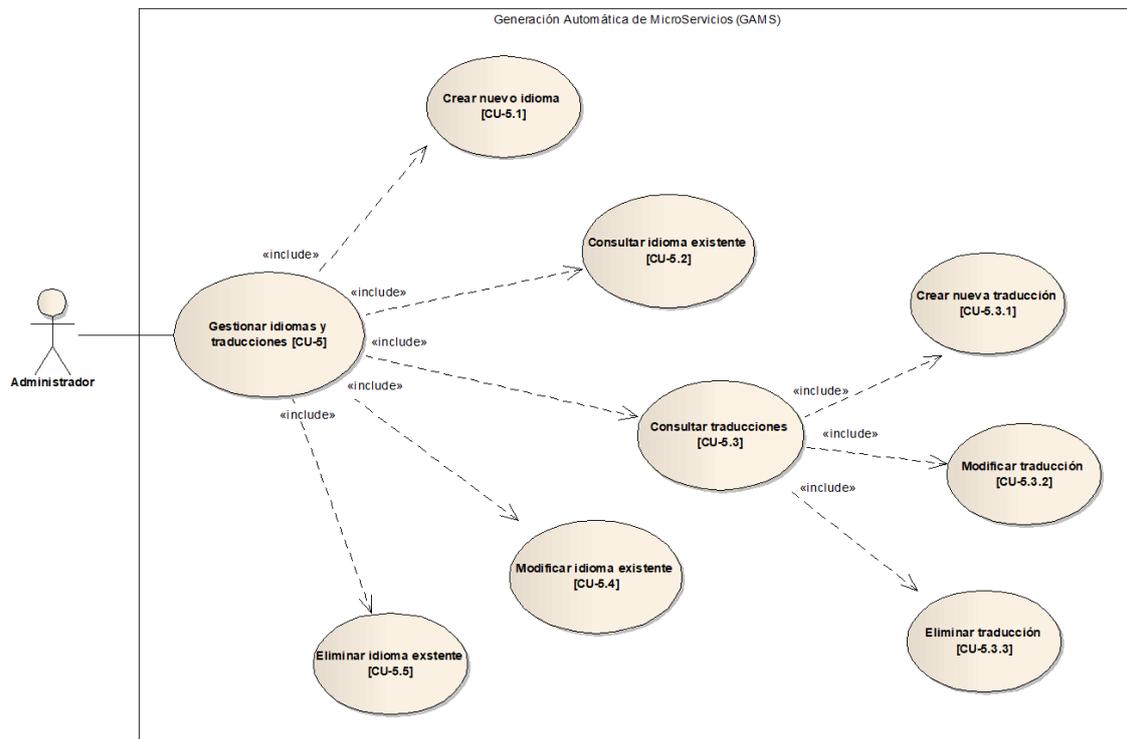


Fig. 8.24.- Diagrama de casos de uso secundarios de "gestionar idiomas y traducciones".

### 8.5.96.- Caso de uso: crear nuevo idioma [CU-5.1].

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador crear (registrar) un nuevo idioma en la solución.

**Referencias:** requisitos [R-1] y [R-3], caso de uso [CU-5].

**Descripción:** el administrador seleccionará "crear nuevo idioma" y establecerá los atributos del mismo (código ISO-639-1 de idioma, nombre y código ISO-3166-1 Alfa-2 de país). Tanto el código de idioma como el nombre de un idioma deben ser únicos.

**Curso típico de eventos:**

1) El administrador selecciona "crear nuevo idioma" y establece los atributos del mismo.

2) El sistema verifica la información especificada y, en caso de ser correcta, crea el nuevo idioma.



#### **8.5.97.- Caso de uso: consultar idioma existente [CU-5.2].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador consultar la información del idioma seleccionado.

**Referencias:** requisitos [R-1] y [R-3], caso de uso [CU-5].

**Descripción:** el administrador dispondrá de la opción de consultar la información de cualquier idioma registrado en la solución.

**Curso típico de eventos:**

- 1) El administrador selecciona uno de los idiomas registrados en la solución.
- 2) El sistema muestra toda la información relacionada con dicho idioma, incluyendo si existen traducciones asociadas al mismo.

#### **8.5.98.- Caso de uso: consultar traducciones [CU-5.3].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador consultar todas las traducciones existentes asociadas al idioma seleccionado.

**Referencias:** requisitos [R-1] y [R-3], caso de uso [CU-5].

**Descripción:** el administrador dispondrá de la opción de consultar todas las traducciones existentes asociadas a cualquier idioma registrado en la solución.

**Curso típico de eventos:**

- 1) El administrador selecciona uno de los idiomas registrados en la solución y selecciona “consultar traducciones”.
- 2) El sistema muestra la información de todas las traducciones asociadas con dicho idioma.

### **8.5.99.- Caso de uso: crear nueva traducción [CU-5.3.1].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador crear (registrar) una nueva traducción en la solución.

**Referencias:** requisitos [R-1] y [R-3], caso de uso [CU-5].

**Descripción:** el administrador seleccionará “crear nueva traducción” especificando un idioma concreto y establecerá los atributos de la misma (elemento a traducir y texto de la traducción).

**Curso típico de eventos:**

- 1) El administrador selecciona “crear nueva traducción”, especificando un idioma y elemento concretos, y establece el texto de la traducción.
- 2) El sistema verifica la información especificada y, en caso de ser correcta, crea el nuevo registro de traducción.

### **8.5.100.- Caso de uso: modificar traducción [CU-5.3.2].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador modificar un registro de traducción existente en la solución.

**Referencias:** requisitos [R-1] y [R-3], caso de uso [CU-5].

**Descripción:** el administrador podrá modificar el texto de una traducción especificada en un idioma determinado para un elemento concreto.

**Curso típico de eventos:**

- 1) El administrador seleccionará “modificar traducción”, especificando un idioma concreto y el elemento a traducir, modificando el valor del texto de la traducción.
- 2) El sistema verifica la información especificada y, en caso de ser correcta, actualizará el valor del registro de traducción.

#### **8.5.101.- Caso de uso: eliminar traducción [CU-5.3.3].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador eliminar un registro de traducción existente en la solución.

**Referencias:** requisitos [R-1] y [R-3], caso de uso [CU-5].

**Descripción:** el administrador podrá eliminar el texto de una traducción especificada en un idioma determinado para un elemento concreto.

**Curso típico de eventos:**

1) El administrador seleccionará “eliminar traducción”, especificando un idioma concreto y el elemento cuya traducción se desea eliminar.

2) El sistema verifica la información especificada y, en caso de ser correcta, eliminará el registro de traducción.

#### **8.5.102.- Caso de uso: modificar idioma existente [CU-5.4].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador modificar la información asociada a un idioma previamente registrado en la solución.

**Referencias:** requisitos [R-1] y [R-3], caso de uso [CU-5].

**Descripción:** el administrador seleccionará “modificar idioma existente” y establecerá nuevos valores a determinados atributos del mismo (código ISO-639-1 de idioma, nombre y código ISO-3166-1 Alfa-2 de país). Tanto el código de idioma como el nombre deben seguir siendo únicos.

**Curso típico de eventos:**

1) El administrador selecciona “modificar idioma existente” y establece los nuevos valores para los atributos del mismo.

2) El sistema verifica la información especificada y, en caso de ser correcta, modificará los datos del idioma especificado.

### **8.5.103.- Caso de uso: eliminar idioma existente [CU-5.5].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador eliminar toda la información de un idioma previamente registrado en la solución.

**Referencias:** requisitos [R-1] y [R-3], caso de uso [CU-5].

**Descripción:** el administrador podrá eliminar cualquier idioma existente, eliminando también todas las traducciones asociadas al mismo. Si se opta por eliminar un idioma, los datos del idioma y de todas las traducciones asociadas al mismo se borrarán y no podrán recuperarse.

**Curso típico de eventos:**

- 1) El administrador selecciona “eliminar idioma existente”.
- 2) El sistema elimina el registro del idioma, junto con todas las traducciones asociadas al mismo.

### **8.5.104.- Caso de uso: gestión de sistemas y despliegues [CU-5].**

**Tipo:** primario.

**Actor:** administrador.

**Propósito:** permitir a un administrador gestionar los sistemas registrados que podrán utilizarse para desplegar las aplicaciones generadas como parte de la solución.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; casos de uso [CU-5.1], [CU-5.2], [CU-5.3], [CU-5.4] y [CU-5.5].

**Descripción:** el administrador dispondrá de la opción de gestionar (crear, modificar y eliminar) sistemas donde posteriormente se podrán desplegar las diferentes aplicaciones generadas para los consumidores. Téngase en cuenta que cuando se opte por eliminar un sistema, toda la información asociada al mismo será eliminada y no podrá recuperarse, incluyendo toda la información “local” de las aplicaciones desplegadas en el mismo, tanto a nivel de metadatos como de datos.

**Curso típico de eventos:**

Este caso de uso primario es el nexo de partida de varios casos de uso secundarios y permite al usuario acceder a estos otros casos, por lo que no existe como tal un único escenario. Por ello, nos remitimos a los casos de uso secundarios del apartado referencias para más información. Se presenta a continuación una ampliación de la vista de casos de uso representando estos nexos:

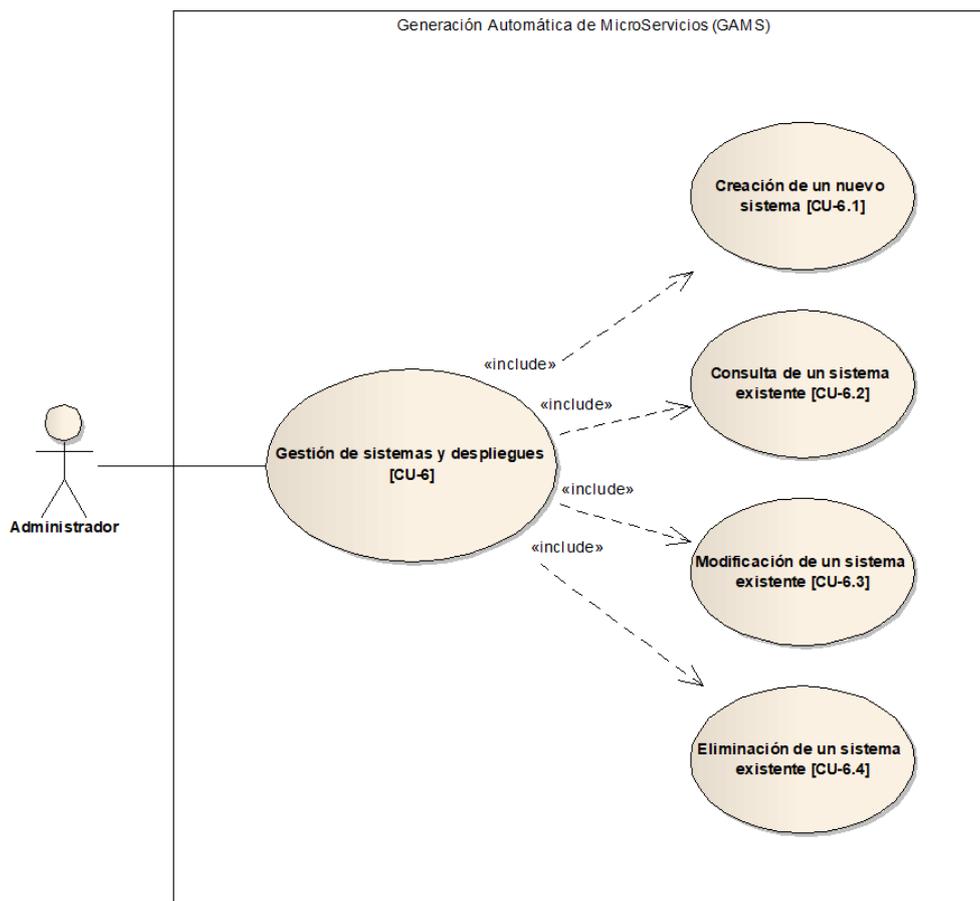


Fig. 8.25.- Diagrama de casos de uso secundarios de "gestión de sistemas y despliegues".

#### 8.5.105.- Caso de uso: creación de un nuevo sistema [CU-6.1].

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador crear (registrar) un nuevo sistema en la solución.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-6].

**Descripción:** el administrador seleccionará “crear nuevo sistema” y establecerá los atributos del mismo (nombre, título, descripción y dirección). Tanto el nombre como el título y la dirección de un sistema deben ser únicos.

**Curso típico de eventos:**

- 1) El administrador selecciona “crear nuevo sistema” y establece los atributos del mismo.
- 2) El sistema verifica la información especificada y, en caso de ser correcta, crea (registra) el nuevo sistema.

#### **8.5.106.- Caso de uso: consulta de un sistema existente [CU-6.2].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador consultar la información del sistema seleccionado.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-6].

**Descripción:** el administrador dispondrá de la opción de consultar la información de cualquier sistema registrado en la solución.

**Curso típico de eventos:**

- 1) El administrador selecciona uno de los sistemas registrados en la solución.
- 2) El sistema muestra toda la información relacionada con dicho sistema, incluyendo las aplicaciones desplegadas en el mismo.

#### **8.5.107.- Caso de uso: modificación de un sistema existente [CU-6.3].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador modificar la información asociada a un sistema previamente registrado en la solución.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-6].

**Descripción:** el administrador seleccionará “modificar sistema existente” y establecerá nuevos valores a determinados atributos del mismo (nombre, título, descripción y dirección). Tanto el nombre como el título y la dirección de un sistema deben ser únicos.

**Curso típico de eventos:**

1) El administrador selecciona “modificar sistema existente” y establece los nuevos valores para los atributos del mismo.

2) El sistema verifica la información especificada y, en caso de ser correcta, modificará los datos del sistema especificado.

**8.5.108.- Caso de uso: eliminación de un sistema existente [CU-6.4].**

**Tipo:** secundario.

**Actor:** administrador.

**Propósito:** permitir a un administrador eliminar toda la información de un sistema previamente registrado en la solución.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-6].

**Descripción:** el administrador podrá eliminar cualquier sistema previamente registrado, eliminando también todos los despliegues realizados en el mismo. Si se opta por eliminar un sistema, los datos y metadatos “locales” de las aplicaciones desplegadas se borrarán y no podrán recuperarse.

**Curso típico de eventos:**

1) El administrador selecciona “eliminar sistema existente”.

2) El sistema elimina el registro del sistema, junto con todos los despliegues de aplicaciones realizados sobre el mismo.

**8.5.109.- Caso de uso: gestionar información [CU-7].**

**Tipo:** secundario.

**Actor:** consumidor.

**Propósito:** permitir a un usuario consumidor y, por extensión, a un proceso por lotes que utilice las credenciales de un usuario consumidor (o administrador), gestionar la información (datos de usuario) que haya sido previamente modelada por un usuario administrador para la aplicación generada correspondiente.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; casos de uso [CU-7], [CU-7.1], [CU-7.2], [CU-7.3], [CU-7.4] y [CU-7.5].

**Descripción:** el usuario consumidor dispondrá de la opción de crear nuevos registros de cada entidad modelada, así como consultar, modificar y eliminar registros de entidades modeladas previamente. Si se opta por eliminar un registro, todos los datos de usuario asociados a dicho registro se eliminarán y no podrán recuperarse. Aunque técnicamente el acceso a través de la interfaz gráfica de usuario difiere del acceso a través de los microservicios generados, desde un punto de vista funcional son equivalentes, de ahí que se traten conjuntamente.

#### Curso típico de eventos:

Este caso de uso secundario es el nexo de partida de varios casos de uso secundarios y permite al usuario acceder a estos otros casos, por lo que no existe como tal un único escenario. Por ello, nos remitimos a los casos de uso secundarios del apartado referencias para más información.

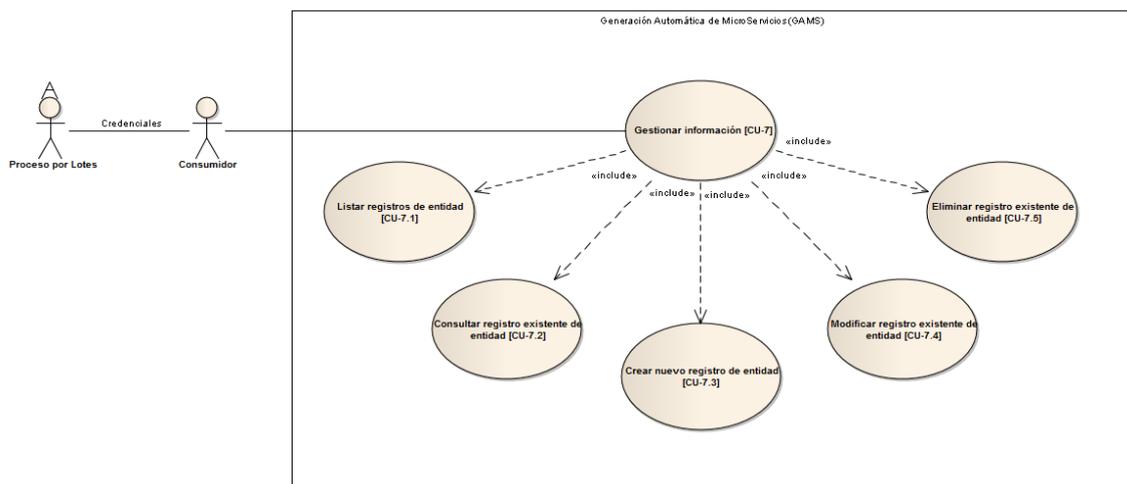


Fig. 8.26.- Casos de uso secundarios de "gestionar información".

#### **8.5.110.- Caso de uso: listar registro existente de entidad [CU-7.1].**

**Tipo:** secundario.

**Actor:** consumidor.

**Propósito:** permitir a un consumidor consultar la información de todos los registros existentes de una entidad previamente modelada, incluyendo por tanto información de sus atributos de entidad, objetos valor, atributos de objetos valor y relaciones.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-7].

**Descripción:** el consumidor dispondrá de la opción de consultar la información de todos los registros de la entidad especificada.

**Curso típico de eventos:**

- 1) El consumidor selecciona una de las entidades que forman parte del dominio de la aplicación generada que está siendo consumida.
- 2) El sistema muestra la información de todos los registros de dicha entidad.

#### **8.5.111.- Caso de uso: consultar registro existente de entidad [CU-7.2].**

**Tipo:** secundario.

**Actor:** consumidor.

**Propósito:** permitir a un consumidor consultar la información de un registro existente de una entidad previamente modelada, incluyendo por tanto información de sus atributos de entidad, objetos valor, atributos de objetos valor y relaciones.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-7].

**Descripción:** el consumidor dispondrá de la opción de consultar la información del registro especificado.

**Curso típico de eventos:**

- 1) El consumidor selecciona uno de los registros que forman parte del dominio de la aplicación generada que está siendo consumida.
- 2) El sistema muestra toda la información relacionada con dicho registro.

### **8.5.112.- Caso de uso: crear nuevo registro de entidad [CU-7.3].**

**Tipo:** secundario.

**Actor:** consumidor.

**Propósito:** permitir a un consumidor crear un nuevo registro de una entidad previamente modelada, incluyendo información de sus atributos de entidad, objetos valor, atributos de objeto valor y relaciones.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-7].

**Descripción:** el consumidor seleccionará “crear nuevo registro” y establecerá los atributos iniciales del mismo (en función del modelo de dominio de la aplicación generada).

**Curso típico de eventos:**

1) El consumidor selecciona “crear nuevo registro” y establece valores para los diferentes atributos, objetos valor, atributos de objeto valor y relaciones que conformen el modelo de dominio de la entidad especificada para el nuevo registro.

2) El sistema verifica la información especificada, validando la corrección de la misma a través del uso de las diferentes políticas modeladas (políticas de entidad, políticas de atributo de entidad, políticas de objeto valor, políticas de atributo de objeto valor y políticas de tipo de datos); en caso de ser correcta toda la información, crea el nuevo registro de entidad.

### **8.5.113.- Caso de uso: modificar registro existente de entidad [CU-7.4].**

**Tipo:** secundario.

**Actor:** consumidor.

**Propósito:** permitir a un consumidor modificar un registro existente de una entidad previamente modelada, incluyendo información de sus atributos de entidad, objetos valor, atributos de objeto valor y relaciones.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-7].

**Descripción:** el consumidor seleccionará “modificar registro existente” y establecerá los nuevos valores para los atributos del mismo (en función del modelo de dominio de la aplicación generada).

**Curso típico de eventos:**

1) El consumidor selecciona “modificar registro existente” y establece valores para los diferentes atributos, objetos valor, atributos de objeto valor y relaciones que conformen el modelo de dominio de la entidad especificada para el registro existente.

2) El sistema verifica la información especificada, validando la corrección de la misma a través del uso de las diferentes políticas modeladas (políticas de entidad, políticas de atributo de entidad, políticas de objeto valor, políticas de atributo de objeto valor y políticas de tipo de datos); en caso de ser correcta toda la información, actualiza la información del registro de entidad especificado.

**8.5.114.- Caso de uso: eliminar registro existente de entidad [CU-7.5].**

**Tipo:** secundario.

**Actor:** consumidor.

**Propósito:** permitir a un consumidor eliminar un registro existente de una entidad previamente modelada.

**Referencias:** requisitos [R-1], [R-2], [R-3], [R-4] y [R-6]; caso de uso [CU-7].

**Descripción:** el consumidor podrá eliminar cualquier registro existente, eliminando consigo a todos y cada uno de los diferentes elementos contenidos en dicho registro de entidad, como puedan ser atributos, objetos valor, atributos de objetos valor y relaciones. Toda la información (datos de usuario) asociados a dicho registro (y de sus elementos contenidos) se borrará y no podrá recuperarse.

**Curso típico de eventos:**

1) El consumidor selecciona “eliminar registro existente” y especifica el registro de entidad a eliminar.

2) El sistema elimina toda la información de usuario asociada a dicho registro de entidad.

## 9.- ANEXO B: DISEÑO TÉCNICO.

*Aquellos que se enamoran sólo de la práctica, sin cuidar de la exactitud o de la ciencia, son como el piloto que se embarca sin timón ni brújula, y nunca sabrá dónde va a parar.*  
Leonardo da Vinci (1452-1519)

Se presentan a continuación los diferentes productos obtenidos durante la fase de Diseño Técnico.

El objetivo del presente Anexo B es sacar fuera del cuerpo principal de la Memoria material voluminoso, de carácter eminentemente técnico, de cara a no romper el hilo conductor del presente documento sin, evidentemente, dejar incompleto el material resultante.

El presente Anexo B cubre en profundidad los siguientes apartados, que ya fueron mencionados en el punto 4.3:

- Arquitectura tecnológica de la solución.
- Documentación de código fuente.
- Modelo de datos.
- Modelo de procesos.

### 9.1.- Arquitectura tecnológica de la solución.

Precisamente los dos requisitos no funcionales definidos para el presente Trabajo sobre Generación Automática de MicroServicios (GAMS) atienden a la capacidad del mismo para ser ejecutado en múltiples sistemas y plataformas, de características diferentes, basándose en soluciones abiertas (véase apartado 8.1):



- [R-5]: se utilizarán cuantos estándares existentes sea posible, frente a mecanismos y tecnologías propietarias.
- [R-6]: se posibilitará el uso de entornos de ejecución de diferente naturaleza y capacidad, como computadores personales y dispositivos embarcables.

Por ello, múltiples son las plataformas que se han utilizado para construir y validar la solución, destacando las siguientes:

- Plataforma PC:
  - Hardware: TOSHIBA Qosmio X70-B-10T [208], con procesador Intel Core i7-4720HQ a 2,60 GHz, 16 GB de memoria RAM y disco duro de 1 TB.
  - Sistema operativo de cliente (físico): Microsoft Windows 10 Professional 64 bits (versión 1803).
  - Entorno de virtualización (hipervisor): Microsoft Hyper-V.
  - Sistema operativo de servidor (virtual): Microsoft Windows Server 2012 R2.
  - Servidor web y de aplicaciones: Microsoft Internet Information Server 8.5 (IIS85).
  - Entorno de ejecución: PHP 7.1.4.
  - Sistemas gestores de bases de datos: Microsoft SQL Server 2017 y SQLite 3.15.1.
  - Entorno de desarrollo: JetBrains PhpStorm 2017.3.
  - Navegadores web: Microsoft Edge 42, Microsoft Internet Explorer 11, Mozilla Firefox ESR 52, Mozilla Firefox Developer Edition 62 y Google Chrome 67.

- Plataforma Synology:
  - Hardware: Synology DiskStation DS415+ [209], con procesador Intel Atom C2538 a 2,4 GHz, con 2 GB de memoria RAM y 4 discos duros en RAID 5 de 6 TB cada uno.
  - Sistema operativo: Synology Disk Station Manager 6.2 (DSM), basado en GNU/Linux.
  - Servidor web y de aplicaciones: Synology WebStation 2.1 (véanse [210] y [211]) y Apache HTTP Server 2.2 [212].
  - Entorno de ejecución: PHP 7.0.30 [213].
  - Sistemas gestores de bases de datos: MariaDB 10 (véanse [214] y [215]) complementado con phpMyAdmin 4.7 [216] y SQLite 3.15.1.
  
- Plataforma Raspberry:
  - Hardware: Raspberry PI modelo B+, con procesador Broadcom BCM2835, 512 MB de memoria RAM y una tarjeta extraíble de memoria Secure Digital (SD) de 16 GB para el almacenamiento persistente (dado que este tipo de dispositivo carece de disco duro).
  - Sistema operativo: Raspbian OS 8, basado en GNU/Linux para procesadores ARM.
  - Servidor web y de aplicaciones: Apache HTTP Server 2.x.
  - Entorno de ejecución: PHP 7.x.
  - Sistemas gestores de bases de datos: SQLite 3.x.

Tal y como se mencionó en el apartado 2.1.2, el objetivo de la arquitectura tecnológica escogida es dar cabida de la mejor manera posible a los diferentes requisitos tanto funcionales como no funcionales, expresados en el apartado 8.1

La arquitectura tecnológica escogida propugna una división funcional basada en capas lógicas, es decir, una división de las tareas en las que se descompone la solución global basada en la funcionalidad que aporta cada capa. En el siguiente diagrama (fig. 9.1), puede verse esta división con mayor detalle:

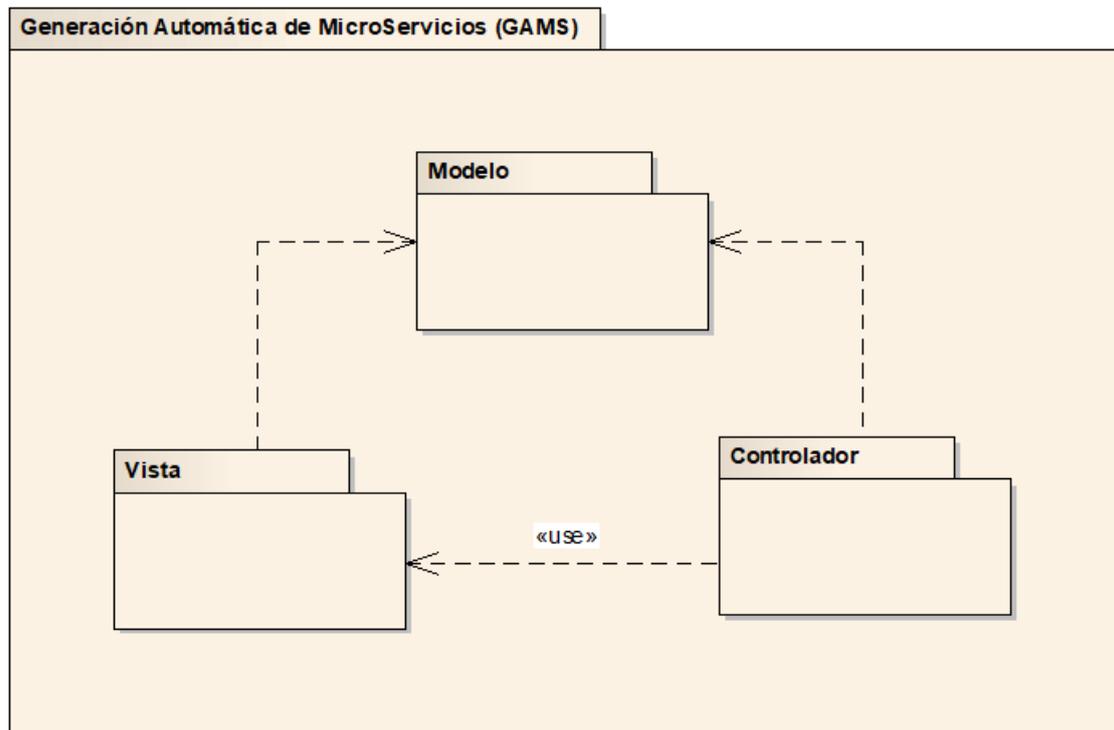


Fig. 9.1.- Estilo arquitectónico de la solución.

Como puede observarse, el estilo arquitectónico escogido es el conocido como Modelo-Vista-Controlador (ampliamente reconocido como MVC, por las siglas en inglés de *Model-View-Controller*).

Este estilo arquitectónico propugna la creación de un “modelo” que encapsule la funcionalidad del dominio de la solución, el cual es gestionado por parte de los usuarios a través de las diferentes acciones permitidas por el “controlador”, que a su vez presenta visualmente dicho “modelo” al usuario a través de la “vista”.

Véase [217] con documentos, podríamos decir hoy que históricos, sobre su concepción a finales de la década de 1970 para el entorno de programación SmallTalk; en [218] se revisa el modelo treinta años más tarde, siendo ya aplicado en entornos Java como el estilo principal para construir interfaces de usuario, especialmente en entornos web; por último, en [27] (que bien puede considerarse un añadido a [219]) se confronta MVC con otras variantes que han aparecido a lo largo de estos años.

Ampliando los detalles de la arquitectura, nos encontramos que a nivel de componentes la solución GAMS se construye como un paquete de aplicación del *framework* Symfony (véase el apartado 4.4.1.13), denominado *application bundle*, que a su vez hace uso de numerosos paquetes de biblioteca (denominados genéricamente como *bundles*) para cuestiones transversales: desde la seguridad, pasando por la funcionalidad propia de una aplicación web, hasta cuestiones más específicas ligadas con los servicios web basados en REST.

En cuanto a la organización detallada del paquete de aplicación (véase fig. 9.2), siguiendo con el estilo MVC, tenemos:

- Modelo:
  - Consta de las clases que definen las diferentes entidades del dominio, esto es, aquéllas que configuran el meta-esquema de las aplicaciones generadas para consumidores.
  - Asimismo, para cada una de estas clases de entidad, se ha definido también una clase de repositorio, la cual permite al subsistema de acceso a la base de datos agilizar consultas especialmente complejas.
  - Véase [220] para obtener información técnica detallada y ejemplos acerca de cómo integrar Doctrine ORM (véase el apartado □) y el acceso a bases de datos con el marco de desarrollo Symfony (véase el apartado 4.4.1.13).
  - En el apartado 8.3 puede consultarse el listado completo de entidades del dominio.

- Controlador:
  - Consta de las clases y métodos que definen cada una de las acciones que pueden realizarse sobre el modelo de dominio por parte de los usuarios.
  - Téngase en cuenta que los cambios en el modelo de dominio tienen su oportuno reflejo en las aplicaciones generadas para consumidores a través de las acciones desencadenadas mediante los correspondientes generadores de artefactos y de esquemas.
  - Estas acciones se definen tanto para las interacciones a través de la interfaz gráfica de usuario, utilizable por usuarios humanos como administradores y consumidores, como para el interfaz API estilo REST generado automáticamente, en este caso, utilizable por terceros sistemas.
  - Véanse [221] y [222] para obtener información técnica detallada y ejemplos acerca de cómo construir este paquete y publicar su funcionalidad a través de direcciones web.
  
- Vista:
  - Consta de un conjunto de plantillas, cada una formada por al menos un fichero Twig (véase el apartado □), que combinadas entre sí devuelve el código HTML, CSS y JavaScript de las diferentes páginas web que componen la interfaz gráfica de usuario de la solución GAMS.
  - Desde un punto de vista tradicional, la vista ya no es directamente programada; en su lugar, cada acción de cada controlador se limita a indicar qué datos serán mostrados y con qué plantilla serán mostrarlos. La infraestructura de la aplicación (componentes como Twig y Symfony, entre otros), se encargan del resto, agilizando enormemente el desarrollo de las aplicaciones.
  - Véase [223] para obtener información detallada y ejemplos acerca de cómo construir interfaces web de usuario utilizando estas tecnologías.

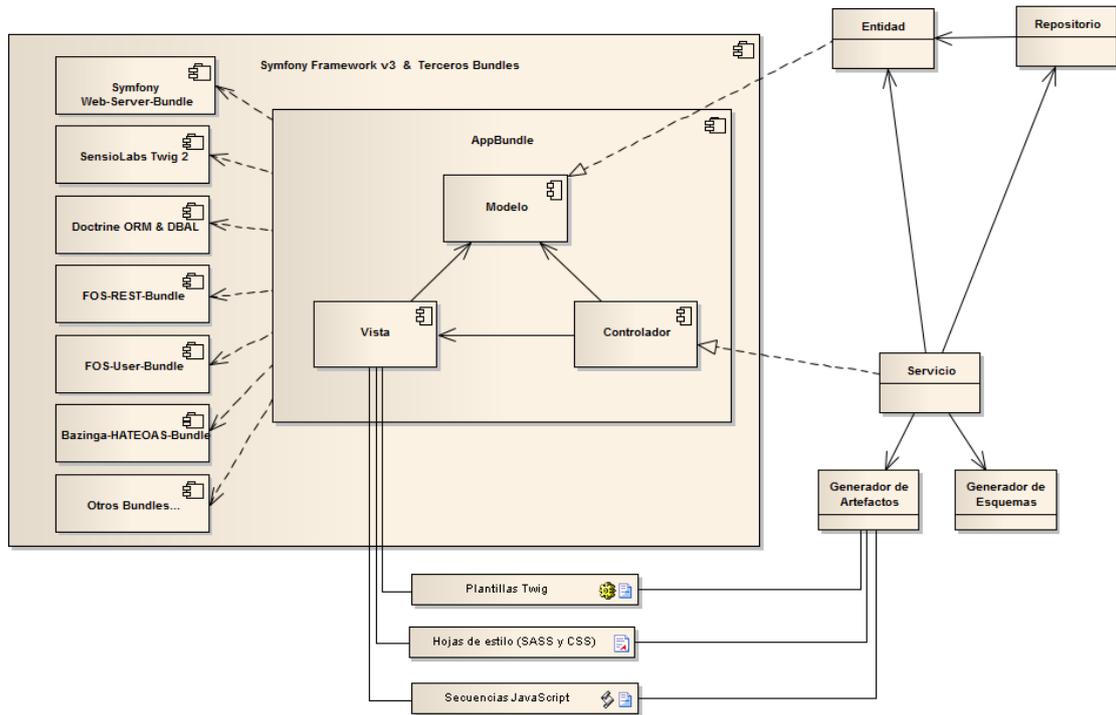


Fig. 9.2.- Diagrama detallado de arquitectura.

Por último, atendiendo además a la especial particularidad de la solución GAMS en cuanto a su capacidad para ser ejecutada en múltiples plataformas diferentes, se presenta un diagrama de despliegue (véase fig. 9.3), donde a modo de ejemplo se han distribuido las tres principales arquitecturas tecnológicas: PC servidor, sistema específico para base de datos y cliente basado en IoT.

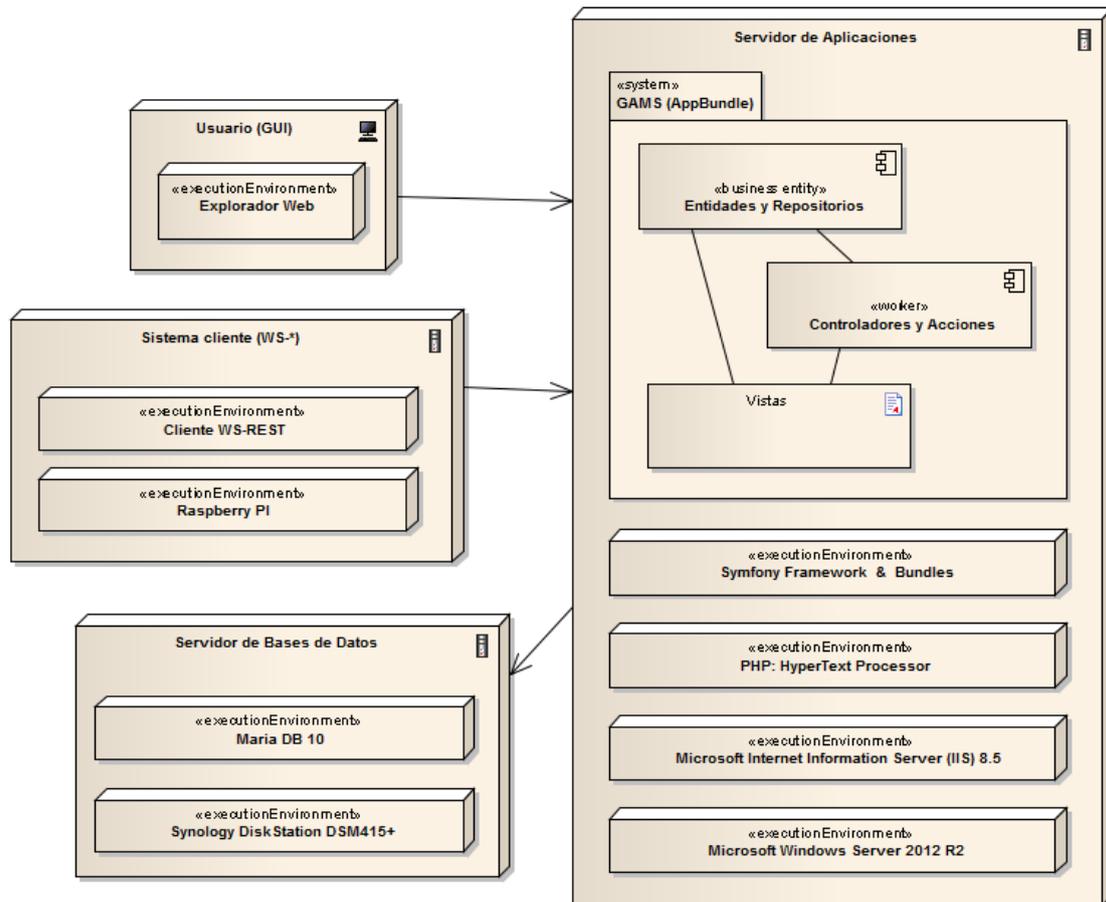


Fig. 9.3.- Diagrama de despliegue.

## 9.2.- Documentación de código fuente.

Mencionamos a continuación los comentarios de documentación de código fuente PHP, por ofrecer éstos una ayuda singular a la hora tanto de entender el código fuente, y con ello la solución GAMS descrita, como a la hora de realizar futuros mantenimientos tanto correctivos como evolutivos.

Utilizando las técnicas descritas en [191], la solución GAMS hace uso de comentarios de documentación de código fuente, conocidos como PHPDOC.

```
39 #
40 #
41 /**
42  * Symfony's MVC controller for "Application" domain entities.
43  *
44  * @Route("Administration/Application")
45  * @package AppBundle\Controller\Administration
46  */
47 class ApplicationController
48     extends SymfonyController
49 {
50     # -----
51     # -----
52     # -----
53     /**
54      * Lists all existing applications.
55      *
56      * @return \Symfony\Component\HttpFoundation\Response
57      *
58      * @Route(
59      *     "/",
60      *     name = "Administration_Application_Index" )
61      * @Method(
62      *     "GET" )
63      */
64     public function indexAction()
65     {
66         # Unit of work.
67         $em = $this->getDoctrine()->getManager();
68         # Repository
69         $applications = $em->getRepository( 'AppBundle\Entity\Application::class' )->findAll();
70         # Render
71         return $this->render(
72             View: 'Administration/Application/Index',
73             parameters: [
```

Fig. 9.4.- Documentación de código fuente PHPDOC; ejemplo de uso en la clase “Application Controller”.

La fig. 9.4 muestra parte del contenido del fichero “ApplicationController.php”, donde se observa el código fuente de la clase homónima comentado de varias maneras distintas:

- La clase “ApplicationController” ha sido comentada utilizando PHPDOC e incluye adicionalmente anotaciones específicas como clase de un “controlador”, como son la ruta base para el acceso web (URL) y los métodos HTTP admitidos.
- El método “indexAction” incluye asimismo código que representa tanto documentación PHPDOC (en este caso, un texto descriptivo y el tipo de datos que será retornado) como anotaciones propias de Symfony (igualmente, relacionadas con la URL de acceso y los verbos HTTP mediante los que acceder a dicha acción).

La solución GAMS hace uso de la herramienta SAMI [224] para recopilar toda esta documentación y generar un completo manual técnico. De hecho, toda la documentación del código fuente del propio *framework* de Symfony está generada mediante esta misma herramienta.

La herramienta genera toda la documentación de los ficheros de código PHP en el directorio “\$GAMS/app/Resources/doc/Sami”, desde donde puede consultarse abriendo el fichero “index.html”.

En la fig. 9.5 puede verse el resultado obtenido al generar la documentación del código de la solución GAMS.

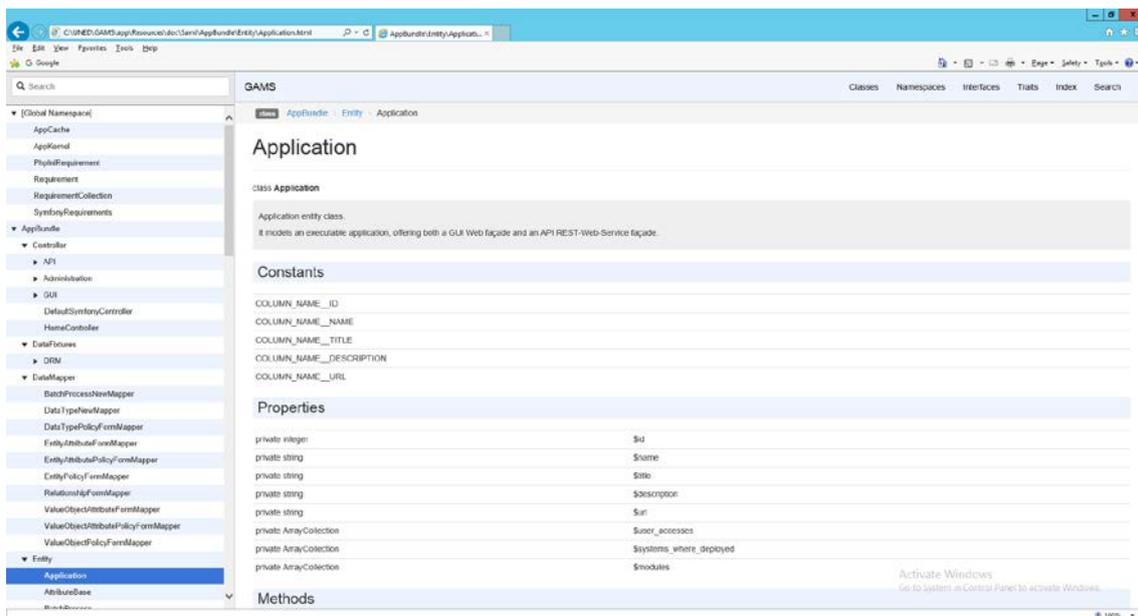


Fig. 9.5.- Documentación del código fuente de la solución GAMS, generada mediante la herramienta SAMI.

### 9.3.- Modelo de datos.

El modelo de datos determinará la organización de la información de la que hace uso el sistema desarrollado. Esta organización será presentada desde dos perspectivas:



- Basada en la organización en memoria RAM, sobre la cual se apoyará el programa en ejecución. Esta organización se representa mediante el diagrama de clases.
- Basada en la persistencia en disco de dicha información, mediante la cual se almacenarán los datos entre diferentes ejecuciones de la aplicación. Esta organización se representa mediante el diagrama entidad/relación y el diagrama relacional.

### **9.3.1.- Diagrama de clases.**

Se muestran a continuación los diagramas de clases que muestran todo el código del paquete de aplicación utilizado para construir GAMS. Téngase presente que, debido al tamaño de la solución, se ha necesitado dividir en múltiples diagramas para poder representarlos con un nivel de detalle suficiente.

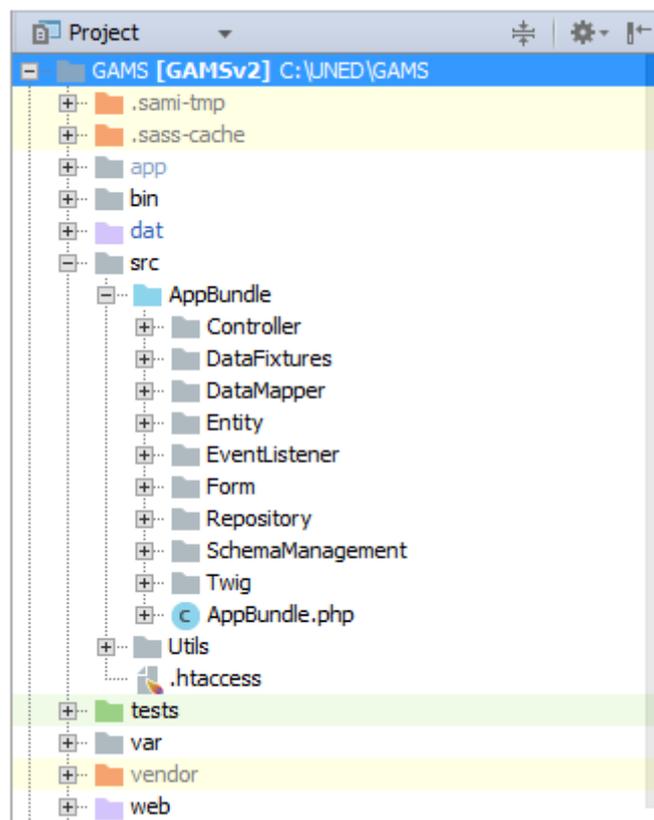


Fig. 9.6.- Estructura de directorio del código fuente del paquete AppBundle de la solución GAMS.

La fig. 9.6 muestra la estructura de directorios del código fuente del paquete de aplicación, denominado “AppBundle”, y los directorios en los que se organizan los diferentes tipos de clases que contiene:

- Fichero “AppBundle.php”: clase que define la configuración específica de la solución GAMS dentro del marco de desarrollo Symfony.
- Subdirectorio “Controller”: contiene todas las clases que actúan como controladores en la interfaz de usuario, tanto gráfica para humanos como de servicios para terceros sistemas. En este subdirectorio se encuentran clases como “DataTypeController”, la cual define las acciones que pueden realizarse sobre la entidad de dominio “DataType”, encargada del modelado de tipos de datos.
- Subdirectorio “DataFixtures”: contiene todas las clases que generan datos iniciales, como parte de la creación primigenia de las bases de datos de administración y de las aplicaciones para consumidores de ejemplo.



- Subdirectorio “DataMapper”: contiene clases que permiten ligar los datos introducidos a través de formularios HTML con las entidades del dominio existentes en un momento dado en la solución.
- Subdirectorio “Entity”: contiene todas las clases que actúan como modelo de dominio. En este subdirectorio se encuentran clases como “EntityAttributePolicy”, la cual define a nivel de meta-esquema una política de atributo de entidad concreta (reglas de validación de un tipo de atributo, siendo el atributo propio de una entidad).
- Subdirectorio “EventListener”: contiene todas las clases que responden ante eventos de Symfony; actualmente sólo se ha definido la clase “AuthenticationSuccessHandler” encargada de gestionar el inicio de sesión.
- Subdirectorio “Form”: contiene todas las clases que gestionan los datos tanto enviados como recibidos mediante formularios HTML.
- Subdirectorio “Repository”: contiene todas las clases que actúan como repositorios para el modelo de dominio. En este subdirectorio se encuentran clases como “GlobalLogInRepository”, la cual define métodos concretos para conocer qué usuarios pueden o no iniciar sesión en una determinada aplicación generada para consumidores.
- Subdirectorio “SchemaManagement”: contiene todas las clases tanto generadoras de artefactos como generadoras de esquemas. En este subdirectorio se encuentran clases como “ApplicationSchemaGenerator”, encargada de crear, actualizar/sincronizar y eliminar la base de datos asociada a cada una de las aplicaciones generadas para consumidores (véanse los casos de usos secundarios del “caso de uso: gestionar aplicaciones [cu-4]” para disponer de más detalles al respecto).
- Subdirectorio “Twig”: contiene todas las clases auxiliares que ofrecen utilidades relacionadas con las plantillas Twig; actualmente sólo se ha definido la clase “GetClassTwigExtension” encargada de obtener el nombre de la clase del parámetro especificado dentro de una plantilla.

Nótese que, debido a su enorme extensión, no se enumerarán los más de 120 ficheros de código fuente PHP que conforma el paquete “AppBundle” de la solución GAMS. Remitimos al lector a dicho material, donde encontrará la pertinente documentación de código fuente descrita en el apartado 9.2.

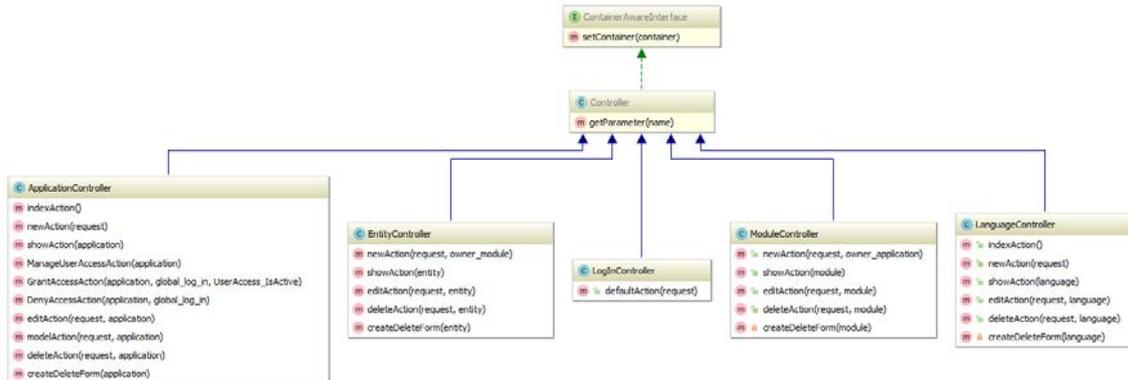


Fig. 9.7.- Diagrama de clases: vista detallada de un extracto de clases del subdirectorio “AppBundle/Controller”.

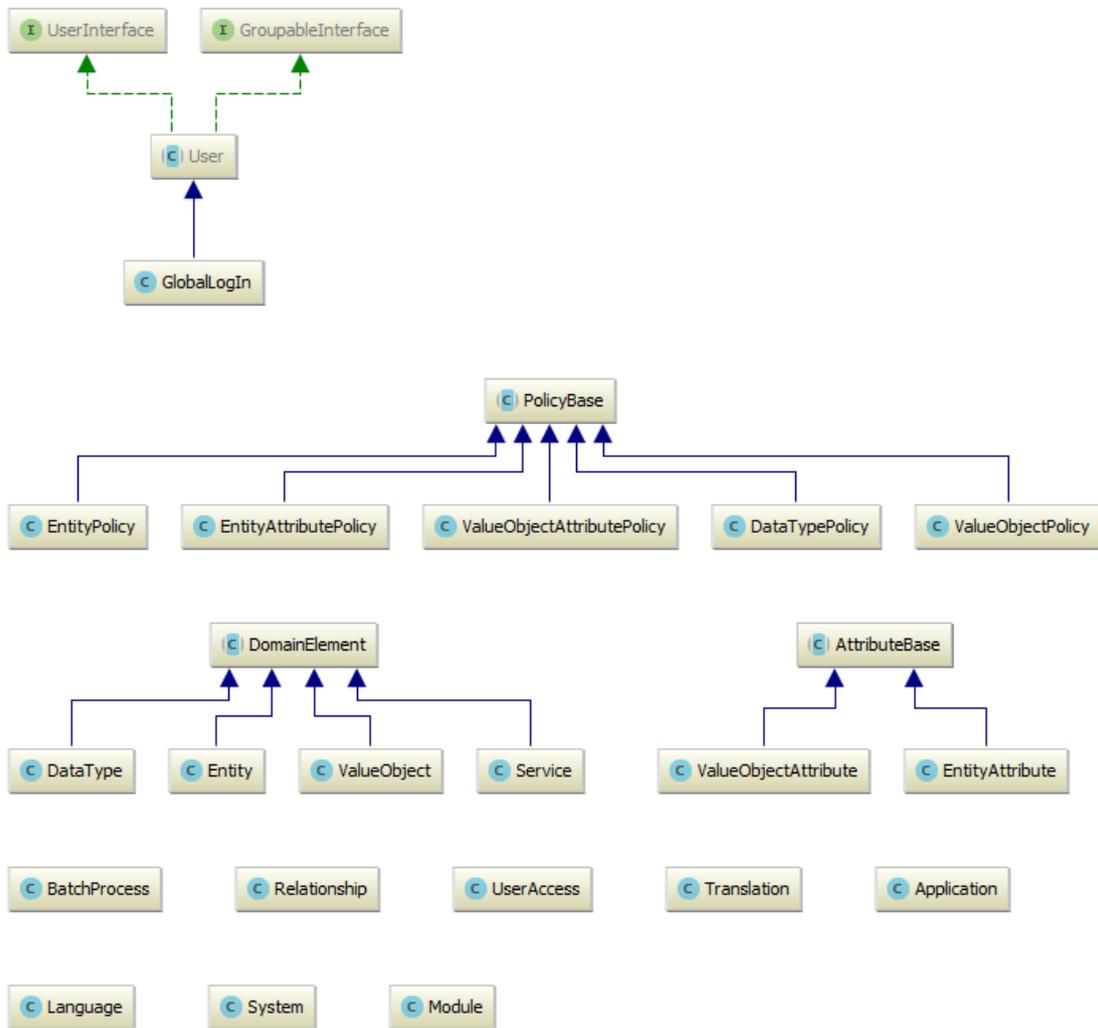


Fig. 9.8.- Diagrama de clases: vista reducida de las clases del subdirectorio “AppBundle/Entity”.

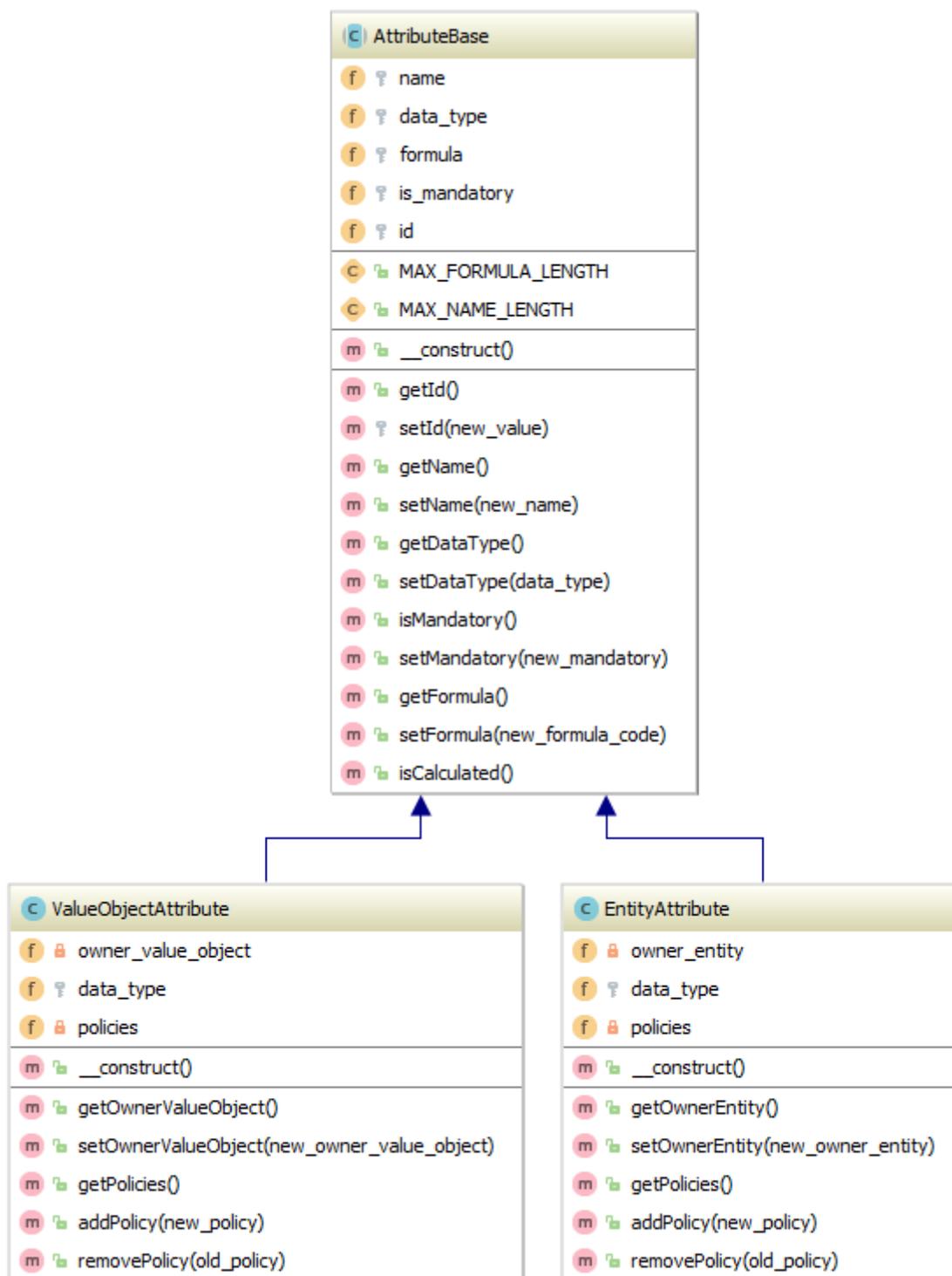


Fig. 9.9.- Diagrama de clases: vista detallada de la jerarquía de clases que modelan los atributos del dominio.



### **9.3.2.- Diagramas de bases de datos.**

Los diagramas de bases de datos muestran cómo se organizará la información dentro de la base de datos. Se presentan los dos diagramas más conocidos:

- A nivel conceptual, el diagrama entidad/relación.
- A nivel lógico, el diagrama relacional.

Téngase en cuenta que a través de la solución GAMS podrán existir múltiples aplicaciones: una única aplicación de administración y varias aplicaciones generadas para consumidores.

Solamente la aplicación de administración tiene un modelo de datos fijado de antemano, a modo de meta-esquema, pues su función es precisamente permitir el modelado de las aplicaciones generadas para consumidores.

9.3.2.1.- *Diagrama entidad/relación de la aplicación de administración.*

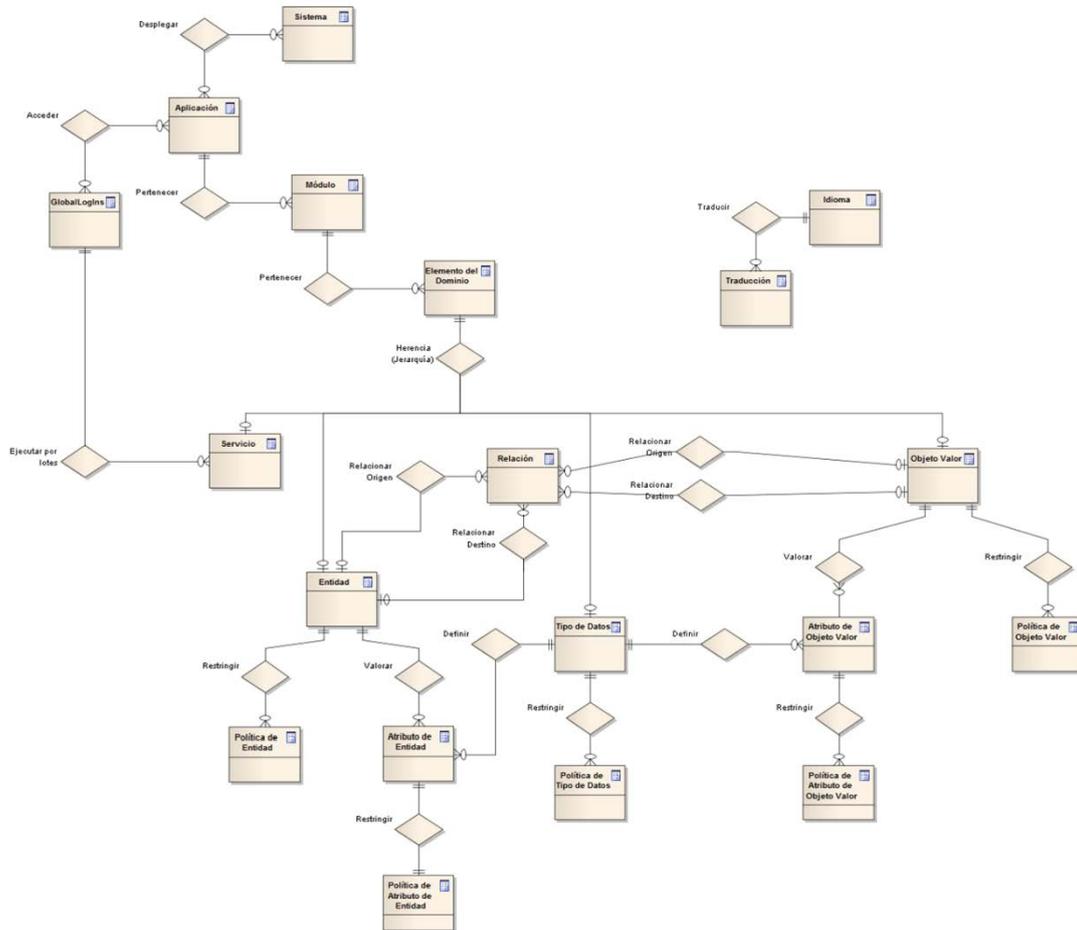


Fig. 9.10.- Diagrama entidad/relación de la aplicación de administración.

9.3.2.2.- *Diagrama relacional de la aplicación de administración.*

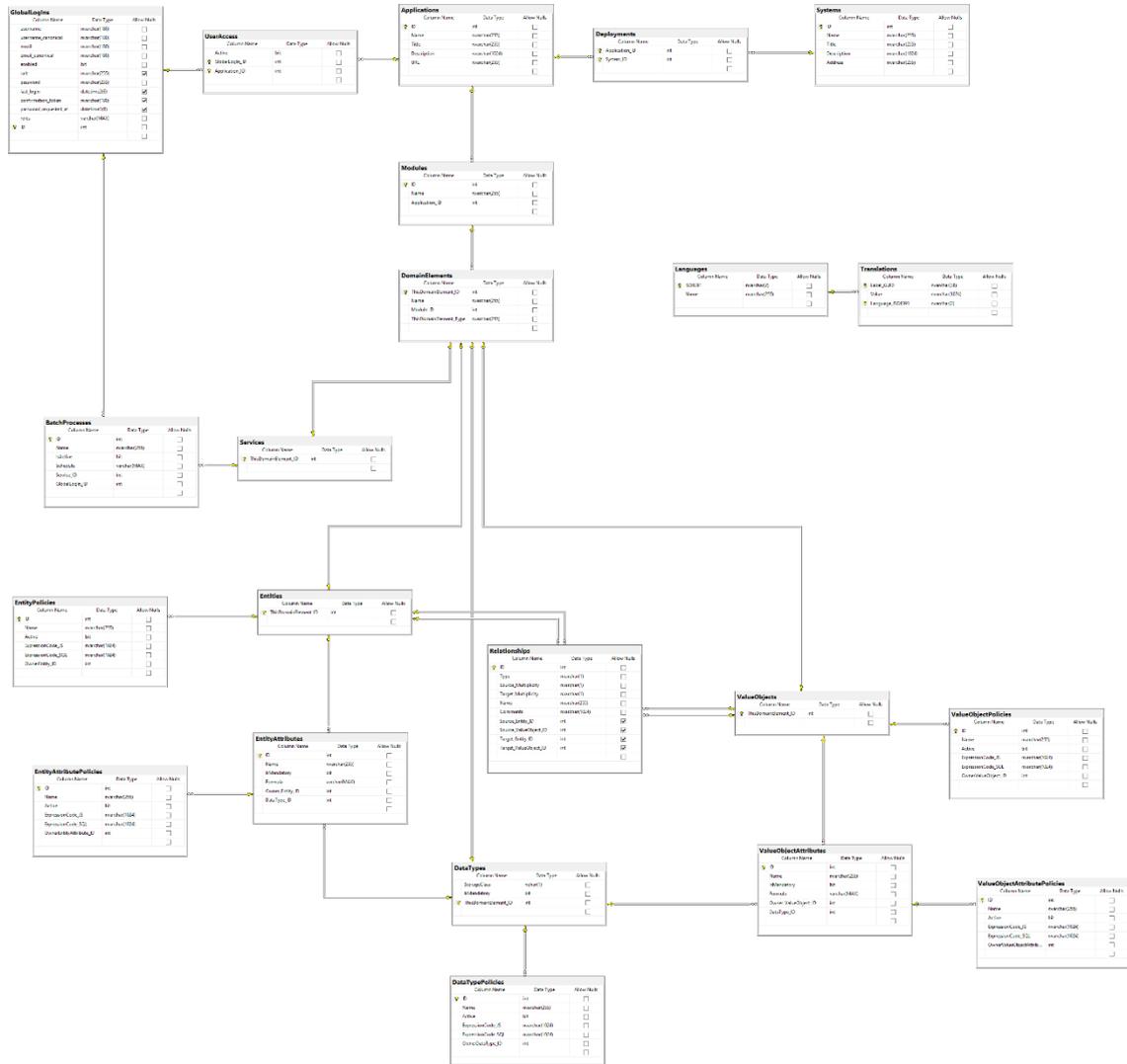


Fig. 9.11.- Diagrama relacional de la aplicación de administración.

La fig. 9.11 muestra el diagrama asociado al modelo relacional de la aplicación de administración que forma parte de la solución GAMS. Este diagrama ha sido obtenido directamente a partir de la base de datos instalada en la plataforma de referencia para PC, la cual está basada en el sistema gestor “Microsoft SQL Server 2017”, utilizando para ello la herramienta de administración “Microsoft SQL Server Management Studio” del propio sistema gestor de bases de datos (véase el apartado 10.2.1.3 para más información).

## 9.4.- Modelo de procesos.

El modelo de procesos determinará la funcionalidad de la aplicación. Este modelo consta de los diferentes procesos, a nivel técnico, que han sido descritos a nivel funcional a través de los cursos típicos de eventos y escenarios alternativos de los casos de uso presentados durante la etapa de Análisis Funcional.

A este modelo de procesos se le denomina también “vista dinámica del sistema”, pues determina el comportamiento del mismo durante su ejecución. El modelo de procesos se ha dividido en dos apartados:

- El diagrama de navegación, donde se presenta cómo el usuario puede navegar a través de la interfaz de usuario basada en Web.
- El catálogo de procesos, donde se desglosan todos los procesos que intervienen en el funcionamiento del sistema.

### 9.4.1.- Diagrama de navegación.

Se presenta en la fig. 9.12 el diagrama de navegación de la solución GAMS, al estar la interfaz gráfica de usuario basada en tecnología Web. Podrá comprobarse en este diagrama de navegación cómo el usuario interacciona con la aplicación y en qué orden puede hacerlo.

Con el objetivo de aumentar considerablemente la legibilidad del diagrama de navegación, se han omitido de los respectivos identificadores los sufijos que, por convención, el *framework* Symfony recomienda utilizar como nombre de las clases de controlador y de los respectivos métodos de acción.

Así, la acción de creación de un nuevo “Tipo de Datos” como parte del modelo de dominio de una aplicación generada para consumidores, tendrá como identificador del método función “newAction”, perteneciente a la clase controladora “DataTypeController”.

En su lugar, este diagrama basado en notación UML 2.1 simplifica dichos identificadores representando a la clase como el componente “DataType” y a la acción con el puerto “new”, lo que en notación PHP escribiríamos, incluyendo el correspondiente espacio de nombres, como: “\AppBundle\Controller\Administration\DataTypeController::newAction”.

Puede confrontarse esta simplificación en los identificadores reales utilizados en el componente “HomeController”, que gestiona la página inicial de la solución GAMS, frente al resto de elementos del diagrama.

Debido a la barra de navegación situada en la cabecera de todas las páginas de la solución GAMS (véase más adelante el apartado 10.3.3), una vez iniciada la sesión, desde cualquier página es posible acceder a determinadas acciones, siempre que el usuario tenga el rol y los permisos necesarios:

- Panel de control “DashboardController::indexAction”.
- Inicio de cada aplicación generada para consumidores “ConsumerController::indexAction”.
- Administración de la Solución “AdministrationController::indexAction”.
- Índice de Administración de Aplicaciones “ApplicationController::indexAction”.
- Índice de Administración de Sistemas “SystemController::indexAction”.
- Índice de Administración de Usuarios “GlobalLogInController::indexAction”.
- Índice de Administración de Idiomas “LanguageController::indexAction”.



- Visualización del propio perfil de usuario “ProfileController::showAction”.
- Modificación de la propia contraseña de usuario “ChangePasswordController::changePasswordAction”.
- Cierre de sesión y abandono de la solución “SecurityController::logoutAction”.

Asimismo, esta barra de navegación incluye, a modo de ejemplo, una serie de enlaces hacia páginas de la UNED, externas a la solución GAMS. Para navegar hacia dichos enlaces, cada uno de ellos abre una nueva ventana (o pestaña) en el navegador del usuario.

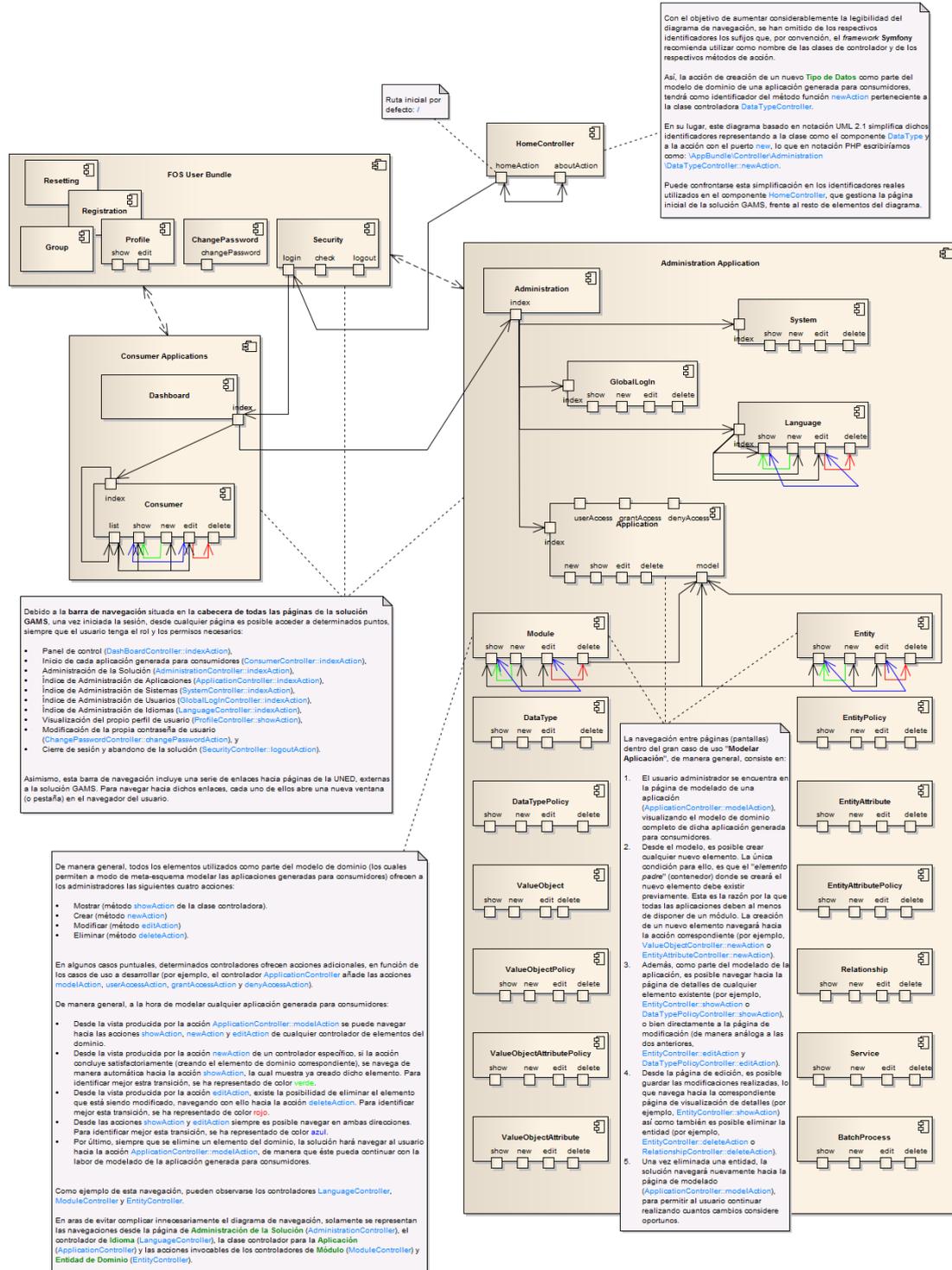


Fig. 9.12.- Diagrama de navegación a través de la interfaz gráfica de usuario, basada en tecnologías web.

Indicar que el concepto de página es realmente una abstracción. Atendiendo a la arquitectura “Modelo-Vista-Controlador” (véase el apartado 9.1 para más información), cuando un usuario navega hacia una URL, ésta es interpretada por componente “Routing” [221] del framework Symfony, encargado de identificar tanto a la clase que actúa como “controlador” [222], como al método (función) que realiza la “acción” esperada por el usuario. Como resultado de dicha acción, el “modelo” (normalmente, los datos almacenados en la base de datos) sufrirá las alteraciones correspondientes y se devolverá al usuario la nueva “vista” [223] apropiada (por ejemplo, una página HTML junto con las hojas de estilo CSS y código JavaScript asociado a la misma, o bien un texto en formato JSON o XML que representa información que deberá ser procesada por la página en la que actualmente se encuentre el usuario).

La página inicial a la que será dirigido el usuario automáticamente cuando se conecte a la aplicación tiene por ruta “/” y se corresponde con la acción “HomeAction” del controlador “HomeController” (se encuentra debidamente identificada en la parte superior central del diagrama).

Mediante la línea de mandatos “`PHP bin/console debug:router`” la plataforma Symfony lista todas las rutas registradas en la solución GAMS, indicando para cada una de ellas el nombre (identificador asignado), los protocolos HTTP aceptados, aquellas restricciones de acceso que hayan sido configuradas y, lo más importante, la URL parametrizada que representa a cada ruta.

Se lista a continuación esta información, extraída según el método indicado en el párrafo anterior:

| <b>Método(s)</b> | <b>Ruta</b>                          |
|------------------|--------------------------------------|
| <b>HTTP</b>      | <b>(URL publicada parametrizada)</b> |
| ANY              | <code>/_wdt/{token}</code>           |
| ANY              | <code>/_profiler/</code>             |
| ANY              | <code>/_profiler/search</code>       |

| <b>Método(s)<br/>HTTP</b> | <b>Ruta<br/>(URL publicada parametrizada)</b>   |
|---------------------------|---|
| ANY                       | /_profiler/search_bar   |
| ANY                       | /_profiler/phpinfo  |
| ANY                       | /_profiler/{token}/search/results   |
| ANY                       | /_profiler/open   |
| ANY                       | /_profiler/{token}  |
| ANY                       | /_profiler/{token}/router   |
| ANY                       | /_profiler/{token}/exception  |
| ANY                       | /_profiler/{token}/exception.css  |
| ANY                       | /_error/{code}._format  |
| GET POST                  | /login  |
| POST                      | /login_check  |
| GET POST                  | /logout   |
| GET                       | /profile/   |
| GET POST                  | /profile/edit   |
| GET POST                  | /register/  |
| GET                       | /register/check-email   |
| GET                       | /register/confirm/{token}   |
| GET                       | /register/confirmed   |
| GET                       | /resetting/request  |
| POST                      | /resetting/send-email   |
| GET                       | /resetting/check-email  |
| GET POST                  | /resetting/reset/{token}  |
| GET POST                  | /profile/change-password  |
| GET                       | /Administration/API/Applications  |
| ANY                       | /Administration/  |
| GET                       | /Administration/Application/  |
| GET POST                  | /Administration/Application/New   |
| GET                       | /Administration/Application/{id}  |
| GET                       | /Administration/Application/{id}/ManageUserAccess   |
| GET                       | /Administration/Application/{Application_ID}/GrantAccess/<br>{GlobalLogIn_ID}/{UserAccess_IsActive} |
| GET                       | /Administration/Application/{Application_ID}/DenyAccess/<br>{GlobalLogIn_ID}                        |
| GET POST                  | /Administration/Application/{id}/Edit   |
| GET POST                  | /Administration/Application/{id}/Model  |
| DELETE                    | /Administration/Application/{id}  |
| GET POST                  | /Administration/BatchProcess/New/{OwnerModule_ID}   |
| GET                       | /Administration/BatchProcess/{id}   |
| GET POST                  | /Administration/BatchProcess/{id}/Edit  |



| <b>Método(s)<br/>HTTP</b> | <b>Ruta<br/>(URL publicada parametrizada)</b>                           |
|---------------------------|---|
| DELETE                    | /Administration/BatchProcess/{id}                                       |
| GET POST                  | /Administration/DataType/New/{OwnerModule_ID}                           |
| GET                       | /Administration/DataType/{id}   |
| GET POST                  | /Administration/DataType/{id}/Edit                                      |
| DELETE                    | /Administration/DataType/{id}   |
| GET POST                  | /Administration/DataTypePolicy/New/{OwnerDataType_ID}                   |
| GET                       | /Administration/DataTypePolicy/{id}                                     |
| GET POST                  | /Administration/DataTypePolicy/{id}/Edit                                |
| DELETE                    | /Administration/DataTypePolicy/{id}                                     |
| GET POST                  | /Administration/EntityAttribute/New/{OwnerEntity_ID}                    |
| GET                       | /Administration/EntityAttribute/{id}                                    |
| GET POST                  | /Administration/EntityAttribute/{id}/Edit                               |
| DELETE                    | /Administration/EntityAttribute/{id}                                    |
| GET POST                  | /Administration/EntityAttributePolicy/New/<br>{OwnerEntityAttribute_ID} |
| GET                       | /Administration/EntityAttributePolicy/{id}                              |
| GET POST                  | /Administration/EntityAttributePolicy/{id}/Edit                         |
| DELETE                    | /Administration/EntityAttributePolicy/{id}                              |
| GET POST                  | /Administration/Entity/new/{OwnerModule_ID}                             |
| GET                       | /Administration/Entity/{id}   |
| GET POST                  | /Administration/Entity/{id}/edit  |
| DELETE                    | /Administration/Entity/{id}   |
| GET POST                  | /Administration/EntityPolicy/New/{OwnerEntity_ID}                       |
| GET                       | /Administration/EntityPolicy/{id}                                       |
| GET POST                  | /Administration/EntityPolicy/{id}/Edit                                  |
| DELETE                    | /Administration/EntityPolicy/{id}                                       |
| GET                       | /Administration/GlobalLogIn/  |
| GET POST                  | /Administration/GlobalLogIn/new   |
| GET                       | /Administration/GlobalLogIn/{id}  |
| GET POST                  | /Administration/GlobalLogIn/{id}/Edit                                   |
| DELETE                    | /Administration/GlobalLogIn/{id}  |
| GET                       | /Administration/Language/   |
| GET POST                  | /Administration/Language/New  |
| GET                       | /Administration/Language/{iso6391}                                      |
| GET POST                  | /Administration/Language/{iso6391}/Edit                                 |
| DELETE                    | /Administration/Language/{iso6391}                                      |
| ANY                       | /LogIn  |
| GET POST                  | /Administration/Module/New/{Application_ID}                             |
| GET                       | /Administration/Module/{id}   |



| <b>Método(s)<br/>HTTP</b> | <b>Ruta<br/>(URL publicada parametrizada)</b>                                     |
|---------------------------|---|
| GET POST                  | /Administration/Module/{id}/edit  |
| DELETE                    | /Administration/Module/{id}   |
| GET POST                  | /Administration/Relationship/New/{Application_ID}                                 |
| GET                       | /Administration/Relationship/{id}   |
| GET POST                  | /Administration/Relationship/{id}/Edit  |
| DELETE                    | /Administration/Relationship/{id}   |
| GET POST                  | /Administration/Service/new/{OwnerModule_ID}                                      |
| GET                       | /Administration/Service/{id}  |
| GET POST                  | /Administration/Service/{id}/edit   |
| DELETE                    | /Administration/Service/{id}  |
| GET                       | /Administration/System/   |
| GET POST                  | /Administration/System/New  |
| GET                       | /Administration/System/{id}   |
| GET POST                  | /Administration/System/{id}/Edit  |
| DELETE                    | /Administration/System/{id}   |
| GET POST                  | /Administration/ValueObjectAttribute/New/{OwnerValueObject_ID}                    |
| GET                       | /Administration/ValueObjectAttribute/{id}   |
| GET POST                  | /Administration/ValueObjectAttribute/{id}/Edit                                    |
| DELETE                    | /Administration/ValueObjectAttribute/{id}   |
| GET POST                  | /Administration/ValueObjectAttributePolicy/<br>New/{OwnerValueObjectAttribute_ID} |
| GET                       | /Administration/ValueObjectAttributePolicy/{id}                                   |
| GET POST                  | /Administration/ValueObjectAttributePolicy/{id}/Edit                              |
| DELETE                    | /Administration/ValueObjectAttributePolicy/{id}                                   |
| GET POST                  | /Administration/ValueObject/new/{OwnerModule_ID}                                  |
| GET                       | /Administration/ValueObject/{id}  |
| GET POST                  | /Administration/ValueObject/{id}/edit   |
| DELETE                    | /Administration/ValueObject/{id}  |
| GET POST                  | /Administration/ValueObjectPolicy/New/{OwnerValueObject_ID<br>}                   |
| GET                       | /Administration/ValueObjectPolicy/{id}  |
| GET POST                  | /Administration/ValueObjectPolicy/{id}/Edit                                       |
| DELETE                    | /Administration/ValueObjectPolicy/{id}  |
| ANY                       | /Symfony  |
| ANY                       | /GUI/DashBoard  |
| ANY                       | /   |
| ANY                       | /about  |
| ANY                       | /acerca   |
| ANY                       | /acercade   |

| <b>Método(s)<br/>HTTP</b> | <b>Ruta<br/>(URL publicada parametrizada)</b> |
|---------------------------|---|
| ANY                       | /acerca-de                                    |
| GET                       | /API/Administration/GlobalLogIn/              |
| GET                       | /API/Administration/GlobalLogIn/{id}          |
| GET POST                  | /API/Prosumer/{Entity}/New                    |
| GET                       | /API/Prosumer/{Entity}/{id}                   |
| GET POST                  | /API/Prosumer/{Entity}/{id}/Edit              |
| DELETE                    | /API/Prosumer/{Entity}/{id}                   |

Tabla 9.1.- Rutas de acceso a cada una de las acciones de los diferentes controladores.

#### 9.4.2.- Catálogo de procesos.

Presentamos a continuación una tabla donde puede comprobarse la relación entre los casos de uso, sus escenarios y el proceso correspondiente que será descrito a continuación a través de los denominados “diagramas de secuencia” o “diagramas de interacción”.

Para la identificación de los procesos, se ha utilizado la identificación del caso de uso del cual provienen; entiéndase que para cada escenario de cada caso de uso habrá un proceso equivalente, es decir el caso de uso [CU-1] tendrá un proceso [P-1-T] para su escenario típico de eventos y un proceso [P-1-A1] para el primer escenario alternativo.

La tabla mostrada a continuación muestra la relación de procesos; para facilitar la claridad de la misma, se han explicitado únicamente aquellos procesos que suponen escenarios alternativos a los casos de uso.

| <b>Caso de uso</b>    | <b>Escenario</b>        | <b>Proceso</b>                     |
|-----------------------|-------------------------|------------------------------------|
| Iniciar sesión [CU-1] | Curso típico de eventos | Autenticar credenciales<br>[P-1-T] |
|                       | Escenario alternativo 1 | Recordar contraseña<br>[P-1-A1]    |



| <b>Caso de uso</b>                                     | <b>Escenario</b>        | <b>Proceso</b>                             |
|--|-------------------------|--|
|  | Escenario alternativo 2 | Rechazar credenciales incorrectas [P-1-A2] |
| Gestionar perfil como usuario [CU-2]                   | Curso típico de eventos |  |
| Modificar datos del perfil [CU-2.1]                    | Curso típico de eventos |  |
| Modificar contraseña de acceso [CU-2.2]                | Curso típico de eventos |  |
| Cerrar sesión [CU-2.3]                                 | Curso típico de eventos |  |
| Gestionar usuarios [CU-3]                              | Curso típico de eventos |  |
| Mostrar usuario existente [CU-3.1]                     | Curso típico de eventos |  |
| Crear nuevo usuario [CU-3.2]                           | Curso típico de eventos |  |
| Modificar usuario existente [CU-3.3]                   | Curso típico de eventos |  |
| Restablecer contraseña de usuario existente [CU-3.3.1] | Curso típico de eventos |  |
| Eliminar usuario existente [CU-3.4]                    | Curso típico de eventos |  |
| Gestionar aplicaciones [CU-4]                          | Curso típico de eventos |  |
| Crear nueva aplicación [CU-4.1]                        | Curso típico de eventos |  |
| Consultar aplicación existente [CU-4.2]                | Curso típico de eventos |  |
| Gestionar accesos de usuario [CU-4.2.1]                | Curso típico de eventos |  |

| <b>Caso de uso</b>                             | <b>Escenario</b>        | <b>Proceso</b>                              |
|--|-------------------------|---|
| Conceder acceso [CU-4.2.1.1]                   | Curso típico de eventos | Conceder acceso habilitado [P-4.2.1.1-T]    |
|  | Escenario alternativo 1 | Conceder acceso deshabilitado [P-4.2.1.1-A] |
| Bloquear acceso [CU-4.2.1.2]                   | Curso típico de eventos |   |
| Eliminar acceso [CU-4.2.1.3]                   | Curso típico de eventos |   |
| Gestionar despliegues [CU-4.2.2]               | Curso típico de eventos |   |
| Instalar aplicación en sistema [CU-4.2.2.1]    | Curso típico de eventos |   |
| Desinstalar aplicación de sistema [CU-4.2.2.2] | Curso típico de eventos |   |
| Modificar aplicación existente [CU-4.3]        | Curso típico de eventos |   |
| Modelar aplicación existente [CU-4.4]          | Curso típico de eventos |   |
| Gestionar módulos [CU-4.4.1]                   | Curso típico de eventos |   |
| Consultar módulo existente [CU-4.4.1.1]        | Curso típico de eventos |   |
| Crear nuevo módulo [CU-4.4.1.2]                | Curso típico de eventos |   |
| Modificar módulo existente [CU-4.4.1.3]        | Curso típico de eventos |   |
| Eliminar módulo existente [CU-4.4.1.4]         | Curso típico de eventos |   |

| <b>Caso de uso</b>   | <b>Escenario</b>        | <b>Proceso</b> |
|--|-------------------------|----------------|
| Gestionar tipos de datos [CU-4.4.2]                          | Curso típico de eventos |                |
| Consultar tipo de datos existente [CU-4.4.2.1]               | Curso típico de eventos |                |
| Crear nuevo tipo de datos [CU-4.4.2.2]                       | Curso típico de eventos |                |
| Modificar tipo de datos existente [CU-4.4.2.3]               | Curso típico de eventos |                |
| Eliminar tipo de datos existente [CU-4.4.2.4]                | Curso típico de eventos |                |
| Gestionar políticas de tipo de datos [CU-4.4.3]              | Curso típico de eventos |                |
| Consultar política existente de tipo de datos [CU-4.4.3.1]   | Curso típico de eventos |                |
| Crear nueva política existente de tipo de datos [CU-4.4.3.2] | Curso típico de eventos |                |
| Modificar política existente de tipo de datos [CU-4.4.3.3]   | Curso típico de eventos |                |
| Eliminar política existente de tipo de datos [CU-4.4.3.4]    | Curso típico de eventos |                |
| Gestionar entidades [CU-4.4.4]                               | Curso típico de eventos |                |
| Consultar entidad existente [CU-4.4.4.1]                     | Curso típico de eventos |                |

| <b>Caso de uso</b>                                   | <b>Escenario</b>        | <b>Proceso</b> |
|--|-------------------------|----------------|
| Crear nueva entidad [CU-4.4.4.2]                     | Curso típico de eventos |                |
| Modificar entidad existente [CU-4.4.4.3]             | Curso típico de eventos |                |
| Eliminar entidad existente [CU-4.4.4.4]              | Curso típico de eventos |                |
| Gestionar políticas de entidad [CU-4.4.5]            | Curso típico de eventos |                |
| Consultar política existente de entidad [CU-4.4.5.1] | Curso típico de eventos |                |
| Crear nueva política de entidad [CU-4.4.5.2]         | Curso típico de eventos |                |
| Modificar política existente de entidad [CU-4.4.5.3] | Curso típico de eventos |                |
| Eliminar política existente de entidad [CU-4.4.5.4]  | Curso típico de eventos |                |
| Gestionar atributos de entidad [CU-4.4.6]            | Curso típico de eventos |                |
| Consultar atributo existente de entidad [CU-4.4.6.1] | Curso típico de eventos |                |
| Crear nuevo atributo de entidad [CU-4.4.6.2]         | Curso típico de eventos |                |
| Modificar atributo existente de entidad [CU-4.4.6.3] | Curso típico de eventos |                |



| <b>Caso de uso</b>   | <b>Escenario</b>        | <b>Proceso</b> |
|--|-------------------------|----------------|
| Eliminar atributo existente de entidad [CU-4.4.6.4]              | Curso típico de eventos |                |
| Gestionar políticas de atributo de entidad [CU-4.4.7]            | Curso típico de eventos |                |
| Consultar política existente de atributo de entidad [CU-4.4.7.1] | Curso típico de eventos |                |
| Crear nueva política de atributo de entidad [CU-4.4.7.2]         | Curso típico de eventos |                |
| Modificar política existente de atributo de entidad [CU-4.4.7.3] | Curso típico de eventos |                |
| Eliminar política existente de atributo de entidad [CU-4.4.7.4]  | Curso típico de eventos |                |
| Gestionar relaciones [CU-4.4.8]                                  | Curso típico de eventos |                |
| Consultar relación existente [CU-4.4.8.1]                        | Curso típico de eventos |                |
| Crear nueva relación [CU-4.4.8.2]                                | Curso típico de eventos |                |
| Modificar relación existente [CU-4.4.8.3]                        | Curso típico de eventos |                |
| Eliminar relación existente [CU-4.4.8.4]                         | Curso típico de eventos |                |
| Gestionar objetos valor [CU-4.4.9]                               | Curso típico de eventos |                |



| <b>Caso de uso</b>   | <b>Escenario</b>        | <b>Proceso</b> |
|--|-------------------------|----------------|
| Consultar objeto valor existente [CU-4.4.9.1]              | Curso típico de eventos |                |
| Crear nuevo objeto valor [CU-4.4.9.2]                      | Curso típico de eventos |                |
| Modificar objeto valor existente [CU-4.4.9.3]              | Curso típico de eventos |                |
| Eliminar objeto valor existente [CU-4.4.9.4]               | Curso típico de eventos |                |
| Gestionar políticas de objeto valor [CU-4.4.10]            | Curso típico de eventos |                |
| Consultar política existente de objeto valor [CU-4.4.10.1] | Curso típico de eventos |                |
| Crear nueva política de objeto valor [CU-4.4.10.2]         | Curso típico de eventos |                |
| Modificar política existente de objeto valor [CU-4.4.10.3] | Curso típico de eventos |                |
| Eliminar política existente de objeto valor [CU-4.4.10.4]  | Curso típico de eventos |                |
| Gestionar atributos de objeto valor [CU-4.4.11]            | Curso típico de eventos |                |
| Consultar atributo existente de objeto valor [CU-4.4.11.1] | Curso típico de eventos |                |
| Crear nuevo atributo de objeto valor [CU-4.4.11.2]         | Curso típico de eventos |                |



| <b>Caso de uso</b>   | <b>Escenario</b>        | <b>Proceso</b> |
|--|-------------------------|----------------|
| Modificar atributo existente de objeto valor [CU-4.4.11.3]             | Curso típico de eventos |                |
| Eliminar atributo existente de objeto valor [CU-4.4.11.4]              | Curso típico de eventos |                |
| Gestionar políticas de atributo de objeto valor [CU-4.4.12]            | Curso típico de eventos |                |
| Consultar política existente de atributo de objeto valor [CU-4.4.12.1] | Curso típico de eventos |                |
| Crear nueva política de atributo de objeto valor [CU-4.4.12.2]         | Curso típico de eventos |                |
| Modificar política existente de atributo de objeto valor [CU-4.4.12.3] | Curso típico de eventos |                |
| Eliminar política existente de atributo de objeto valor [CU-4.4.12.4]  | Curso típico de eventos |                |
| Gestionar servicios [CU-4.4.13]  | Curso típico de eventos |                |
| Consultar servicio existente [CU-4.4.13.1]                             | Curso típico de eventos |                |
| Crear nuevo servicio [CU-4.4.13.2]                                     | Curso típico de eventos |                |
| Modificar servicio existente [CU-4.4.13.3]                             | Curso típico de eventos |                |

| <b>Caso de uso</b>                                 | <b>Escenario</b>        | <b>Proceso</b> |
|--|-------------------------|----------------|
| Eliminar servicio existente [CU-4.4.13.4]          | Curso típico de eventos |                |
| Gestionar procesos por lotes [CU-4.4.14]           | Curso típico de eventos |                |
| Consultar proceso por lote existente [CU-4.4.14.1] | Curso típico de eventos |                |
| Crear nuevo proceso por lote [CU-4.4.14.2]         | Curso típico de eventos |                |
| Modificar proceso por lote existente [CU-4.4.14.3] | Curso típico de eventos |                |
| Eliminar proceso por lote existente [CU-4.4.14.4]  | Curso típico de eventos |                |
| Eliminar aplicación existente [CU-4.5]             | Curso típico de eventos |                |
| Gestionar idiomas y traducciones [CU-5]            | Curso típico de eventos |                |
| Crear nuevo idioma [CU-5.1]                        | Curso típico de eventos |                |
| Consultar idioma existente [CU-5.2]                | Curso típico de eventos |                |
| Consultar traducciones [CU-5.3]                    | Curso típico de eventos |                |
| Crear nueva traducción [CU-5.3.1]                  | Curso típico de eventos |                |
| Modificar traducción [CU-5.3.2]                    | Curso típico de eventos |                |
| Eliminar traducción [CU-5.3.3]                     | Curso típico de eventos |                |

| <b>Caso de uso</b>                               | <b>Escenario</b>        | <b>Proceso</b> |
|--|-------------------------|----------------|
| Modificar idioma existente [CU-5.4]              | Curso típico de eventos |                |
| Eliminar idioma existente [CU-5.5]               | Curso típico de eventos |                |
| Gestión de sistemas y despliegues [CU-5]         | Curso típico de eventos |                |
| Creación de un nuevo sistema [CU-6.1]            | Curso típico de eventos |                |
| Consulta de un sistema existente [CU-6.2]        | Curso típico de eventos |                |
| Modificación de un sistema existente [CU-6.3]    | Curso típico de eventos |                |
| Eliminación de un sistema existente [CU-6.4]     | Curso típico de eventos |                |
| Gestionar información [CU-7]                     | Curso típico de eventos |                |
| Listar registro existente de entidad [CU-7.1]    | Curso típico de eventos |                |
| Consultar registro existente de entidad [CU-7.2] | Curso típico de eventos |                |
| Crear nuevo registro de entidad [CU-7.3]         | Curso típico de eventos |                |
| Modificar registro existente de entidad [CU-7.4] | Curso típico de eventos |                |
| Eliminar registro existente de entidad [CU-7.5]  | Curso típico de eventos |                |

Tabla 9.2.- Relación entre escenarios de casos de uso y procesos.

Una vez enumerados todos los procesos de los que consta el sistema, pasaremos a modelar el comportamiento de los mismos (modelo dinámico) a través de los correspondientes “diagramas de secuencia”, donde se representan todos los elementos que intervienen en un proceso, así como las sucesivas interacciones que suceden entre los mismos.

Es importante resaltar la enorme similitud entre la mayoría de los diagramas de secuencia que representan a cada uno de los procesos. Esto se debe a que gracias al *framework* de desarrollo Symfony, es posible obtener un alto grado de coherencia a la hora de desarrollar una aplicación de gestión de datos, pues los programadores son capaces de eliminar de su propio código aquellas tareas repetitivas similares cuya única variación sean los datos de las entidades que participan en ellos así como las operaciones con la interfaz de usuario (como, por ejemplo, el enlace de datos entre los controles de formulario y sus respectivos valores).

Por tanto, en la práctica, no es interesante obtener todo este grado de detalle, pues Symfony libera al programador de las tareas tediosas y repetitivas gracias a simples mecanismos de configuración establecidos tanto en las clases como en sus métodos PHP que actúan como controladores y acciones, respectivamente.

Dichos mecanismos de configuración se basan en “anotaciones” dentro de los comentarios PHPDOC [191] asociados a cada clase y método, estableciendo así el comportamiento en tiempo de ejecución del *framework* Symfony para con dichos elementos.

Con todos estas consideraciones, un diagrama de secuencia altamente elaborado (que incluyese tanto las operaciones de encaminamiento basado en URL, resolución e inyección de dependencias, materialización de entidades de dominio, así como el enlace de datos y posterior persistencia de los mismos) podría presentar la forma de la fig. 9.13, teniendo en cuenta que participarían tantas entidades como fuese necesario (el diagrama presenta el ejemplo de una única entidad) y que dichas entidades dispondrían generalmente de un número representativo de atributos (de ahí que hayamos mostrado un proceso iterativo en las operaciones de lectura y escritura de dichas propiedades, conocidas como “getters” o “accessors” y “setters” o “mutators”, respectivamente):

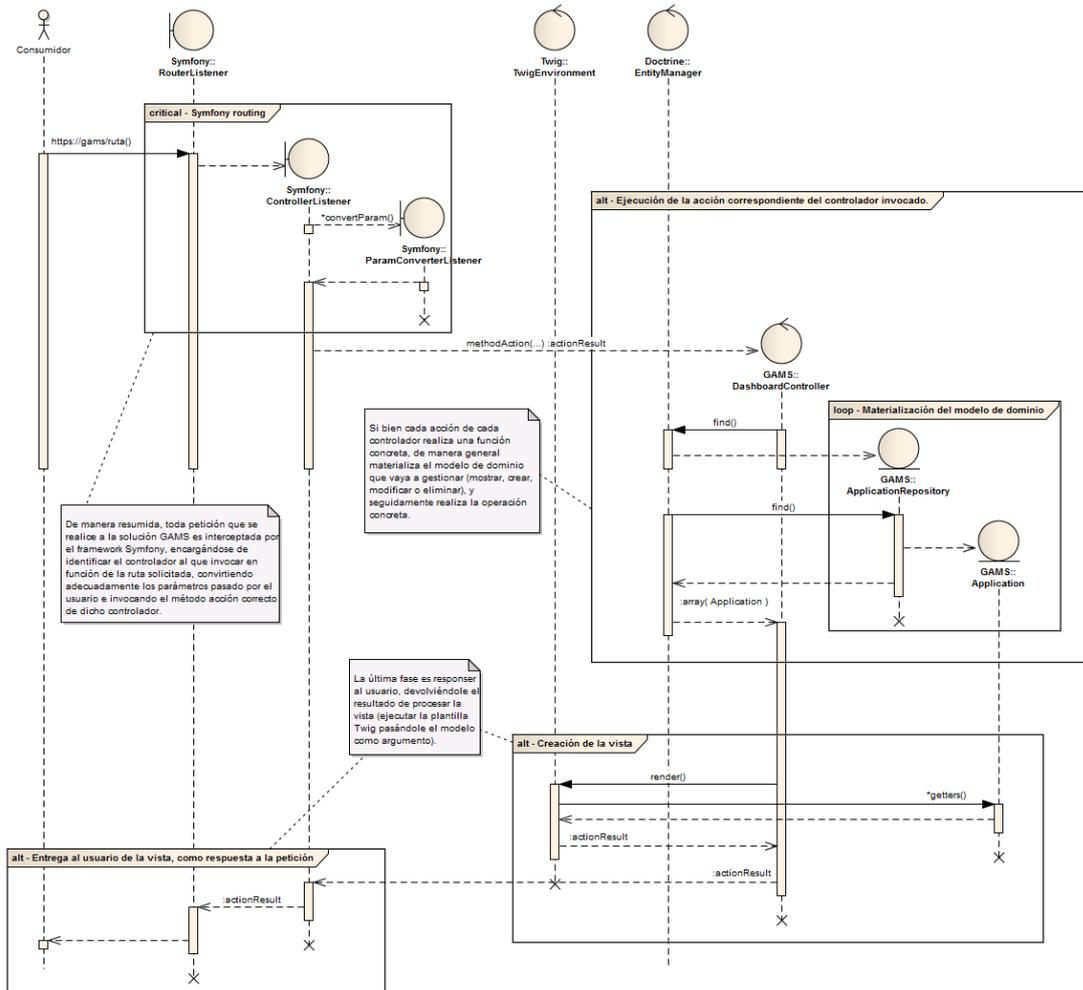


Fig. 9.13.- Diagrama de secuencia general para las operaciones con la interfaz de usuario.

## 9.5.- Diseño de la interfaz de usuario.

El diseño de la interfaz de usuario determinará cómo se expone la funcionalidad de la aplicación a los diferentes usuarios del sistema. Este diseño hace uso del diagrama de navegación, así como de los diferentes procesos descritos en el modelo anterior, con el objetivo de describir cómo se presenta el sistema ante sus diferentes usuarios.

A este modelo de procesos se le denomina también “vista dinámica del sistema”, pues determina el comportamiento del mismo durante su ejecución. El modelo de procesos se ha dividido en tres apartados:

- Usabilidad, donde se presenta cómo el usuario puede navegar a través de la interfaz de usuario basada en Web.
- Diseño web adaptativo, donde se determina cómo se presentará la interfaz web en función de las capacidades del dispositivo utilizado por el usuario.
- API de servicios, donde se presenta cómo interactúa el sistema con terceros sistemas.

### 9.5.1.- Usabilidad.

Los principales criterios de usabilidad empleados durante el diseño de la interfaz gráfica de usuario basada en web han sido:

- Uso de elementos de bloque, en lugar de tablas.
- Uso de iconografía textual, basada en Font Awesome, versión 4.7 (véase [225]).
- Uso de diseño web adaptativo (en el siguiente apartado se explica esta técnica con más detalle).



### **9.5.2.- Diseño web adaptativo (responsive web design).**

Toda la interfaz gráfica de usuario hará uso de diseño web adaptativo (*responsive web design*) [136]. Gracias a este tipo de diseño, las diferentes pantallas se visualizarán adaptadas en función de la resolución del dispositivo utilizado en cada ocasión.

El diseño adaptativo se encarga de mostrar los elementos de manera que siempre se priorice la comodidad de acceso frente a una posición rígida de los mismos en la pantalla, aprovechando al máximo el espacio disponible en la pantalla.

En el apartado 10.3 del Manual de Usuario se muestran numerosas capturas de pantalla de la solución GAMS donde se pueden apreciar los resultados de aplicar esta técnica.

## 10.- ANEXO C: MANUAL DE USUARIO.

*No trabajes más duro. Hazlo con más  
inteligencia.*

Ken Blanchard (1939-)

Presentamos a continuación el Manual de Usuario de la solución de **Generación Automática de MicroServicios (GAMS)**. Este Manual pretende describir el funcionamiento completo de la solución desde un punto de vista principalmente funcional (qué hace) y no tanto desde un punto de vista técnico (cómo lo hace).

No debe entenderse como un documento aislado del resto de la presente Memoria, por lo que no cubre exhaustivamente todos los aspectos conceptuales tratados en el proyecto software resultado del Trabajo Fin de Máster, ampliando la información anterior con cuestiones propias del usuario, como son la instalación de la solución en la plataforma final escogida (utilizándose tres diferentes a modo de referencia), los elementos generales de la interfaz de usuario y las pantallas que componen la solución, explicando brevemente el funcionamiento de cada una de ellas.

Así pues, remitimos al lector a los capítulos anteriores de la Memoria, especialmente los Anexos A y B, de Análisis Funcional y de Diseño Técnico respectivamente, de cara a repasar los conceptos, entidades y características que forman parte de la funcionalidad ofrecida por la solución de **Generación Automática de MicroServicios**, desde un punto de vista de usuario final, ya sea éste administrador (eminentemente técnico) o consumidor (eminentemente funcional).

Nótese que si bien a lo largo del presente Trabajo Fin de Máster se han utilizado tres plataformas diferentes sobre las que ejecutar la solución (referidas como: PC, NAS e IoT), éstas deben entenderse con el único objetivo de buscar la diversidad a la hora de utilizar la solución, como un ejercicio que valide el requisito de servir en múltiples entornos de características muy diferentes, tanto a nivel hardware como de software.

El objetivo de este Anexo C es que cualquier persona con interés por crear su propio *MicroServicio* pueda hacerlo a través de la solución objeto del presente Trabajo Fin de Máster.

Los puntos que cubre este Manual de Usuario son los siguientes:

- Requisitos del sistema.
  - Requisitos generales del sistema.
  - Requisitos específicos para cada una de las diferentes plataformas de referencia: PC, NAS e IoT.
- Proceso de instalación.
  - Descripción del proceso de instalación y configuración inicial en cada una de las plataformas de referencia: PC, NAS e IoT.
- Pantallas principales del programa (con sus correspondientes pantallas secundarias relacionadas):
  - Inicio.
  - Acerca de.
  - Autenticación y restablecimiento de contraseña.
  - Panel de Control (principal).
  - Cierre de sesión.
  - Modificación del perfil de usuario.
  - Gestión de usuarios.
  - Gestión de idiomas.
  - Gestión de aplicaciones.
  - Modelado de aplicaciones.
  - Utilización de aplicaciones.

## 10.1.- Requisitos del sistema.

La solución de **Generación Automática de MicroServicios (GAMS)** se puede concebir como una aplicación web basada en la plataforma PHP versión 7, con dos interfaces de usuario: una interfaz gráfica basada en HTML5/CSS/JavaScript, y otra interfaz API basada en servicios web REST.

### 10.1.1.- Requisitos generales del sistema.

Por ello, los principales requisitos son los expresados por la propia plataforma PHP [226], destacando los siguientes:

- Un servidor web con capacidad para ejecutar PHP, ya sea como biblioteca dinámica (a través del módulo SAPI) o a través de una interfaz CGI (o FastCGI).
- El propio motor de PHP.
- Un navegador web, que sirva de cliente HTTP, con capacidad para visualizar HTML y CSS, así como ejecutar JavaScript.

A estos requisitos habrá que añadir la capacidad del servidor para acceder a bases de datos. La solución GAMS se ha desarrollado haciendo uso de la biblioteca Doctrine ORM, la cual a su vez hace uso del paquete “*PHP Data Objects*” (PDO) [227] propio de PHP. Gracias a PHP y al paquete PDO, es posible acceder a una gran multitud de sistemas gestores de bases de datos [228] de diferentes naturalezas, ya sean: jerárquicos, relacionales, documentales e incluso orientados a objetos, por mencionar algunos.

Indicar que se han utilizado las siguientes tres plataformas como referencia en las que se ha desarrollado y validado la solución GAMS:

- Plataforma PC, basada en Intel y Windows.
- Plataforma NAS, basada en Synology DSM 6.

- Plataforma IoT, basada en Raspberry Pi versión 1 revisión B.

## **10.2.- Proceso de instalación.**

Se describen a continuación los diferentes procesos de instalación de la solución de **Generación Automática de MicroServicios** (GAMS), específicos para cada una de las tres plataformas de referencia en las que se ha desarrollado y validado: PC, NAS e IoT.

Recordemos que, adicionalmente, el sistema cliente requiere únicamente un navegador Web para utilizar hacer uso de la solución, tanto de la aplicación de administración, como de todas y cada una de las aplicaciones generadas para consumidores.

En el caso de que el sistema cliente sea una máquina, será necesario un cliente de servicios web estilo REST.

En su defecto, pueden utilizarse herramientas y utilidades como por ejemplo cURL [229] o POSTMAN [230], evitando así la necesidad de programar específicamente un cliente web.

### **10.2.1.- Instalación de referencia en la plataforma PC, basada en Intel y Windows.**

El proceso de instalación para plataforma PC se basa en la consecución del siguiente conjunto de actividades:

- Instalación del sistema operativo.
- Instalación del servidor web.
- Instalación del sistema gestor de bases de datos.



- Descarga de las distribuciones de software a instalar.
- Creación de la estructura de directorios.
- Configuración del servidor web.
- Configuración inicial de la solución.

#### *10.2.1.1.- Instalación del sistema operativo.*

La plataforma PC de referencia está basada en la arquitectura Intel, ya sea para procesadores x86 (32 bits) o x64 (64 bits) sobre los que se ejecutará el sistema operativo Microsoft Windows.

El procedimiento de instalación variará dependiendo de la versión y edición concreta del sistema operativo Microsoft Windows utilizado para ejecutar la solución GAMS.

Remitimos al lector a [231] para instalar un sistema operativo Microsoft Windows Server 2012 R2, propuesto como sistema de referencia.

Nótese que, si bien la instalación de referencia para la plataforma PC ha considerado un entorno Microsoft Windows, existen multitud de variantes igualmente válidas a nivel de sistema operativo, siendo ejemplos notables las diferentes distribuciones abiertas de LINUX (o sus equivalentes empresariales), así como Mac OS X de Apple.

Dejamos como tarea al lector la instalación del sistema operativo que considere más acorde a sus preferencias o adecuado para sus intereses.

### 10.2.1.2.- Instalación del servidor web.

La plataforma PC de referencia utiliza el servidor web propio de Windows, denominado Microsoft Internet Information Server (IIS). Indicar que cada versión de Microsoft Windows dispone de una versión específica y diferente del servidor web IIS.

En el caso de Windows Server 2012 R2, la versión de IIS es la 8.5. En [232] pueden consultarse las instrucciones precisas, paso a paso, para instalar IIS 8.5 en Windows Server 2012 R2.

Nótese que, si bien la instalación de referencia para la plataforma PC ha considerado un entorno Microsoft Windows con IIS, existen multitud de opciones a nivel de servidor web, destacando Apache HTTP Server [19] y NGIX [233] como principales pero no únicas alternativas.

Dejamos igualmente al lector la tarea de instalación del servidor web que considere más acorde a sus preferencias o adecuado para sus intereses.

### 10.2.1.3.- Instalación del sistema gestor de bases de datos.

Nuevamente, la instalación de referencia para la plataforma PC hace uso de productos de software del fabricante Microsoft. En este caso, el sistema gestor de bases de datos relacionales SQL Server, en su versión 2017. Se ha escogido la edición Express [234] por ser suficiente para la solución GAMS en su desarrollo actual, pues exclusivamente consta del motor de bases de datos y las herramientas de administración, prescindiendo de los sistemas de integraciones (denominado *Integration Services*, SSIS), inteligencia empresarial (*Analysis Services*, SSAS) e informes (*Reporting Services*, SSRS).

Si el lector lo considera interesante, es posible instalar la edición Developer, la cual incluye sin coste todas estas herramientas para entornos de desarrollo y pruebas (puede obtenerse más información en [235] y [236]).

Tal y como se ha mencionado en el apartado 10.1.1, las opciones en cuanto al sistema gestor de bases de datos a instalar son muy amplias, tanto por lo que ofrece de base el entorno PHP [237], como por la abstracción que permiten las bibliotecas Doctrine ORM [238] y Doctrine DBAL [239] a la solución GAMS, limitándose las necesidades del administrador de la solución a configurar el acceso a datos específico para cada caso.

El primer paso consiste en descargar el programa cargador web (*web bootstrapper*) de la versión Express (véase fig. 10.1), que al ser gratuita está disponible en la web de fabricante [240]. Dicho programa en realidad nos permite seleccionar la modalidad real de instalación (véase fig. 10.2).

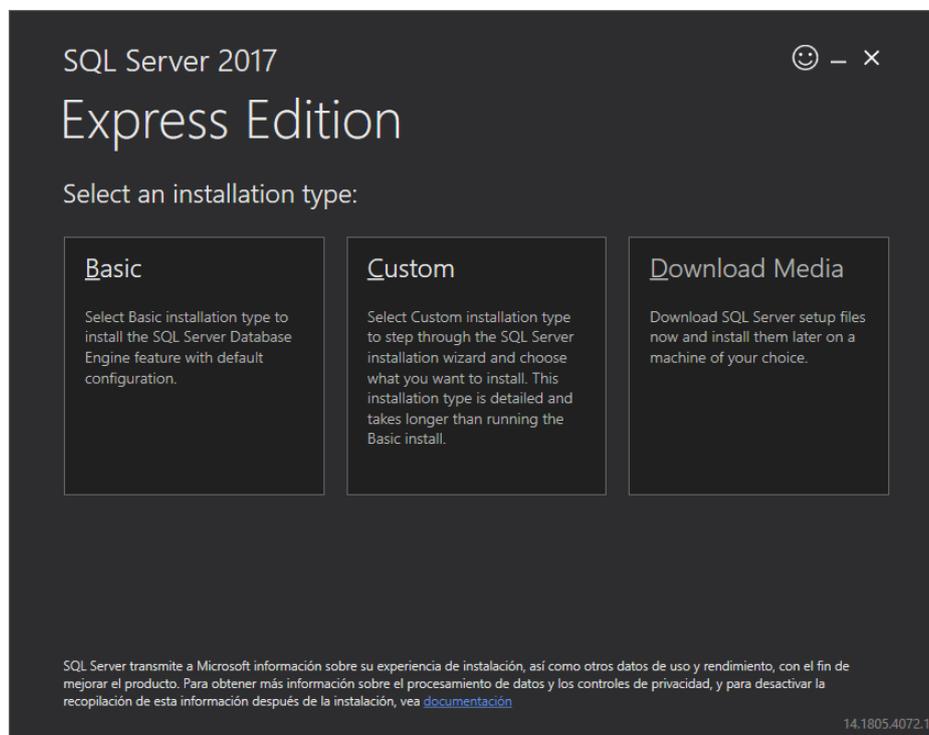


Fig. 10.1.- Programa cargador web de Microsoft SQL Server 2017 edición Express.

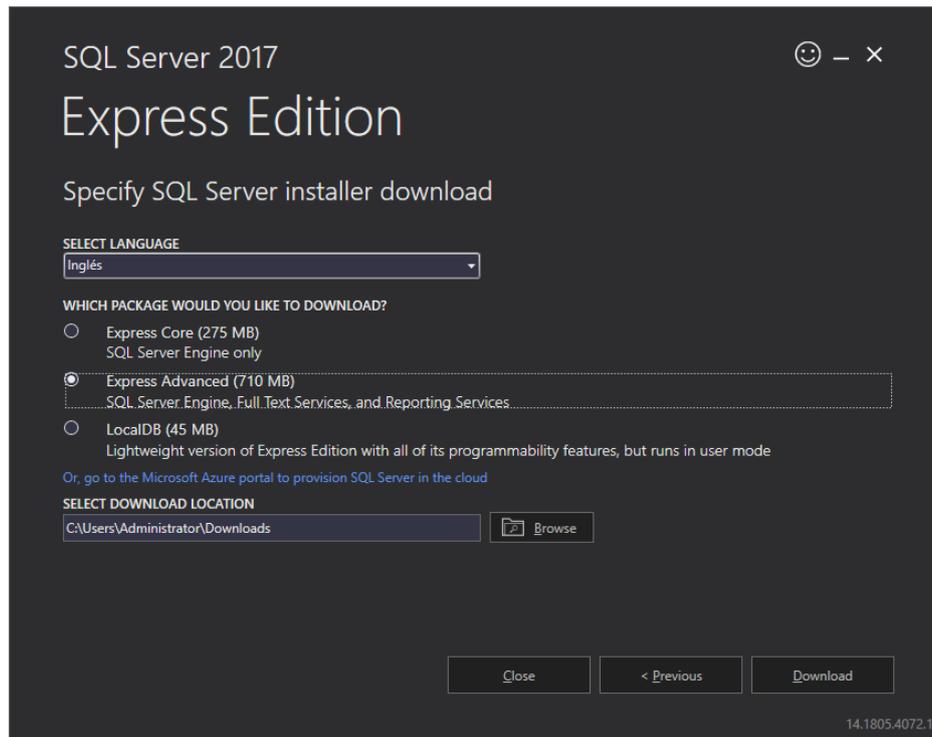


Fig. 10.2.- Selección de la opción a descargar e instalar de Microsoft SQL Server 2017 edición Express.

El siguiente paso de la instalación será ejecutar el auténtico instalador (véase fig. 10.3) y seleccionar la opción de realizar una nueva instalación de un servidor aislado.

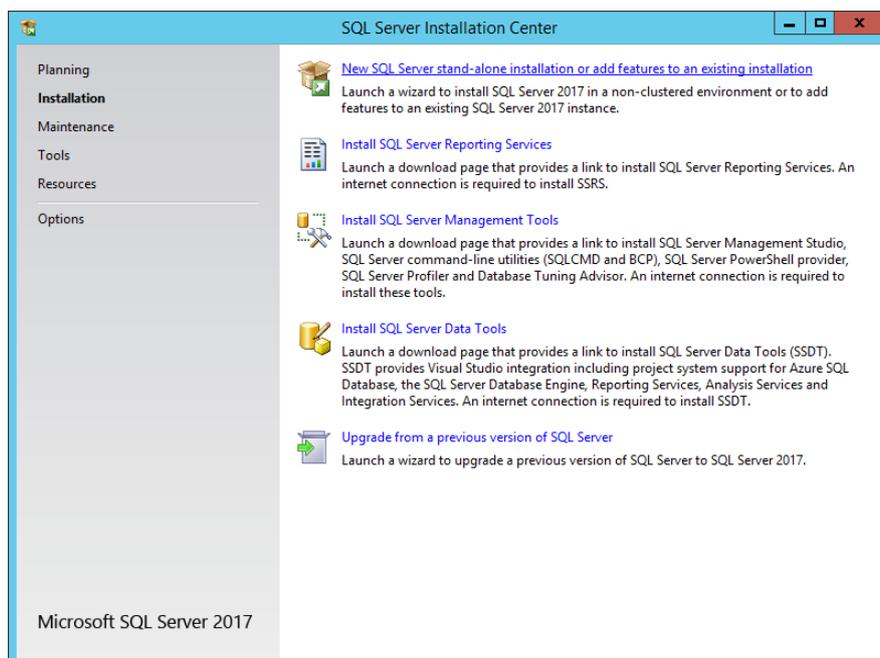


Fig. 10.3.- Instalador de Microsoft SQL Server 2017 edición Express.

En [241] puede consultarse la guía de instalación completa, paso a paso, de Microsoft SQL Server 2017, realizada a través del “Asistente para la instalación” (nombre que recibe el programa de instalación).

Debido a que la versión Express solamente contiene el motor de base de datos y las herramientas, explicaremos a continuación aquellos parámetros de la instalación que sean interesantes para el despliegue inicial de la solución GAMS.

La fig. 10.4 muestra las características (módulos funcionales) a instalar, destacando “*Database Engine Services*”, que representa al motor de bases de datos, y las “*Shared Features*” que representan las herramientas de administración y consulta.

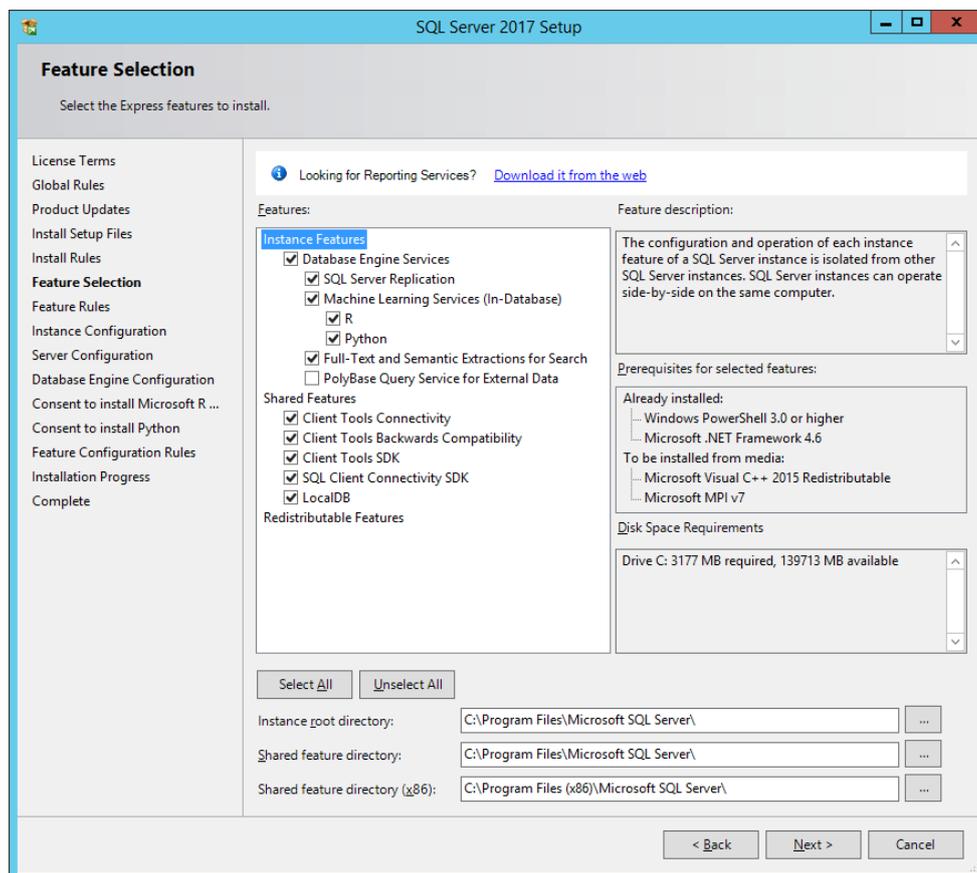


Fig. 10.4.- Características a instalar de Microsoft SQL Server 2017 edición Express.

La fig. 10.5 muestra el nombre de instancia que se otorgará a la instalación de SQL Server. La importancia de este nombre radica en que será necesario posteriormente para configurar adecuadamente la “cadena de conexión” con la base de datos, es decir, identificar al servidor de bases de datos en la configuración de la solución GAMS.

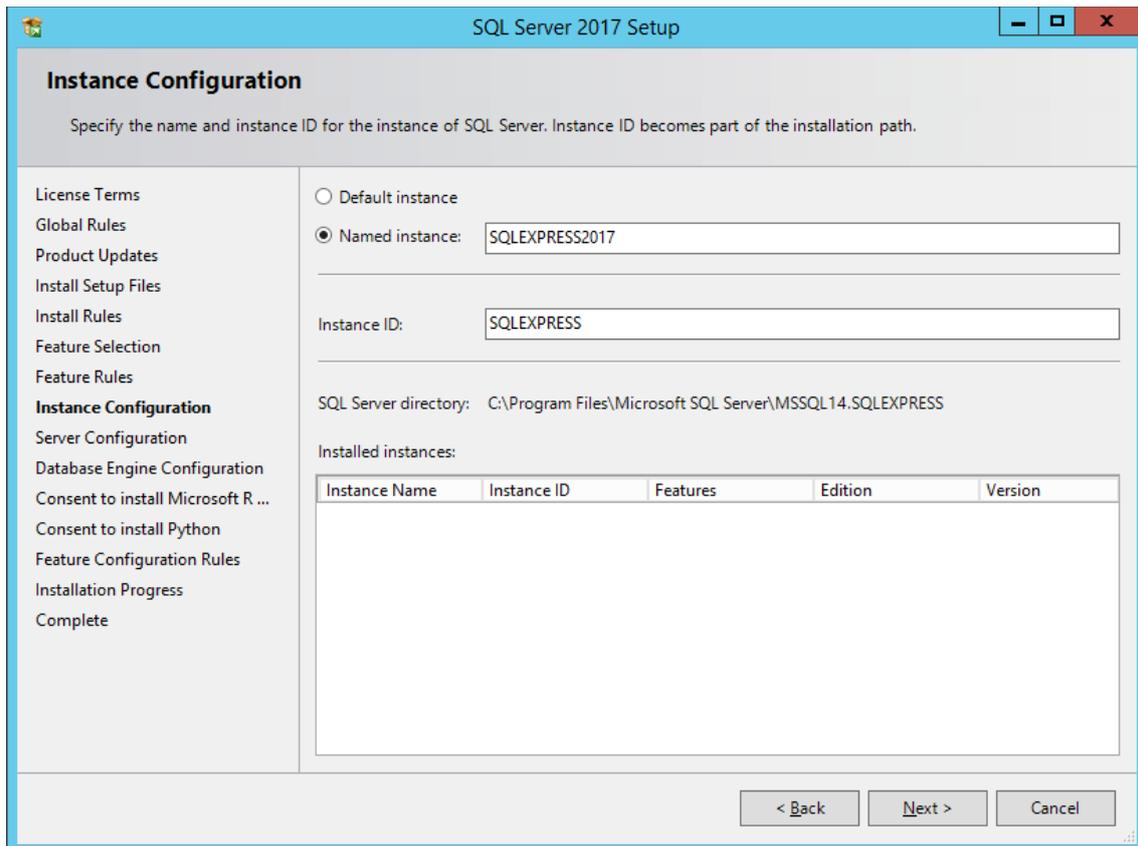


Fig. 10.5.- Nombre de la instalación del servidor Microsoft SQL Server 2017 Express.

La fig. 10.6 muestra la configuración de autenticación en modo mixto, es decir, el motor de bases de datos admitirá tanto usuario de Windows como credenciales propias de SQL Server. Gracias al modo mixto será posible crear usuarios de base de datos que no existan como usuario del sistema operativo.

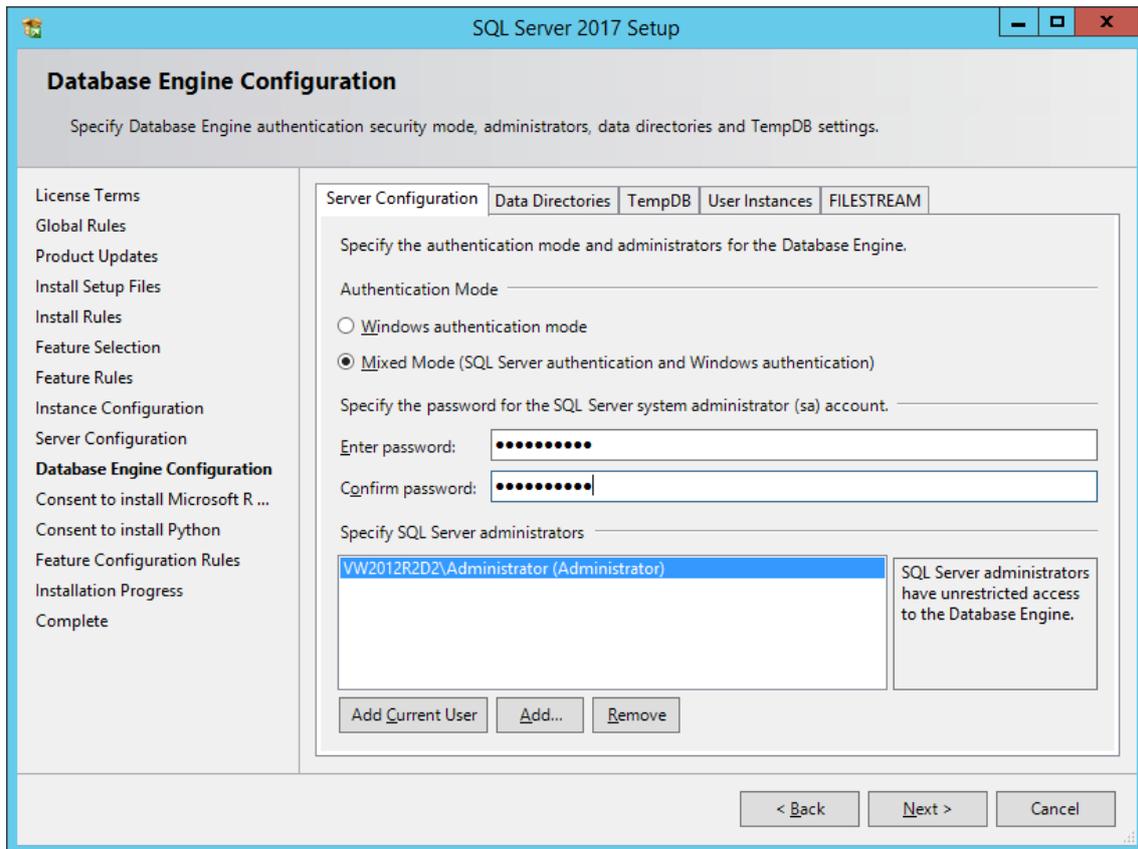


Fig. 10.6.- Modo mixto de autenticación para Microsoft SQL Server 2017 edición Express.

Una vez completada la instalación, se nos mostrará una pantalla similar a la fig. 10.7 donde se indicará el resultado exitoso (o no) del proceso.

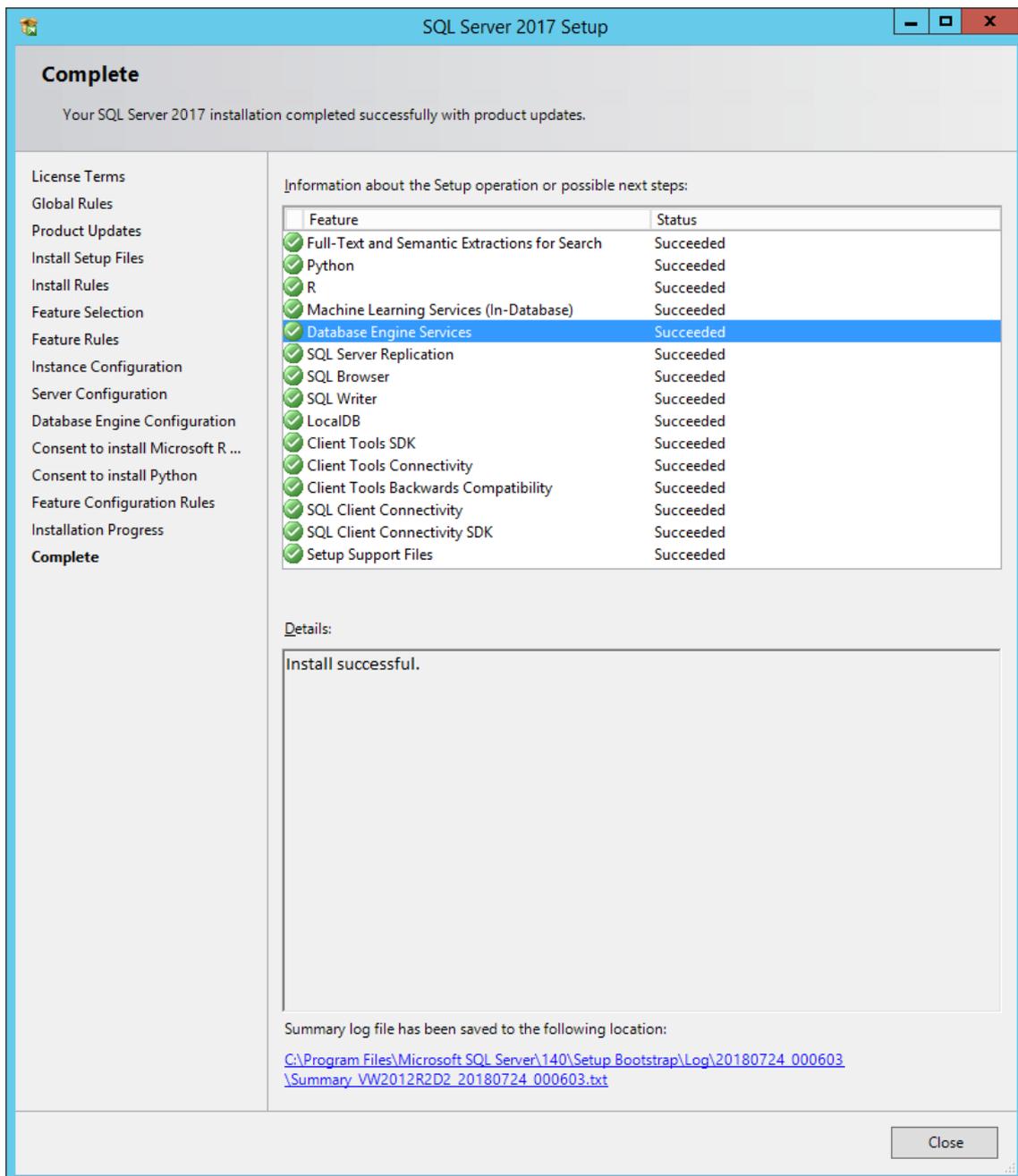


Fig. 10.7.- Resultado satisfactorio de la instalación de Microsoft SQL Server 2017 edición Express.

El siguiente paso, recomendable en cualquier caso, consistirá en instalar las herramientas de administración del sistema de base de datos, denominadas “*SQL Server Management Tools*” (normalmente referidas como *Management Studio*, *SSMS*). Para ello, bastará con descargar el paquete de instalación desde [242] y ejecutarlo inmediatamente (véase fig. 10.8).

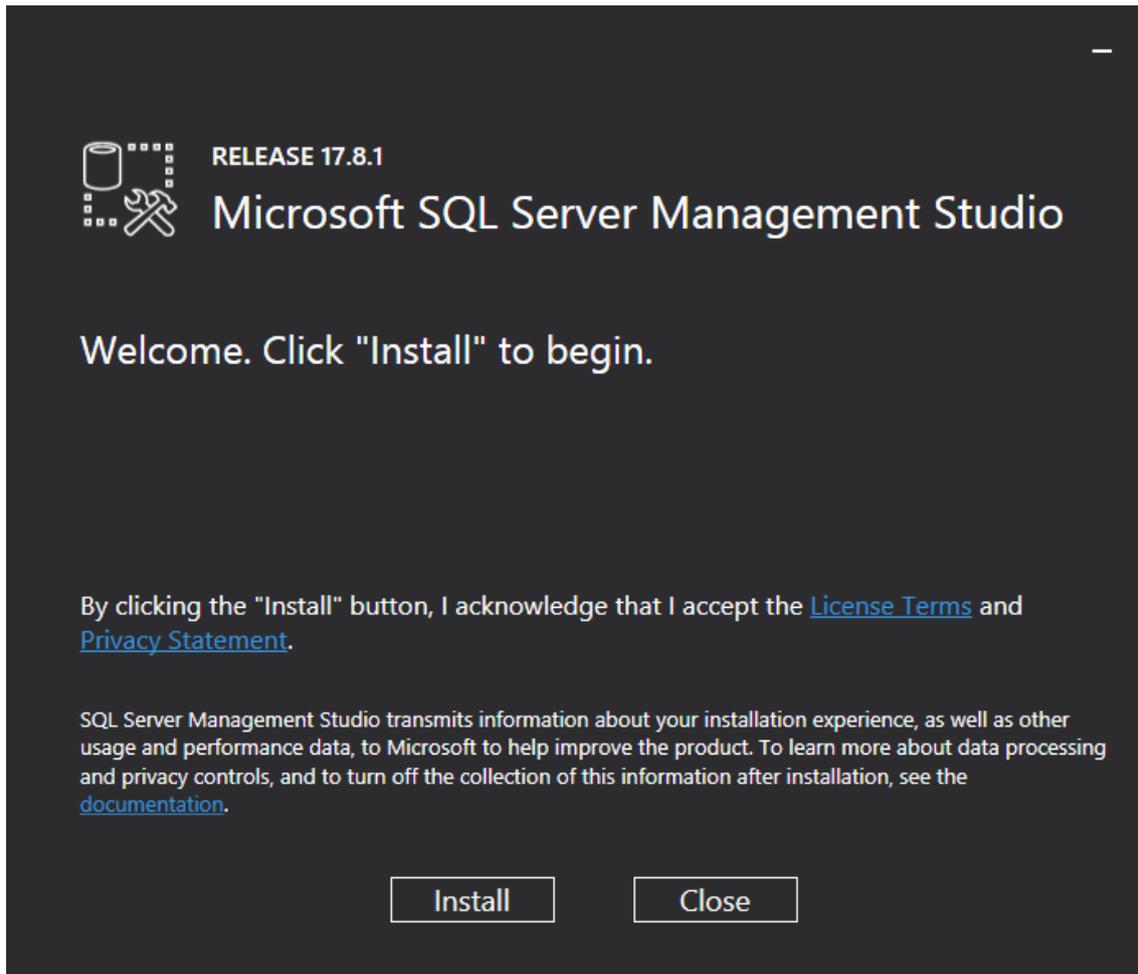


Fig. 10.8.- Instalación de Microsoft SQL Server 2017 Management Studio.

#### *10.2.1.4.- Instalación manual de los controladores de acceso a la base de datos.*

Una vez instalado el sistema gestor de bases de datos Microsoft SQL Server 2017, es necesario instalar una serie de controladores para acceder desde PHP a las bases de datos gestionadas por dicho servidor. Estas bibliotecas adicionales son las siguientes (en el apartado 6 se encuentran las respectivas referencias para descargar cada uno de los mencionados controladores):

- Controlador ODBC versión 17 para Microsoft SQL Server 2017 [243].
- Controlador Microsoft versión 5.3 para acceso desde PHP a SQL Server 2017 (puede descargarse desde [244], obtener el código fuente en [245], así como obtener más documentación en [246]).
- La documentación de referencia del controlador PDO\_SQLSRV puede consultarse en [247].
- En [248] se detallan pasos necesarios para configurar en el entorno PHP los controladores previamente descargados para SQL Server, en función de las características del entorno donde vayan a ser ejecutados.

Para verificar la correcta instalación de la versión 17 del controlador ODBC de SQL Server, basta con ejecutar el programa “Administrador de Orígenes de Datos ODBC” propio del sistema operativo Microsoft Windows (habitualmente este ejecutable se encuentra en el directorio “C:\Windows\System32\” con el nombre “ODBCAD32.EXE”, incluso en versiones de 64 bits como la referida en esta Memoria).

En la fig. 10.9 puede observarse la disponibilidad de dicho controlador ODBC al seleccionar “Crear un nuevo DSN de sistema”.

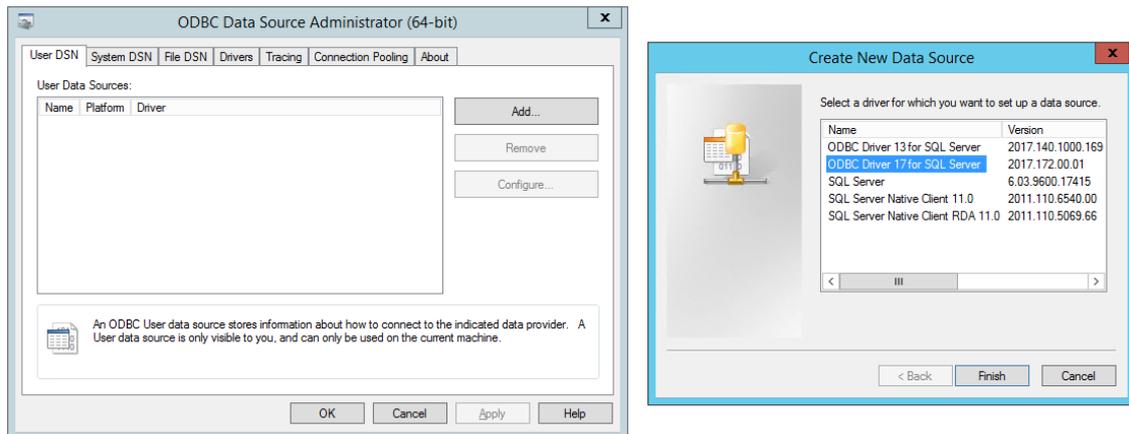


Fig. 10.9.- Creación de un DSN para verificar la instalación del controlador ODBC v17 para Microsoft SQL Server 2017.

Asimismo, en la fig. 10.10 puede observarse el estado final de la configuración del origen de datos (*data source name*, DSN) de sistema que hace uso de este controlador.

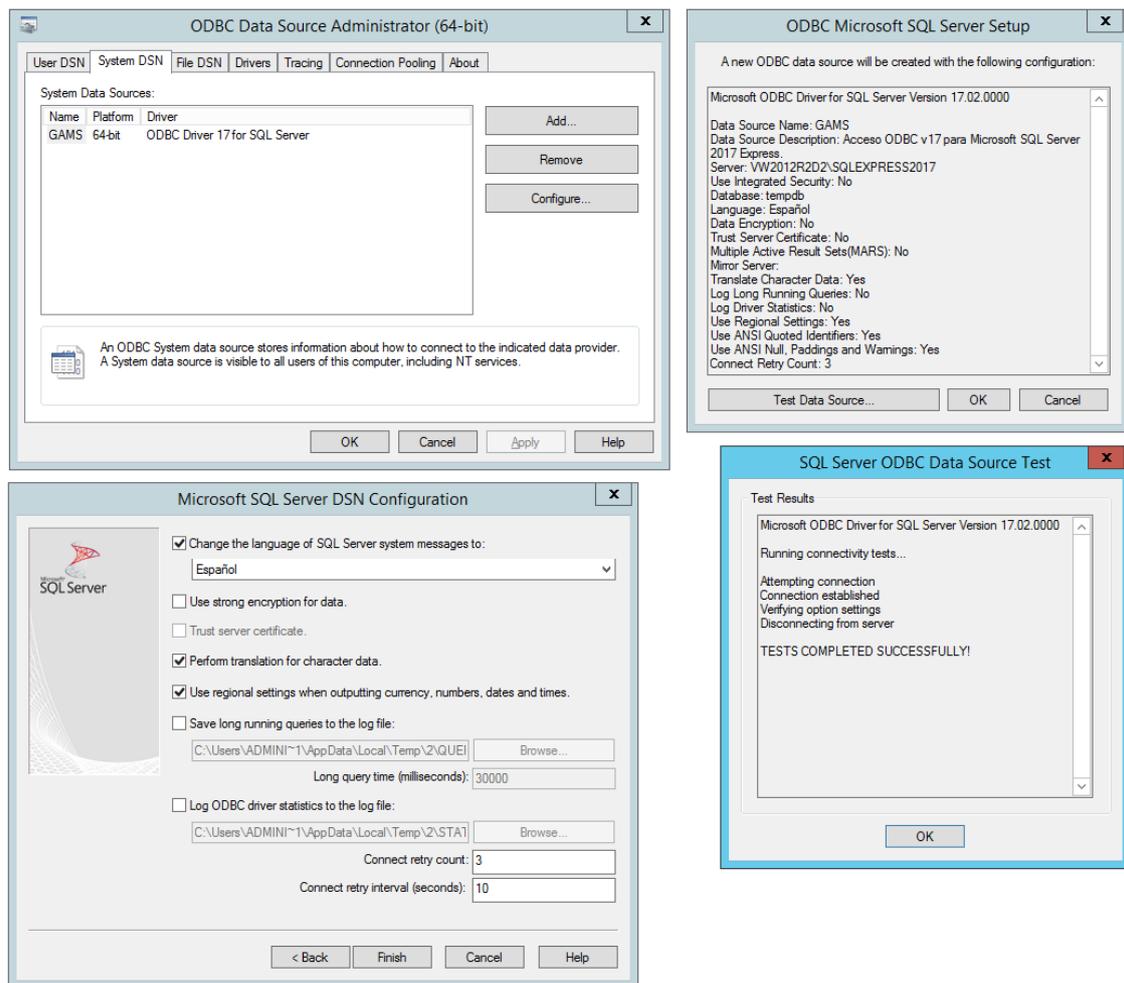


Fig. 10.10.- DSN de sistema configurado para el acceso a Microsoft SQL Server 2017.

En cuanto al controlador de SQL Server específico para PHP, para poder usarse debe descargarse del sitio web del fabricante e, inmediatamente después, debe configurarse el entorno PHP para que incluya las correspondientes bibliotecas dinámicas, modificando para ello el fichero “PHP.INI”. Remitimos al lector al apartado 10.2.1.6, donde se explicará cómo instalar y configurar el entorno PHP 7.1 en el sistema operativo Microsoft Windows.

#### 10.2.1.5.- Creación del inicio de sesión en el sistema gestor de bases de datos.

En este punto, será necesario crear el “inicio de sesión” con el que accederá la aplicación web a la base de datos. Microsoft SQL Server utiliza este término para referirse a las credenciales de acceso al propio sistema gestor.

Este sistema gestor de bases de datos requiere la creación previa del inicio de sesión antes de permitir la ejecución de la aplicación PHP en la que se basa GAMS, exigiendo además la asignación de determinados permisos al mismo [249].

Esto es debido a que mediante dicho inicio de sesión tanto se crearán bases de datos (bien como parte de la instalación de la aplicación de gestión, bien a medida que se creen nuevas aplicaciones gestionadas para consumidores), como se accederá a la información de dichas bases de datos.

Los pasos necesarios para la creación del nuevo inicio de sesión, al que llamaremos “GAMS”, así como la asignación al mismo del rol de servidor “dbcreator”, e incluso la generación opcional del *script* Transact-SQL equivalente, se detallan en: fig. 10.11, fig. 10.12, fig. 10.13, fig. 10.14 y fig. 10.15

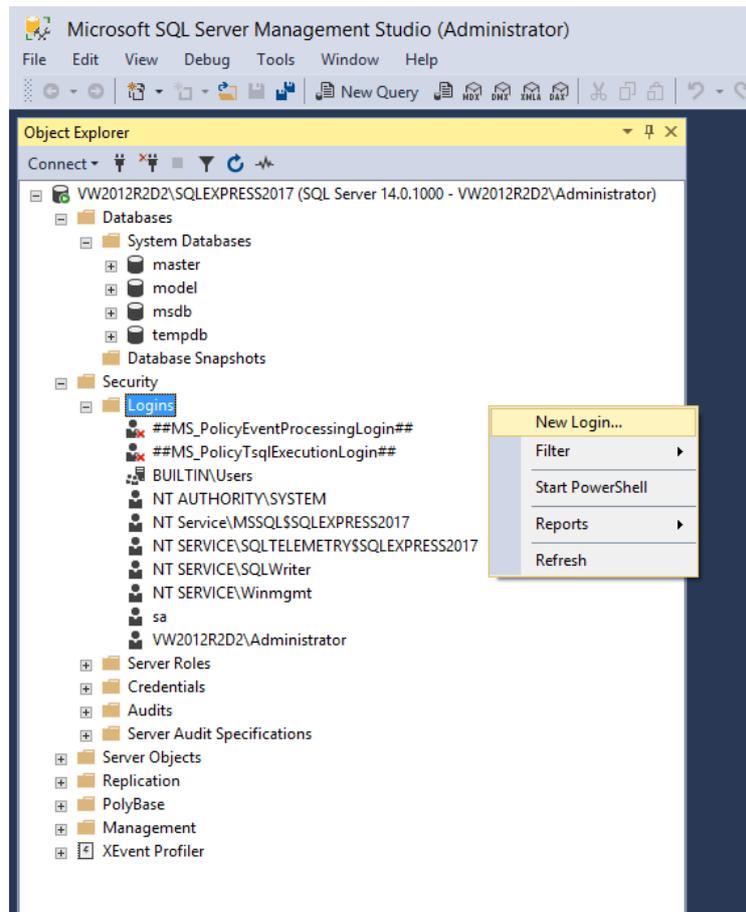


Fig. 10.11.- Creación del inicio de sesión en Microsoft SQL Server (paso 1 de 4).

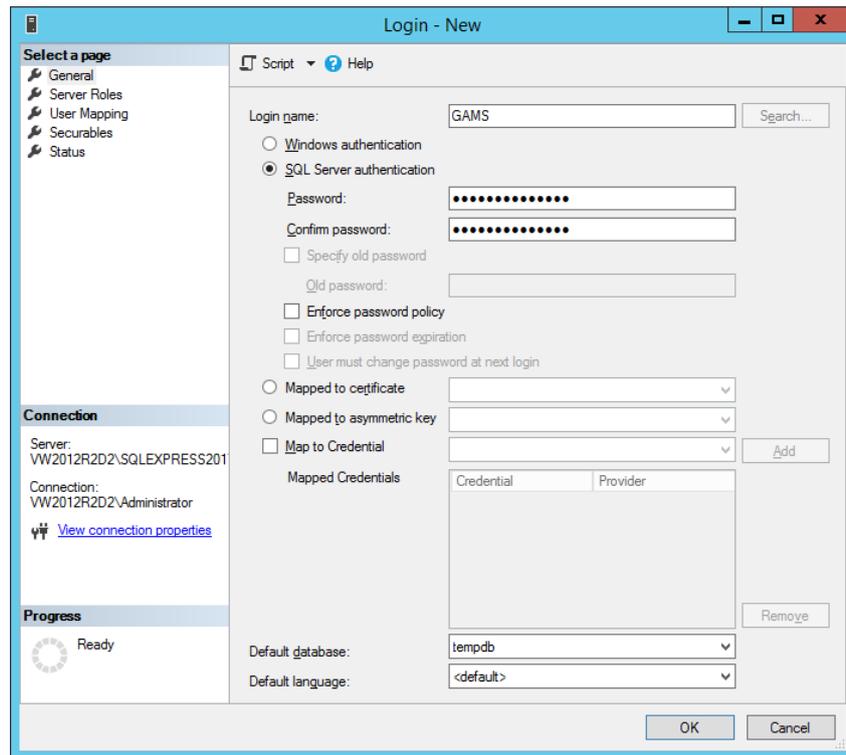


Fig. 10.12.- Creación del inicio de sesión en Microsoft SQL Server (paso 2 de 4).

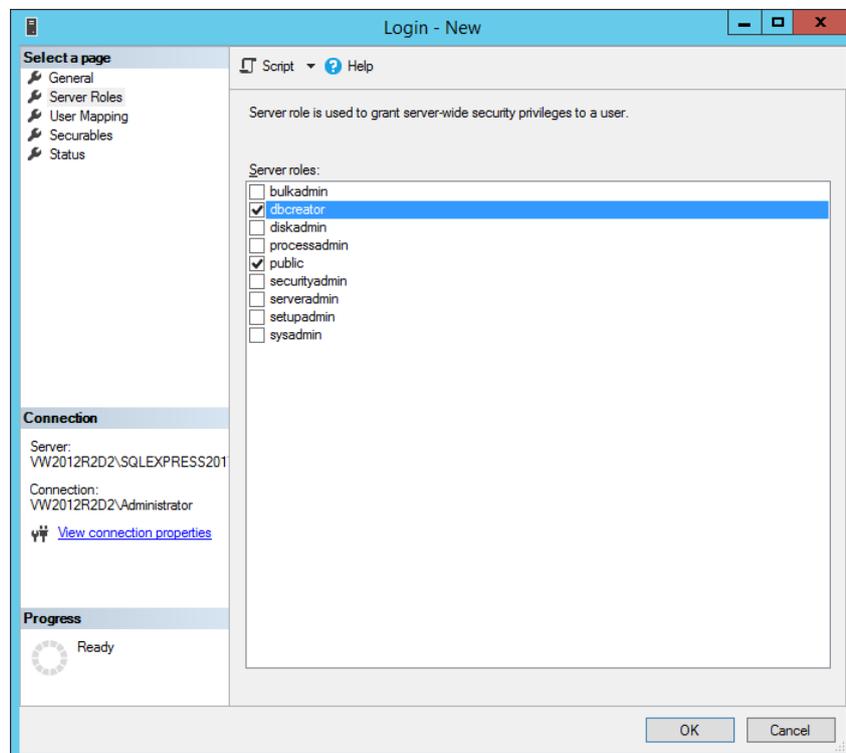


Fig. 10.13.- Creación del inicio de sesión en Microsoft SQL Server (paso 3 de 4).

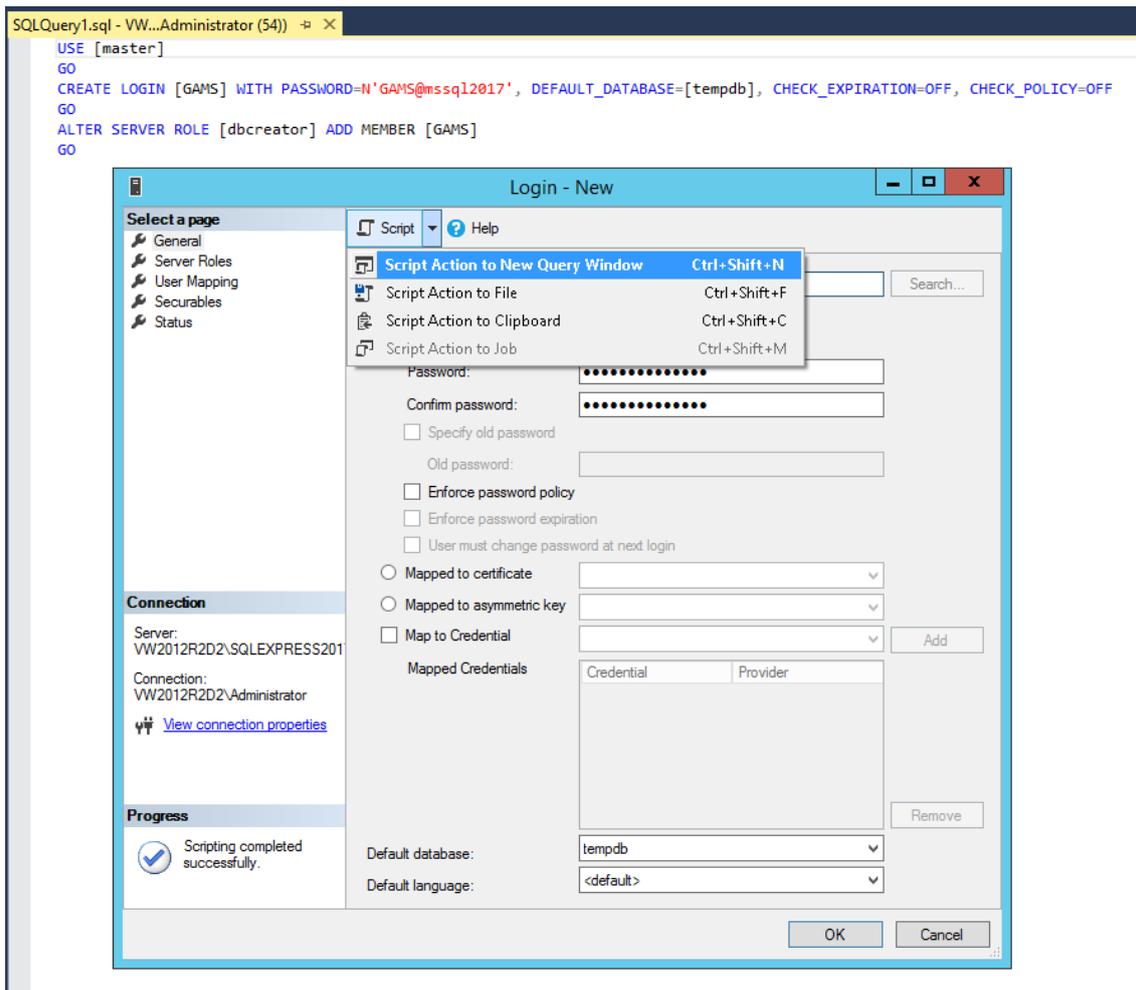


Fig. 10.14.- Creación del inicio de sesión en Microsoft SQL Server (paso 4 de 4).

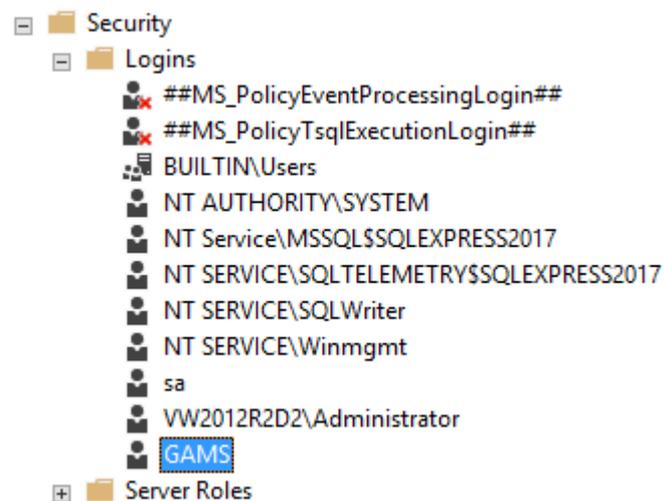
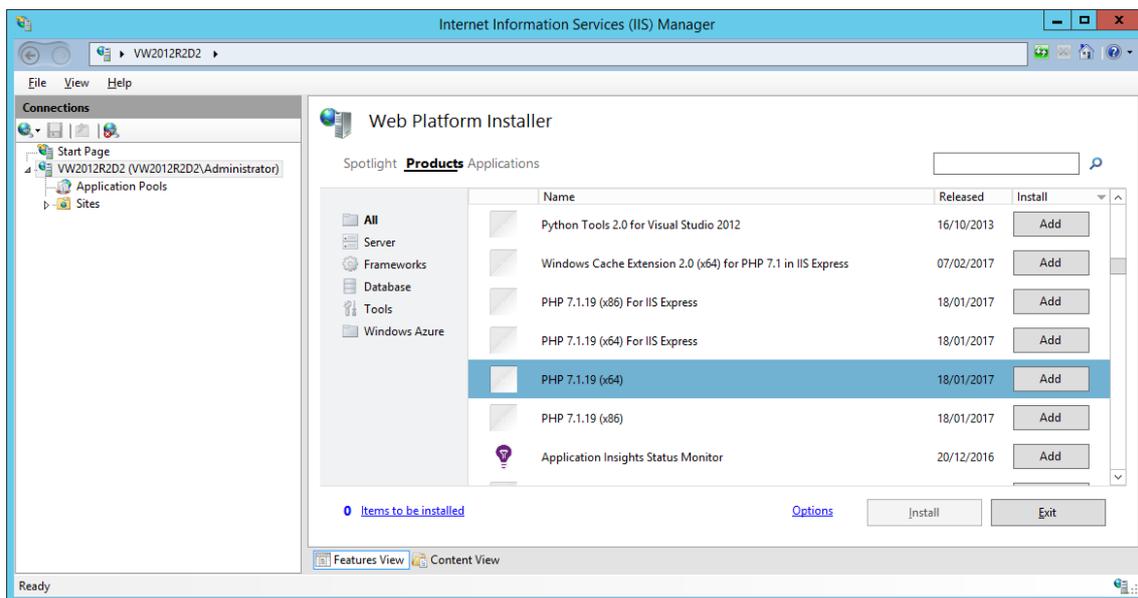


Fig. 10.15.- Inicio de sesión creado para GAMS en Microsoft SQL Server 2017.

### 10.2.1.6.- Instalación de PHP para Windows.

Existen dos procedimientos para instalar PHP en entorno Microsoft Windows: el primero hace uso de una herramienta propia de Microsoft, denominada “Web Platform Installer” [250], que a modo de asistente permite instalar de manera guiada innumerables entornos y aplicaciones, todas ellas relacionadas con el servidor web de Microsoft: Internet Information Server (IIS).



**Fig. 10.16.- Instalación de PHP 7 desde Microsoft Web Platform Installer.**

En la fig. 10.16 puede observarse la herramienta “Web Platform Installer 5”, integrada dentro de la consola de administración “IIS Manager” del servidor web, de modo que explorando a través de las diferentes plataformas y productos basta un *clik* para instalar y configurar cualquiera de los entornos que ofrece; en el caso que nos ocupa: PHP 7.1. Puede aprenderse más acerca de esta herramienta consultando en [251].

Si por el contrario preferimos descargar, instalar y configurar manualmente el entorno PHP 7, deberemos acudir a [252] para su descarga. En este punto cabe mencionar que en el caso de ejecutar PHP 7 en el servidor web IIS, tal y como se explica en este Manual de Usuario, se debe utilizar la versión marcada como “PHP 7.1 VC14 x64 Non Thread Safe”. Por favor, consúltese la documentación (véase [253]) para conocer los detalles exactos acerca de esta versión, así como qué otras versiones existen y deben instalarse en función del sistema operativo y servidor web que se utilicen.

En cualquier caso, conviene recordar la idoneidad de configurar la variable de entorno “PATH” del sistema operativo, añadiendo la ruta del directorio donde se encuentran los ejecutables de PHP (ficheros “PHP.EXE” para línea de mandatos y “PHP-CGI.EXE” para su ejecución vía FastCGI [254]). Habitualmente este directorio es “C:\Program Files\PHP\v7.1”.

Asimismo, será interesante revisar el contenido del fichero “PHP.INI” para configurar el entorno de ejecución. Destacaremos la posibilidad de ampliar la memoria de trabajo (entrada “memory\_limit”), mensajes de error (entradas: “error\_reporting”, “display\_errors” y “log\_errors”, entre otras), tamaño máximo de aceptación para cualquier petición POST (entradas “post\_max\_size” y “upload\_max\_filesize”, útiles por ejemplo para recibir ficheros desde formularios HTML a través de un control INPUT FILE), y sobre todo la sección “[ExtensionList]” donde podremos incluir (activar) tantas extensiones como deseemos.

En el caso de la solución GAMS, se necesitará activar las extensiones propias del controlador de PHP para Microsoft SQL Server (si se han seguido los pasos del punto 10.2.1.4, serán las bibliotecas dinámicas “php\_sqlsrv\_71\_nts\_x64.dll” y “php\_pdo\_sqlsrv\_71\_nts\_x64.dll”).

#### *10.2.1.7.- Creación de la estructura de directorios.*

Este punto es muy dependiente del sistema operativo elegido para alojar la solución GAMS. A continuación, mostraremos los pasos necesarios para la creación de la estructura de directorios, basándonos en la plataforma PC de referencia (arquitectura Intel bajo sistema operativo Microsoft Windows).

El primer paso consiste en crear una “carpeta raíz” de la instalación, donde residirá toda la estructura de directorios y ficheros que conforman la aplicación. En este Manual se utilizará el directorio “C:\UNED\GAMS\” pero puede utilizarse cualquier otra unidad de disco o ruta de acceso.

El segundo paso consiste en copiar los ficheros de programa en los subdirectorios correspondientes. Existen varias opciones para ello:

- En el CD-ROM que acompaña a esta Memoria se encuentra el fichero “GAMS\_FULL.ZIP” que contiene todos estos ficheros debidamente organizados. Bastará con descomprimir el contenido de dicho directorio.
- Instalando GIT y clonando el proyecto asociado a la solución GAMS:
  - Instalar GIT
  - Clonar el proyecto GAMS.git
  - Instalar Composer.
  - Instalar las dependencias mediante Composer.

### 10.2.1.8.- Publicación en el servidor Web.

Una vez copiadas las carpetas y ficheros correspondientes, deberemos publicar la carpeta “C:\GAMS\bin\” a través de la Web mediante el servidor Microsoft IIS disponible en Windows (éste forma parte de los requisitos de infraestructura necesarios, véase apartado 10.1). Para ello, abriremos dentro del “Panel de Control” el icono de “Herramientas Administrativas” y ejecutaremos la Consola de Administración (MMC) “Administrador de Internet Information Services (IIS)”:

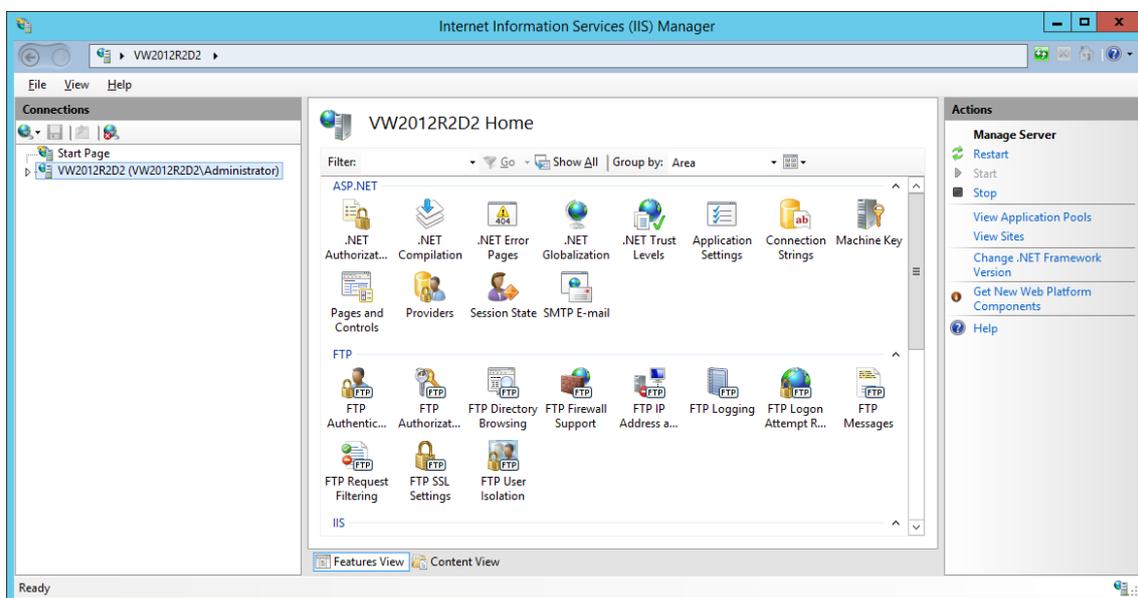


Fig. 10.17.- Consola de administración (MMC) de "Administrador de Internet Information Services (IIS)".

Procederemos a publicar el directorio “C:\GAMS\web\” para lo que existen dos alternativas:

- Crear un nuevo Sitio Web que aloje por completo la aplicación de **Generación Automática de MicroServicios**.
- Crear un nuevo Directorio Virtual en algún Sitio Web previamente existente.

### 10.2.1.9.- *Creación de un nuevo sitio Web.*

Para crear un nuevo Sitio Web que aloje por completo la aplicación de **Generación Automática de MicroServicios**, pulsaremos con el botón secundario (derecho) del ratón sobre el nodo “Sitios Web” del árbol que se encuentra en la parte izquierda del complemento de IIS y seleccionaremos la opción “Nuevo ▶ Sitio Web...” (véase fig. 10.18).

A continuación, deberemos establecer el nombre identificativo del sitio web, la ruta de acceso al directorio físico donde se encontrarán los ficheros a publicar y el protocolo al que responderá el servidor web. Inicialmente la ruta deberá ser un directorio local, aunque con posterioridad podrá establecerse una ruta de red o una redirección mediante URL. En el caso de la aplicación de **Generación Automática de MicroServicios**, este directorio será “C:\GAMS\web”.

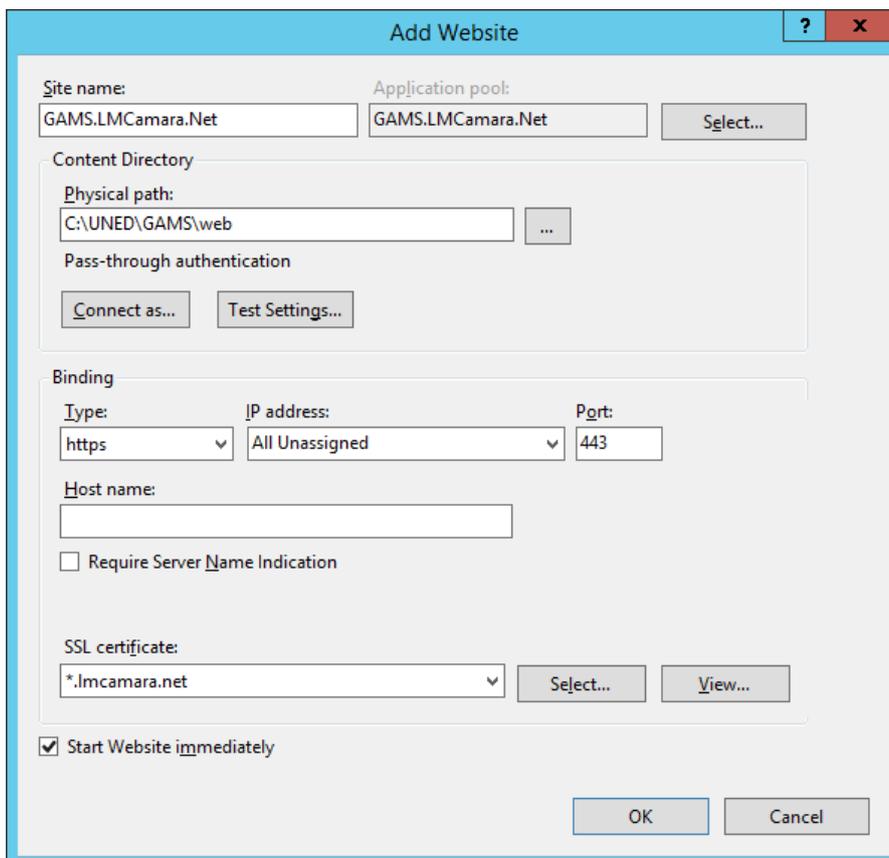


Fig. 10.18.- Creación de un nuevo Sitio Web para publicar la aplicación GCVN.

En este punto, es recomendable modificar las direcciones de aceptación de peticiones (*bindings*) para aceptar tanto el protocolo HTTP (típicamente por el puerto 80, véase Fig. 10.19) como a través del protocolo S/HTTP (puerto 443, siempre que dispongamos de algún tipo de certificado de seguridad SSL/TLS para sitios web, véase FIG).

The screenshot shows a dialog box titled "Add Site Binding". It has three input fields at the top: "Type" with a dropdown menu set to "http", "IP address" with a dropdown menu set to "All Unassigned", and "Port" with a text box containing "80". Below these is a "Host name" text box containing "gams.lmcamara.net". A small text example "Example: www.contoso.com or marketing.contoso.com" is visible below the host name field. At the bottom right, there are "OK" and "Cancel" buttons.

Fig. 10.19.- Configuración HTTP del nuevo sitio web.

The screenshot shows a dialog box titled "Edit Site Binding". It has three input fields at the top: "Type" with a dropdown menu set to "https", "IP address" with a dropdown menu set to "All Unassigned", and "Port" with a text box containing "443". Below these is a "Host name" text box containing "gams.lmcamara.net". There is an unchecked checkbox labeled "Require Server Name Indication". Below that is an "SSL certificate" dropdown menu set to "\*.lmcamara.net", with "Select..." and "View..." buttons to its right. At the bottom right, there are "OK" and "Cancel" buttons.

Fig. 10.20.- Configuración S/HTTP del sitio web, indicando el certificado SSL/TLS.



#### *10.2.1.10.- Instalación de la base de datos.*

Si bien en el CD-ROM que acompaña al presente Trabajo Fin de Carrera se incluye una copia de seguridad de la base de datos SQLite (fichero “GAMS\_Core.sqlite3”) y Microsoft SQL Server (fichero “GAMS\_MsSql2017.bak”), es muy probable que la aplicación evolucione con el tiempo o sea necesario instalarla en otra máquina diferente.

Esto hace que deba en todo momento disponerse de una copia de seguridad de la base de datos con los datos más actualizados disponibles.

Los procedimientos de instalación de la aplicación y de restauración de la base de datos son independientes entre sí, por lo que el orden en el que se realicen es indiferente. Eso sí, ambos deberán completarse correctamente para que la aplicación funcione de manera adecuada.

#### *10.2.1.11.- Pruebas de funcionamiento.*

Una vez realizados todos los pasos para instalar manualmente la aplicación, ésta estará disponible para su ejecución. Conviene en cualquier caso realizar una prueba para verificar que se pueda usar libre de problemas por parte de los usuarios.

Para ello, al tratarse de una aplicación Web, deberá utilizarse un navegador de Internet y visitar la página de inicio de la aplicación de **Generación Automática de MicroServicios**, cuya URL será (utilizando “localhost” como servidor o el nombre especificado durante la creación del sitio web): “http://localhost/”. En caso de realizarse la instalación con éxito, el navegador nos mostrará la pantalla de inicio (véase apartado 10.4).

## 10.3.- Elementos generales de la interfaz de usuario.

Se describen a continuación los elementos generales de los que consta la interfaz gráfica de usuario (GUI [27]).

### 10.3.1.- Acceso web.

La interfaz gráfica de usuario está basada en tecnologías web, haciendo uso de estándares *de facto* como son HTML, CSS y JavaScript. Gracias a ello, el usuario de la solución solamente necesitará un navegador web para acceder a la solución.

### 10.3.2.- Diseño web adaptativo (*responsive*).

Toda la interfaz gráfica de usuario hace uso de diseño web adaptativo (*responsive web design*) [136]. Gracias a este tipo de diseño, las diferentes pantallas se visualizarán adaptadas en función de la resolución del dispositivo utilizado en cada ocasión.

El diseño adaptativo se encarga de mostrar los elementos de manera que siempre se priorice la comodidad de acceso frente a una posición rígida de los mismos en la pantalla, aprovechando al máximo el espacio disponible.

Gracias a las herramientas de desarrollo integradas en la mayoría de los navegadores web, es posible visualizar desde un ordenador personal las páginas web como si éstas fuesen consultadas desde un dispositivo móvil, por ejemplo, limitando la resolución de la visualización (denominada “*viewport*”) o la velocidad máxima de descarga de contenidos.

En fig. 10.21 y fig. 10.22 puede apreciarse el uso de la herramienta Responsive Design Mode [255] del navegador Mozilla Firefox Developer Edition [256] para simular el acceso desde dispositivos con diferentes capacidades, observando el comportamiento adaptativo que muestra la solución.

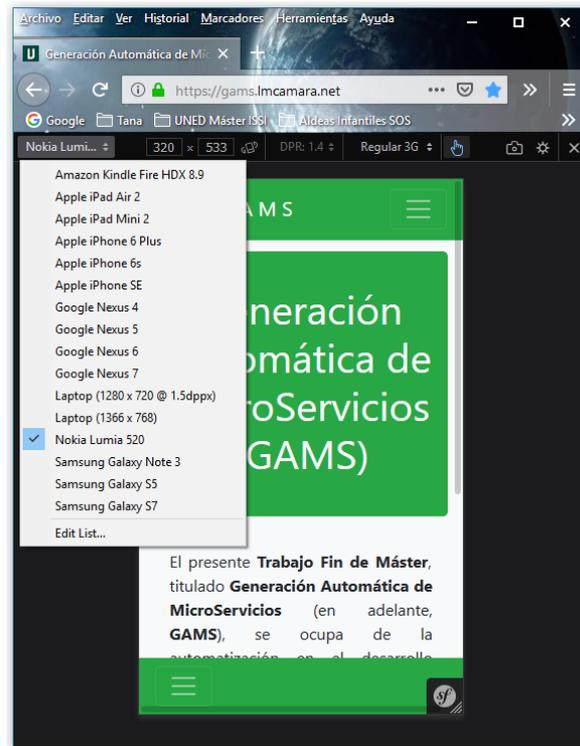


Fig. 10.21.- Ejemplo de visualización web adaptada a teléfonos móviles de con baja resolución.

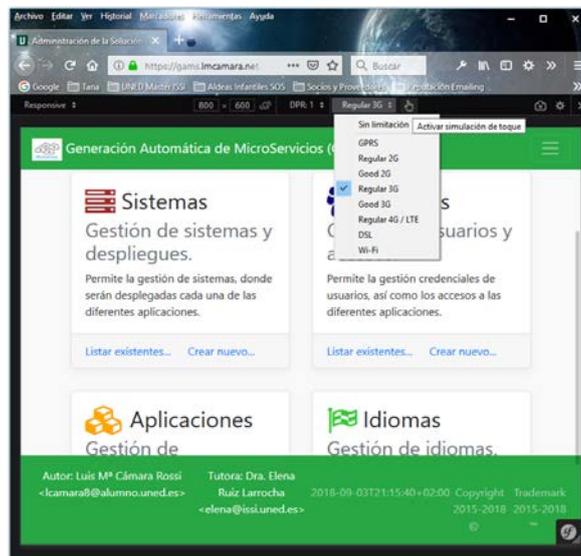


Fig. 10.22.- Ejemplo de visualización web adaptada a una tableta con conexión inalámbrica.

### 10.3.3.- Cabecera: barra superior de navegación.

Todas las páginas de la solución exhiben una cabecera similar, la cual hace uso del diseño web adaptativo (véase la fig. 10.23 para la barra inicial, la fig. 10.24 para una barra completamente extendida y la fig. 10.25 para una barra comprimida).



Fig. 10.23.- Cabecera, barra inicial.



Fig. 10.24.- Cabecera, barra completamente extendida.

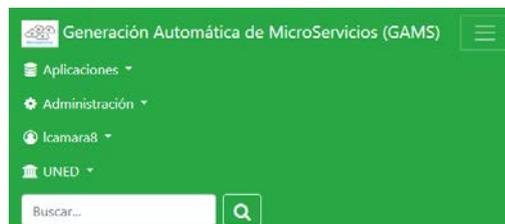


Fig. 10.25.- Cabecera, barra comprimida.

Esta cabecera muestra una barra de navegación mediante la cual es posible navegar a las principales secciones de la solución, a través de un conjunto de cuatro menús (aplicaciones, administración, usuario y enlaces UNED), los cuales se exponen a continuación.

#### 10.3.3.1.- Menú de aplicaciones.

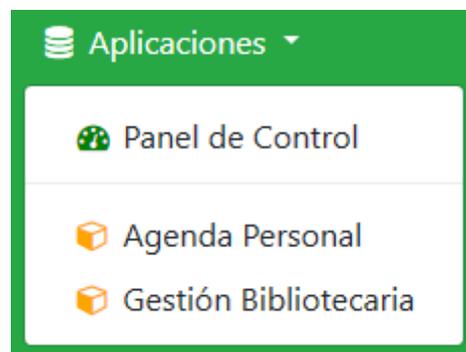


Fig. 10.26.- Menú de aplicaciones, extendido.

Gracias al menú de aplicaciones (véase fig. 10.26) es posible acceder tanto al panel de control (véase apartado 10.7), como a cada una de las diferentes aplicaciones generadas para consumidores que se hayan construido como parte de la solución y a las que tenga acceso el usuario que haya iniciado sesión.

#### 10.3.3.2.- Menú de administración.

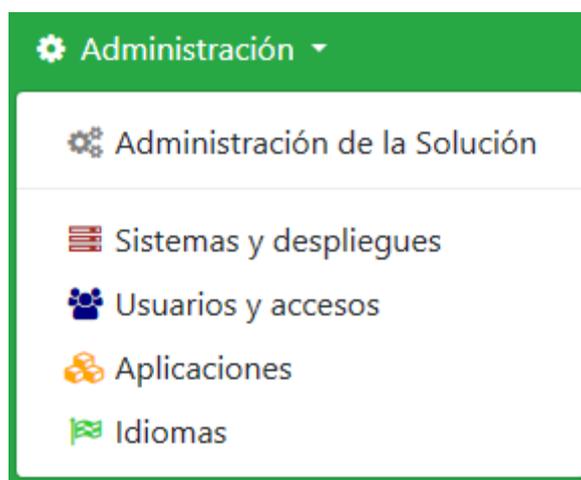


Fig. 10.27.- Menú de administración, extendido.

Mediante el menú de administración (véase fig. 10.27) es posible acceder desde cualquier parte de la solución a las pantallas de: administración de sistemas y despliegues (véase apartado 10.9), usuarios y accesos (véase apartado 10.12), aplicaciones generadas para consumidores (véase apartado 10.16) y administración de idiomas y traducciones (véase apartado 10.18).

Resaltar que este menú solamente estará visible para aquellos usuarios que tengan el rol de administrador.

### 10.3.3.3.- Menú de usuario.

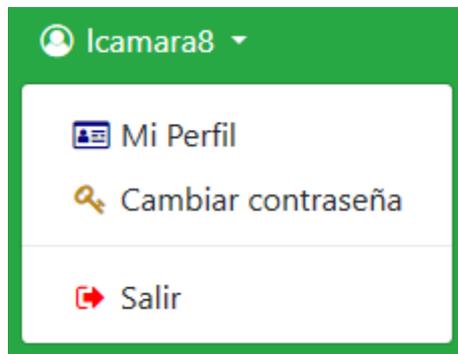


Fig. 10.28.- Menú asociado al perfil del usuario, extendido.

Gracias al menú de usuario, que se presenta bajo el nombre del usuario que haya iniciado sesión (en el caso de la fig. 10.28, se muestra el nombre de usuario “Icamara8”), desde cualquier parte de la solución es posible acceder a las páginas con la información del perfil del usuario y la página para el cambio de contraseña, así como cerrar sesión y salir de la solución (lo que nos conducirá a la página de inicio, véase apartado 10.4).

### 10.3.3.4.- Menú de enlaces de la UNED.

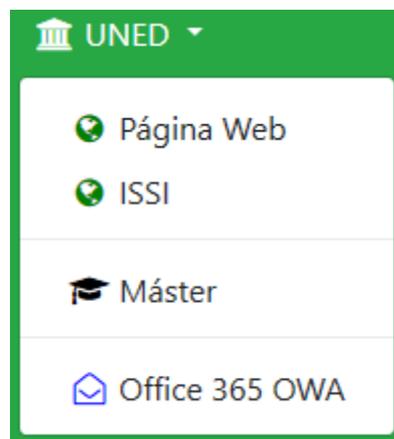


Fig. 10.29.- Menú de la UNED, extendido.

El menú de enlaces de la UNED (véase fig. 10.29) contiene una serie de enlaces de interés a páginas web de la Universidad, relacionadas con el Máster para el que se desarrolla el presente Trabajo.

### 10.3.3.5.- Control de búsqueda global.



Fig. 10.30.- Control de búsqueda global.

Actualmente la función de búsqueda global no está implementada. Por tanto, este control no realiza en estos momentos ninguna función especial.

### 10.3.4.- Pie de página: barra inferior de información.

Todas las pantallas de la aplicación muestran un pie de página similar. Gracias a este pie de página se muestra información general de la aplicación, como son: autor del Trabajo, tutora, fecha actual del sistema, texto básico de copyright y marca comercial (estos dos últimos, a modo de ejemplo).

El pie de página exhibe igualmente un comportamiento adaptativo, en función del ancho de pantalla disponible en cada caso, llegándose a ocultar por defecto en las resoluciones inferiores (véanse: la fig. 10.31 para una barra completamente extendida, la fig. 10.32 para una barra ligeramente comprimida y la fig. 10.33 para una barra completamente contraída y cómo se expande al hacer uso del icono de expansión).



Fig. 10.31.- Pie de página inferior, completamente extendido.



Fig. 10.32.- Pie de página inferior, parcialmente contraído.



Fig. 10.33.- Pie de página inferior, completamente contraído.

### 10.3.5.- Contenidos en múltiples idiomas.

La solución hace uso de las capacidades de traducción de textos literales que ofrece la tecnología subyacente. Por ello, en función del idioma configurado en el navegador web del usuario, la interfaz de usuario podrá visualizarse en un idioma u otro.

La solución hace actualmente uso de esta capacidad multi-idioma para presentar los textos en castellano y en inglés.

## 10.4.- Pantalla de inicio.

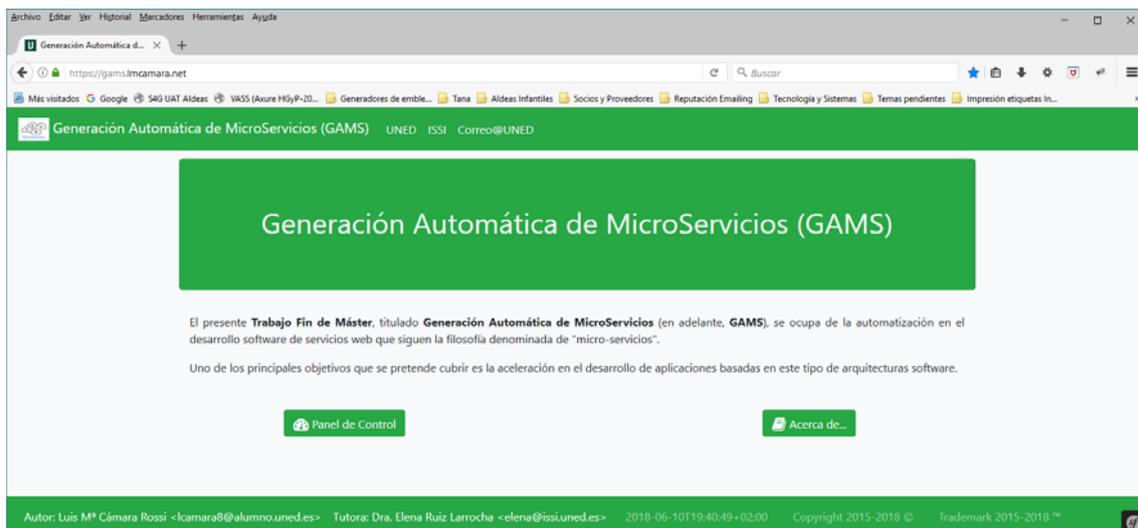


Fig. 10.34.- Pantalla de inicio.



El sistema da la bienvenida a los usuarios a través de una pantalla de inicio (véase fig. 10.34) donde se describen los objetivos fundamentales de la aplicación.

Esta pantalla contiene un botón en la parte inferior izquierda denominado “Panel de Control” que dirige a los usuarios hacia la pantalla principal (véase apartado 10.7), así como un botón denominado “Acerca de...” que muestra información más detallada acerca de la solución GAMS (véase apartado 10.5).

Mientras que los usuarios no se validen previamente, serán siempre redirigidos automáticamente a la pantalla de autenticación (véase apartado 10.6) para realizar el inicio de sesión.

## **10.5.- Pantalla de información acerca de la solución.**

Cuando se pulse sobre el botón “Acerca de...” de la pantalla de inicio, el sistema mostrará la pantalla de información acerca de la solución.

Esta pantalla muestra información de interés acerca de la solución GAMS y, en general, acerca del presente Trabajo.

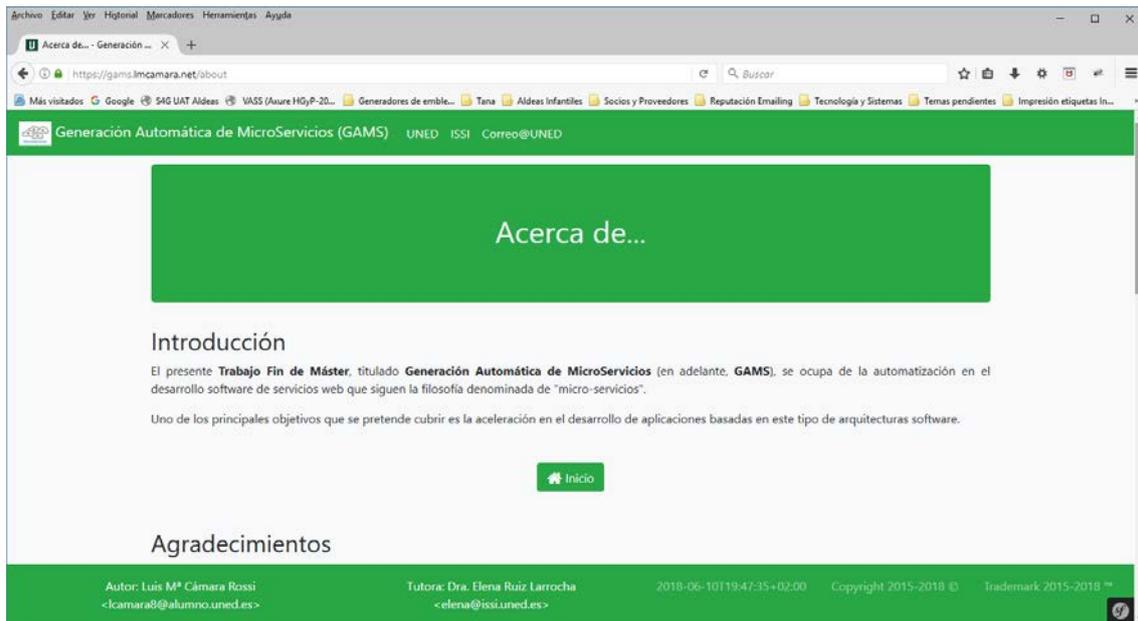


Fig. 10.35.- Pantalla de información acerca de la solución.

Pulsando sobre el botón “Inicio”, se volverá a la pantalla de inicio de la solución.

## 10.6.- Pantalla de autenticación.

La pantalla de autenticación es la encargada de velar por que los datos de los usuarios se mantengan confidenciales durante la ejecución de la aplicación. Para ello establece un mecanismo de autenticación mediante el cual todos los usuarios registrados en el sistema deberán introducir sus credenciales de acceso (nombre de usuario y contraseña), previamente al uso del mismo. A este proceso se le denomina igualmente *inicio de sesión*.

Esta pantalla presenta dos posibles formas de autenticación, a saber:

- Introducción de credenciales.
- Restablecimiento de contraseñas olvidadas.

### 10.6.1.- Introducir credenciales.

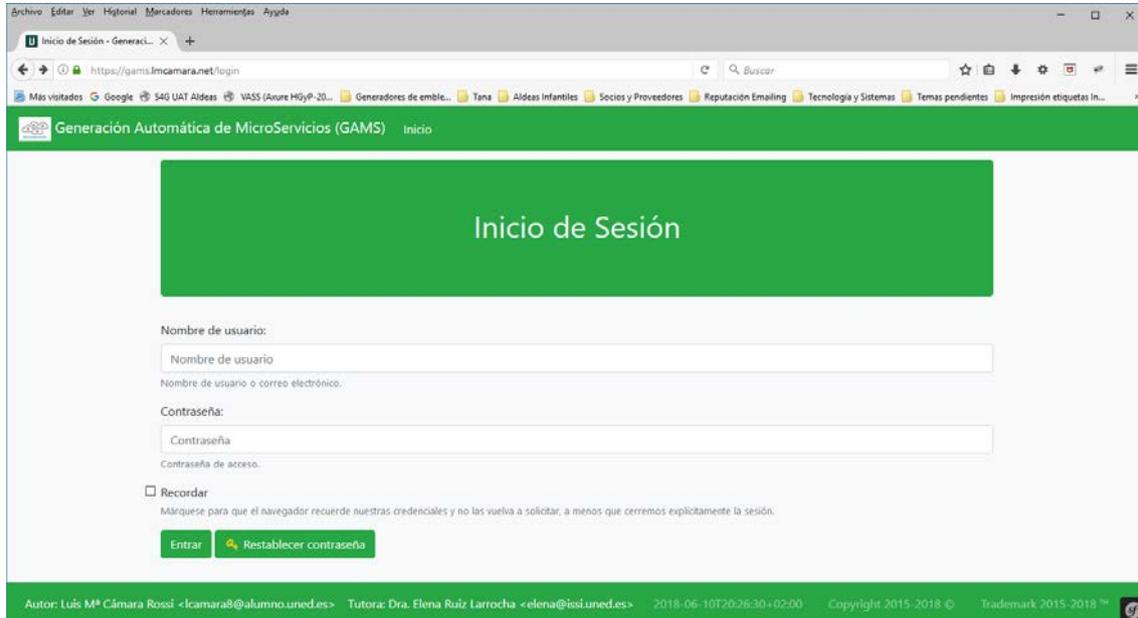


Fig. 10.36.- Pantalla de autenticación: introducir credenciales.

Por defecto, la pantalla de autenticación muestra los controles “nombre de usuario”, “contraseña” y “recordar”, de cara a determinar el usuario que utilizará la aplicación (véase fig. 10.36).

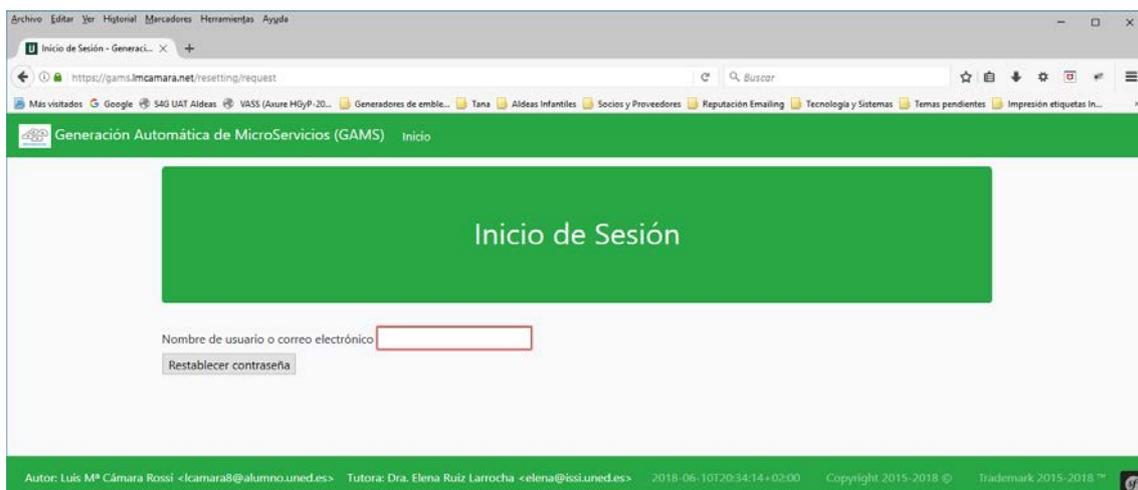
La función “recordar” permite acceder a la aplicación con posterioridad, sin necesidad de volver a introducir las credenciales; por ejemplo, al cerrar el navegador de internet. Para lograr que el sistema olvide nuestras credenciales, tendremos que cerrar la sesión de manera explícita.

Si las credenciales son correctas, se dará al usuario paso a la aplicación, mostrando por defecto la pantalla denominada “Cuadro de Mando”.

Si las credenciales son incorrectas o el usuario se encuentra bloqueado, se mostrará una advertencia indicando dicha situación, impidiendo el acceso a la aplicación. Un usuario puede encontrarse bloqueado cuando, por ejemplo, se introducen mal las credenciales un número determinado de ocasiones seguidas; también sucede cuando el usuario ha sido expresamente bloqueado por algún administrador del sistema.

### 10.6.2.- Restablecer contraseña.

La segunda opción, “restablecer contraseña”, tiene por objetivo permitir a un usuario que haya olvidado su clave acceder al sistema. El sistema solicitará al usuario que introduzca su correo electrónico, utilizado durante la creación de su registro de usuario (véase fig. 10.37).



**Fig. 10.37.-** Pantalla de autenticación: recordar contraseña.

Si el sistema determina que la dirección de correo electrónico especificada se encuentra registrada en el sistema, enviará un correo electrónico al usuario, mostrando una pantalla avisando de que dicho correo ha sido enviado y las restricciones de seguridad que impone el sistema para restablecer contraseñas (véase fig. 10.38).

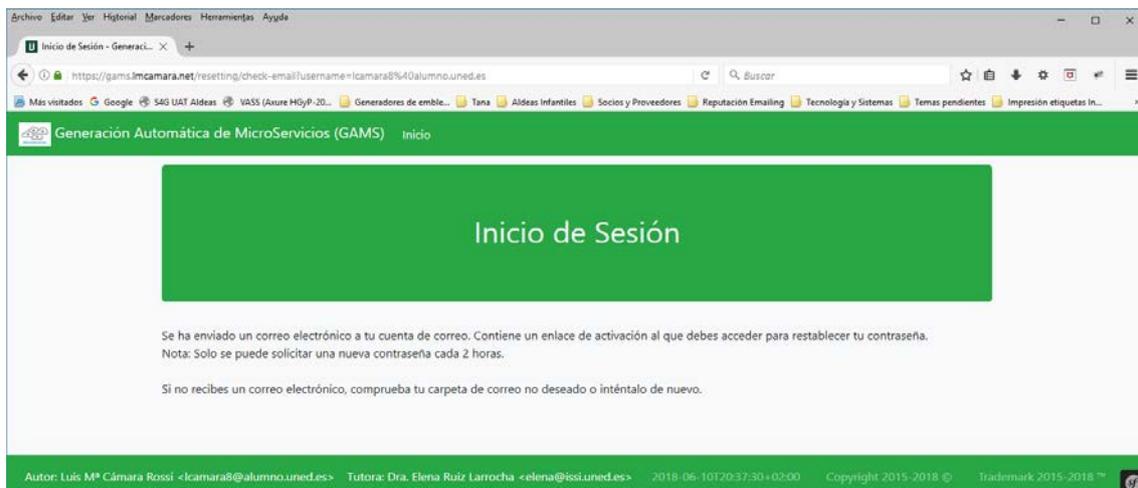


Fig. 10.38.- Pantalla de autenticación: envío de correo para restablecer contraseña.

Se muestra a continuación (véase fig. 10.39) un ejemplo del correo electrónico enviado al usuario, cuando éste lo visualiza en el cliente de correo electrónico de un teléfono móvil. Obsérvese el enlace que dirige a la “pantalla de restablecimiento de contraseña”.

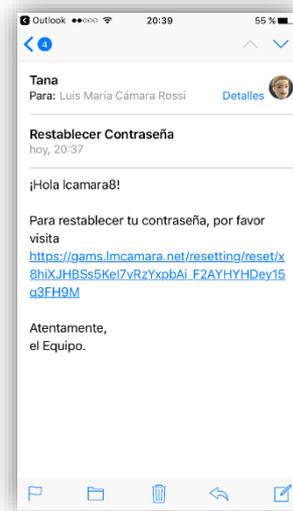


Fig. 10.39.- Correo electrónico para restablecimiento de contraseñas.

Al utilizar el enlace mencionado en el párrafo anterior, se abrirá una pantalla (véase fig. 10.40) donde se pedirá al usuario que establezca su nueva contraseña, debiendo repetirla para evitar equívocos.



Fig. 10.40.- Pantalla de restablecimiento de contraseña, vista desde un teléfono móvil.

## 10.7.- Panel de control (pantalla principal).

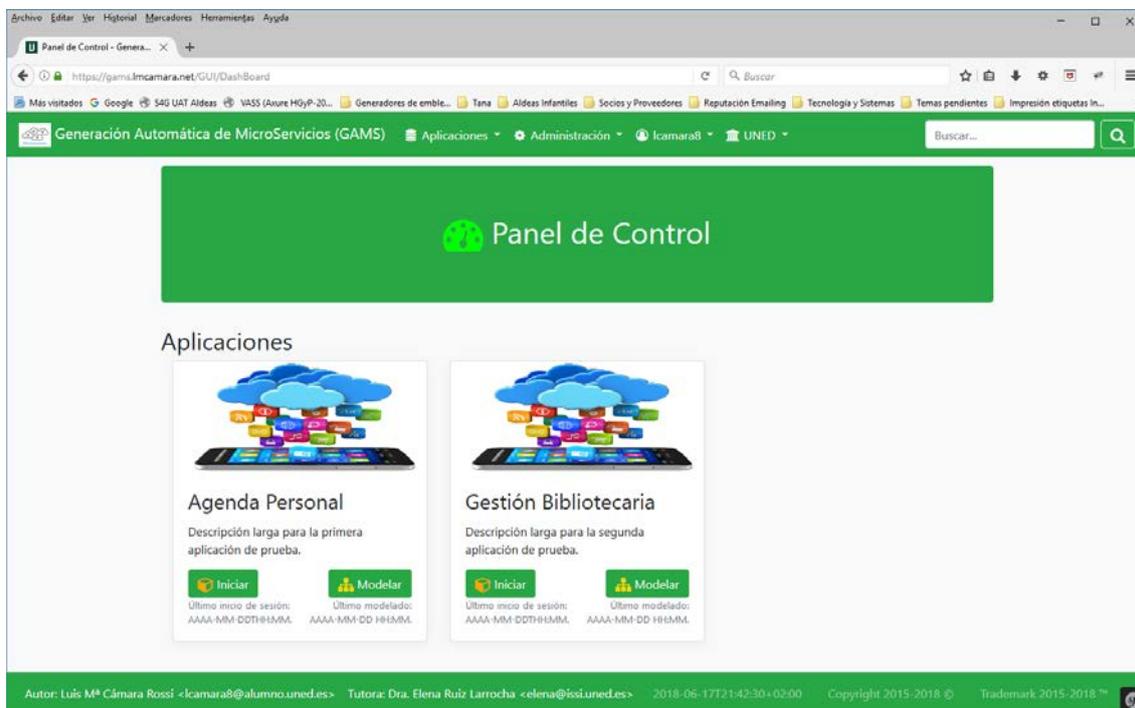


Fig. 10.41.- Pantalla principal, denominada panel de control.

La pantalla principal de la aplicación, denominada panel de control (véase fig. 10.41), presenta los accesos a todas y cada una de las aplicaciones generadas para consumidores a las que el usuario tenga acceso. Bastará al usuario pulsar sobre el botón “Iniciar” para acceder a la correspondiente aplicación.

Si dicho usuario tiene además el rol de administrador, la solución mostrará adicionalmente el botón “Modelar” para gestionar el modelo de dominio de la aplicación (véase apartado 10.17), permitiendo al usuario acceder directamente a dicha funcionalidad sin necesidad de hacer uso del menú de administración (véase apartado 10.3.3.2) ni la pantalla de administración de aplicaciones (véase apartado 10.16).

## 10.8.- Pantalla de administración de la solución.

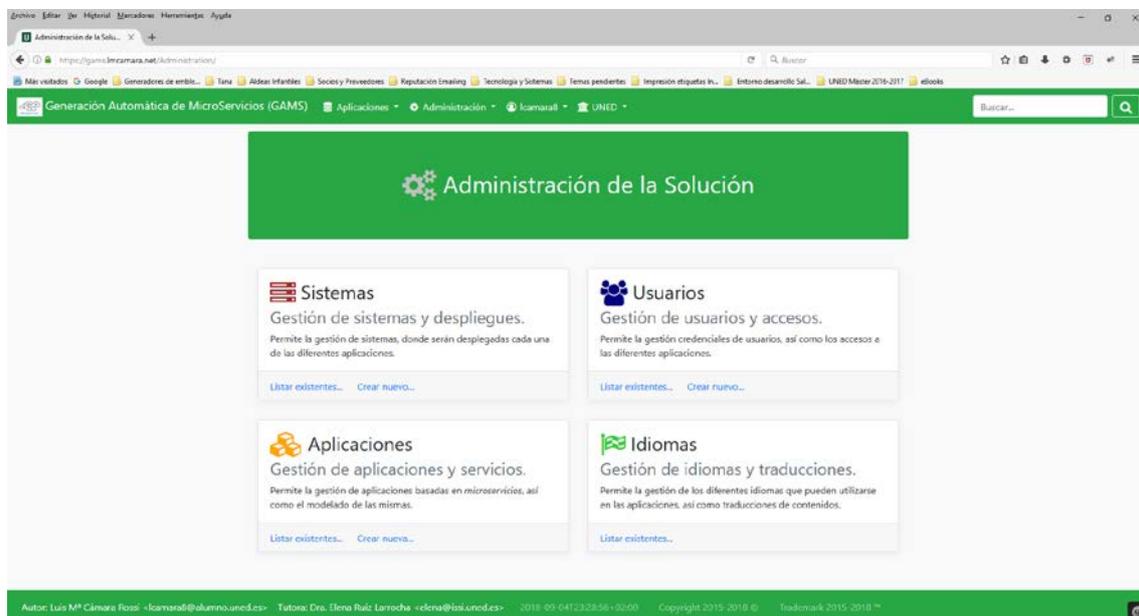


Fig. 10.42.- Pantalla de administración general de la solución.

La pantalla de administración de la solución (véase fig. 10.42) sirve de punto de encuentro de toda la funcionalidad que los usuarios con el rol de administración pueden necesitar, como parte de sus labores de gestión de la solución.

A través de esta pantalla un administrador puede acceder a: la pantalla para gestionar sistemas y despliegues (véase apartado 10.9), la pantalla para gestionar aplicaciones generadas para consumidores (véase apartado 10.16), la pantalla para gestionar usuarios y conceder permisos a las diferentes aplicaciones (véase apartado 10.12), así como la pantalla para gestionar idiomas y traducciones (véase apartado 10.18).

## 10.9.- Pantalla de administración de sistemas y despliegues.

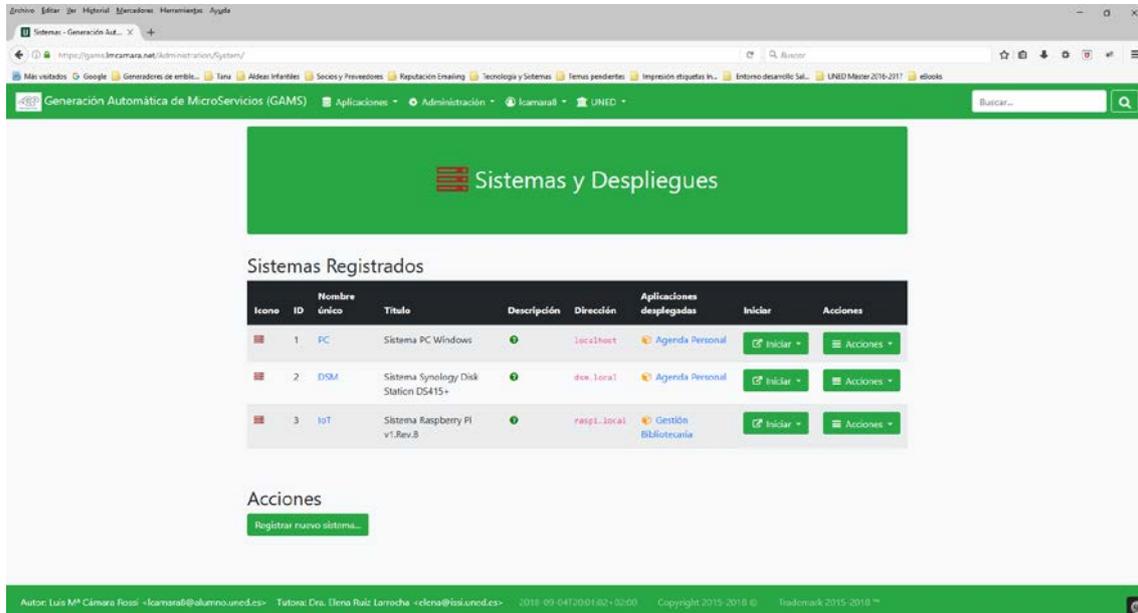


Fig. 10.43.- Pantalla de administración de sistemas y despliegues.

Mediante la pantalla de administración de sistemas y despliegues (véase fig. 10.43), los administradores pueden registrar en la solución GAMS los diferentes sistemas que podrán ser gestionados.

Para cada uno de estos sistemas, la solución permitirá visualizar (pulsando sobre el nombre único del sistema, véase apartado 10.10) y modificar (accediendo a través del menú desplegable “Acciones” según fig. 10.44, véase apartado 10.11) individualmente los datos del registro correspondiente.

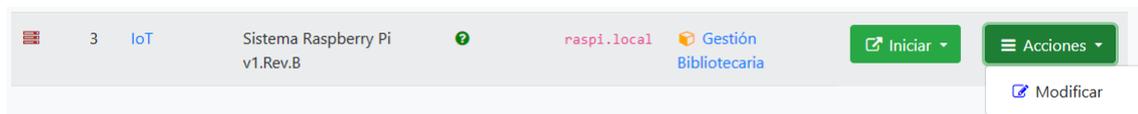


Fig. 10.44.- Menú desplegable de acciones a realizar sobre un sistema registrado en la solución GAMS.

Asimismo, desde esta pantalla se facilita el acceso a cualquiera de las combinaciones posibles de ejecución de una aplicación en los sistemas en los que haya sido desplegada. Para ello, a través del menú desplegable “Iniciar” (véase fig. 10.45) se muestran las direcciones URL para navegar hacia la aplicación correspondiente.

| Icono | ID | Nombre único | Título                               | Descripción | Dirección   | Aplicaciones desplegadas | Iniciar | Acciones |
|-------|----|--------------|--------------------------------------|-------------|-------------|--------------------------|---------|----------|
|       | 1  | PC           | Sistema PC Windows                   |             | localhost   | Agenda Personal          |         |          |
|       | 2  | DSM          | Sistema Synology Disk Station DS415+ |             | dsm.local   | Agenda Personal          |         |          |
|       | 3  | IoT          | Sistema Raspberry Pi v1.Rev.B        |             | raspi.local | Gestión Bibliotecaria    |         |          |

Fig. 10.45.- Menú desplegable para el inicio de una aplicación, desde un sistema registrado en la solución GAMS.

Indicar que tanto el despliegue (instalación) como el repliegue (desinstalación) de las diferentes aplicaciones modeladas para consumidores, se deberá hacer uso de la pantalla de visualización de aplicaciones.

## 10.10.- Pantalla de administración para la consulta de un sistema existente.

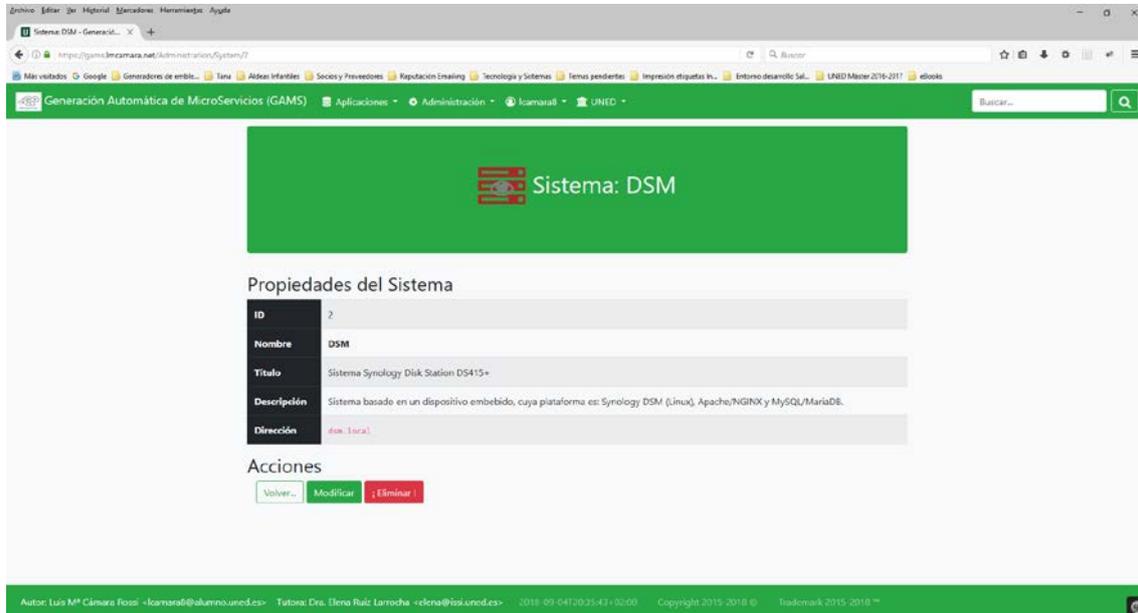


Fig. 10.46.- Pantalla de visualización de un sistema existente.

Mediante la pantalla de visualización de un sistema existente (véase fig. 10.46) es posible consultar las propiedades del registro de dicho sistema en la solución GAMS sin realizar ninguna modificación a las mismas.

A través de esta pantalla es posible navegar hacia las pantallas para “Modificar” dicho registro, así como “Eliminar” el sistema.

## 10.11.- Pantalla de administración para la modificación de un sistema existente.

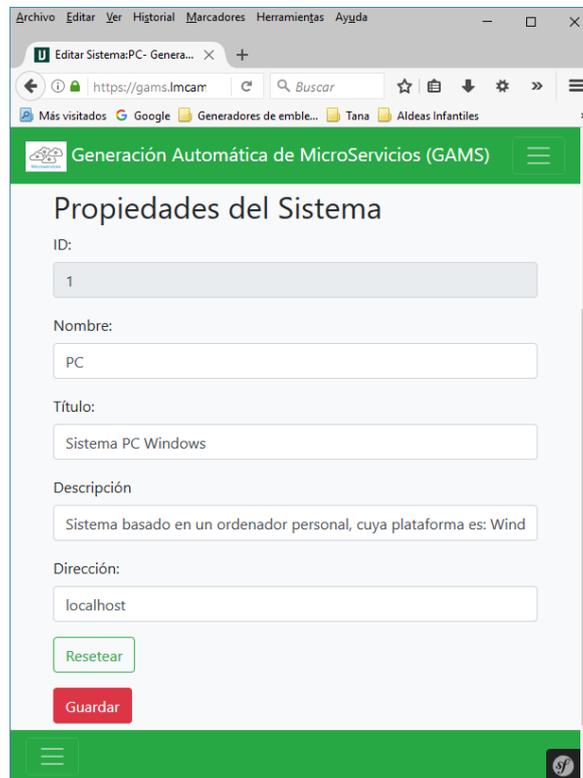


Fig. 10.47.- Pantalla de modificación de un sistema existente.

La pantalla de modificación de un sistema existente (véase fig. 10.47) permite modificar las propiedades del registro del sistema en edición.

A través de esta pantalla es igualmente posible “Eliminar” dicho registro de la solución.

## 10.12.- Pantalla de administración de usuarios y accesos.

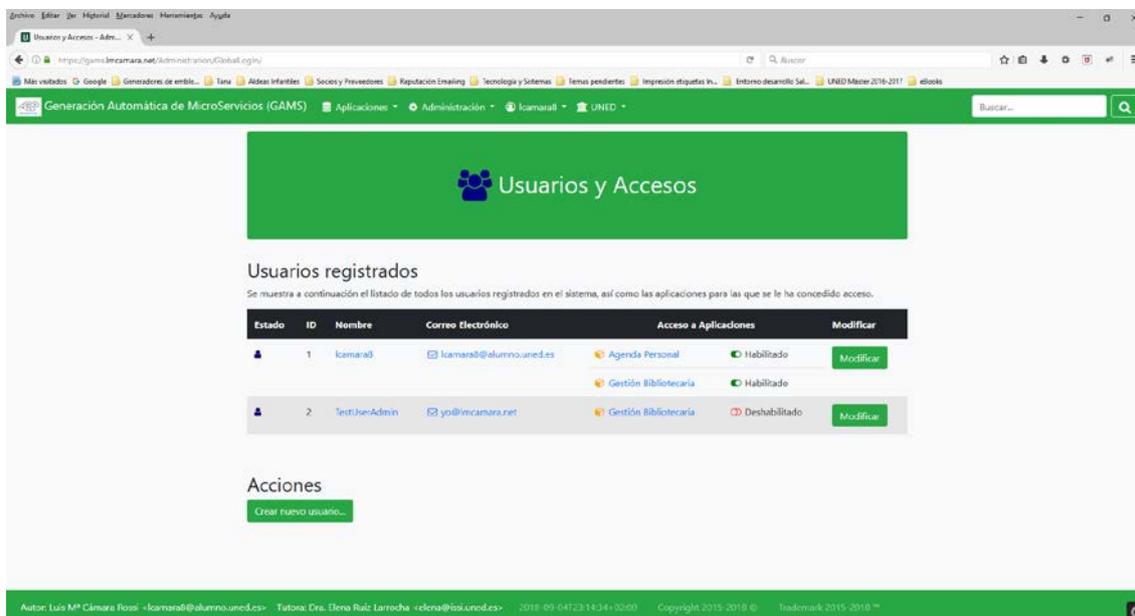


Fig. 10.48.- Pantalla de administración de usuarios y accesos.

La pantalla de administración de usuarios y accesos (véase Fig. 10.48) permite consultar la información de todos los usuarios registrados en la solución GAMS, así como verificar el acceso de cada usuario a las diferentes aplicaciones generadas para consumidores.

A través de esta pantalla es posible:

- Crear nuevos usuarios, pulsando sobre el botón “Crear nuevo usuario...” situado en la parte inferior de la pantalla.
- Visualizar la información de un usuario, pulsando sobre el enlace asociado al nombre del mismo.
- Enviar automáticamente un correo electrónico a dicho usuario, pulsando sobre su dirección de correo electrónico (que realmente es un enlace y abrirá automáticamente el cliente de correo electrónico que tengamos asociado en nuestro dispositivo).

- Consultar la información de una aplicación generada para consumidores, pulsando sobre el enlace asociado al nombre de la misma.
- Pulsando sobre el correspondiente botón “Modificar”, acceder a la pantalla de modificación de un usuario concreto (véase apartado fig. 10.51).

### 10.13.- Pantalla de administración para la consulta de un usuario existente.

The screenshot displays the GAMS user administration interface. The left panel shows the user's profile information, and the right panel shows the user's access permissions.

**Usuario: Icamara8**

**Propiedades del Usuario**

| Propiedad                 | Valor   |
|---------------------------|---|
| Id :                      | 1   |
| Nombre de usuario :       | Icamara8  |
| Correo electrónico :      | Icamara8@alumno.uned.es                                   |
| Roles :                   | FOSUserBundle SuperAdmin<br>ROLE_SUPER_ADMIN<br>ROLE_USER |
| Grupos :                  |   |
| Último inicio de sesión : | 2018-09-04T21:32:29+02:00                                 |
| Activo :                  | <input checked="" type="checkbox"/> Habilitado            |

**Accesos del Usuario**

| Aplicación | Acceso   |
|------------|--|
| Test-01    | <input checked="" type="checkbox"/> Acceso concedido |
| Test-DOS   | <input checked="" type="checkbox"/> Acceso concedido |

**Acciones**

[Volver al listado de usuarios...](#) [Modificar...](#)

Fig. 10.49.- Pantalla de administración para la consulta de un usuario.

La pantalla de administración para la consulta de un usuario existente (véase fig. 10.49) muestra los datos relacionados con un usuario en concreto, divididos en dos grandes bloques de información:

- Propiedades del usuario:
  - Identificador (único en toda la solución, generado automáticamente).
  - Nombre de usuario (único en toda la solución).
  - Correo electrónico del usuario (también debe ser único en toda la solución).
  - Roles de usuario: indica si el usuario es administrador (mostrando el texto “ROLE\_SUPER\_ADMIN”) o consumidor (mostrando “ROLE\_USER”).
  - Grupos de seguridad: este atributo estará siempre vacío, dado que esta funcionalidad se ha dejado fuera de la solución (véase 5.1).
  - Último inicio de sesión: muestra la fecha y la hora del último inicio de sesión realizado por el usuario para acceder a la solución.
  - Activo: indica si se concederá acceso al usuario a la solución o, por el contrario, éste se encuentra bloqueado.
- Accesos del usuario:
  - Para cada aplicación generada para consumidores a la que se haya concedido acceso al usuario, independientemente de si sus credenciales se encuentran activas, indica si dicho acceso está vigente (y por tanto el usuario puede utilizar la aplicación) o bloqueado (y se le negará el acceso exclusivamente a dicha aplicación).

Desde esta pantalla es posible “Modificar” la información personal del usuario, así como reconfigurar sus accesos (véase apartado 10.15).

## 10.14.- Pantalla de administración para la creación de un nuevo usuario.

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

Creación de Nuevo Usuario

https://gams.lcamara.net/Administration/GlobalLogin/new

80%

Buscar

Más visitados Google Generadores de emble... Tana Aldeas Infantiles Socios y Proveedores Reputación Emailing Tecnología y Sistemas Temas pendientes Impresión etiquetas In...

Generación Automática de MicroServicios (GAMS) Aplicaciones Administración Icamara8 UNED

Buscar...

### Creación de Nuevo Usuario

#### Propiedades del Usuario

ID:

0

Correo electrónico:

Activado:

Nombre único de usuario:

Contraseña:

Resetear contraseña

¡ Resetear !

¡ Modificar !

Autor: Luis Mª Cámara Rossi <lcamara@alumno.uned.es> Tutora: Dra. Elena Ruiz Larrocha <elena@isi.uned.es> 2018-09-04T23:38:46+02:00 Copyright 2015-2018 © Trademark 2015-2018™

Fig. 10.50.- Pantalla de administración para la creación de un nuevo usuario.

Mediante la pantalla de administración para la creación de un nuevo usuario (véase fig. 10.50), es posible que cualquier usuario administrador pueda dar de alta a nuevos usuarios, indicando las credenciales de acceso para los mismo:

- Nombre de usuario.
- Correo electrónico.
- Contraseña inicial de acceso.
- Si el usuario recién creado se encontrará activado desde el principio, o por el contrario se creará bloqueado.

## 10.15.- Pantalla de administración para la modificación de un usuario existente y sus accesos.

The screenshot shows a web browser window with the URL `https://gams.lmcamara.net/Administration/GlobalLogin/1/Edit`. The page title is "Modificación de Usuario: Icamara8". Below the title, there is a section for "Credenciales del Usuario" with the following fields:

- ID: 1
- Correo electrónico: Icamara8@alumno.uned.es
- Activado:
- Nombre único de usuario: Icamara8
- Contraseña: (empty field)

The footer of the page contains the text: "Autor: Luis Mª Cámara Rossi <Icamara8@alumno.uned.es> Tutora: Dra. Elena Ruiz Larrocha <elena@issi.uned.es> 2018-09-05T05:46:14+02:00 Copyright 2015-2018 © Trademark 2015-2018™"

Fig. 10.51.- Pantalla de administración de modificación de un usuario y sus accesos (primera parte).

The screenshot shows the "Datos adicionales del usuario" section of the user modification page. It contains the following information:

| Propiedad                 | Valor   |
|---------------------------|---|
| Roles :                   | FOSUserBundle SuperAdmin<br>ROLE_SUPER_ADMIN<br>ROLE_USER |
| Grupos :                  |   |
| Último inicio de sesión : | 2018-09-04T21:32:29+02:00                                 |

Below this is the "Accesos del Usuario" section, which is a table with the following data:

| Aplicación | Acceso concedido                                     | Acciones   |
|------------|--|--|
| Test-01    | <input checked="" type="checkbox"/> Acceso concedido | <a href="#">Bloquear</a> <a href="#">Eliminar acceso</a> |
| Test-DOS   | <input checked="" type="checkbox"/> Acceso concedido | <a href="#">Bloquear</a> <a href="#">Eliminar acceso</a> |

The footer of the page contains the text: "Autor: Luis Mª Cámara Rossi <Icamara8@alumno.uned.es> Tutora: Dra. Elena Ruiz Larrocha <elena@issi.uned.es> 2018-09-05T05:48:59+02:00 Copyright 2015-2018 © Trademark 2015-2018™"

Fig. 10.52.- Pantalla de administración de modificación de un usuario y sus accesos (segunda parte).

La pantalla de administración para la modificación de un usuario existente y sus accesos (véanse fig. 10.51 y fig. 10.52), permite a cualquier administrador de la solución gestionar las credenciales, el rol y los diferentes accesos de cualquier usuario a las aplicaciones generadas para consumidores.

Asimismo, se ofrece la posibilidad de eliminar, mediante el botón homónimo situado al final de la pantalla, el registro asociado a un usuario en términos de credenciales de acceso. La información generada por dicho usuario permanecerá en las diferentes aplicaciones, no viéndose afectada si dicho usuario fuese eliminado de la solución.

## 10.16.- Pantalla de administración de aplicaciones generadas para consumidores.

| Icono | ID | Nombre único | Título                | URL de acceso          | Desplegada en  | Iniciar                 | Acciones                 |
|-------|----|--------------|-----------------------|------------------------|--|-------------------------|--------------------------|
|       | 1  | Test-01      | Agenda Personal       | /Agenda-Personal       | Sistema PC Windows<br>Sistema Synology Disk Station DS415+ | <a href="#">Iniciar</a> | <a href="#">Acciones</a> |
|       | 2  | Test-DOS     | Gestión Bibliotecaria | /Gestion-Bibliotecaria | Sistema Raspberry Pi v1.Rev.B                              | <a href="#">Iniciar</a> | <a href="#">Acciones</a> |

Fig. 10.53.- Pantalla de administración de aplicaciones generadas para consumidores.

La pantalla de administración de aplicaciones generadas para consumidores (véase fig. 10.53) muestra un listado con todas las aplicaciones generadas, permitiendo crear una nueva aplicación o, sobre una aplicación existente: modificarla, modelarla, eliminarla o acceder a ella (iniciarla).



Fig. 10.54.- Opciones para ejecutar una aplicación en función del sistema donde se haya desplegado.

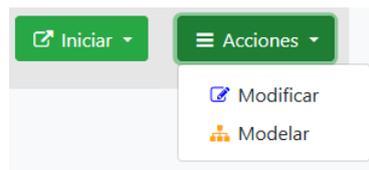


Fig. 10.55.- Acciones para modificar o modelar una aplicación generada para consumidores.

## 10.17.- Pantalla de administración para el modelado de una aplicación generada para consumidores.

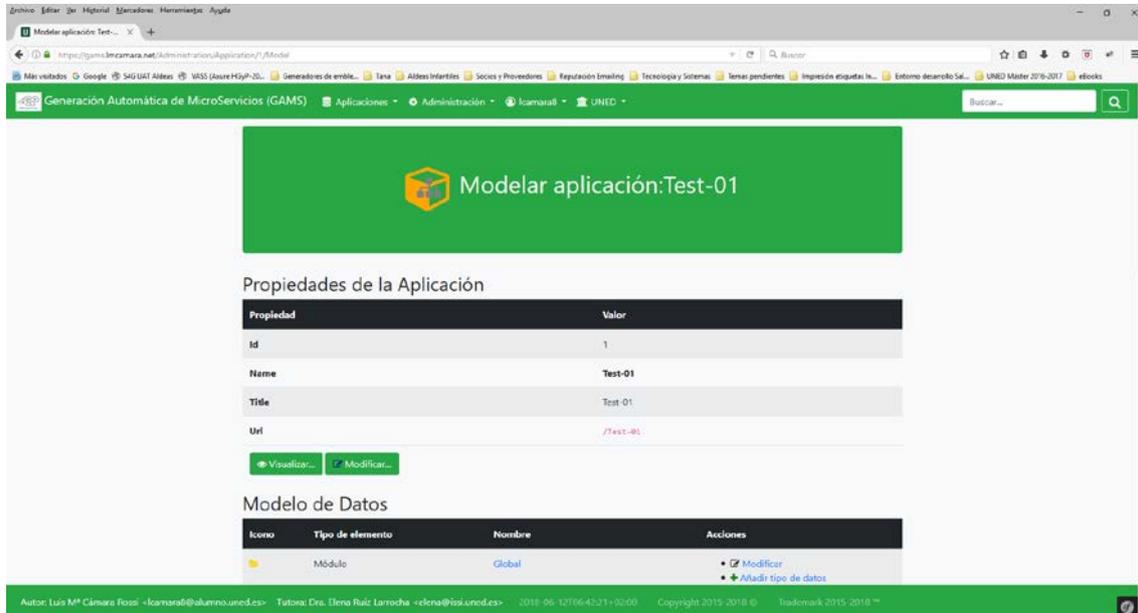


Fig. 10.56.- Pantalla de administración para el modelado de una aplicación, tabla de propiedades.

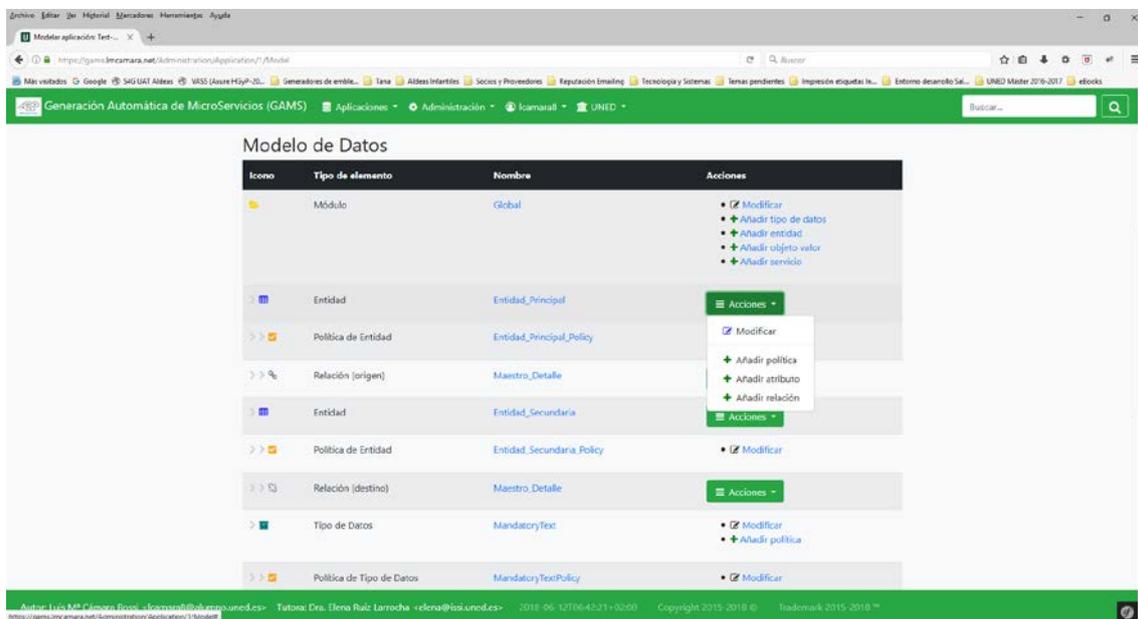


Fig. 10.57.- Pantalla de administración para el modelado de una aplicación, árbol de elementos.

Mediante la pantalla de administración para el modelado de una aplicación (véase fig. 10.57) se visualizan los diferentes elementos del modelo de dominio, pudiendo gestionar los mismo. Remitimos al lector al Anexo A, donde a través de los diferentes casos de uso del Análisis Funcional se detallan todas y cada una de las acciones disponibles.

## 10.18.- Pantalla de administración de idiomas y traducciones.

The screenshot shows a web browser window displaying the 'Idiomas y Traducciones' administration page. The page title is 'Idiomas y Traducciones' and the URL is 'https://gams.lmcamara.net/Administration/Language/'. The page content includes a header with the GAMS logo and navigation menu, and a main section titled 'Listado de idiomas definidos'. This section contains a table with the following data:

| Código ISO-6391 | Nombre                | Modificar |
|-----------------|-----------------------|-----------|
| aa              | Afar                  | Modificar |
| ab              | Abjasio (o Abjasiano) | Modificar |
| ae              | Avéstico              | Modificar |
| af              | Afrikáans             | Modificar |
| ak              | Akano                 | Modificar |
| am              | Amhárico              | Modificar |

The footer of the page contains contact information for the author (Luis Mª Cámara Rossi) and tutor (Dra. Elena Ruiz Larrocha), along with copyright and trademark information for the years 2015-2018.

Fig. 10.58.- Pantalla de administración de idiomas y traducciones.

Desde la “Pantalla de Administración de Idiomas” es posible consultar la lista actual de idiomas registrados en la solución GAMS.



Nótese que, de manera predeterminada, la solución GAMS incorpora todos los idiomas actualmente definidos en el estándar internacional ISO-639-1. Así, por mencionar algunos códigos definidos, indicar que para el castellano dicho estándar ha definido el código “es”, mientras que para el inglés se ha definido el código “en” y para el francés el código “fr”.

A través del botón “Modificar”, situado en cada una de las filas asociadas a cada idioma, es posible acceder a la “Pantalla de Administración de Modificación de Idioma”, donde dicho idioma podrá ser redefinido por cualquier usuario con el rol de “Administrador”, acorde a las necesidades específicas de las aplicaciones a modelar.

Al final de la lista de idiomas, se encuentra el botón “Crear un nuevo idioma...”, gracias al cual se accederá a la “Pantalla de Administración para la Creación de un Nuevo Idioma”.

## 10.19.- Pantalla de Administración de Consulta de un Idioma.

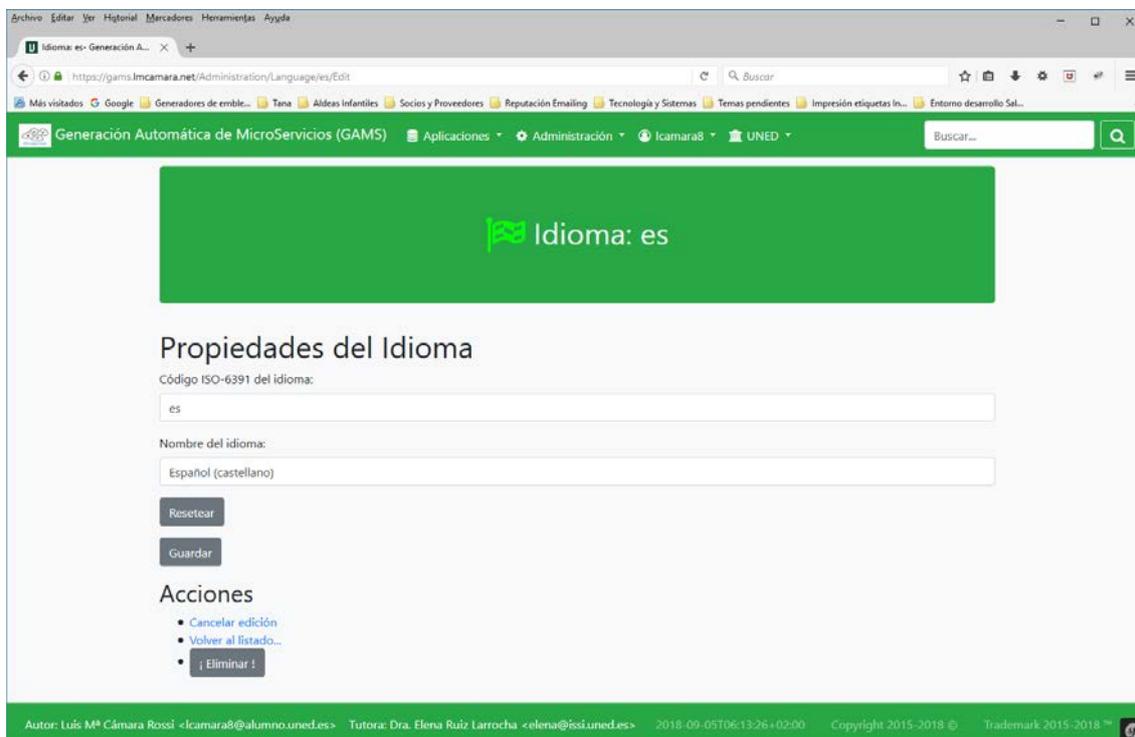


Fig. 10.59.- Pantalla de administración de consulta de un idioma.

La “Pantalla de Administración de Consulta de un Idioma” tiene por objetivo presentar a los usuarios con el rol de “Administrador” la información asociada a un determinado idioma.

En la fig. 10.59 se muestra como ejemplo la definición del idioma “Afar”, cuyo código según el estándar ISO-639-1 es “aa”.

Pulsando sobre el botón “Volver al listado...”, el sistema conducirá al usuario hasta la “Pantalla de Administración de Idiomas”.

Pulsando sobre el botón “Modificar”, el sistema mostrará la “Pantalla de Modificación de un Idioma” para el idioma que estuviese visualizándose; con esta opción será posible actualizar los campos de datos del idioma.

Pulsando sobre el botón “Eliminar”, el sistema procederá a la completa eliminación del idioma que se estuviese visualizando.

## 10.20.- Pantalla de Administración para la Creación de un Nuevo Idioma.

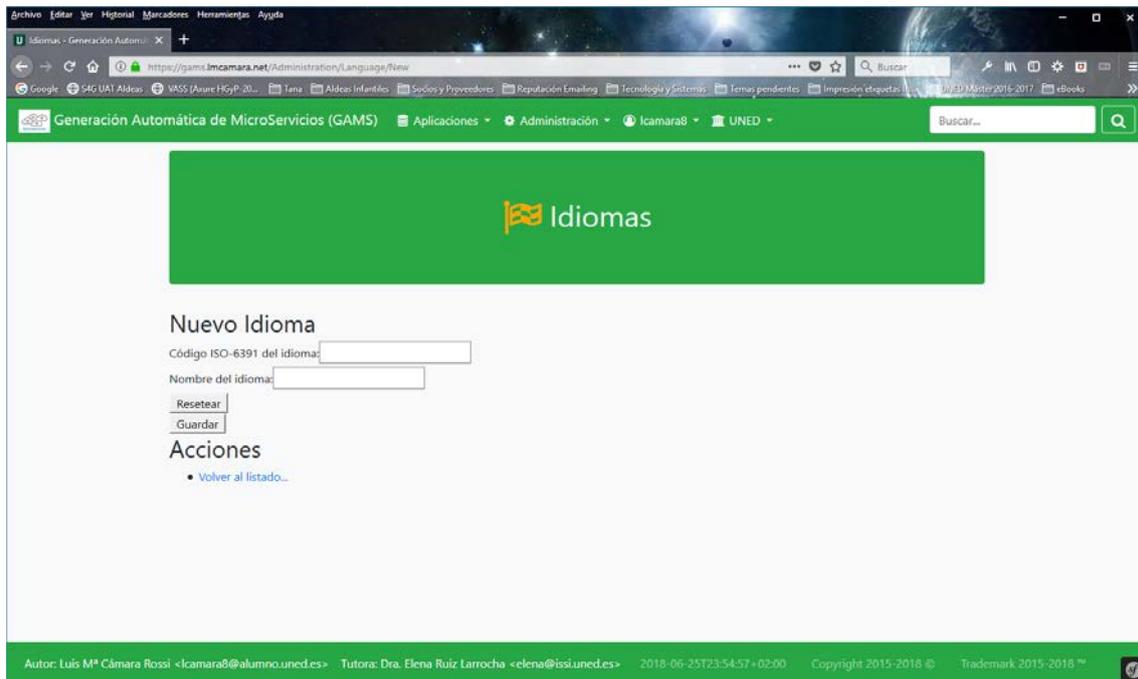


Fig. 10.60.- Pantalla de administración para la creación de un nuevo idioma.

Mediante la “Pantalla de Administración para la Creación de un Nuevo Idioma” (véase fig. 10.60), es posible establecer los valores iniciales para los campos del nuevo idioma que será creado.

Pulsando sobre el botón “Resetear”, los valores de cada campo volverán al estado original, es decir, en blanco (vacíos), en caso de haber sido modificados.

Pulsando sobre el botón “Guardar”, se creará un nuevo idioma y se almacenarán los valores escritos en los correspondientes campos de datos.

Téngase en cuenta que el sistema exige que el valor del campo “Código ISO-639-1” sea único, por lo que avisará mediante un mensaje de error en caso de repetir dicho valor.

## 10.21.- Pantalla de Administración de Modificación de un Idioma.

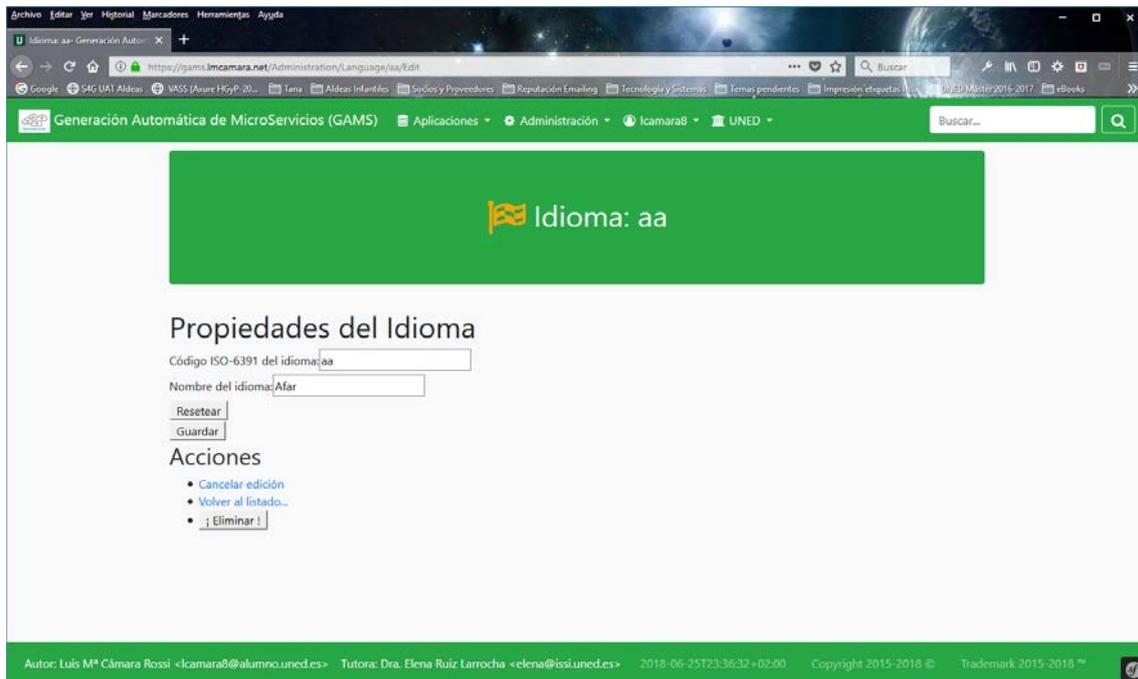


Fig. 10.61.- Pantalla de administración de modificación de idioma.

Mediante la “Pantalla de Administración de Modificación de Idioma” (véase fig. 10.61), la solución ofrece la posibilidad de establecer el código ISO-639-1 y el nombre de un determinado idioma.

Gracias a esta pantalla es posible realizar el mantenimiento de los idiomas, a medida que el estándar evolucione con el tiempo.

Pulsando sobre el botón “Resetear”, los valores de cada campo volverán al estado original, en caso de haber sido modificados.



Pulsando sobre el botón “Guardar”, se almacenarán los cambios realizados en los campos de datos. Téngase en cuenta que el sistema exige que el valor del campo “Código ISO-639-1” sea único, por lo que avisará mediante un mensaje de error en caso de repetir dicho valor.

Pulsando sobre el botón “Cancelar edición”, el sistema mostrará la “Pantalla de Consulta de Idioma” para el idioma que se estuviese editando; con esta opción no se modifica ninguno de los campos de datos del idioma, ni siquiera cuando estos se hubieran visto alterados.

Pulsando sobre el botón “Volver al listado...”, el sistema conducirá al usuario hasta la “Pantalla de Administración de Idiomas”, nuevamente, sin alterar ningún valor del idioma que estuviera en edición. Téngase en cuenta que cualquier modificación de datos se perderá al abandonar la edición de dicho idioma.