

# UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA



MASTER UNIVERSITARIO DE INVESTIGACIÓN EN INGENIERÍA DE  
SOFTWARE Y SISTEMAS INFORMÁTICOS

ITINERARIO DE INGENIERÍA DE SOFTWARE. 31105151

---

## **Diseño e implementación de una intranet basada en la arquitectura de microservicios para ambientes empresariales**

---

*Autor:*  
LUIS ALBERTO  
HERRERA IZQUIERDO

*Director:*  
ELENA  
RUIZ LARROCHA

**SEPTIEMBRE 2018/2019**

MÁSTER UNIVERSITARIO DE INVESTIGACIÓN EN  
INGENIERÍA DE SOFTWARE Y SISTEMAS  
INFORMÁTICOS

ITINERARIO DE INGENIERÍA DE SOFTWARE.  
31105151

**Diseño e implementación de una intranet  
basada en la arquitectura de microservicios  
para ambientes empresariales**

Tipo B

*Autor:*  
LUIS ALBERTO  
HERRERA IZQUIERDO

*Director:*  
ELENA  
RUIZ LARROCHA

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

Fecha: 10/09/2018

Quién suscribe:

Autor(a): Luis Alberto Herrera Izquierdo  
D.N.I: 0803786680

Hace constar que es el autor(a) del trabajo:

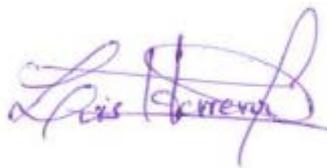
Diseño e implementación de una intranet basada en la arquitectura de microservicios para ambientes empresariales

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.

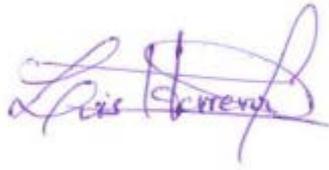
Fdo.



# 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

A handwritten signature in purple ink, appearing to read "Luis [unintelligible]". The signature is stylized and somewhat illegible due to the cursive nature of the handwriting.

## Resumen

En las empresas para la toma de decisiones tener la información adecuada en el momento adecuado puede ser algo de vital importancia que repercute directamente en su factor de éxito o que le permitirá sobresalir frente a las demás empresas, es por esta razón que tener un centro de conocimiento y de reportes en tiempo real se ha vuelto algo necesario en el día a día de las empresas. Por otra parte, el desarrollo de aplicaciones software de manera monolítica ha demostrado que es una manera ineficiente de desarrollo que termina generando mucho código espagueti y las convierte en aplicaciones difícil de mantener, por esta razón ha surgido el paradigma de diseño de microservicios que se ha vuelto popular por su enfoque granular, acoplamiento ligero en la arquitectura y despliegue frecuente de manera independiente, así mismo para las administración de una arquitectura de microservicios resulta imprescindible tener un miembro del equipo con conocimientos sólidos en prácticas DevOps, dentro de las cuales está teniendo una fuerte acogida el uso de pipelines de integración continua (CI) y de entrega continua (CD), esto es porque proporcionan los mecanismos necesarios para ciclos de desarrollo cortos, automatizando varios pasos después de que cualquier miembro del equipo de desarrollo haya emitido un commit en el repositorio del proyecto teniendo como resultado final luego de pasar satisfactoriamente las pruebas automatizadas esta nueva versión del proyecto en producción.

En esta tesis de master se presenta una arquitectura para el desarrollo de un sistema intranet que permitirá compartir de información y obtener reportes en tiempo real, bajo una arquitectura de microservicios utilizando las metodologías de desarrollo ágil: Programación Extrema (XP) y Scrum. El desarrollo de la intranet bajo la metodología XP y Scrum permitió obtener los servicios acordados con lo cliente en el tiempo estimado y además una distribución adecuada del conocimiento técnico en el equipo de desarrollo, así mismo la implementación de la arquitectura bajo el paradigma de microservicios con la utilización de contenedores Docker en modo swarm permitió desarrollar los distintos módulos de manera independiente permitiendo así implementar procesos de CI y CD de manera satisfactoria.

### Palabras Claves.

Intranet, microservicios, contenedores, swarm, control de versiones, programación extrema, scrum, pruebas automatizadas, integración continua, entrega continua, directorio activo, NoSQL, AMQP

# Índice de Contenido

<b>Capítulo 1: Introducción</b> .....	11
Objetivos .....	13
Metodología .....	13
<b>Capítulo 2: Estado del arte</b> .....	15
2.1. Arquitectura orientada a servicios.....	15
2.2. Microservicios.....	16
2.3. Transmisión asíncrona de mensajes .....	17
2.3.1 Flujo.....	18
2.4. Docker .....	19
2.4.1. Compose .....	20
2.4.2. Orquestación de contenedores .....	20
2.5. Control de accesos y seguridades.....	21
2.6. Base de datos NoSQL .....	21
2.7. Control de versiones.....	22
2.7.1. Integraciones continuas (CI).....	23
2.7.2. Entrega continua (CD).....	24
2.8. Metodología de desarrollo ágil.....	24
2.8.1. Scrum .....	25
2.8.2. Programación extrema (XP) .....	26
2.8.3. Desarrollo rápido de aplicaciones (DRA).....	27
2.8.4. Lean .....	28
<b>Capítulo 3: Resultados</b> .....	30
3.1. Análisis de la entrevista.....	30
3.2. Análisis de las encuestas .....	31
3.3. Selección de la metodología de desarrollo.....	31

3.4. Análisis del caso.....	33
3.4.1 Implementación de XP y Scrum.....	33
3.4.1.1. Planeación.....	33
3.4.1.2. Diseño de la arquitectura .....	34
3.4.1.3. Herramientas seleccionadas para el desarrollo del sistema .....	36
3.5 Resolución del problema.....	36
3.5.1. Registro de aplicaciones en Office 365 .....	37
3.5.2. Configuración del AD local.....	38
3.5.3 Configuración del middleware de mensajería.....	40
3.5.3.1. Creación del intercambiador .....	40
3.5.3.2 Creación de las colas.....	41
3.5.4. Fase 1 módulo de evaluación de competencias .....	42
3.5.5. Fase 2 módulo de reportes biométrico.....	45
3.5.6. Fase 3 módulo de reportes de uso de la plataforma aula virtual.....	46
3.5.7. Integración de los módulos a la intranet. ....	48
3.5.7.1. Autenticación en la intranet. ....	49
3.5.7.2. Interfaces de la aplicación intranet. ....	50
3.5.7.2.1 Interfaces para evaluación laboral .....	52
3.5.7.2.2 Interfaces para reportes de marcaciones Biométrico .....	54
3.5.7.2.3 Interfaces para reportes de uso aula virtual.....	55
3.5.8. Implementación de CI y CD .....	56
3.5.8.1 Control de versiones de los proyectos .....	56
3.5.8.2 Pruebas automatizadas .....	57
3.5.8.3 Imágenes multicapas de los proyectos mediante la utilización de Docker.....	57
3.5.8.4 Orquestación de contenedores .....	59
3.5.8.5 Configuración de Pipelines en Jenkins .....	60

3.5.9. Implementación de monitoreo y comunicación.....	62
<b>Capítulo 4: Evaluación de la solución</b> .....	64
<b>Capítulo 5: Conclusiones y trabajos futuro</b> .....	65
<b>Bibliografía</b> .....	66
<b>Anexos</b> .....	71

## Índice de Figuras

Figura 1 Arquitectura no orientada a SOA.....	15
Figura 2 Arquitectura SOA.....	16
Figura 3 Ejemplo de arquitectura de microservicios [18] .....	17
Figura 4 Arquitectura básica de servicio de mensajería asíncrona [20] .....	17
Figura 5 Estándar de flujo de mensajes en RabbitMQ [22][23].....	18
Figura 6 Ejemplo de creación de contenedor [24].....	19
Figura 7 Arquitectura de Docker [27] .....	19
Figura 8 Ejemplo Archivo Docker Compose .....	20
Figura 9 Arquitectura swarm [31] .....	21
Figura 10 Ejemplo de documento almacenado en MongoDB [36].....	22
Figura 11 Ejemplo de proceso de despliegue automatizado [44].....	24
Figura 12 Proceso de desarrollo del departamento de TIC's.....	30
Figura 13 Arquitectura MSA propuesta para desarrollo del proyecto.....	35
Figura 14 Formulario de registro de aplicación en el portal de Microsoft .....	37
Figura 15 Registro de URLs de los distintos ambientes de la aplicación para autenticación.....	38
Figura 16 Ingreso de datos obtenidos del portal de Microsoft en nuestra aplicación ....	38
Figura 17 Vista principal del AD local.....	39
Figura 18 Datos de configuración para el acceso al AD local desde la aplicación .....	39
Figura 19 Configuración para acceso a los grupos del AD local .....	40
Figura 20 Intercambiador del proyecto queue_multiple.....	40
Figura 21 Enlace de colas al intercambiador queue_multiple.....	41
Figura 22 Colas creadas en RabbitMQ para el intercambio de datos entre los microservicios.....	42

Figura 23 Política de códigos organigrama .....	43
Figura 25 Vista de evaluaciones entre los trabajadores.....	44
Figura 26 Interfaz gráfica de la vista vEvaluacionesRealizar .....	44
Figura 27 Respuesta de consumo del microservicio evaluación laboral .....	45
Figura 28 Vista para obtener las marcaciones de todos los trabajadores .....	46
Figura 29 Pruebas de comunicación con el middleware de mensajería para los reportes de biométrico .....	46
Figura 30 Colección Moodle en MongoDB .....	47
Figura 31 Interfaz de inicio de sesión en microservicio Moodle .....	47
Figura 32 Interfaz de lista de documentos en la colección Moodle .....	48
Figura 33 Respuesta del microservicio reportes moodle.....	48
Figura 34 Lista de URLs de los microservicios a los que se conecta la aplicación intranet.....	49
Figura 35 Formulario de aceptación de acceso a la cuenta del usuario.....	49
Figura 36 Control de acceso a las distintas interfaces basado en el rol del directorio activo local.....	50
Figura 37 Interfaz de la aplicación intranet para iniciar sesión.....	51
Figura 38 Interfaz principal de la aplicación intranet para usuarios autenticados.....	51
Figura 39 Interfaz para selección del tipo de evaluación a realizar.....	52
Figura 40 Interfaz de selección de la persona a evaluar .....	53
Figura 41 Formulario de evaluación.....	53
Figura 42 Reporte de resultados de evaluaciones.....	54
Figura 43 Reporte de las marcaciones de un trabajador en el sistema biométrico.....	55
Figura 44 Reporte total de horas y días laborados durante el mes .....	55
Figura 45 Interfaz de reporte de uso del aula virtual.....	56
Figura 46 Repositorios creados en Bitbucket.....	57
Figura 47 Resultado de ejecución de pruebas automatizadas en el proyecto intranet....	57
Figura 48 Imágenes Docker multicapas .....	58
Figura 49 Archivo Dockerfile del proyecto intranet .....	58
Figura 50 Imágenes del proyecto alojadas en la nube .....	59
Figura 51 Definición del servicio mssql con la red micro_net y volumen compartido..	60
Figura 52 Lista de servicios corriendo .....	60
Figura 53 Pipeline build-image .....	61
Figura 54 Pipeline micro-intranet.....	61

Figura 55 Resumen de estado de nuestros pipelines .....	62
Figura 56 Dashboard principal de nuestro sistema de monitoreo Grafana.....	62
Figura 57 Alertas enviadas al canal de comunicación general por Alert Manager .....	63
Figura 58 Reporte de consumo de CPU por el microservicio mssql.....	63

## Índice de Tablas

Tabla 1 Diferencia entre metodologías ágiles y tradicionales [53] .....	25
Tabla 2 Comparación de métodos ágiles basados en parámetros importantes [15] .....	32
Tabla 3 Comparación entre los métodos más ágiles en base a los criterios definidos [16] .....	32
Tabla 4 Tabla de valoración para la evaluación de desempeño .....	43
Tabla 5 Evaluación de la solución implementada para el desarrollo del proyecto.....	64

# Capítulo 1: Introducción

En las empresas modernas ofrecer las herramientas necesarias para la toma de decisiones en tiempo real se ha vuelto algo crucial, es aquí donde la arquitectura de intranet ha tomado fuerza puesto que son sistemas de información que ayudan a adquirir, recordar, distribuir e interpretar información y conocimiento en las empresas [1]. Especialmente en grandes empresas distribuidas, la aplicación de intranet es beneficiosa [2], debido a que un sitio web de intranet está disponible para los usuarios finales en cualquier momento, es un recurso ideal para diseminar información sobre cómo proteger los datos confidenciales de las empresas y cómo obtener acceso autorizado a la información que está protegida [3]. En una arquitectura de intranet la administración de la seguridad es un aspecto clave puesto que se maneja información confidencial a la cual solo deberán acceder los usuarios con los respectivos privilegios. En efecto un sistema de administración de seguridad para intranet debería contar con las siguientes funciones: seguridad de datos mejorada, respaldo de archivos en tiempo real, control flexible del permiso de archivos, control de documentos distribuidos, seguridad en el uso fuera de línea, administración perfecta de registros y alta compatibilidad [4].

Otro aspecto clave en una intranet es la identidad del usuario, esta información es valiosa en esta era digital, por esta razón la identidad digital de la persona debe ser confiable todo el tiempo [5]. Implementar un controlador de dominio Active Directory o Directorio Activo (AD) facilita enormemente las tareas de la gestión de identidades de usuarios, grupos y de supervisar toda la información relacionada en una empresa cuando un usuario de la intranet trata de asignar o adquirir diversos recursos y servicios [6]. En la integración del AD con aplicaciones, la autenticación es el proceso mediante el cual los usuarios finales se identifican a sí mismos en el AD mediante la utilización de protocolos específicos y posteriormente se les brinda acceso personalizados dentro de la aplicación en función del rol que desempeñan en la empresa [5]. El motivo principal de la integración de las aplicaciones con AD es centralizar el proceso de autenticación cuya funcionalidad se denomina Single Sign-On (SSO) de esta manera los usuarios sólo manejan una cuenta única para acceder a todos los sistemas e inclusive al equipo informático asignado dentro de la empresa y así se evita la fatiga de tener que ingresar varias veces las mismas credenciales en los diferentes sistemas.

Para las empresas el rápido proceso de digitalización [7] exige flexibilidad para adaptarse a los requisitos empresariales que cambian rápidamente y a las nuevas oportunidades comerciales emergentes que se basan en el aprovechamiento de las nuevas tecnologías. Para hacer frente a estas exigencias de la era moderna muchas empresas de desarrollo de software han optado por la selección de la arquitectura basada en microservicios (MSA) [8]. El enfoque hacia MSA se basa en capacidades empresariales separadas y se puede construir utilizando diferentes lenguajes de programación con diferentes tecnologías de almacenamiento de datos [9]. Este enfoque para el desarrollo de aplicaciones desafía la aplicación y los servicios monolíticos tradicionales y también

vienen acompañado con la necesidad de una fuerte cultura de DevOps para manejar el mayor nivel de distribución y la frecuencia de implementación posible [10] haciendo uso de la tecnología de contenedores como Docker. Sin embargo para que una MSA se pueda implementar de manera correcta se debe cumplir con ciertas características, una de las características principales es que debe ser de implementación independiente con respecto a los demás microservicios esto quiere decir que debe tener su propia arquitectura de implementación evitando compartir sus recursos como: contenedores, cachés y almacenes de datos con otros componentes de software empresariales [11].

La disciplina que se encarga de determinar, analizar, describir y establecer las diferentes metodologías y lineamientos para mejorar u optimizar el desarrollo de sistemas informáticos se la denomina Ingeniería de Software [12]. En la ingeniería de software para el desarrollo de proyectos software se utilizan metodologías que facilitan las tareas de desarrollo, algunos autores definen una metodología como una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información. Una metodología está formada por fases, cada una de las cuales se puede dividir en sub-fases, que guiarán a los desarrolladores de sistemas a elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo. Sin embargo, una metodología es algo más que una colección de pasos, puesto que se basa en una filosofía, distinguiéndose de los métodos o de las simples recetas, que marcan unos pasos a seguir de manera secuencial para obtener un producto final. Así, las metodologías difieren ya sea por la cantidad de fases, las técnicas de cada fase, el contenido de la fase o en su base filosófica, todo esto se aplica, dependiendo del contexto de desarrollo, tamaño del proyecto o del equipo de trabajo, cultura organizacional, entre otros aspectos [13].

Las empresas buscan constantemente formas y mecanismos de automatizar tareas repetitivas para agilizar procesos que se desarrollan a diario en el ámbito laboral mediante la utilización de tecnologías que permiten desarrollar sistemas en periodos cortos de tiempo mediante la utilización de metodologías ágiles, a pesar de los grandes esfuerzos que se llevan adelante para lograr este objetivo hay áreas que no se logran cubrir o automatizar por completo. En la Pontificia Universidad Católica del Ecuador Sede Esmeraldas (PUCESE) se han venido desarrollando sistemas que han permitido agilizar varios procesos internos tanto a nivel académico como administrativo, sin embargo, para el desarrollo no se han usado metodologías que permitan establecer un tiempo estimado de duración, asegurar la calidad del software y disponer de una documentación adecuada. Debido a la constante necesidad de sistematizar procesos en la PUCESE se requiere que en el Departamento de TIC's se apliquen procesos guiados a través de una metodología, de tal forma que faciliten el desarrollo de software, y con ello cumplir con los requerimientos de la institución, obteniendo así mejores productos de software. Actualmente frente a la expansión que está teniendo la PUCESE con la apertura de nuevas carreras y un nuevo campus, resulta necesario contar con un sistema de información gerencial que permita tener información en tiempo real de la situación de la empresa y de

esta manera brindar un soporte técnico para facilitar la toma de decisiones. En efecto para resolver esta problemática se plantea implementar una metodología ágil para el desarrollo de proyectos software y evaluar su implementación mediante el desarrollar un sistema intranet basado en MSA en el cual se va a proveer a los usuarios dependiendo del rol que posean dentro de la empresa reportes en tiempo real de la situación de sus respectivos departamentos o dirección. En este proyecto software la utilización de nuevas tecnologías es un factor clave para asegurar su evolución y escalabilidad en futuras versiones con la implementación de más funcionalidades o características.

## **Objetivos**

En base a lo expuesto anteriormente el objetivo principal de este trabajo de fin de máster consiste en: Establecer una metodología para el desarrollo de proyectos software y evaluarlo mediante el desarrollo de un sistema intranet basado en MSA para la obtención de información gerencial en la PUCESE.

Los objetivos secundarios son:

- Analizar la situación actual de la forma de desarrollo de proyectos software en la PUCESE.
- Evaluar las distintas metodologías ágiles disponibles para el desarrollo de proyectos software y establecer la que más se ajuste a los procesos de automatización que se requieren en la PUCESE.
- Desarrollar un sistema intranet basado en MSA siguiendo las pautas de la metodología ágil seleccionada para medir la factibilidad de implementación de la metodología.
- Definir los mecanismos de seguridad para el acceso a la intranet.
- Identificar los reportes necesarios que deberán estar disponibles en el sistema de información gerencial.
- Implementar procesos de integraciones continuas (CI) y despliegues continuos (CD) para el desarrollo del sistema.
- Establecer un protocolo de comunicación y monitoreo entre los microservicios que permita conocer el estado de los mismos en tiempo real.

## **Metodología**

Para el desarrollo del proyecto se realizó una investigación de tipo transversal cuali-cuantitativa la cual nos permitió conocer la situación actual del Departamento de TIC's, puesto que este tipo de investigación se basa en la recolección de datos durante una cantidad de tiempo limitada y además suele ser más descriptiva que experimental. Estos tipos de estudios son útiles para describir un efecto particular en una población, en un momento determinado del tiempo [14].

Para conocer la situación actual de la forma de desarrollo de proyectos software que se utiliza y los procesos de automatización que se requieren en el Departamento de TIC's de la PUCESE se procedió a realizar una entrevista (Ver anexo 1) al jefe de este departamento con la finalidad de obtener datos relevantes que contribuyan a la investigación para la identificación del problema. Así mismo se realizaron encuestas (Ver anexo 2) a cada uno de los miembros del equipo de desarrollo, la encuesta fue diseñada en base a un análisis crítico del problema planteado y tomando en cuenta las diferentes características de la población objeto de estudio. Los datos generados por las diferentes técnicas de investigación utilizadas, fueron procesados haciendo uso de la herramienta de cálculo numérico Microsoft Excel, con el cual se generaron las diferentes tablas y gráficos, los mismos que permitieron realizar los análisis respectivos.

Luego de los resultados obtenidos producto de la aplicación de las técnicas de investigación antes mencionadas se procedió a evaluar las distintas metodologías ágiles disponibles para el desarrollo de proyectos software y de esta manera identificar la que más se ajuste a los procesos de automatización que se requieren en la PUCESE, para ello se realizó una búsqueda exhaustiva de fuentes de información en: artículos científicos, libros y revistas relacionadas al área de estudio. Los criterios para la evaluación de la metodología de desarrollo se aplicaron en base a las investigaciones realizadas por [15] y [16], estos criterios evaluación fueron los siguientes:

- Cantidad de personas mínimas para conforman el equipo
- Documentación clara
- Tiempo de entrega de proyectos
- Estimación de tiempo y costes
- Gestión de los cambios en los requisitos
- Capacidad de equilibrar las necesidades del negocio con las necesidades de los clientes

Una vez que se definió la metodología ágil de desarrollo que se va a utilizar se procedió a identificar los módulos a desarrollar, la arquitectura MSA y a definir cuáles serían las tecnologías factibles para establecer los mecanismos de seguridad, comunicación, monitoreo, integración y despliegue del software que se van a utilizar para el desarrollo del proyecto y de esta manera evaluar la metodología ágil aplicada.

## Capítulo 2: Estado del arte

### 2.1. Arquitectura orientada a servicios

Un servicio es una funcionalidad de alto nivel reutilizable a la que se puede acceder mediante la red, la misma que describe lo que se puede hacer y de que forma el cliente puede interactuar con ella. Un servicio desde el punto de vista de las empresas representa una tarea u operación específica, cómo puede ser la obtención de información de un producto, al cumplir con una operación determinada se puede reutilizar el servicio en conjunto con otros servicios para la realización de operaciones complejas que requiera la interacción entre ellos de manera armónica, y de esta manera cumplir con los objetivos del negocio o empresa.

No hay una definición estándar para la arquitectura orientada a servicios (SOA), hay varias definiciones de este estilo de arquitectura, una de ellas es la que publica IBM [17] afirmando que SOA es “*Un conjunto de principios, modelos y criterios arquitectónicos que abordan características como modularidad, encapsulación, acoplamiento abierto, separación de elementos de interés y reutilización.*” basados en esta definición SOA es un estilo de arquitectura que busca el desarrollo de piezas de software en forma de componentes los mismos que ofrecen sus funcionalidades desde la red en forma de servicio y son consumidos o ejecutados mediante los protocolos web estándares (como HTTP, HTTPS). Es un estilo que hizo su aparición a inicios del 2000 y provocó un gran impacto en la forma en la que se desarrollaba software puesto que permite descomponer aplicaciones monolíticas en un conjunto de servicios es decir desarrollar funcionalidades en forma modular las mismas que se pueden reutilizar para otras operaciones, en la figura 1 y 2 se puede observar la diferencia entre una arquitectura monolítica y una arquitectura SOA. El objetivo de SOA es separar la lógica de integración de negocio de la implementación, de esta manera el desarrollador se encarga de crear las funcionalidades del sistema en forma de módulos mientras que el integrador de aplicaciones solo se encarga de ensamblar estos módulos.

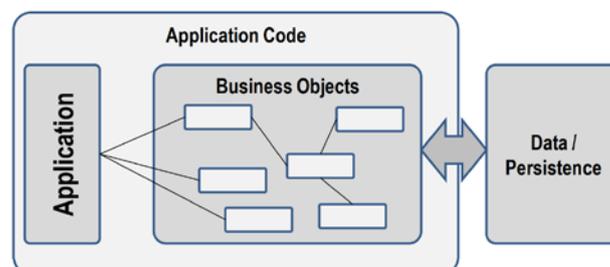
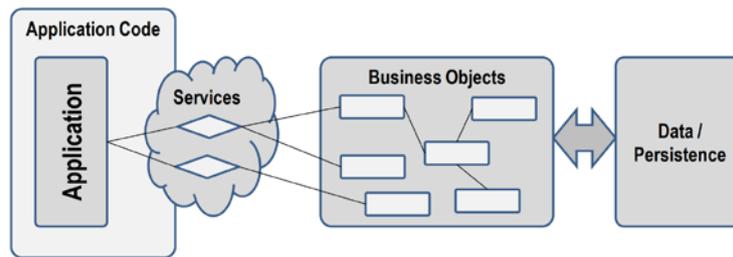


Figura 1 Arquitectura no orientada a SOA



*Figura 2 Arquitectura SOA*

En el contexto en el que se va a desarrollar el proyecto los sistemas actuales poseen una arquitectura monolítica, con lo cual resulta complejo agregar nuevas funcionalidades, la migración y así mismo la escalabilidad.

## **2.2. Microservicios**

El término microservicios ha sido usado en varias formas por varios años, aunque no es una tecnología nueva puesto que tiene que ver mucho con SOA, las definiciones que se le han atribuido están relacionadas a la forma de construir software luego de la reunión de un grupo de arquitectos de software por el año 2011, entre ellos James Lewis y Martin Fowler quienes acuñaron el término. Según [11] el enfoque de microservicios consisten en muchos servicios desacoplados independientes entre sí. Cada uno de estos servicios realiza una tarea específica que se desarrolla, despliega y escala de forma independiente.

Cuando se trata de MSA se refiere a aplicaciones distribuidas en la cual todos sus componentes son microservicios [18], pero su comportamiento distintivo deriva de la composición y coordinación de sus componentes a través de mensajes como se puede apreciar en la figura 3. Con MSA las aplicaciones individuales son desarrolladas para realizar validaciones de tarjetas de créditos de un cliente y realizar una compra, por ejemplo, estas aplicaciones luego son agrupadas y como resultados se obtendría un sistema de compras [11]. De la misma forma si luego la empresa requiere hacer un sistema para pago de servicios básicos vía internet estos módulos o funcionalidades desarrollados previamente se pueden reutilizar para nuevos sistemas sin ningún problema, permitiendo a la empresa desarrollar grandes sistemas de software en periodos cortos de tiempo.

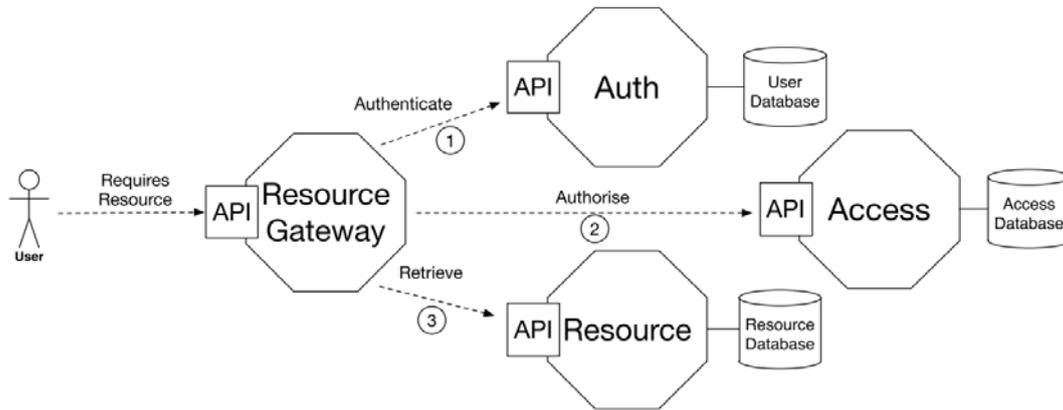


Figura 3 Ejemplo de arquitectura de microservicios [18]

El desarrollo con enfoque en microservicios está relacionado con el uso de tecnologías que facilitan su implementación, despliegue y administración como lo son: transmisión asíncrona de mensajes, contenedores, control de versiones o git, integraciones con despliegues continuos y bases de datos no relacionales como mongodb, más adelante en los siguientes apartados se hablara de cada una de ellas y como se usan en este proyecto.

### 2.3. Transmisión asíncrona de mensajes

En una MSA hay que establecer comunicación segura entre los distintos componentes que la conforman y asegurar que la información no se pierda o pueda ser interceptada en el trayecto, por esta razón la transmisión de mensajes de manera asíncrona tiene un papel importante para mantener las comunicado todos los componentes en este tipo de arquitecturas. Tradicionalmente, los negociadores de mensajes como RabbitMQ resuelven esta problemática mediante la utilización del protocolo AMQP que es la abreviatura de Advanced Message Queuing Protocol, AMQP crea la interoperabilidad entre Productor, Negociador de mensajes y Consumidor. La interacción entre los distintos elementos antes mencionados es la siguiente; Los productores usan una API para publicar un evento al negociador de mensajes. El negociador de mensajes maneja las suscripciones, lo que permite que los consumidores estén informados cuando llega un evento [19] en la Figura 4 se puede ver un ejemplo de la interacción entre estos tres componentes.



Figura 4 Arquitectura básica de servicio de mensajería asíncrona [20]

En vista de que la información es un bien muy valioso para cualquier institución o empresa, la utilización de RabbitMQ en el proyecto va a garantizar que la información

transmitida entre los distintos componentes llegue a su destino de forma segura, aun cuando ciertos componentes queden fuera de servicio, la información transmitida mediante RabbitMQ siempre se almacena en una pila y es enviada a su destino cuando el servicio se restablezca lo cual incrementa la confiabilidad en el sistema.

### 2.3.1 Flujo

La idea central en el modelo de mensajería de RabbitMQ es que el productor nunca envía ningún mensaje directamente a una cola en su lugar lo envía a un intercambiador que actúa como enrutador del mensaje hacia las colas, muy a menudo el productor ni siquiera sabe si un mensaje será entregado a cualquier cola en absoluto [21]. En RabbitMQ, hay cuatro tipos diferentes de intercambio (direct, fanout, topic y headers) que direccionan el mensaje de forma diferente utilizando diferentes parámetros y configuraciones de enlaces. Los clientes pueden crear sus propios intercambiadores o utilizar los intercambiadores predefinidos, a continuación, se describe de manera detallada el flujo de transmisión de mensajes seguido en RabbitMQ.

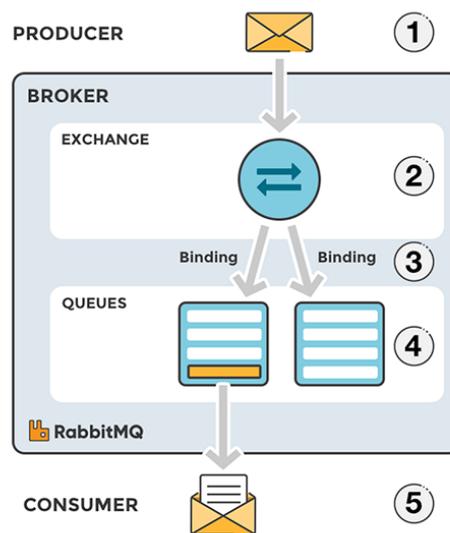


Figura 5 Estándar de flujo de mensajes en RabbitMQ [22][23]

El flujo del mensaje como se describe en la figura 5 está compuesto por cinco pasos según [22] y [23] los cuales se describen a continuación.

1. El productor publica un mensaje para el intercambio.
2. El negociador/mediador recibe el mensaje y ahora es responsable del enrutamiento del mensaje.
3. Se debe establecer un enlace entre la cola y el intercambio. En este caso, tenemos enlaces a dos colas diferentes del negociador. El intercambiador enruta el mensaje a las colas.
4. Los mensajes permanecen en la cola hasta que son manejados por un consumidor.
5. El consumidor maneja el mensaje.

## 2.4. Docker

La tecnología de contenedores docker es una plataforma que permite a los desarrolladores construir, asegurar y administrar una gran gama de aplicaciones, desde el desarrollo hasta la puesta en producción ya sea en instalaciones locales como en la nube [24], contiene las bibliotecas y software necesarios para la ejecución de servicios, además del código fuente de la aplicación para su portabilidad, en la Figura 6 se muestra un ejemplo de la creación de un contenedor.

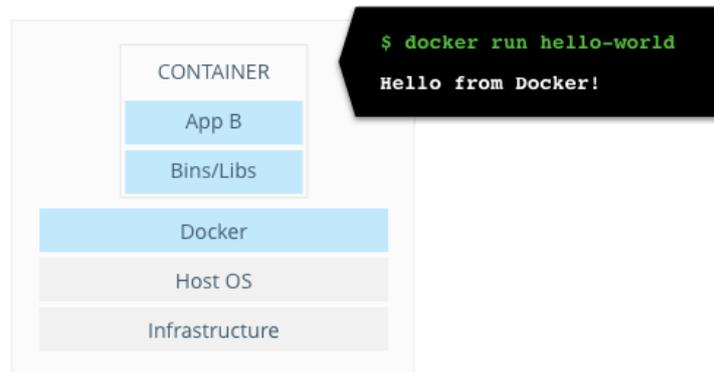


Figura 6 Ejemplo de creación de contenedor [24]

A diferencia de las máquinas virtuales que necesitan de un hypervisor para su correcto funcionamiento docker comparte los recursos con el sistema operativo host, lo cual lo hace más ligero [25][26] y adicionalmente cuenta con un servicio de registro de imágenes gratuito denominado Docker Hub que permite subir o descargar imágenes de contenedores preconfigurados desde la nube. Por su diseño liviano y portabilidad permiten acelerar el desarrollo, despliegue y la revisión de una gran cantidad de contenedores como una sola aplicación lo cual lo hace ideal para el desarrollo de aplicaciones de microservicios en la Figura 7 se muestra más detalles sobre la arquitectura de docker.

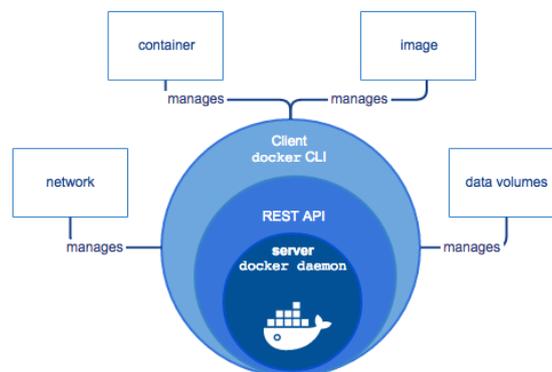


Figura 7 Arquitectura de Docker [27]

Con la utilización de Docker en el proyecto será más sencillo construir con sus respectivas dependencias los distintos microservicios, esto también permite que el

proyecto sea portable, y además agiliza los procesos de despliegue ya que cada microservicio requiere un despliegue o puesta en producción de manera independiente.

### 2.4.1. Compose

La utilización de la tecnología de contenedores tiene un efecto en la arquitectura de los servicios, de la misma manera en la estructura de los ambientes que se utilizan para generarlos y distribuirlos. Esto es debido a que el desarrollador tiene la libertad de elegir que aplicaciones necesita y la forma en la que esta se comunican, lo cual provoca un entorno complejo de desarrollo [28]. Para solucionar esta problemática Docker desarrolló una herramienta denominada Docker Compose la cual permite definir o configurar y ejecutar múltiples contenedores mediante el uso de archivos YAML ya sea en ambientes de desarrollo, pruebas o producción. De esta manera el desarrollo de software se vuelve más sencillo y efectivo. En la siguiente Figura se muestra un ejemplo de archivo YAML para levantar un ambiente de desarrollo con PHP y MySQL basado en contenedores.

```
1 version: '2'|
2 services:
3   db_mysql:
4     build: ./mysql
5     restart: always
6     volumes:
7       - db_data:/var/lib/mysql
8     environment:
9       MYSQL_ROOT_PASSWORD: root-pass
10  php:
11    build: ./php
12    ports:
13      - '90:80'
14    restart: always
15    volumes:
16      - /var/www/html/web/:/var/www/html
17    depends_on:
18      - db_mysql
19  volumes:
20    db_data:
```

Figura 8 Ejemplo Archivo Docker Compose

Docker Compose tiene un rol importante en el proyecto puesto que permitirá agilizar los procesos de despliegue y además facilita la escalabilidad a medida que se vayan agregando más funcionalidades.

### 2.4.2. Orquestación de contenedores

Cuando se trabaja con una MSA se necesitan administrar sistemas distribuidos y a su vez recursos de informáticos, Docker en modo swarm permite realizar estas operaciones uniendo varios equipos que tengan ejecutando Docker a un cluster, de esta manera se puede ejecutar o lanzar contenedores de manera centralizada [29], como se puede apreciar en la Figura 9. Docker en modo swarm ofrece varias características importantes, como disponibilidad, confiabilidad, seguridad, facilidad de mantenimiento y escalabilidad, lo que es una ventaja adicional para la tecnología de contenedor estándar y hace que el sistema basado en contenedores sea un sistema confiable [30]

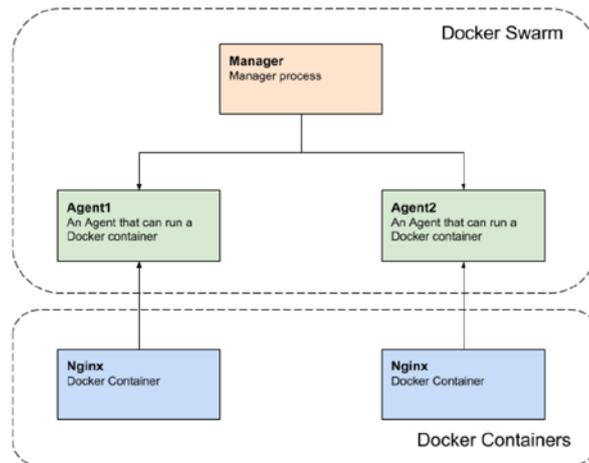


Figura 9 Arquitectura swarm [31]

## 2.5. Control de accesos y seguridades

Las empresas que operan en entornos distribuidos tienen la necesidad de administrar los recursos y servicios de la red, además en una aplicación de intranet garantizar la seguridad y el acceso es algo crítico. Con la implementación de un Active Directory Domain Service (AD) desarrollado por Microsoft se puede resolver esta problemática puesto que proporciona una ubicación centralizada para almacenar información en un entorno distribuido sobre dispositivos, servicios en red y las personas que los usan, además implementa los servicios y mecanismos de seguridad que hacen que esta información esté disponible para usuarios, computadoras y aplicaciones [32]

En nuestro proyecto para el proceso de autenticación y acceso a las diferentes opciones de la intranet se va utilizar el protocolo ligero de acceso a directorios (LDAP) en la comunicación con el AD ya que le brinda al usuario métodos que le permiten conectarse, desconectarse y buscar información (grupos asignados). Además, permite consolidar información para toda una organización dentro de un repositorio central [33], de esta manera la gestión de usuario y asignación de credenciales se va a realizar de manera centralizada, por lo tanto, el acceso de las distintas opciones de la intranet dependerá del grupo al cual pertenece el usuario (docentes, directivos, administradores, entre otros).

## 2.6. Base de datos NoSQL

Otro aspecto importante con la implementación de microservicios está relacionado con la incorporación de dependencias, en este caso la persistencia de datos, cada microservicio debe estar en la capacidad de administrar e integrar su base de datos, almacén, clave valor, índices de búsqueda, etc. A pesar de que las bases de datos relacionales actuales admiten replicación y escalado, las soluciones similares aún no

proporcionan la accesibilidad y la coherencia adecuadas de los datos. En estas situaciones [34] sostiene que utilizar las denominadas bases de datos NoSQL es la más adecuado .

Para el sistema de información gerencial se requiere procesar información documental y que además se pueda cambiar con facilidad los esquemas conforme la aplicación y la empresa vayan evolucionando, por esta razón se va a utilizar MongoDB el cual es una base de datos orientada a documentos que representa almacenamientos de datos estructurados jerárquicamente y que brinda escalabilidad, alto rendimiento y gran disponibilidad [35]. De hecho, se trata de una base de datos basada en el estándar JSON lo cual la hace adecuada para su uso en aplicaciones web, en la siguiente figura se muestra un ejemplo de un documento almacenado en MongoDB.

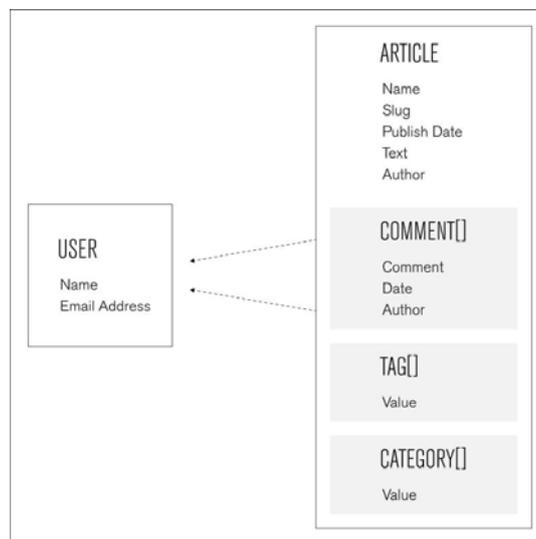


Figura 10 Ejemplo de documento almacenado en MongoDB [36]

Según [37] algunas características importantes de MongoDB son:

- Admite tipos de datos complejos como BSON
- Tiene un lenguaje de consulta muy potente ya que es una base de datos indexable y consultable.
- Se puede acceder a datos masivos a gran velocidad. Se puede acceder a datos que superan los 50 GB 10 veces más rápido que el MySQL ordinario. Debido a esta velocidad, los proyectos con datos crecientes se están cambiando hacia MongoDB.

## 2.7. Control de versiones

Hace algún tiempo para tener un histórico de versiones sobre los cambios realizados ya sea en documentos o proyectos software el método común utilizado era copiar los archivos a otro directorio y utilizar un mecanismo para identificar los cambios que puede ser por fechas o número de versión. Este mecanismo está altamente propenso a errores, puesto que resulta complicado identificar los cambios y cual es realmente la

versión más reciente. Para hacer frente a esta problemática está disponible de forma gratuita el sistema de control versiones comúnmente conocido como git el cual registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo, de modo que se pueda recuperar versiones específicas más adelante [38]. Una de las principales características de git según [39] es que los proyectos se comparten como un todo, por lo que los proyectos complejos con múltiples archivos que se vinculan entre sí son fáciles de compartir. Compartir proyectos con otros desarrolladores está integrado en git, y puede ser facilitado con servicios en línea como GitHub: una vez que un proyecto usa git, se puede copiar fácilmente en GitHub, desde donde otros pueden descargar fácilmente todo el proyecto en sus computadoras locales. El flujo de trabajo de Git es de la siguiente manera [38]:

- Modificas una serie de archivos en tu directorio de trabajo.
- Preparas los archivos, añadiéndolos a tu área de preparación (stage).
- Confirmas los cambios (commit), lo que toma los archivos en su último estado que se encuentran en el área de preparación, y almacena esos cambios de manera permanente en tu directorio de git.

### **2.7.1. Integraciones continuas (CI)**

La metodología ágil XP fue la primera en promover las integraciones continuas [40], apoyando la necesidad de confirmar y probar pequeños cambios en lugar de grandes, para que de esta manera se pueda reducir el riesgo de errores y facilitar la depuración al mismo tiempo.

El control de versiones es el punto de partida de la integración continua (CI) la cual fue un primer paso que aceleró significativamente el ciclo de vida de un producto, empujando a los desarrolladores a comprometerse o integrarse con mayor frecuencia a un repositorio compartido, lo mismo que provocó pruebas unitarias automatizadas después de cada commit o actualización del software; como consecuencia directa, esto ayudó a detectar problemas inmediatamente después de una actualización con errores [41]. [42] Sostiene que CI es un proceso clave para garantizar la calidad del servicio puesto que incluye control de código fuente o versiones, análisis de calidad, pruebas unitarias y de integración del código fuente (automatización), creación de servicio e implementación y así mismo [43] lo define como una práctica de desarrollo que requiere que los desarrolladores integren código en un repositorio compartido varias veces al día. Luego, cada registro se verifica mediante una compilación automática, lo que permite que los equipos detecten problemas con anticipación.

Según [44] los principales beneficios de CI son los siguientes:

- Mejora la productividad de desarrollo
- Permite encontrar y arreglar errores con mayor rapidez
- Entrega las actualizaciones con mayor rapidez

## 2.7.2. Entrega continua (CD)

La entrega continua (CD) va de la mano con CI, puesto que inicia luego de que se ha realizado una nueva integración del software pasando de manera correcta todas las pruebas automatizadas, esta técnica usa los mecanismos necesarios para que de forma segura todos los cambios pasen a los distintos entornos ya sea de: aseguramiento de calidad, listo para producción y producción sin afectar el rendimiento ni la calidad del sistema. En efecto [45] sostiene que CD se centra en maximizar la automatización, incluida la infraestructura, como el código, la gestión del entorno y la automatización del despliegue para garantizar que el sistema esté siempre listo para la producción. De acuerdo a [46] y [47] los beneficios de CD son:

- Acelera el tiempo hacia la comercialización del producto
- Lanzamientos de nuevas versiones de bajo riesgo o propenso a fallas
- Entrega de productos con alta calidad
- Aseguramiento de la construcción del producto correcto.
- Aseguramiento de que el producto cumple con las expectativas del cliente.

En la siguiente figura se muestra un ejemplo del proceso de despliegue automatizado de un producto utilizando las técnicas antes descritas.

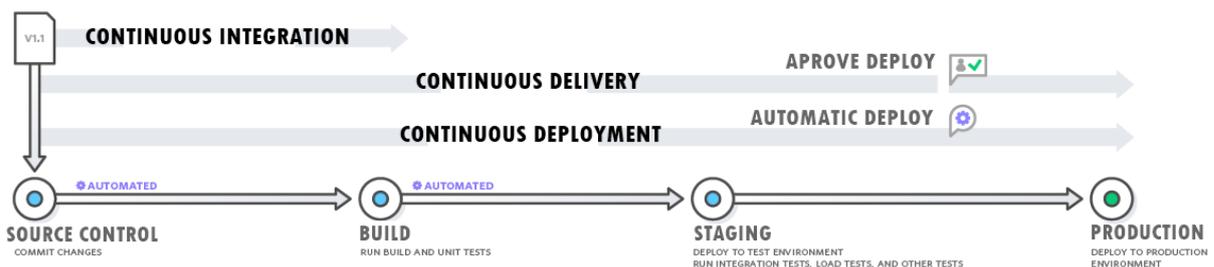


Figura 11 Ejemplo de proceso de despliegue automatizado [44]

Para llevar a cabo las tareas de CI y CD actualmente existen varias herramientas que permiten automatizar estas tareas como son: Jenkins, GoCD, TravisCI, CircleCI entre otra. Una de las herramientas más populares usada para las tareas de CI y CD actualmente es Jenkins, se ha ganado su popularidad por su facilidad y adaptación a las necesidades de los proyectos por su amplio soporte mediante la instalación de complementos. En Jenkins para la automatización de las tareas de CI y CD es común el uso de pipelines, en los cuales se definen los pasos a seguir mediante el uso de stages y además nos muestra de manera intuitiva el estado actual de cada tarea. Otro aspecto importante a tomar en cuenta para la utilización de Jenkins es que es una herramienta de código abierto, con lo cual no hace falta el pago de licencias adicionales que puedan incrementar los costos del proyecto.

## 2.8. Metodología de desarrollo ágil

En el área del desarrollo de software los arquitectos e ingenieros de software

constantemente buscan nuevas o mejores metodologías que permitan agilizar el desarrollo de productos y que a su vez estén lo suficientemente documentadas [13], para garantizar que el proceso de desarrollo cumpla sus objetivos. El desarrollo ágil de software representa una alternativa para las metodologías tradicionales de desarrollo como se puede apreciar en la tabla 1, esto es porque pone menos énfasis en el control previo y estricto para darle más prioridad a la colaboración, la coordinación y el aprendizaje [48]. El desarrollo de software de manera ágil se caracteriza por la entrega constante de software que funciona en periodos cortos de tiempo, cambiando la manera tradicional que entrega nuevas versiones del proyecto software al cliente en meses a semanas, por lo que resulta importante descomponer las distintas funcionalidades del sistema en pequeñas partes para luego priorizarlas junto al cliente, de tal forma que se pueda desarrollar y entregar valor constantemente el mismo que es visible por parte del cliente [49]. Aunque existe una variedad de metodologías y marcos de trabajo para el desarrollo ágil de software [50] [51] [52], las metodologías ágiles Scrum, programación extrema (XP), desarrollo rápido de aplicaciones (DRA) y Lean son las más representativas en este campo por su flexibilidad y facilidad de implementación, lo cual genera buenos resultados para quienes deciden usarlas.

*Tabla 1 Diferencia entre metodologías ágiles y tradicionales [53]*

<b>Metodologías ágiles</b>	<b>Metodologías tradicionales</b>
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Grupos pequeños (< 10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Orientados a procesos	Orientado a personas
Pocos artefactos	Más artefactos
Pocos roles	Más roles

### **2.8.1. Scrum**

En proyectos de entornos complejos donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales, usar la metodología Scrum es lo más adecuado según [54], Scrum es ideal cuando se necesita obtener resultados pronto y cuando los requisitos son cambiantes o pocos definidos lo cual sucede frecuentemente en empresas pequeñas. El equipo Scrum consiste en un dueño de producto (Product Owner), el equipo de desarrollo (Development Team) y un Scrum master que es el responsable de asegurar que Scrum es entendido y adoptado [55].

Según [56] en Scrum la pieza clave es el sprint, el cual se define como cada ciclo o iteración de trabajo que produce una parte del producto terminada y funcionalmente

operativa (incremento), permitiendo así tener un ritmo de avance constante. Otro aspecto importante son los artefactos, los mismo que representan trabajo o valor en diversas formas que son útiles para proporcionar transparencia y oportunidades para la inspección y adaptación. A continuación se presentan los artefactos según [56]:

- Pila del producto: (product backlog) lista de requisitos de usuario, que a partir de la visión inicial del producto crece y evoluciona durante el desarrollo.
- Pila del sprint: (sprint backlog) lista de los trabajos que debe realizar el equipo durante el sprint para generar el incremento previsto.
- Incremento: resultado de cada sprint

Con la utilización de la metodología Scrum el cliente puede empezar a utilizar las características más importantes del proyecto antes de que esté completamente terminado, esto resulta interesante para nuestro proyecto de microservicios puesto que se maneja los componentes de manera modular y por lo tanto sería factible mostrarle al cliente los avances que se vayan realizando.

### **2.8.2. Programación extrema (XP)**

La metodología XP se enfoca en la programación de pares, haciendo revisión extensiva del código, así como en la simplicidad y claridad. Unos de los puntos fuertes de XP es que mantiene un seguimiento frecuente de la comunicación con el cliente y programador [57], de esta manera si hubiera una posible desviación de los objetivos del proyecto, se la puede detectar a tiempo y hacer las respectivas correcciones.

Según [58] las herramientas de XP son las siguientes: historias de usuarios, las cuales se utilizan para realizar estimaciones de tiempo y representan una característica esencial del sistema, otra herramienta importante en XP son las pruebas de aceptación o pruebas del cliente, durante estas pruebas el cliente verifica si las historias de usuarios de cada iteración cumple con las funcionalidades esperadas. Las tarjetas Clase-Responsabilidad-Colaboración (CRC) también forman parte de las herramientas de XP, las tarjetas CRC permiten conocer que clases componen el sistema y cuales interactúan entre sí. De acuerdo a [59] las reglas de XP son simples, juntas conforman la esencia de XP pero sin embargo, si se manejan individualmente carecen de sentido, a continuación se presentan las reglas de XP según [59]:

- Planeación

En esta fase el proyecto es dividido en interacciones, las mismas que tienen como resultados pequeños releases o entregas del proyecto y las entregas se realizan en base a la culminación de un grupo de historias de usuarios en cada interacción.

- Gestión

En la gestión se mide la velocidad del proyecto, esta medición se realiza en base a los resultados obtenidos de las reuniones breves que se llevan a cabo todos los días, también se encarga de mantener un ambiente de trabajo agradable para el equipo y que

de esta manera el ritmo de trabajo sea sostenible.

- **Diseño**

Durante esta etapa es recomendable utilizar tarjetas CRC en las sesiones de diseño, durante estas sesiones se debe dejar claro que las soluciones deben ser simples y esto se logra abordando problemas específicos, así mismo que ninguna funcionalidad se debe añadir antes de tiempo y que se debe refactorizar el código las veces que sea posible.

- **Desarrollo**

Durante el desarrollo se debe fomentar la participación colectiva de los desarrolladores y asegurarse de que el código es escrito de acuerdo a los estándares acordados, así mismo se debe asegurar la participación activa por parte del cliente. XP fomenta la integración continua de código por esta razón un aspecto clave en este punto es la configuración de un servidor que este dedicado a la integración de código.

- **Pruebas**

En esta fase se debe tener claro que todas las piezas de códigos desarrolladas deben tener su respectiva prueba unitaria, así mismo que todo el código antes de ser liberado debe pasar los distintos tipos de pruebas que son: unitarias, componentes, contrato y funcionales. Otro aspecto importante en esta fase es que si se detecta un fallo se deben crear pruebas que garanticen que se supera el problema durante cada mejora realizada y finalmente que las pruebas de aceptación además de las unitarias se deben realizar con frecuencia.

En el presente trabajo la utilización de la metodología XP permitiría delimitar el proyecto con el cliente es decir definir el alcance, también se podría definir mediante la utilización de historias de usuarios cuales son las características esenciales que debe tener el sistema, adicionalmente se podría garantizar un producto final de alta calidad puesto que todos los componentes desarrollados tendrían su respectiva prueba unitaria y el cliente estaría involucrado en los avances con la realización de las pruebas de aceptación.

### **2.8.3. Desarrollo rápido de aplicaciones (DRA)**

El término desarrollo rápido de aplicaciones se hizo popular luego de su aparición en el libro Rapid Application Development de James Martin [60] en cual expone que DRA es un ciclo de vida de desarrollo diseñado para proporcionar resultados de desarrollo mucho más rápidos y de mayor calidad que los obtenidos con el ciclo de vida tradicional. Está diseñado para aprovechar al máximo las ventajas del potente software de desarrollo que ha evolucionado recientemente. Así mismo Martin sostiene que hay cuatro aspectos fundamentales en DRA los cuales son: las herramientas, las metodologías, las personas y la gestión [61].

Las principales ventajas de DRA según [62] son las siguientes:

- **Rapidez en el desarrollo**

- Visibilidad temprana debido al uso de técnicas prototipo
- Mayor flexibilidad que otros modelos
- Ciclos de desarrollo más cortos

DRA requiere el uso de prototipos para definir los requerimientos del usuario y diseñar el sistema final. Así mismo [63] afirma que la metodología funciona mejor para proyectos donde el alcance es pequeño o donde el trabajo puede dividirse en módulos manejables, esto implica que los equipos de proyecto también deben ser pequeños, preferiblemente de dos a seis personas, y el equipo debe tener experiencia en todas las tecnologías que se vayan a utilizar.

#### **2.8.4. Lean**

Lean nació como parte del renacimiento industrial de la manufactura japonesa después de la Segunda Guerra Mundial en la década de 1940, pero el término Lean se aplicó públicamente por primera vez a un proceso de gestión de la producción y luego al desarrollo de productos en el MIT a mediados de la década de 1980 [64]. El objetivo principal de Lean es implementar los principios de fabricación, ajustada en un modelo de desarrollo de software para reducir el desperdicio en un sistema y producir un mayor valor para el cliente [65]. Esta es quizás la causa de que los conceptos Lean hayan sido ampliamente adoptados por la comunidad ágil y que algunos hablen de "Leagile" [66].

Según [67] el desarrollo de software Lean proporciona la teoría detrás de las prácticas ágiles de desarrollo de software y da a las empresas un conjunto de principios a partir de los cuales pueden diseñar procesos de ingeniería de software que funcionen mejor en el contexto de sus clientes, su dominio, su capacidad de desarrollo y su situación única. De acuerdo a [67] y [68] hay siete principios que guían el desarrollo de software Lean, los cuales son:

1. Eliminar el desperdicio, entendiendo primero qué es el valor.
2. Construir calidad, probando lo antes posible la automatización y la refactorización.
3. Crear conocimiento, a través de la retroalimentación rápida y la mejora continua.
4. Diferir el compromiso, manteniendo las opciones y tomando decisiones irreversibles en el último momento responsable cuando la mayor parte de la información está disponible.
5. Entrega rápida, a través de lotes pequeños.
6. Respetar a la gente que hace el trabajo.
7. Optimizar todo el sistema, implementando Lean en todo un flujo de valor.

Como ya se mencionó anteriormente Lean en el desarrollo de software reduce el desperdicio, a continuación, se presenta los siete tipos de desperdicios que Lean reduce:

1. Trabajo parcialmente realizado
2. Características adicionales o extras
3. Reaprendizaje

4. Transferencia de información o ciclos de vida en cascada
5. Cambio de tareas
6. Retrasos
7. Defectos.

La implementación de los principios Lean en el presente trabajo generarían un beneficio tanto al cliente como al equipo de trabajo, puesto que se entregaría software funcional al cliente de manera frecuente, haciendo visible las actividades que realiza el equipo de trabajo, mientras que el beneficio que generaría para el equipo de trabajo sería la creación de software escalable que resultaría fácil de mantener y refactorizar por su amplia cobertura de pruebas en los distintos componentes desarrollados y al mismo tiempo les permitirá generar conocimiento a través de la retroalimentación. De tal forma que al final del proyecto el cliente pueda obtener el producto deseado en el periodo de tiempo acordado al inicio del proyecto.

# Capítulo 3: Resultados

## 3.1. Análisis de la entrevista

Los datos obtenidos de la entrevista realizada al jefe de TIC's de la PUCESE indican que cuando un cliente solicita el desarrollo de un sistema la mayoría de las veces no tiene claro lo que realmente necesita. esta problemática se ha venido dando desde hace algún tiempo atrás, indica el entrevistado. Puesto que en las distintas operaciones internas de la Universidad hay una carencia de procesos, con lo cual se dificulta la identificación de las operaciones claves en las distintas actividades que realizan los demás departamentos o clientes de TIC's. Otra problemática identificada por el jefe de TIC's es que los sistemas desarrollados por el departamento son altamente personalizados, con lo cual se dificulta la escalabilidad y su descomposición en módulos o componentes.

El entrevistado también indica que por la carencia de procesos y planificación no se utilizan metodologías de desarrollo, sin embargo, los sistemas se han venido desarrollando en base al proceso del modelo lineal como se puede apreciar en la Figura 12, en el cual se pueden evidenciar las siguientes etapas: análisis, diseño, programación y pruebas manuales.

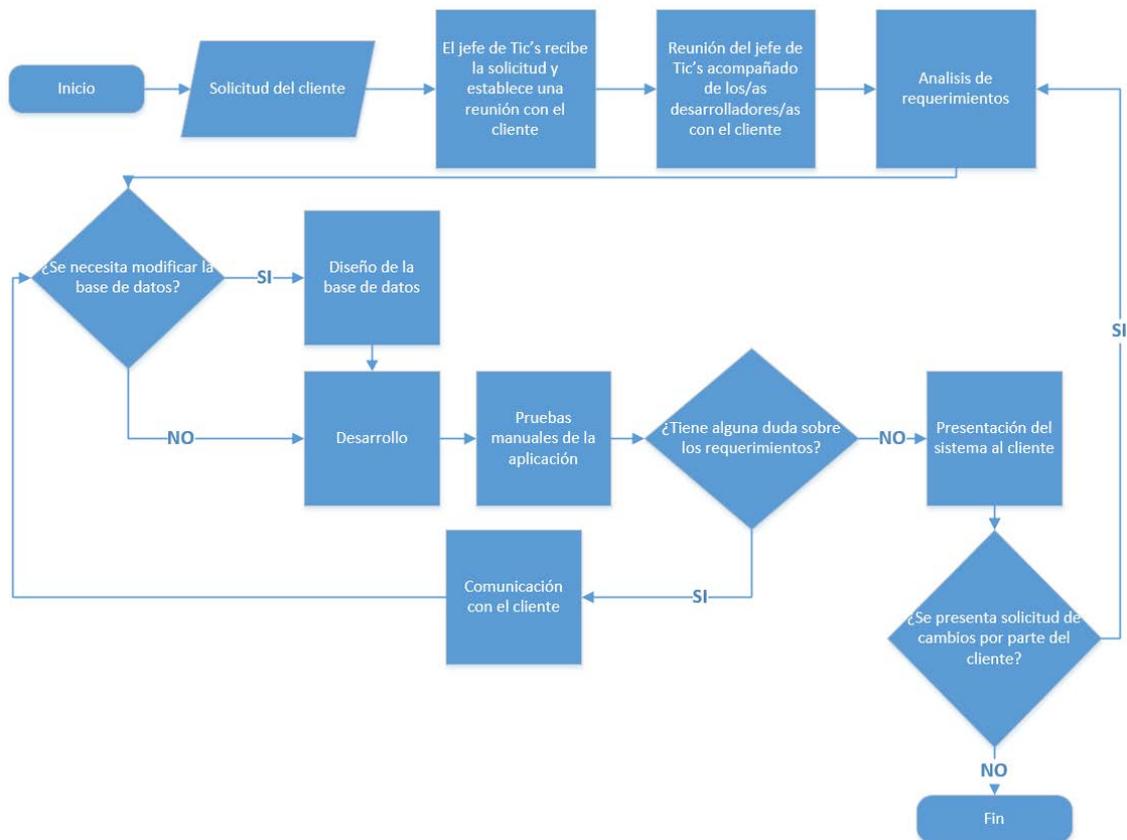


Figura 12 Proceso de desarrollo del departamento de TIC's

De la entrevista también se pudo conocer que los sistemas actuales de la Universidad no están orientados a SOA, el sistema solo se encarga de automatizar un proceso, pero sus funcionalidades no pueden ser reutilizadas por otro sistema. Sobre los lenguajes de programación utilizados en el departamento para el desarrollo de software se nos supo indicar que los que más se utilizan son: Visual Basic, C#, y PHP. A su vez sostienen que la utilización de frameworks facilita la reutilización de funcionalidades, complementos y mejora la seguridad, por esta razón utilizan los frameworks: .NET y Yii2.

### **3.2. Análisis de las encuestas**

Luego de realizar la tabulación y la interpretación de las encuestas, se ha llegado a las siguientes conclusiones.

- Los desarrolladores consideran que los problemas más comunes en los retrasos de los proyectos software se originan por: falta de información durante el levantamiento de requerimientos, fallas por omisión de detalles y constantes cambios por parte del cliente.
- Consideran que es importante la utilización de un framework durante la etapa de desarrollo.
- Rara vez o nunca se establecen estimaciones de esfuerzo, alcance y costo en el desarrollo de un proyecto.
- Durante la etapa de desarrollo de un proyecto software hay poco compromiso con el plan por parte del cliente.
- Algunas veces se suele identificar todos los requisitos del cliente.
- Los requisitos del proyecto software del cliente, no siempre se validan completamente.

Los resultados obtenidos demuestran que los desarrolladores tienen amplias habilidades técnicas y utilizan herramientas que le permiten optimizar el tiempo de desarrollo, en este caso hablamos de frameworks, sin embargo, frente al poco compromiso por parte del cliente, falta de planificación, ausencia de metodología de desarrollo y falencias en la comunicación durante el levantamiento de requerimientos, los productos generados no son de alta calidad como espera la gerencia, lo cual ha generado insatisfacción hacia el departamento de TIC's.

### **3.3. Selección de la metodología de desarrollo**

Luego del análisis de la información obtenida de la entrevista y las encuestas, los resultados demuestran que se necesita aplicar una metodología de desarrollo ágil para los proyectos software. En las siguientes tablas se muestra la comparativa realizada entre las diferentes metodologías ágiles encontradas en varias fuentes de información.

Tabla 2 Comparación de métodos ágiles basados en parámetros importantes [15]

Feature	XP	Scrum	DSDM	ASD	FDD	Crystal
Team (members)	2-12 (colocation preferred)	5-7	2-6 per team	Variable	4-20 per team	1-200
Iteration/ Sprint (weeks)	1-3	2-4	20% time for 80% solution	4-8	Up to 2	Up to 17
Documentation	Minimum	Minimum	Moderate	Minimum	Maximum	Minimum

Tabla 3 Comparación entre los métodos más ágiles en base a los criterios definidos [16]

Criteria	Attributes	XP	Scrum	Crystal	DSDM
<b>Project Setup</b>	Environment	Highly satisfied	Highly satisfied	Satisfied	Not satisfied
	Location				
	Design				
	Development team				
<b>Project Complexity</b>	Manage changes in requirements	Highly satisfied	Highly satisfied	Satisfied	Satisfied
	Handle project time and cost estimated	Satisfied	Highly satisfied	Satisfied	Satisfied
	Handle project scalability	Not satisfied	Not satisfied	Highly satisfied	Not satisfied
	Handle unapproved customer decisions	Highly satisfied	Satisfied	Satisfied	Highly satisfied
	Balance the business needs with customers' needs	Satisfied	Not satisfied	Not satisfied	Highly satisfied
<b>Contract Management</b>	Changes in requirements and needs	Highly satisfied	Not satisfied	Not satisfied	Not satisfied
	Time and cost estimation				
	Unapproved customer decisions				
	Delay in the delivery time				

En base a la información encontrada de las siguientes investigaciones [69] [15] [70] [71] [16] [72] [73] [74] , teniendo en cuenta que el personal de desarrollo del departamento de TIC's es relativamente pequeño (3 desarrolladores) y que se necesita realizar entregas de productos software en periodos cortos de tiempo, se llegó a la conclusión de que la metodología ágil que más se ajusta a este tipo de ambientes es: La metodología ágil XP. Otro aspecto a tomar en cuenta en este tipo de ambientes es que los clientes siempre solicitan entregas de manera temprana luego de que inicia el proyecto, por esta razón se va adoptar el proceso de entregas por sprint de la metodología ágil Scrum.

### **3.4. Análisis del caso**

En el departamento de TIC's de la PUCESE se ha venido desarrollando software que han permitido agilizar procesos internos tanto de áreas administrativas como académicas, sin embargo, durante las fases de desarrollo e incluso en el cierre del proyecto se han presentado varios inconvenientes producto de la carencia de planificación. En varios estudios que se han realizado sobre el departamento de TIC's e inclusive en este estudio se demuestra que es necesario la utilización de una metodología ágil para el desarrollo de software que permita identificar las principales necesidades de los nuevos sistemas que se vayan a desarrollar teniendo en cuenta siempre los objetivos de la Universidad. Es así que luego de la entrevista realizada al jefe del Departamento de TIC's, se nos dio a conocer la necesidad de un sistema integral al que dominarían intranet el cual debe permitir llevar a cabo el proceso de evaluación de competencias del personal administrativo, reportes que permitan conocer el uso que le dan los docentes a la plataforma de aprendizaje en línea denominada Aula Virtual, y además necesitan tener acceso a las marcaciones que se han realizado en el sistema biométrico tanto del personal administrativo como docente.

#### **3.4.1 Implementación de XP y Scrum**

Luego de identificar que la metodología XP en conjunto con la metodología Scrum se ajusta a las necesidades actuales del departamento de TIC's de la PUCESE se procedió a formar el equipo de trabajo y a establecer los procesos que se deberían llevar a cabo para la realización del nuevo proyecto software.

##### **3.4.1.1. Planeación.**

Luego de conocer las necesidades se procedió a la descomposición del problema para establecer la arquitectura adecuada la cual debe permitir agregar más reportes o módulos que la Universidad requiera en futuras versiones o mejoras. De esta manera se llegó a la conclusión de que el desarrollo de este proyecto se va a realizar en tres fases, cada una de las cuales debe cumplir con las diferentes necesidades del cliente, a continuación, se presentan las fases del proyecto y que funcionalidades o entregas debe tener cada una.

- **Fase uno (Primer entrega).**

En esta entrega se debe implementar el modulo para la evaluación de competencias del personal administrativo, este sistema debe ser configurable por parte del cliente en cuanto a las fechas de apertura y cierre para la evaluación, además deberá presentar los reportes individuales y generales de la evaluación realizada a los distintos trabajadores, en el anexo 3 se pueden apreciar las historias de usuarios relacionadas a esta entrega.

- **Fase dos (Segunda entrega).**

Se deberá entregar el modulo para reportes del sistema biométrico, cada uno de los trabajadores tanto de personal administrativo como docente deben tener acceso a su información, esto les va a permitir conocer si sus marcaciones del día laborado están correctas y además cuantas horas ha laborado durante día y durante el mes, tal y como se indica en las historias de usuario del anexo 4.

- **Fase tres (Tercer entrega).**

Para la tercera entrega se deberá entregar el módulo de reportes sobre el uso de la plataforma Aula Virtual, esta información solo debe ser accesible por los directores Académicos y de Escuela, para lo cual se deben implementar filtros de información de acuerdo al rol que tenga el usuario que accede al reporte, Ejm: Director de la escuela de Sistemas y Computación solo tendrá acceso a los reportes generados en la plataforma referente a los docentes de la escuela de sistema, Director general académico tendrá acceso al reporte generado por la plataforma de todas las escuelas de la Universidad. En el anexo 5 se pueden apreciar los detalles de esta funcionalidad mediante las historias de usuarios.

#### **3.4.1.2. Diseño de la arquitectura**

Para dar solución a la necesidad de la Universidad se planteó la implementación de una MSA para que de esta manera el sistema se pueda manejar de forma modular y sea más sencilla su portabilidad y escalabilidad. Desarrollar el sistema con MSA también le va a dar un manejo independiente de cada una de las funcionalidades que tenga el sistema y por ende el cumplimiento de los objetivos del proyecto se podrán realizar en periodos cortos de tiempos, en la siguiente figura se muestra la arquitectura propuesta.

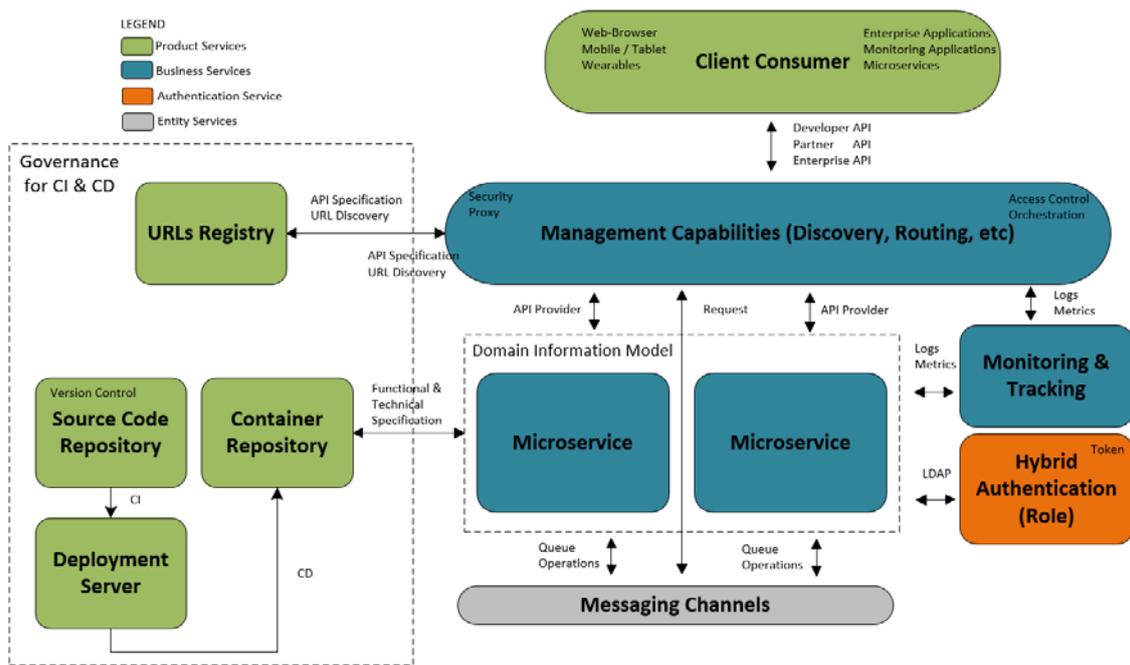


Figura 13 Arquitectura MSA propuesta para desarrollo del proyecto

Los componentes en forma de bloques presentados en la figura anterior son la clave la implementación y gestión de los microservicios en el contexto de una arquitectura empresarial, a continuación, se va a explicar el rol de los componentes que se encuentran clasificados por colores.

- **Product Services**

Son los servicios que están relacionados con los procesos de la gestión de la comunicación desde el exterior hacia en el interior de la arquitectura. Permiten establecer la comunicación con el controlador principal de los microservicios y a su vez es el espacio de trabajo donde se realizan los procesos de integración y despliegues de los microservicios.

- **Business Services**

Aquí se maneja los procesos de negocios de la empresa, desde la gestión de los distintos microservicios hasta los procesos de monitorio y control de acceso hacia los distintos módulos. Cada uno de estos servicios representa una tarea específica del proceso de negocio.

- **Authentication Service**

En este servicio se maneja el proceso de autenticación combinando un servicio externo con un servicio interno de la empresa, en nuestro caso se refiere a Office 365 y al directorio activo local de la empresa.

- **Entity Services**

La función de este servicio es garantizar la persistencia de datos y administrar todo lo relacionado con almacenamiento, recuperación, transacciones y gestión de bloques.

### **3.4.1.3. Herramientas seleccionadas para el desarrollo del sistema**

En base a los datos obtenidos de la entrevista con el jefe de TIC's y las encuestas de los desarrolladores, en la cual se pudo identificar que los nuevos sistemas se están desarrollando bajo el lenguaje de programación PHP la versión 7, se escogió el framework Yii2 para el desarrollo del proyecto, así mismo para el levantamiento de la MSA se escogió la tecnología de contenedores Docker en su versión 18.03. Por otra parte, para el intercambio asíncronos de mensajes entre los microservicios se seleccionó RabbitMQ 3.7.6, para el almacenamiento de los reportes se seleccionó MongoDB con la versión 3.4.10 puesto que no es tan rígido como las bases de datos relacionales y nos permite establecer estructuras dinámicas que pueden cambiar con la evaluación de la empresa. Para los procesos automatizados de integración y despliegues continuos se escogió la herramienta Jenkins en la versión 2.106. Todas las tecnologías mencionadas anteriormente se manejan bajo libre distribución y cuentan con una amplia comunidad de desarrolladores y documentación, por lo cual son las más populares y a la vez las más estables en el área del desarrollo de software.

Actualmente la Universidad tiene acuerdos con Microsoft mediante el pago de licencias para el uso de sus herramientas en ambientes educativos. Aprovechando la disponibilidad de las herramientas Microsoft como mecanismo de seguridad para el proceso de autenticación y controles de accesos mediante roles de los usuarios, se implementó una solución híbrida que permite al usuario autenticarse mediante office 365 en la nube y a la vez se verifica que rol tiene asignado en el directorio activo local de la empresa. Esta solución facilita el acceso del usuario al sistema puesto que una vez que inicie su sesión de correo corporativo en la nube no tendrá que volver a ingresar sus credenciales o pasar por un formulario de autenticación para el acceso a la intranet.

## **3.5 Resolución del problema**

Luego de establecer el acuerdo con el cliente de que el proyecto se va a entregar en tres fases, la primera actividad que se procedió a realizar para la resolución del problema fue la estimación de las distintas tareas que son necesarias llevar a cabo en cada fase. Luego de la estimación de las tareas se determinó que en cada fase es necesario ejecutar dos sprints, del cual el primero está conformado en su mayoría por tareas técnicas, este tipo de tareas no son visibles por el cliente pero son necesarias para el desarrollo del mismo, en este proyecto algunas de las tareas técnicas son: preparación de los ambientes, levantamiento de la base de datos, creación de vistas en las bases de datos de los sistemas actuales que permitan obtener la información necesaria para elaborar de los reportes solicitados, registro de las URLs de los ambientes del cliente en los servicios de Microsoft Apps para el inicio de sesión, entre otras. Por otra parte, el segundo sprint está conformado por tareas relacionadas con el negocio es decir tareas que generan valor

al negocio y es visible por el cliente como: interfaz de inicio de sesión en la aplicación, interfaces que muestran los distintos reportes, entre otras.

### 3.5.1. Registro de aplicaciones en Office 365

Para la autenticación híbrida es necesario registrar nuestras aplicaciones en el portal de aplicaciones de Microsoft, luego de registrar nuestra aplicación el portal nos genera un client id y un password token, lo cual lo debemos configurar en nuestra aplicación. En la siguiente figura se muestra el formulario de registro de aplicación.

## Intranet micro Registration

[Click here for help integrating your application with Microsoft.](#)

This application will be registered in the Azure Active Directory instance used to manage your luis.herrera@pucese.edu.ec account. ✕

### Properties

Name

Application Id

7048fad3-6f08-417b-8ddf-b36cc231fe9e

### Application Secrets

[Generate New Password](#) [Generate New Key Pair](#) [Upload Public Key](#)

Type	Password/Public Key	Created	
Password	ZJQ*****	Jun 28, 2018 9:24:32 PM	<a href="#">Delete</a>

*Figura 14 Formulario de registro de aplicación en el portal de Microsoft*

Una parte importante del registro de la aplicación es el ingreso de la URL que va a tener configurada nuestra aplicación en los distintos ambientes (dev, qa, stg, prod), para el uso del servicio de Microsoft, en la siguiente figura se muestra la sección de configuración de URL.

## Platforms

**Add Platform**

Web **Delete**

Allow Implicit Flow

Redirect URLs ⓘ **Add URL**

https://dev-msa.pucese.edu.ec/backend/web/site/office365

https://qa-msa.pucese.edu.ec/backend/web/site/office365

https://stg-msa.pucese.edu.ec/backend/web/site/office365

https://msa.pucese.edu.ec/backend/web/site/office365

Figura 15 Registro de URLs de los distintos ambientes de la aplicación para autenticación

En la siguiente figura se muestra la sección de configuración de nuestra aplicación donde se ingresa los datos obtenidos del portal de Microsoft.

```
'office365-api' => [  
  'clientId' => '7048fad3-6f08-417b-8ddf-b36cc231fe9e',  
  'clientSecret' => 'ZJQ*****',  
  'redirectUri' => 'https://msa.pucese.edu.ec/backend/web/site/office365',  
  'urlAuthorize' => 'https://login.microsoftonline.com/common/oauth2/v2.0/authorize',  
  'urlAccessToken' => 'https://login.microsoftonline.com/common/oauth2/v2.0/token',  
  'urlResourceOwnerDetails' => '',  
  'scopes' => 'openid profile user.read mail.send'  
],
```

Figura 16 Ingreso de datos obtenidos del portal de Microsoft en nuestra aplicación

### 3.5.2. Configuración del AD local

La segunda parte del proceso de autenticación está basado en la verificación de rol que tiene asignado el usuario en el AD local de la empresa, cuyos parámetros de acceso deben ser configurados en la aplicación para realizar la consulta de datos. En la siguiente figura se muestra la página principal del AD local que se levantó para el uso con la aplicación.

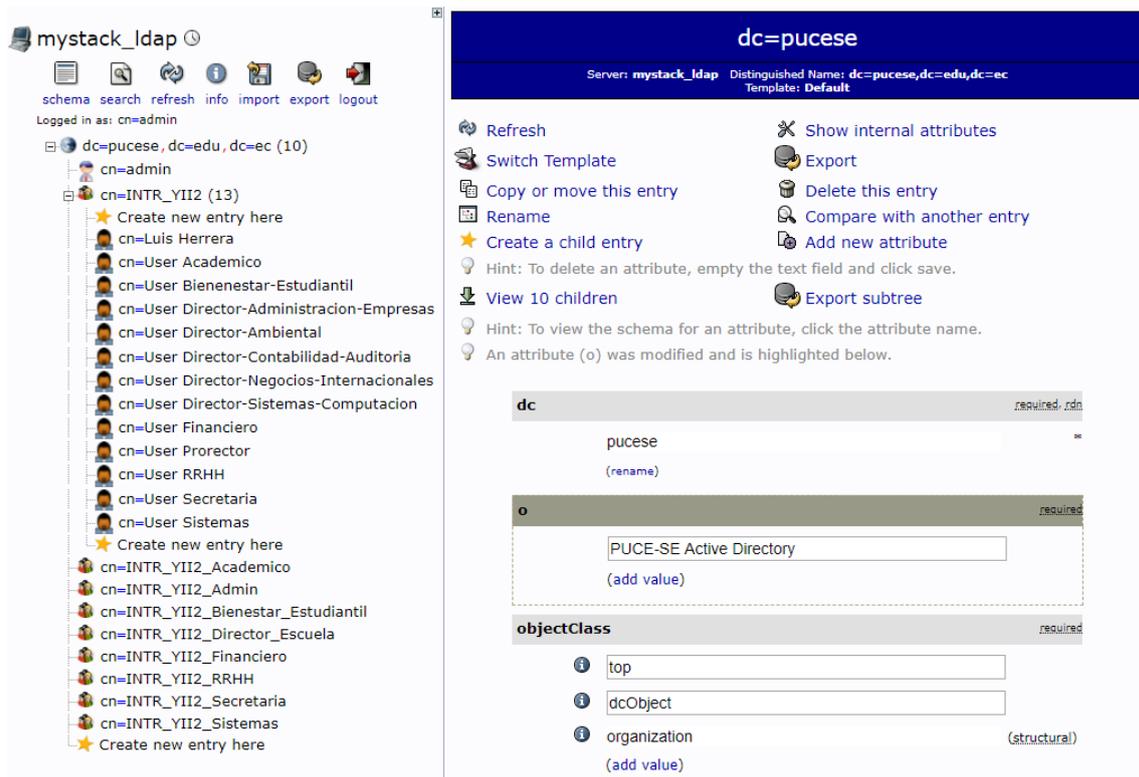


Figura 17 Vista principal del AD local

En la siguiente figura se muestra los datos de configuración para el acceso al AD desde la aplicación.

```
'ad' => [
  'class' => 'Edv1erblog\Adldap2\Adldap2Wrapper',
  'providers' => [
    'default' => [
      'autoconnect' => true,
      'config' => [
        'account_suffix' => '@pucese.edu.ec',
        'domain_controllers' => ['mystack_ldap'],
        'base_dn' => 'dc=pucese,dc=edu,dc=ec',
        'admin_username' => 'cn=admin,dc=pucese,dc=edu,dc=ec',
        'admin_password' => 'JonSn0w',
      ]
    ]
  ],
], //close ad
```

Figura 18 Datos de configuración para el acceso al AD local desde la aplicación

Para el acceso a distintos grupos del AD local se usó expresiones regulares las cual permite a la aplicación acceder a los miembros de los grupos que empiecen con INTR\_YII2 y de estar manera hacer la verificación del rol de cada usuario, en la siguiente figura se muestra la configuración implementada en la aplicación para el acceso a los grupos.

```
'LDAP-Group-Assignment-Options' => [
  'LOGIN_POSSIBLE_WITH_ROLE_ASSIGNED_MATCHING_REGEX' => '/^(INTR_YII2)(.*)/',
  'REGEX_GROUP_MATCH_IN_LDAP' => '/^(INTR_YII2)(.*)/', // start with
  'ADD_GROUPS_FROM_LDAP_MATCHING_REGEX' => true,
  'REMOVE_ALL_GROUPS_NOT_FOUND_IN_LDAP' => true,
  'REMOVE_ONLY_GROUPS_MATCHING_REGEX' => false,
  'SEARCH_NESTED_GROUPS' => true
],
```

Figura 19 Configuración para acceso a los grupos del AD local

### 3.5.3 Configuración del middleware de mensajería

Para establecer la comunicación segura entre los distintos componentes de la arquitectura se implementó el middleware de mensajería RabbitMQ, para lo cual se llevó a cabo configuraciones en el servidor de mensajería y en nuestras aplicaciones.

#### 3.5.3.1. Creación del intercambiador

El intercambiador es quien recibe el mensaje, que es enviado desde el productor, luego se encarga de direccionar el mensaje a las colas correctas, las cuales se definen en la Información de direccionamiento del mensaje. En nuestro proyecto creamos un intercambiador denominado: queue\_multiple, el cual se puede apreciar en la siguiente figura.

Name	Type	Features	Message rate in	Message rate out	+/-
(AMQP default)	direct	D			
amq.direct	direct	D			
amq.fanout	fanout	D			
amq.headers	headers	D			
amq.match	headers	D			
amq.rabbitmq.trace	topic	D I			
amq.topic	topic	D			
queue_multiple	direct	AD	0.40/s	0.40/s	

Figura 20 Intercambiador del proyecto queue\_multiple

Como se había mencionado anteriormente en el flujo de los mensajes de RabbitMQ es necesario establecer un enlace entre el intercambiador y las colas, en la siguiente figura podemos apreciar los enlaces establecidos entre el intercambiador y las colas que se configuraron para nuestro proyecto las cuales son:

- evaluacion\_laboral.asignada\_por\_trabajador
- evaluacion\_laboral.guardar
- evaluacion\_laboral.resultados\_generales
- evaluacion\_laboral.resultados\_por\_trabajador
- evaluacion\_laboral.resultados\_resumen\_trabajador
- biometrico.marcaciones\_generales
- biometrico.marcaciones\_trabajador
- reportes\_moodle.escuela\_director
- reportes\_moodle.todas\_escuelas

The screenshot shows the RabbitMQ Admin interface for the 'queue\_multiple' exchange. The 'Bindings' section is expanded, showing a table of bindings. Above the table, a box labeled 'This exchange' has a double-headed arrow pointing down to the table. The table has four columns: 'To', 'Routing key', 'Arguments', and 'Unbind'.

To	Routing key	Arguments	Unbind
biometrico.marcaciones_generales	biometrico.marcaciones_generales		Unbind
biometrico.marcaciones_trabajador	biometrico.marcaciones_trabajador		Unbind
evaluacion_laboral.asignada_por_trabajador	evaluacion_laboral.asignada_por_trabajador		Unbind
evaluacion_laboral.guardar	evaluacion_laboral.guardar		Unbind
evaluacion_laboral.resultados_generales	evaluacion_laboral.resultados_generales		Unbind
evaluacion_laboral.resultados_por_trabajador	evaluacion_laboral.resultados_por_trabajador		Unbind
evaluacion_laboral.resultados_resumen_trabajador	evaluacion_laboral.resultados_resumen_trabajador		Unbind
reportes_moodle.escuela_director	reportes_moodle.escuela_director		Unbind
reportes_moodle.todas_escuelas	reportes_moodle.todas_escuelas		Unbind

Figura 21 Enlace de colas al intercambiador queue\_multiple

### 3.5.3.2 Creación de las colas

El funcionamiento de las colas de mensajes en RabbitMQ es similar a los sistemas de tareas de colas es decir que almacenan los mensajes de una manera FIFO (First-In-First-Out), la diferencia de RabbitMQ es que si hay varios lectores (consumidores) de una cola activos, a veces uno de ellos tiene prioridad sobre otro y por esta razón, la cola de mensajes en el modelo AMQ se denomina weak-FIFO, en la siguiente figura se muestra las colas creadas en RabbitMQ para ser utilizado en la aplicación.

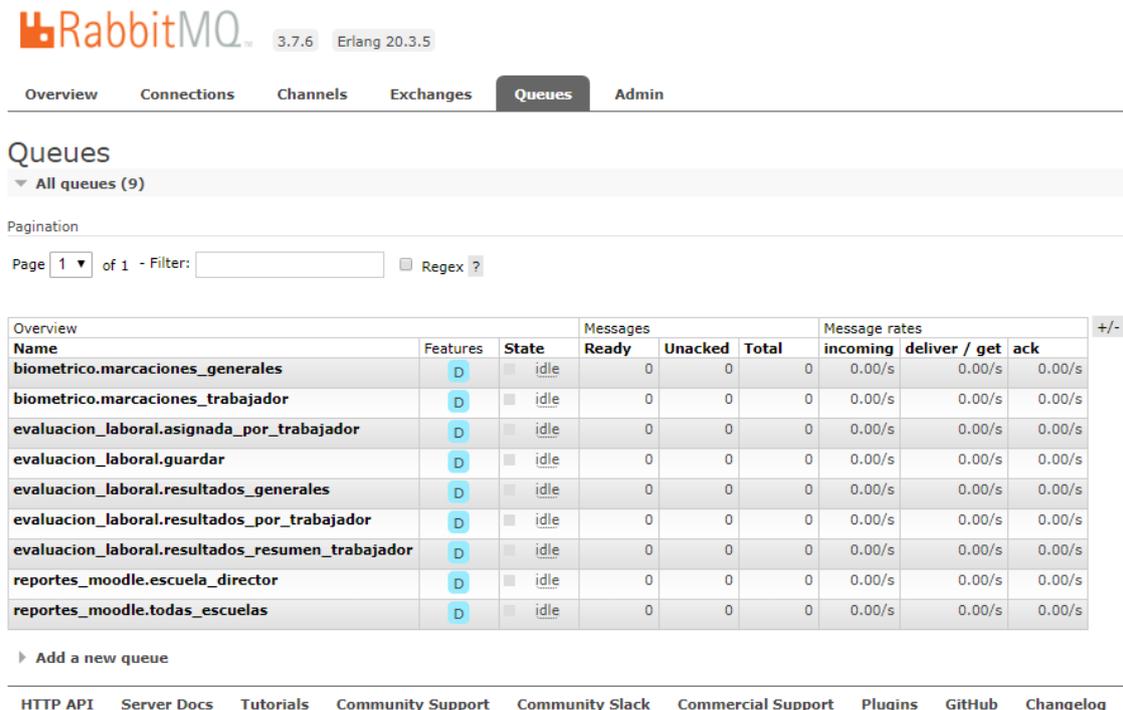


Figura 22 Colas creadas en RabbitMQ para el intercambio de datos entre los microservicios

### 3.5.4. Fase 1 módulo de evaluación de competencias

Para desarrollar este módulo se implementaron vistas que se conectan a las distintas tablas de la base de datos actual de RRHH de la universidad, así mismo se implementó una política de codificación de los distintos cargos del personal administrativo (ver figura 13) lo cual facilito la asignación de los distintos tipos de evaluación a los distintos trabajadores, los tipos de evaluación que se implementaron son los siguientes: autoevaluación, evaluación de jefe, evaluación de subalternos, evaluación de par y clientes.

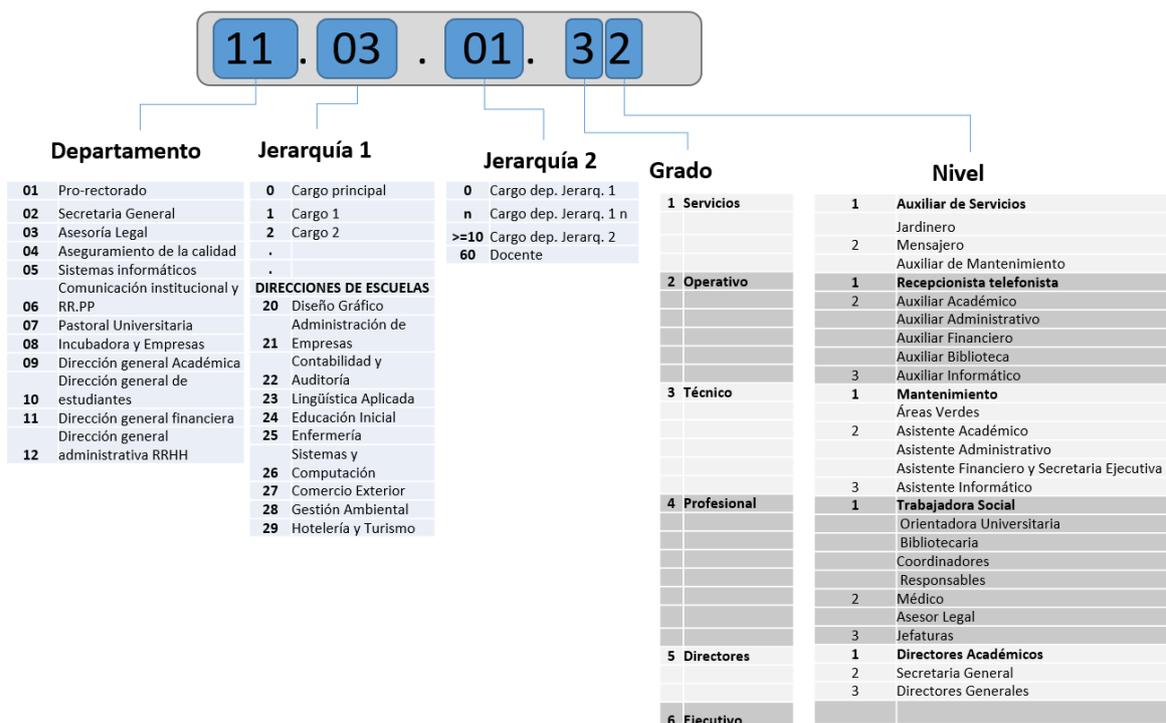


Figura 23 Política de códigos organigrama

Además, para el cálculo del promedio obtenido de las evaluaciones de cada trabajador se implementó una tabla de valoración en la cual se contempla los distintos casos de evaluación en los trabajadores, la cual se utilizó para la generación de reportes, en la siguiente tabla se puede apreciar la valoración implementada en el sistema.

Tabla 4 Tabla de valoración para la evaluación de desempeño

	1	2	3	4	5	6	7	8	9
AUTOEVALUACIÓN	5,00%	5,00%	5,00%	5,00%	5,00%	5,00%	5,00%	5,00%	5,00%
JEFE	35,00%	40,00%	55,00%	55,00%	60,00%	55,00%		80,00%	95,00%
PARES	20,00%	25,00%	25,00%		35,00%				
SUBALTERNOS	25,00%	30,00%		25,00%		40,00%	80,00%		
CLIENTES	15,00%		15,00%	15,00%			15,00%	15,00%	
MAX PORCENTAJE	100%	100%	100%	100%	100%	100%	100%	100%	100%
Max calificación	5								
Max porcentaje	100								
<b>EJEMPLO 1</b>									
	Calificación	Porcentaje tabla	Cálculos						
autoevaluación	4	5%	4						
jefe	4	60%	48						
pares	5	35%	35						
<b>SUBTOTAL</b>	-	<b>100%</b>	<b>87</b>						
<b>PROMEDIO</b>	-	-	<b>4,35</b>						

Luego de tener definido el organigrama para conocer cómo se van a evaluar los trabajadores y la tabla de valoración de las evaluaciones, se procedo a crear la base de

datos para almacenar los resultados de las evaluaciones y las vistas que permitan obtener la información de los trabajadores en la actual base de datos de RRHH, en la siguiente figura se muestra la vista que permite al sistema identificar como los trabajadores se deben evaluar.

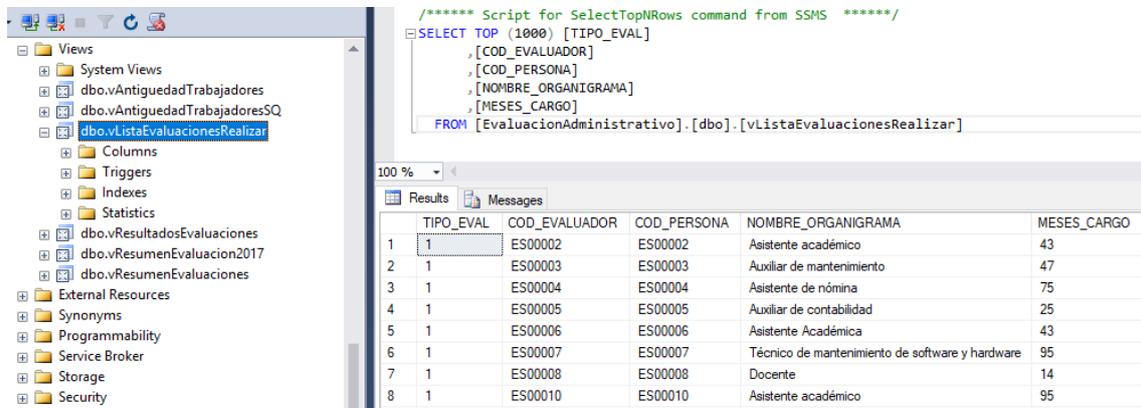


Figura 24 Vista de evaluaciones entre los trabajadores

Una vez culminada la preparación de la base de datos con las tablas y vistas necesarias, se procedió a desarrollar las interfaces en la aplicación Evaluación Laboral que se conectan a las diferentes tablas y vistas de la base de datos, en la siguiente figura se muestra la interfaz gráfica de la vista vListaEvaluacionesRealizar.

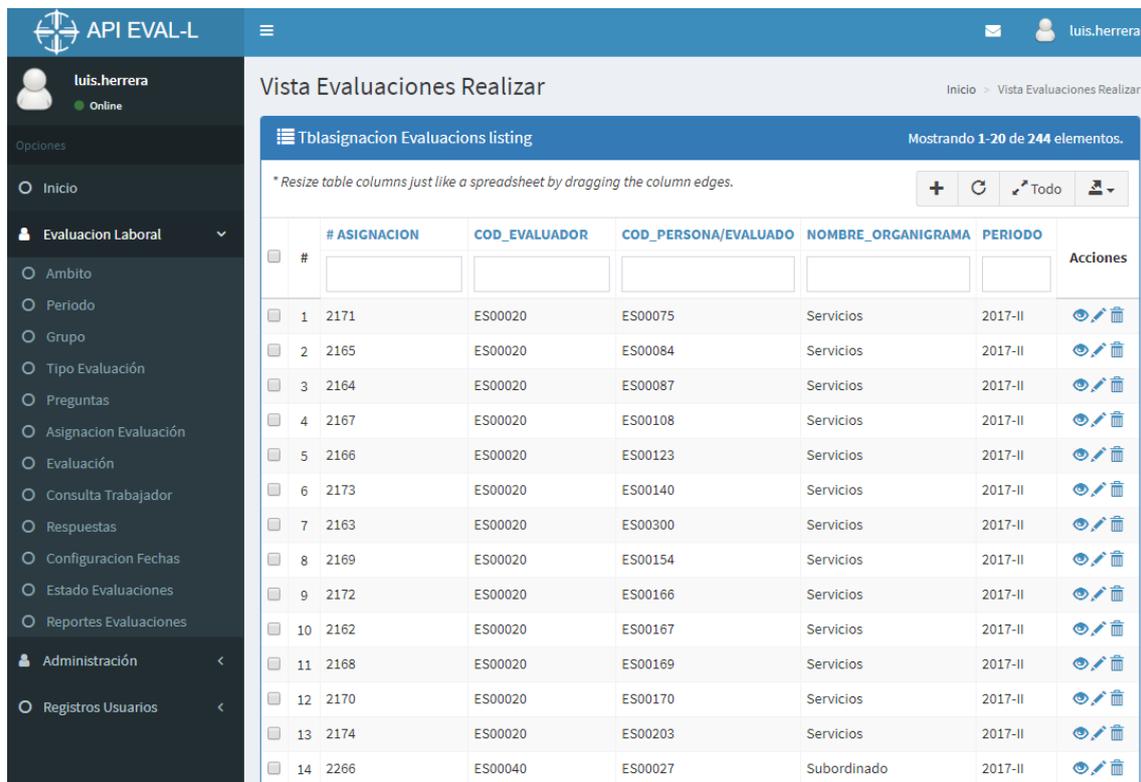


Figura 25 Interfaz gráfica de la vista vEvaluacionesRealizar

Como se puede apreciar en la figura anterior, esta aplicación se desarrolló en forma de microservicio con la intención de que pueda ser usada por otra aplicación y la información que retorne se pueda mostrar de una manera más amigable al usuario final o

cliente. En la siguiente figura se muestra la información que retorna la aplicación cuando es consumida como endpoint.

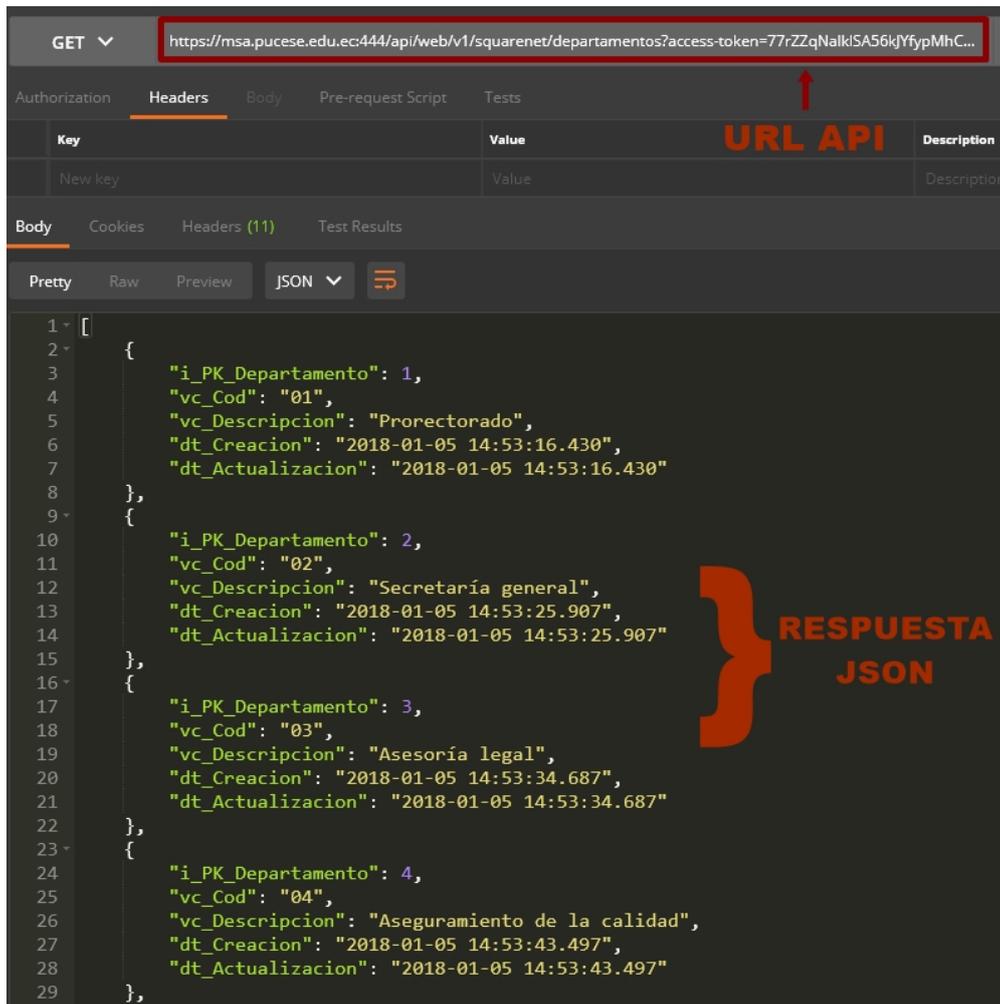


Figura 26 Respuesta de consumo del microservicio evaluación laboral

### 3.5.5. Fase 2 módulo de reportes biométrico.

En el desarrollo del módulo reportes biométrico se siguió el mismo proceso que en el módulo de evaluación por competencias, con la excepción de que en este módulo solo correspondía mostrar los datos recolectados en la base de datos biométrico del sistema que ya tenía implementado la universidad. Para solucionar la necesidad de ver reportes más representativos solicitados por el cliente se desarrollaron vistas que conectan a varias tablas del sistema biométrico actual, para luego procesarla en la aplicación y mostrarla al cliente de una manera más amigable. En la siguiente figura se muestra la vista creada para obtener las marcaciones de todos los trabajadores.

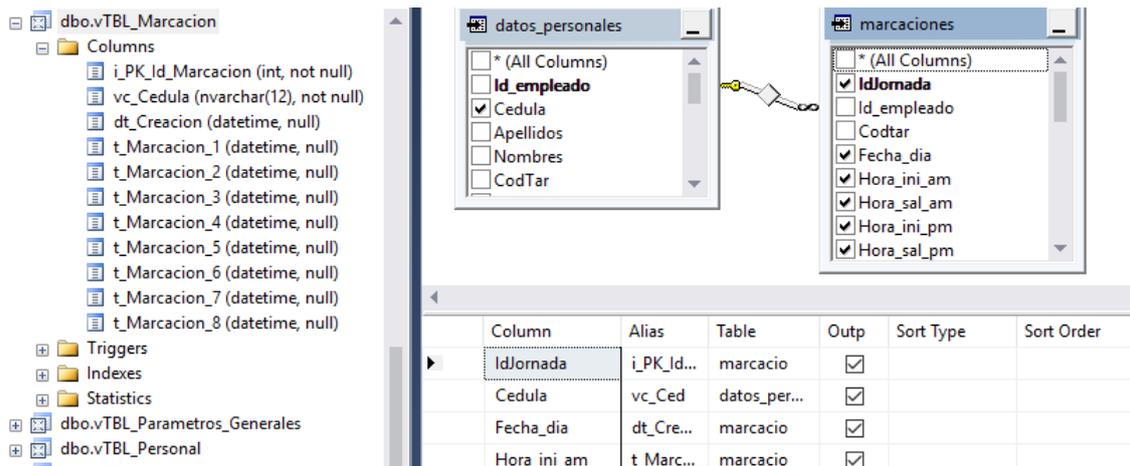


Figura 27 Vista para obtener las marcaciones de todos los trabajadores

Una vez creada las vistas necesarias en la base de datos para los reportes solicitados por el cliente se procedió a crear el microservicio de biométrico y a su vez realizar las pruebas de comunicación con el middleware de mensajería, en la siguiente figura se muestra un ejemplo de la comunicación con el con el middleware para la consulta del reporte por trabajador.



Figura 28 Pruebas de comunicación con el middleware de mensajería para los reportes de biométrico

### 3.5.6. Fase 3 módulo de reportes de uso de la plataforma aula virtual

Para el desarrollo de este módulo se utilizó una base de datos NoSQL, con la finalidad de mantener un histórico de la información generada por la plataforma aula virtual, en vista de que en la plataforma las materias o aulas al culminar el semestre en la mayoría de los casos se asignan a otros docentes y esto ocasiona la sobre escritura de

datos, con lo cual se pierde la información de los semestres anteriores. En la siguiente figura se muestra la colección Mongo en la cual existe dos documentos, en el primer documento se almacena toda la información generada por la plataforma durante el semestre 2018 I y en el segundo la información del semestre 2017 II.

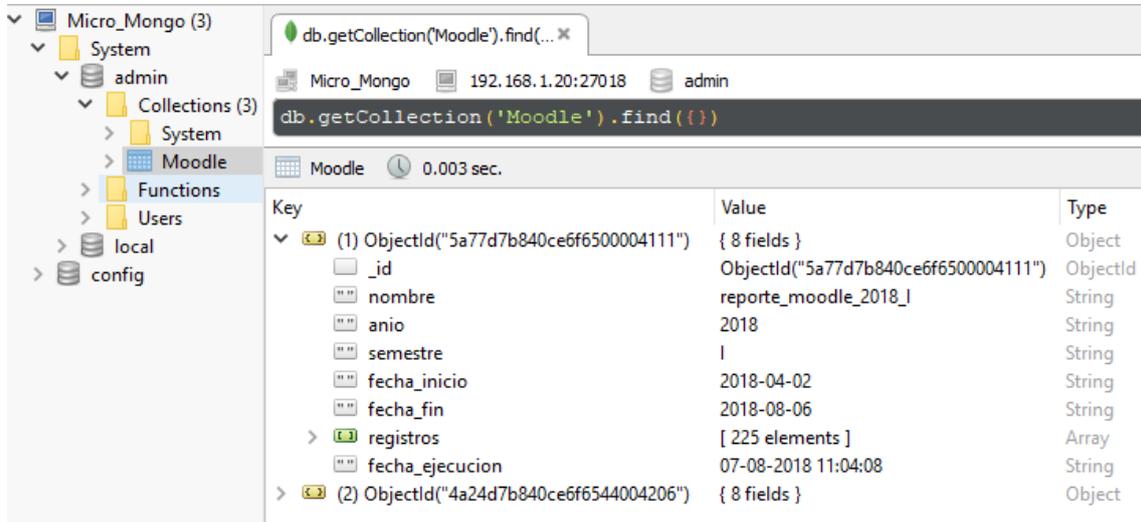


Figura 29 Colección Moodle en MongoDB

Luego de levantar la base de datos se procedió a crear el microservicio Moodle API, en el cual se siguieron los mismos pasos que en las fases anteriores. En la siguiente figura se muestra la interfaz de inicio de sesión y la interfaz que muestra la lista de colecciones disponibles en nuestra base de datos.

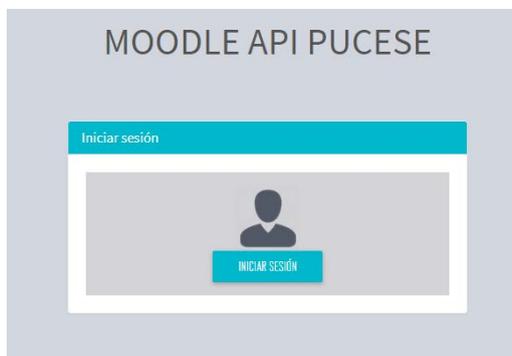


Figura 30 Interfaz de inicio de sesión en microservicio Moodle

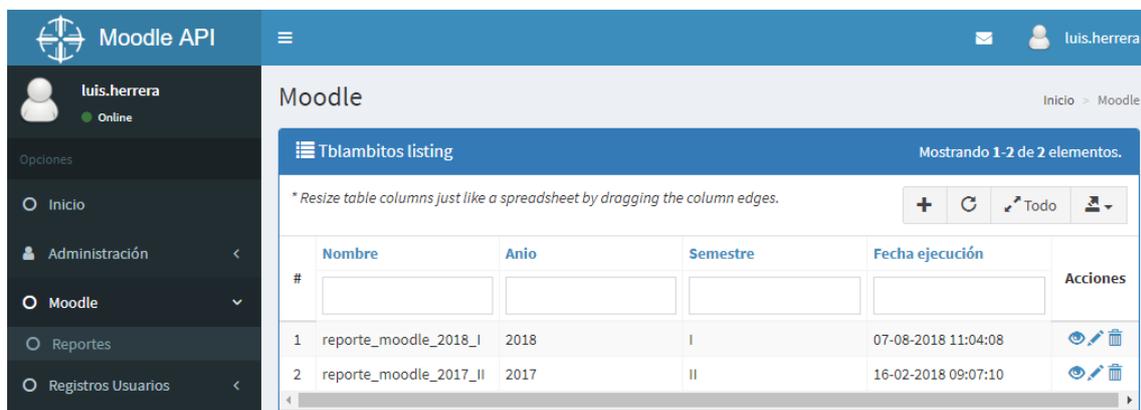


Figura 31 Interfaz de lista de documentos en la colección Moodle

La funcionalidad de este microservicio es retornar información de la colección Moodle a los otros servicios que hagan peticiones. En la siguiente figura se muestra un ejemplo de retorno de información cuando se consume el microservicio.

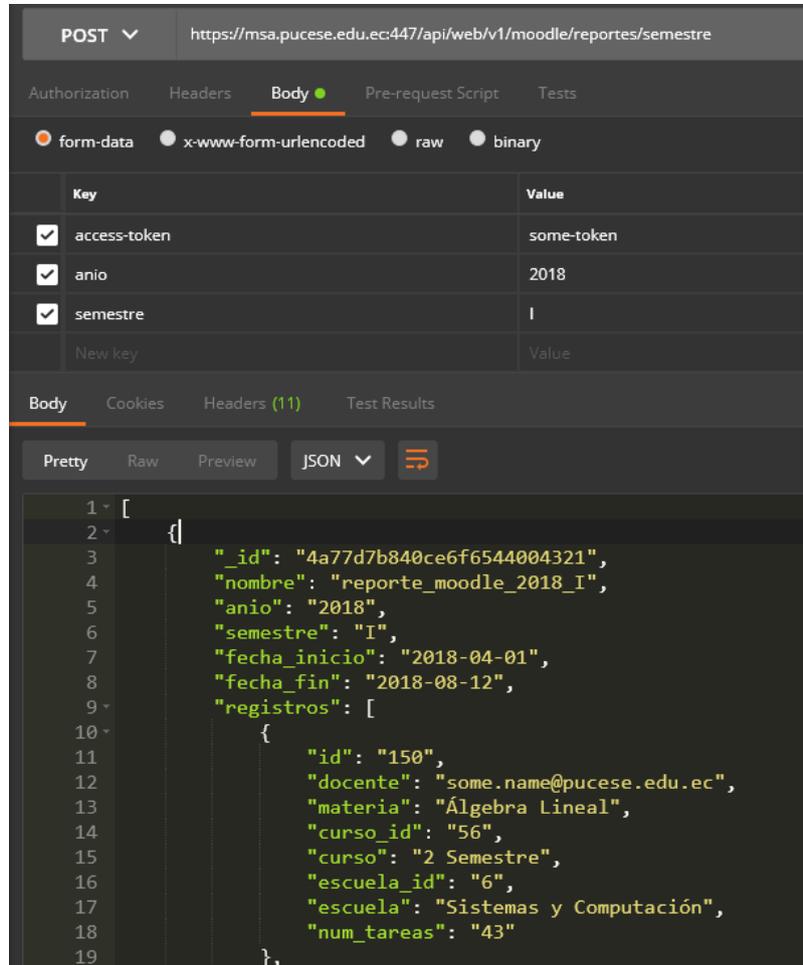


Figura 32 Respuesta del microservicio reportes moodle

### 3.5.7. Integración de los módulos a la intranet.

Luego de culminar los módulos se procedió establecer la conexión de los mismos con la aplicación central denominada intranet. Para que la aplicación intranet sepa cuál es la dirección de cada microservicio se utilizó un archivo de configuración denominado `params.php` donde se almacenaron las distintas URLs, en la siguiente figura se muestra la lista de los servicios a los que se conecta la aplicación intranet mediante REST API.

```

'api-evaluacion-laboral' =>
[
  'clientSecret' => '77rZZqNalklSA56kJYfypMhCZjg5q16d',
  'urlAsigEvaluaciones' => 'https://msa.pucese.edu.ec:444/api/web/v1/squarenet/asignacion-evaluaciones',
  'urlAsigEvalGetRegistro' => 'https://msa.pucese.edu.ec:444/api/web/v1/squarenet/asignacion-evaluaciones/get-registry',
  'urlEvalActualiza' => 'https://msa.pucese.edu.ec:444/api/web/v1/squarenet/evaluaciones/actualiza-evaluacion',
  'urlEvaluaciones' => 'https://msa.pucese.edu.ec:444/api/web/v1/squarenet/evaluaciones',
  'urlPreguntas' => 'https://msa.pucese.edu.ec:444/api/web/v1/squarenet/preguntas',
  'urlAmbitos' => 'https://msa.pucese.edu.ec:444/api/web/v1/squarenet/ambitos',
  'urlResultados' => 'https://msa.pucese.edu.ec:444/api/web/v1/squarenet/resultados',
  'urlEvalResultados' => 'https://msa.pucese.edu.ec:444/api/web/v1/squarenet/resultados/result',
  'urlJefesOrganigrama' => 'https://msa.pucese.edu.ec:444/api/web/v1/squarenet/jefes-organigrama',
  'urlParesOrganigrama' => 'https://msa.pucese.edu.ec:444/api/web/v1/squarenet/pares-organigrama',
  'urlSubordinadosOrganigrama' => 'https://msa.pucese.edu.ec:444/api/web/v1/squarenet/subordinados-organigrama',
  'urlTrabajadorOrganigrama' => 'https://msa.pucese.edu.ec:444/api/web/v1/squarenet/trabajador-organigrama',
  'urlPeriodoActivo' => 'https://msa.pucese.edu.ec:444/api/web/v1/squarenet/periodos/periodo-activo',
  'urlGrupos' => 'https://msa.pucese.edu.ec:444/api/web/v1/squarenet/grupos',
  'urlGruposList' => 'https://msa.pucese.edu.ec:444/api/web/v1/squarenet/grupos/list',
  'urlCuentaEvaluacionesTipo' => 'https://msa.pucese.edu.ec:444/api/web/v1/squarenet/cuenta-evaluaciones-tipo',
  'urlConfiguracionFechas' => 'https://msa.pucese.edu.ec:444/api/web/v1/squarenet/configuracion-fechas',
  'urlResultadosEvaluaciones' => 'https://msa.pucese.edu.ec:444/api/web/v1/squarenet/resultados-evaluaciones',
  'urlReportesEvaluaciones' => 'https://msa.pucese.edu.ec:444/api/web/v1/squarenet/reportes-evaluaciones',
  'urlTipoEvaluacionList' => 'https://msa.pucese.edu.ec:444/api/web/v1/squarenet/tipo-evaluaciones/list',
  'urlDepartamentos' => 'https://msa.pucese.edu.ec:444/api/web/v1/squarenet/departamentos',
],

'api-indicadores' =>
[
  'clientSecret' => '77rZZqNalklSA56kJYfypMhCZjg5q16d',
  'urlMoodleReporteSemestre' => 'https://msa.pucese.edu.ec:447/api/web/v1/moodle/reportes/semestre',
],

```

Figura 33 Lista de URLs de los microservicios a los que se conecta la aplicación intranet.

### 3.5.7.1. Autenticación en la intranet.

Para la autenticación se implementó un módulo híbrido como se mencionó en los apartados anteriores. La primera vez que un usuario inicia sesión en la aplicación intranet con el correo de la institución, en la pantalla aparece un formulario de confirmación donde el usuario acepta que la aplicación pueda acceder a su información como: datos del perfil y correo. En la siguiente figura se muestra el formulario de confirmación.

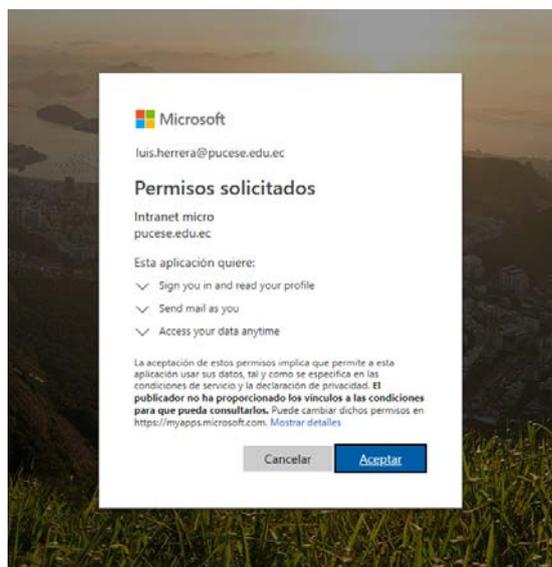


Figura 34 Formulario de aceptación de acceso a la cuenta del usuario.

Realizar este proceso por parte del usuario es importante para darle más funcionalidades a la aplicación intranet como: leer o enviar correos y conocer información relevante de la institución mediante noticias que son enviadas a los distintos grupos. Luego de iniciar sesión con el correo dentro de la aplicación se hace el otro control basado en el rol que tenga asignado el usuario en el directorio activo de la institución, el acceso a las diferentes interfaces se configuró mediante la implementación de la función behaviors dentro de cada controlador de la aplicación intranet, en la siguiente figura se muestra el controlador de evolución laboral y los permisos que tienen asignados los diferentes roles.

```
public function behaviors()  
{  
    return [  
        'access' => [  
            'class' => AccessControl::className(),  
            'rules' => [  
                [  
                    'actions' => ['reportes'], // vista reportes  
                    'allow' => true,  
                    'roles' => ['INTR_YII2_Admin', 'INTR_YII2_RRHH'],  
                ],  
                [  
                    'actions' => ['office365'], // iniciar sesión con correo  
                    'allow' => true,  
                    'roles' => ['?'], // usuario que todavía no inicia sesión  
                ],  
                [  
                    'actions' => ['index', 'login'], // vista principal para evaluar  
                    'allow' => true,  
                    'roles' => ['INTR_YII2'],  
                ],  
                [  
                    'actions' => ['logout'], // cerrar sesión  
                    'allow' => true,  
                    'roles' => ['INTR_YII2'],  
                ],  
            ],  
        ],  
    ];  
}
```

Figura 35 Control de acceso a las distintas interfaces basado en el rol del directorio activo local

### 3.5.7.2. Interfaces de la aplicación intranet.

La aplicación intranet es a la que todos los usuarios de la institución van a acceder para obtener información o herramientas que les permita realizar sus tareas cotidianas y además reportes relacionados con las actividades de sus departamentos. En la siguiente figura se muestra la interfaz para iniciar sesión en la intranet.

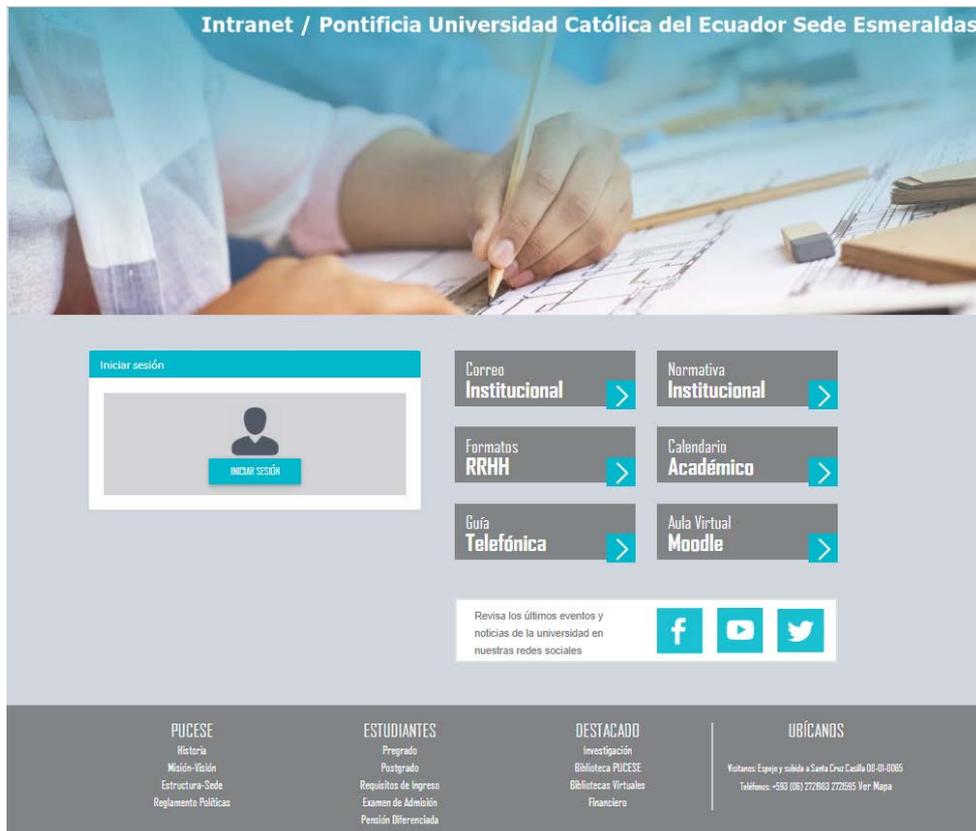


Figura 36 Interfaz de la aplicación intranet para iniciar sesión

Luego de que el usuario ha iniciado sesión en la intranet se presenta la interfaz principal para usuarios autenticado en la cual se presentara las distintas opciones de la aplicación basado en el rol que tenga asignado el usuario. En la siguiente figura se muestra la interfaz principal para el usuario administrador, el cual tiene acceso a todas las opciones de la aplicación.



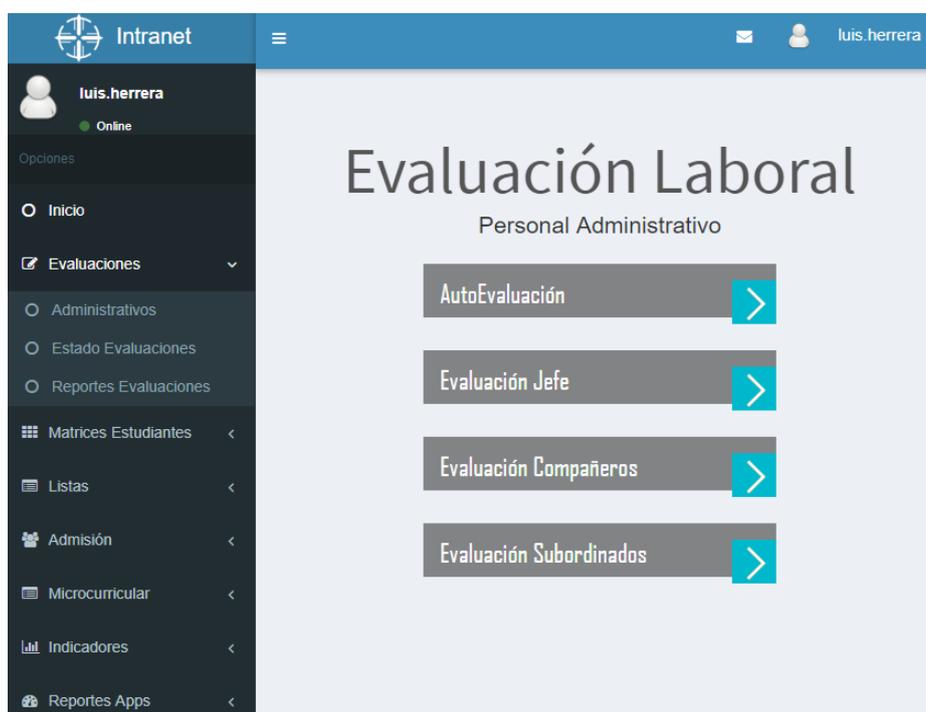
Figura 37 Interfaz principal de la aplicación intranet para usuarios autenticados

### 3.5.7.2.1 Interfaces para evaluación laboral

El proceso de evaluación laboral fue estructurado en cuatro partes las cuales son:

1. Selección del tipo de evaluación
2. Selección de la persona a evaluar
3. Responder el formulario de evaluación
4. Guardar evaluación.

Para la selección del tipo de evaluación se desarrolló la interfaz que se muestra a continuación para que de esta manera el usuario pueda seleccionar el tipo de evaluación que quiere realizar.



*Figura 38 Interfaz para selección del tipo de evaluación a realizar.*

Luego de seleccionar el tipo de evaluación, se presenta en pantalla las personas a las que el trabajador deberá evaluar, esta asignación de las personas que debe evaluar se basa en el organigrama de la institución el cual se mencionó en los apartados anteriores. A continuación, se muestra la interfaz de selección de persona a evaluar.



Figura 39 Interfaz de selección de la persona a evaluar

Una vez seleccionada la persona que el usuario quiere evaluar el sistema carga el formulario de evaluación con las respectivas preguntas y opciones de respuesta como se puede observar en la siguiente figura.

**Desempeño en el trabajo**

**APLICACIÓN DE CONOCIMIENTOS:** Las técnicas, métodos y procedimientos que el empleado utiliza en la ejecución de los trabajos asignados, son:

**INFORMES TÉCNICOS:** Capacidad para elaborar informes técnicos en el área de su especialidad, es:

**DISPOSICIÓN PARA EL TRABAJO:** La calidad de los resultados de acuerdo con los objetivos y las directrices de la unidad, es:

**ACTITUD PERSONAL:** Grado de cumplimiento de las normas y disposiciones establecidas; así como las relaciones interpersonales, así como sus relaciones con superiores, compañeros y público

**Calificación**

- Seleccione –
- Seleccione –
- Deficiente
- Regular
- Bueno
- Muy Bueno
- Sobresaliente

– Seleccione –

Figura 40 Formulario de evaluación

Luego de que el usuario ha evaluado a su compañero la opción evaluar en la interfaz de selección de la persona a evaluar se oculta, esto se implementó para que los reportes no sean adulterados luego de la fecha de evaluación.

Otra de las interfaces implementadas en la aplicación de intranet es la de: reportes de los resultados de las evaluaciones, a esta interfaz solo tienen acceso los usuarios con el rol de Administrador y jefe de RRHH. En la siguiente figura se muestra la interfaz de resultados de las evaluaciones.

**Evaluación Laboral**  
Personal Administrativo  
Resultados Evaluaciones

Resultados Evaluaciones Mostrando 1-261 de 261 elementos.

Guía porcentajes [Refresh] [Filter] [List] [Print]

[FILTAR DATOS]

Nombres Persona	Periodo	Tipo Evaluación	Departamento	Grado	Cod Nivel	Promedio	Resultado calculado %
Trabajador 1	2017	Autoevaluación	Dirección general académica	Técnico	32	4.82	4.59
Trabajador 1	2017	Subordinado	Dirección general académica	Técnico	32	4.82	4.59
Trabajador 1	2017	Servicios	Dirección general académica	Técnico	32	3.30	4.59
Trabajador 2	2017	Autoevaluación	Dirección general de recursos humanos RRHH	Servicios	12	4.44	4.02
Trabajador 2	2017	Compañeros	Dirección general de recursos humanos RRHH	Servicios	12	4.00	4.02
Trabajador 2	2017	Subordinado	Dirección general de recursos humanos RRHH	Servicios	12	4.00	4.02
Trabajador 3	2017	Autoevaluación	Dirección general de recursos humanos RRHH	Técnico	32	3.82	3.91
Trabajador 3	2017	Subordinado	Dirección general de recursos humanos RRHH	Técnico	32	3.91	3.91
Trabajador 4	2017	Autoevaluación	Dirección general financiera	Operativo	22	4.82	4.91
Trabajador 4	2017	Subordinado	Dirección general financiera	Operativo	22	4.91	4.91
Trabajador 5	2017	Autoevaluación	Dirección general académica	Técnico	32	3.91	3.65

*Figura 41 Reporte de resultados de evaluaciones*

### 3.5.7.2.2 Interfaces para reportes de marcaciones Biométrico

La interfaz de reportes de marcaciones biométrico permite obtener la información de las marcaciones de los trabajadores como se puede observar en la siguiente figura.

#	Fecha de Marcación	Marcación 1	Marcación 2	Marcación 3	Marcación 4	Total
1	2018-03-30	07:59:00	12:49:00	13:11:00		04 horas 50 minutos
2	2018-03-29	08:15:00	13:25:00	13:47:00	16:57:00	08 horas 20 minutos
3	2018-03-28	07:54:00	14:03:00	14:28:00	17:55:00	09 horas 36 minutos
4	2018-03-27	07:46:00	13:15:00	13:45:00	17:55:00	09 horas 39 minutos
5	2018-03-26	08:18:00	13:33:00	14:00:00	18:31:00	09 horas 46 minutos
6	2018-03-23	07:54:00	13:56:00	15:21:00	17:45:00	08 horas 26 minutos
7	2018-03-22	08:23:00	13:04:00	13:32:00	17:49:00	08 horas 58 minutos
8	2018-03-21	07:26:00	13:25:00	13:41:00	18:32:00	10 horas 50 minutos
9	2018-03-20	08:15:00	13:04:00	13:24:00	16:21:00	07 horas 46 minutos
10	2018-03-19	08:17:00	13:35:00	20:27:00	22:37:00	07 horas 28 minutos

Figura 42 Reporte de las marcaciones de un trabajador en el sistema biométrico

Una de las funcionalidades del sistema de reportes es que, si un trabajador acumula menos de 8 horas de trabajo, o si hace marcaciones impares, en la columna Total se muestra el total de tiempo en color rojo, caso contrario se muestra color verde. Así mismo en la parte de abajo del reporte se muestra el total de horas y días laborados durante el mes.

12	Modificar Marcación	2018-03-15	08:25:00	13:11:00	13:32:00	18:00:00	09 horas 14 minutos
13	Modificar Marcación	2018-03-14	07:48:00	13:23:00	13:41:00	17:59:00	09 horas 53 minutos
14	Modificar Marcación	2018-03-13	07:55:00	13:02:00	13:35:00	17:52:00	09 horas 24 minutos
15	Modificar Marcación	2018-03-12	08:25:00	13:52:00	14:16:00	17:57:00	09 horas 8 minutos
<b>Total de horas y días laborados durante el mes</b>							<b>133 Horas y 29 Minutos - 15 días</b>

Figura 43 Reporte total de horas y días laborados durante el mes

### 3.5.7.2.3 Interfaces para reportes de uso aula virtual

La interfaz para presentación de uso de Moodle presenta la opción de filtrar por año y por semestre, toda esta información es tomada del microservicio API-MOODLE el cual almacena la información en una base de datos NoSQL como se mencionó anteriormente. En la siguiente figura se muestra la interfaz de reporte de uso del aula virtual.

Docente	Materia	Curso	Escuela	# Tareas
docente7@pucese.edu.ec	IS211 - Electrología y Circuitos Lógicos - Ing. Manuel Nevárez	3 Semestre	SISTEMAS Y COMPUTACIÓN	106
docente8@pucese.edu.ec	IS211 - Electrología y Circuitos Lógicos - Ing. Manuel Nevárez	3 Semestre	SISTEMAS Y COMPUTACIÓN	106
docente9@pucese.edu.ec	IS825 - Sistemas de Información Geográfica - Mgt. Juan Casierra	9 Semestre	SISTEMAS Y COMPUTACIÓN	87
docente7@pucese.edu.ec	Laboratorio de Física II	2 Semestre	SISTEMAS Y COMPUTACIÓN	70
docente10@pucese.edu.ec	Procesamiento de Imágenes	7 Semestre	SISTEMAS Y COMPUTACIÓN	45
docente10@pucese.edu.ec	Álgebra Lineal	2 Semestre	SISTEMAS Y COMPUTACIÓN	43
docente9@pucese.edu.ec	IS703 - Redes - Mgt. Juan Casierra	7 Semestre	SISTEMAS Y COMPUTACIÓN	39
docente9@pucese.edu.ec	SIG - Mgt. Juan Casierra	6 Semestre	GESTIÓN AMBIENTAL	38
docente11@pucese.edu.ec	Estructura de Datos I	3 Semestre	SISTEMAS Y COMPUTACIÓN	36
docente7@pucese.edu.ec	Física II	2 Semestre	SISTEMAS Y COMPUTACIÓN	32

Figura 44 Interfaz de reporte de uso del aula virtual.

### 3.5.8. Implementación de CI y CD

Para el llevar acabo CI y CD en el proyecto se siguieron los siguientes pasos:

1. Realizar control de versiones de los proyectos
2. Pruebas automatizadas
3. Imágenes multicapas de los proyectos mediante la utilización de Docker
4. Orquestación de contenedores
5. Configuración de Pipelines en Jenkins

En las siguientes secciones se va a detallar las actividades realizadas en cada uno de los puntos mencionados anteriormente.

#### 3.5.8.1 Control de versiones de los proyectos

Para llevar un control de los cambios realizados en los proyectos y a su vez para la implementación de CI y CD es de vital importancia la implementación de control de versiones en el proyecto, así mismo es necesario la utilización de servicio de alojamiento basado en web para que los proyectos con sus respectivas actualizaciones puedan ser descargados en equipos que posean estos privilegios. Actualmente hay dos servicios de alojamientos populares basados en web los cuales son: GitHub y Bitbucket. Para el proyecto se utilizaron los servicios de Bitbucket esto es porque permite crear repositorios privados de manera gratuita a diferencia de GitHub que sólo permite crear repositorios públicos de manera gratuita, aunque también ofrece la opción de crear repositorios

privados, pero para esto se deberá pagar suscripción mensual. En la siguiente figura se muestra los repositorios creados en Bitbucket para el desarrollo del proyecto.

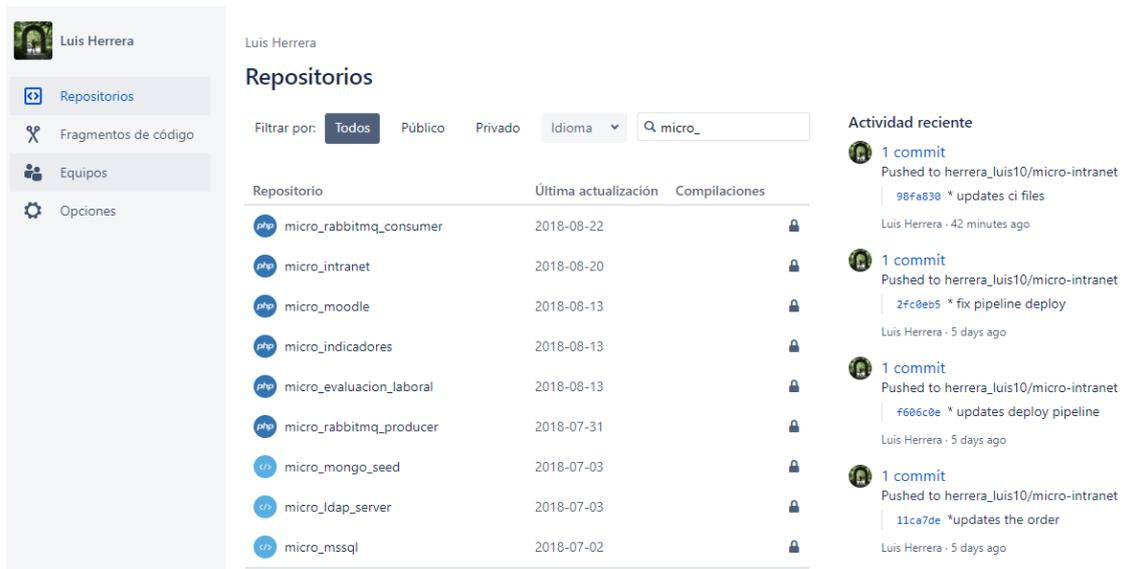


Figura 45 Repositorios creados en Bitbucket

### 3.5.8.2 Pruebas automatizadas

En los procesos de CI para probar que los nuevos cambios o funcionalidades que se agregan al sistema no afecta las funcionalidades que se habían desarrollado previamente es necesario implementar pruebas automatizadas. En CI los cambios se agregan al sistema luego de que se haya realizado un commit y un push al repositorio de control de versiones y que al integrarlo a la aplicación el proyecto pasa todas las pruebas automatizadas, así mismo el éxito y fiabilidad de las pruebas automatizadas está dado por su constante actualización conforme el proyecto avanza. En la siguiente figura se muestra un ejemplo del resultado de la ejecución de las pruebas automatizadas implementadas en el proyecto intranet.

```

clon@jedi:~$ docker exec -i mystack intranet.1.rj9c0cmlwqexag8d213uu3s /var/www/html/vendor/bin/codecept run --env dev -c /var/www/html/
Codeception PHP Testing Framework v2.1.11
Powered by PHPUnit 5.3.5 by Sebastian Bergmann and contributors.

Acceptance (dev) Tests (3) -----
Probar ambiente AulaVirtual (AulaVirtualCept)           ok
Probar ambiente Biometrico (BiometricoCept)            ok
Probar ambiente Evaluacion Laboral (EvaluacionLaboralCept) ok
-----
Unit Tests (0) -----
-----
Functional Tests (0) -----
-----
Time: 21.26 seconds, Memory: 27.86MB

OK (0 tests, 35 assertions)

```

Figura 46 Resultado de ejecución de pruebas automatizadas en el proyecto intranet

### 3.5.8.3 Imágenes multicapas de los proyectos mediante la utilización de Docker

Una de las ventajas de la creación de imágenes Docker basado en multicapas es que nos permite separar el motor de arranque del sistema con el código fuente, en este

sentido dado que nuestra arquitectura está basada en el lenguaje de programación PHP la cual es servida al cliente por el servidor web Apache, se realizó la separación en capas al momento de construir nuestras imágenes como se puede apreciar en la siguiente figura.

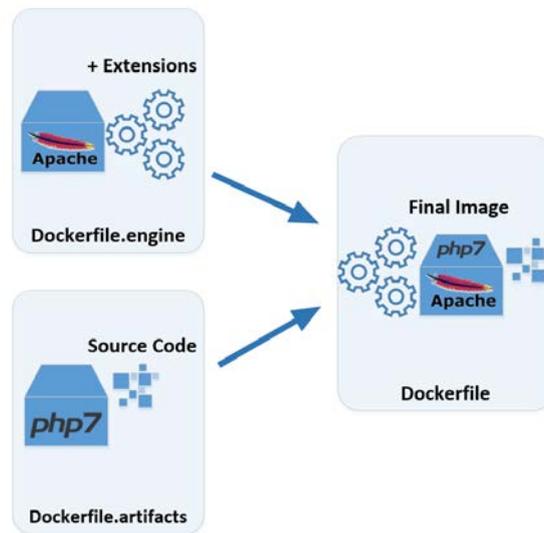


Figura 47 Imágenes Docker multicapas

La figura anterior muestra como nuestro archivo Dockerfile hace uso de dos imágenes que fueron construidas con anterioridad para la construcción de la imagen final en la que se une el motor web que tiene todas las extensiones necesarias para el correcto funcionamiento de nuestro proyecto con el código fuente del proyecto, en la siguiente figura se muestra el contenido de nuestro archivo Dockerfile para el proyecto intranet.

```
Dockerfile
1 FROM herreraluis/micro_intranet_artifacts as intranet-artifacts
2 FROM herreraluis/micro_web_engine as intranet-engine
3
4 COPY --from=intranet-artifacts /var/www/html /var/www/html
5 COPY --from=intranet-artifacts /etc/apache2/ /etc/apache2/
6 COPY --from=intranet-artifacts /usr/local/bin/a2f /usr/local/bin/a2f
7
8 RUN chmod 777 -R /var/www/html/
9
10 EXPOSE 80 443
11 CMD ["/usr/local/bin/a2f"]
```

Figura 48 Archivo Dockerfile del proyecto intranet

Al igual que el servidor de control de versiones las imágenes Docker se pueden ser alojadas en la nube, esto también es un proceso necesario para realizar CI y CD en nuestra MSA, para el alojamiento de nuestras imágenes se utilizó los servicios de DockerHub, en la siguiente figura se puede observar nuestras imágenes alojadas en la nube.

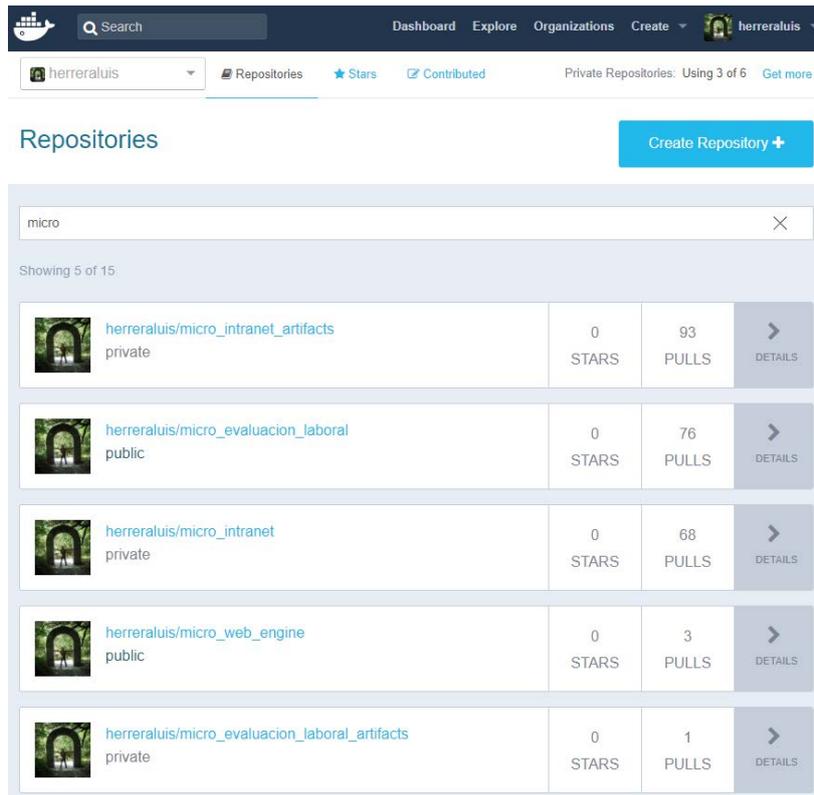


Figura 49 Imágenes del proyecto alojadas en la nube

### 3.5.8.4 Orquestación de contenedores

Para el levantamiento de nuestra MSA basada en contenedores se utilizó las herramientas docker-compose y Docker en modo swarm. Es necesario mencionar que para el funcionamiento de nuestra arquitectura fue necesario levantar sistemas terceros como los que mencionan a continuación:

- Motor de Base de datos SQL Server
- Servidor Jenkins
- Servidor RabbitMQ
- Servidor LDAP
- Motor de Base de datos MongoDB
- Servidor Grafana
- Servidor AlertManager

Otro aspecto importante a tomar en cuenta en nuestra arquitectura, fue la compartición de datos entre los distintos servidores y la comunicación entre los distintos microservicios, esta problemática fue solucionada con la utilización de volúmenes y una red interna que permite configurar Docker en modo swarm. En el proyecto la red creada para la comunicación entre los distintos nodos y microservicios se denominó: micro\_net. Luego de crear la red en el orquestador de contenedores, cuando se define los servicios en el archivo docker-compose.yml se debe ingresar en el atributo o parámetro networks

el nombre de la red, en la siguiente figura se muestra un ejemplo de la definición del servicio mssql que tiene configurada la red micro\_net y el volumen compartido.

```

docker-compose.yml
1  version: '3.6'
2  services:
3  mssql:
4      container_name: mssql_db
5      build: ./mssql/.
6      image: herreraluis/micro_mssql
7      restart: always
8  environment:
9      - ACCEPT_EULA=Y
10     - MSSQL_SA_PASSWORD=ACCESS_db_5555
11     - MSSQL_PID=Developer
12  ports:
13     - "1433:1433"
14  networks:
15     - micro_net
16  deploy:
17     replicas: 1
18  volumes:
19     - mssql_volume:/var/opt/mssql/data/

```

Figura 50 Definición del servicio mssql con la red micro\_net y volumen compartido

Cuando se orquesta servicios utilizando Docker en modo swarm es necesario establecer un nombre de pila con el que se pueda identificar a los servicios, para nuestro proyecto el nombre de la pila que se estableció fue: mystack. En la siguiente figura se muestran los servicios que están corriendo en nuestra arquitectura.

```

clon@jedi:/media/sf_htdocs/proyectos_tesis$ docker service ls
ID                NAME                MODE                REPLICAS                IMAGE
4jeqcbnre64j     mystack_alertmanager  replicated          1/1                     stefanprodan/swarmprom-alertmanager:v0.14.0
inwu73snk3sa     mystack_caddy         replicated          1/1                     stefanprodan/caddy:latest
owwrskf8m3h0     mystack_advisor      global             1/1                     google/cadvisor:latest
z3e4sjz2b1jg     mystack_dockerd-exporter  global            1/1                     stefanprodan/caddy:latest
c36xhq60seoc     mystack_evaluacion_laboral  replicated         1/1                     herreraluis/micro_evaluacion_laboral:latest
lglwmtscwc4d     mystack_grafana       replicated          1/1                     stefanprodan/swarmprom-grafana:5.0.1
16xeuknitz5x     mystack_indicadores    replicated          1/1                     herreraluis/micro_indicadores:latest
yr9mgzuyfs50     mystack_intranet      replicated          1/1                     herreraluis/micro_intranet:latest
19yy8lphuo1z     mystack_jenkins       replicated          1/1                     herreraluis/micro_jenkins:latest
xsb34744n871     mystack_ldap          replicated          1/1                     herreraluis/micro_ldap:latest
s49abpuwcmwn     mystack_mongodb       replicated          1/1                     mongo:4.0.0-xenial
1jto4cxtcc91     mystack_mongodb-seed  replicated          0/1                     herreraluis/mongodb-seed:latest
xz4s99wpsodj     mystack_moodle        replicated          1/1                     herreraluis/micro_moodle:latest
c5qv9gcyod15     mystack_mssql         replicated          1/1                     herreraluis/micro_mssql:latest
zpwfo4ajruog     mystack_node-exporter  global             1/1                     stefanprodan/swarmprom-node-exporter:v0.15.2
mldpdy1ewc0a     mystack_portal_ldap   replicated          1/1                     osixia/phpldapadmin:0.7.1
hndluga492cl     mystack_prometheus    replicated          1/1                     stefanprodan/swarmprom-prometheus:v2.2.0-rc.0
zqdr46i43548     mystack_rabbitmq      replicated          1/1                     herreraluis/micro_rabbitmq:latest
6779762wyn7p     mystack_unsee         replicated          1/1                     cloudflare/unsee:v0.8.0

```

Figura 51 Lista de servicios corriendo

### 3.5.8.5 Configuración de Pipelines en Jenkins

Los pipelines de Jenkins nos permiten definir el flujo de CI y CD mediante stages, estos son definidos en un fichero utilizando el lenguaje Groovy, una de las ventajas que ofrecen los pipelines es que permiten visualizar de forma atractiva e intuitiva dándonos un feedback rápido del proceso. En la siguiente figura se muestra el uno de los pipelines implementados en nuestro proyecto el cual se denomina build-image.

## Pipeline build-image



Desactivar el Proye...

### Stage View

	Checkout code	Build Artifacts Docker Image	Build Docker Image	Push Artifacts Image	Push Image	Send to Deploy New Images Docker
Average stage times: (Average full run time: ~4min 14s)	3min 3s	28s	11s	15s	12s	123ms
#5 Aug 29 22:04 1 commit	3min 5s					
#4 Aug 29 21:27 No Changes	4min 2s	22s	6s	11s	10s	144ms
#3 Aug 28 23:10 No Changes	1min 56s	14s	3s	7s	4s	75ms
#2 Aug 28 22:33 No Changes	2min 37s	10s	3s	6s	5s	77ms
#1 Aug 28 22:03 No Changes	3min 33s	55s	33s	37s	30s	197ms

Figura 52 Pipeline build-image

El pipeline build-image se encarga de tomar la última versión del código, actualizar la imagen de los artefactos, luego construye la imagen final del proyecto uniendo la imagen del motor web con los artefactos para proceder a enviarla al repositorio de imágenes, finalmente en el último paso ejecuta el pipeline para hacer deploy de la nueva imagen en el nodo principal de nuestra MSA. Uno de los pipelines importantes en nuestro servidor de CI y CD es el de micro-intranet puesto que este se encarga de ejecutar las pruebas automatizadas en la nueva imagen construida previamente y con esto verifica que los nuevos cambios implementados no afectan al proyecto. En la siguiente figura se muestra nuestro pipeline micro-intranet.

## Pipeline micro-intranet



### Stage View

	Checkout code	Running Tests
Average stage times: (Average full run time: ~2min 23s)	5s	30s
#5 Aug 28 23:13 No Changes	3s	27s
#4 Aug 28 22:37 No Changes	3s	28s
#3 Aug 28 22:21 No Changes	4s	27s
#2 Aug 28 21:54 No Changes	12s	36s

Figura 53 Pipeline micro-intranet.

En el proyecto hay varios pipelines los cuales están comunicados entre sí, al ser varios los pipelines implementados es necesario buscar una manera de ver de manera resumida el estado de los mismo ya que estos nos indican como se están dando las integraciones y los despliegues del proyecto, para solucionar esta problemática en Jenkins se instaló el complemento: build monitor view, este nos permitirá ver de manera resumida y por colores (verde: bien, rojo: mal) el estado de nuestros pipelines, en la siguiente figura se muestra la interfaz del complemento mencionado anteriormente.

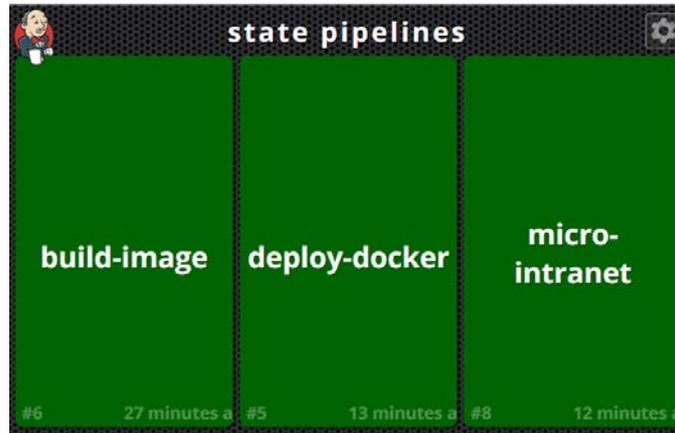


Figura 54 Resumen de estado de nuestros pipelines

### 3.5.9. Implementación de monitoreo y comunicación.

Para el monitoreo de la MSA se implementó el sistema de monitoreo gráfico Grafana, esta es una herramienta de código libre que permite visualizar de forma elegante datos o métricas relacionados con nuestra arquitectura como: consumo de RAM, CPU, Disco, el tráfico de red de los servicios y el estado de cada uno. En la siguiente figura se muestra el Dashboard principal creado para el monitorio de nuestra infraestructura.

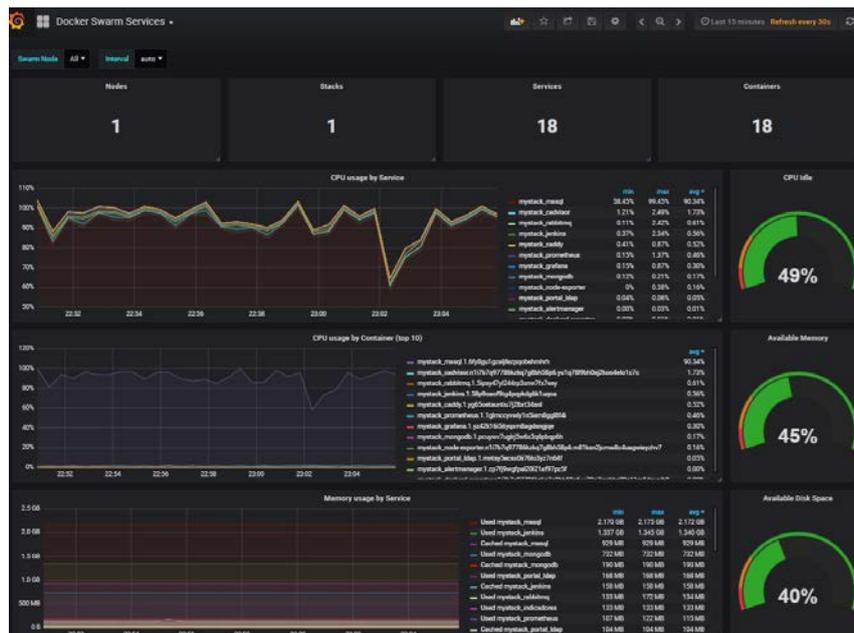


Figura 55 Dashboard principal de nuestro sistema de monitoreo Grafana

Así mismo para la comunicación de incidencia se utilizó la herramienta Alert Manager la cual se encarga de analizar alertas o eventos notables disparadas por nuestra herramienta de monitoreo y tiene la capacidad de enviarlas a centros de comunicación de espacios de trabajos, en nuestro caso utilizamos Slack como herramienta de comunicación para el departamento de TIC's. En la siguiente figura se muestra las alertas enviadas a nuestro centro de comunicación por Alert Manager.

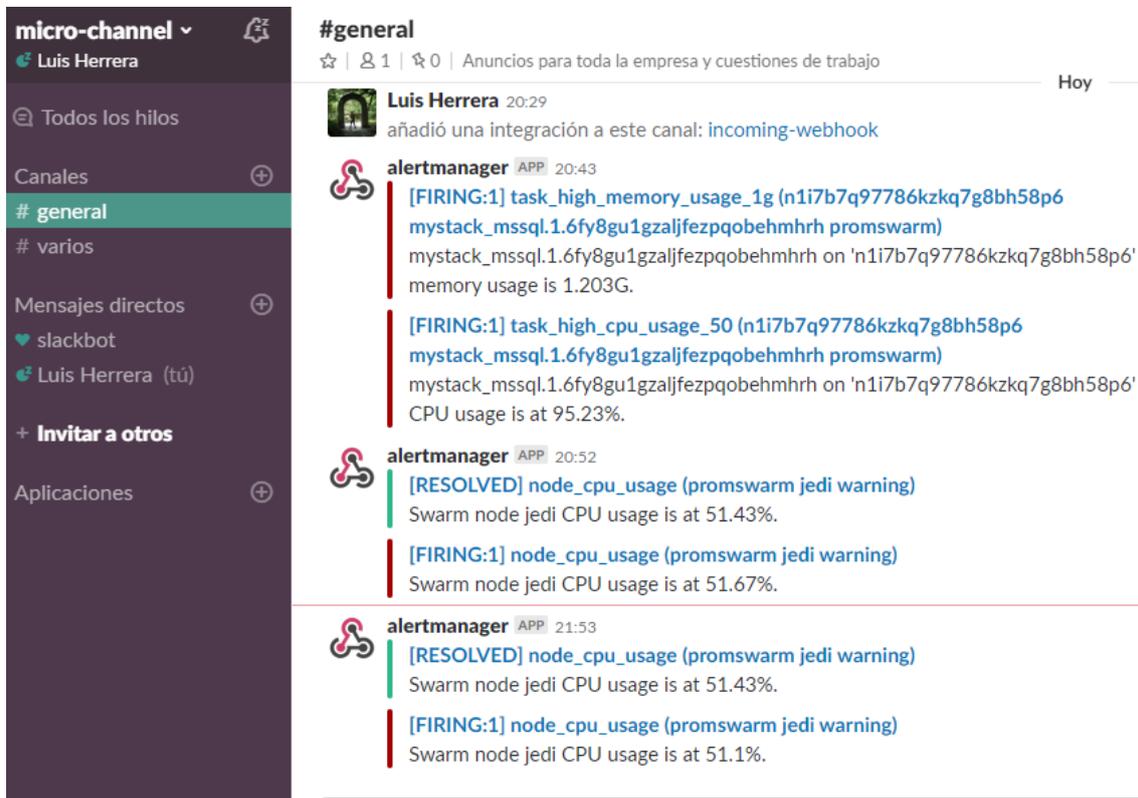


Figura 56 Alertas enviadas al canal de comunicación general por Alert Manager

En la figura mostrada anteriormente se puede observar los mensajes de alertas enviados al canal general, estas alertas están relacionadas por el consumo excesivo de CPU por parte de uno de los microservicios en nuestra arquitectura, en este caso se trata del microservicio mssql (SQL Server). Si regresamos a nuestro sistema de monitoreo podemos verificar el mensaje de alerta que esta enviado el Alert Manager.

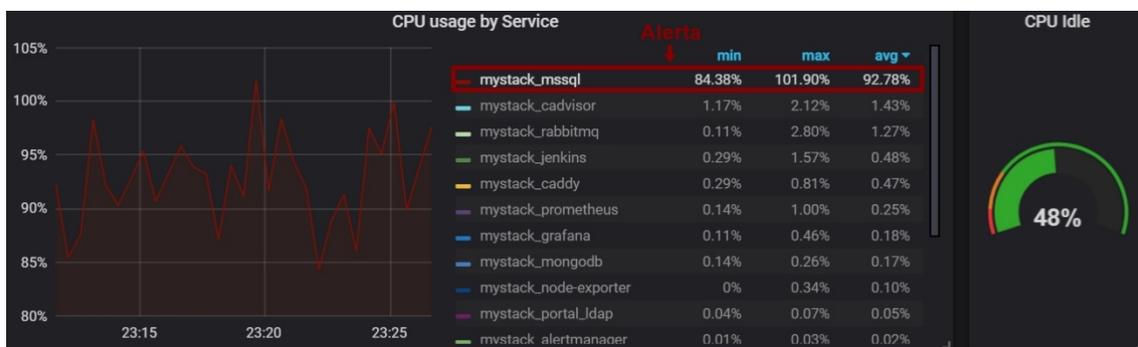


Figura 57 Reporte de consumo de CPU por el microservicio mssql

## Capítulo 4: Evaluación de la solución

En esta sección se va analizar las opiniones de los clientes sobre las metodologías implementadas para el desarrollo del proyecto. Para evaluar la nueva metodología utilizada con los clientes se elaboraron encuestas en las que se pudo recolectar la siguiente información.

*Tabla 5 Evaluación de la solución implementada para el desarrollo del proyecto*

Preguntas	Cumplimiento de las necesidades del cliente				Usabilidad de la aplicación				Tiempo de entrega del proyecto				Gestión de cambios o modificaciones				
	Muy insatisfactorio	Insatisfactorio	Aceptable	Satisfactorio	Muy insatisfactorio	Insatisfactorio	Aceptable	Satisfactorio	Muy insatisfactorio	Insatisfactorio	Aceptable	Satisfactorio	Muy insatisfactorio	Insatisfactorio	Aceptable	Satisfactorio	Muy insatisfactorio
Valoración Sistemas																	
Evaluación Laboral				X					X				X				X
Reportes Biométrico			X						X				X		X		
Reportes Aula Virtual				X					X				X				X

En la tabla mostrada anteriormente se observa el resultado de la valoración en cada sistema desarrollado bajo la nueva metodología, en esta evaluación participaron los clientes directos de cada sistema los cuales fueron:

- Evaluación Laboral: Jefe de RRHH
- Reportes Biométrico: Jefe de RRHH
- Reportes de Aula Virtual: Director de Escuela

Como resultado general se puede observar que los clientes están muy satisfechos con la nueva metodología de desarrollo implementada en el departamento de TIC's para desarrollar los nuevos sistemas, puesto que ha cumplido con sus expectativas, la usabilidad que tienen las aplicaciones hacen que sea fácil de entender sus funciones, los tiempos de entrega de los sistemas se realizaron acorde a lo planeado y en cuanto a la gestión de cambio o modificaciones durante el desarrollo, la metodología implementada hizo que sea fácil de adaptar al proyecto.

## Capítulo 5: Conclusiones y trabajos futuro

Implementar una intranet bajo MSA permitió desarrollar el proyecto de manera modular y de manera que el cliente pudo evidenciar las nuevas funcionalidades que se agregaban en cada entrega, así mismo la implementación de las metodologías XP para desarrollar el proyecto y Scrum para la estimación del esfuerzo por parte del equipo de desarrollo, con lo cual se pudo clasificar el proyecto en iteraciones fue algo importante porque permitió estructurar, estimar y compartir conocimiento en el equipo de desarrollo de una manera más adecuada. Otro aspecto importante en el desarrollo del proyecto fue la utilización de la tecnología de contenedores, puesto que permitió preparar los ambientes de desarrollo y ponerlos en producción en periodos cortos de tiempos, esto es porque la configuración se hacía una sola vez, esta configuración se almacenaban en imágenes y luego se utilizaban en los distintos ambientes de desarrollo y producción.

En cada fase del proyecto se pudo entregar los módulos acordados con el cliente, el mismo que pudo ver los beneficios de manera inmediata puesto que le permitió obtener reportes en tiempo real para la toma de decisiones.

En una MSA realizar las distintas funciones del sistema de manera modular es algo vital para su implementación, es por esta razón que el monitoreo y la implementación de procesos CI y CD son aspectos claves que se deben tener en cuenta al momento de desarrollar un sistema bajo esta arquitectura, en nuestro caso la utilización de Grafana para el monitoreo y Jenkins para los procesos automatizados de CI y CD permitieron solventar esta problemática y generaron un gran aporte tanto para el cliente como para el equipo de desarrollo.

En trabajos futuros tendría un gran valor para el proyecto agregar más tipos de pruebas automatizadas basándose en la pirámide de pruebas (unitarias, componentes, contrato y funcionales), esto daría más valor al producto puesto que la calidad estaría garantizada al cubrir toda la pirámide de pruebas. Así mismo sería interesante agregar más tipos de reportes de los sistemas como: contar las veces que los clientes acceden a los distintos reportes, atrapar errores, contar la cantidad de veces que los reportes son descargados, entre otros. Estos reportes permitirían conocer más a fondo las necesidades reales del cliente y a su vez con los reportes de errores conocer por parte del equipo de desarrollo los errores ocultos que posee el sistema o bajo que parámetros o circunstancias estos ocurren.

## Bibliografía.

- [1] B. Morschheuser, C. Henzi, and R. Alt, “Increasing intranet usage through Gamification - Insights from an experiment in the banking industry,” *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, vol. 2015–March, pp. 635–642, 2015.
- [2] H.-J. Bullinger, J. Ziegler, and W. Bauer, “Intranets and Organizational Learning: A Research and Development Agenda,” *Int. J. Hum. Comput. Interact.*, vol. 14, no. 1, pp. 1–23, 2002.
- [3] K. Steele, “Case study: Developing an intranet web site for information security: The evolution of a corporate intranet web site in the age of cyberwarfare,” *IEEE Int. Prof. Commun. Conf.*, 2013.
- [4] Y. Ye, W. Sun, S. Ning, C. Guo, and W. Xi, “Design of intranet information security management,” *Proc. - 2016 Int. Conf. Robot. Intell. Syst. ICRIS 2016*, pp. 155–158, 2016.
- [5] J. Kadlec, D. Jaros, and R. Kuchta, “Implementation of an advanced authentication method within microsoft active directory network services,” *Proc. - 6th Int. Conf. Wirel. Mob. Commun. ICWMC 2010*, pp. 453–456, 2010.
- [6] C.-H. Hsieh, C.-M. Lai, C.-H. Mao, T.-C. Kao, and K.-C. Lee, “AD2: Anomaly detection on active directory log data for insider threat monitoring,” *2015 Int. Carnahan Conf. Secur. Technol.*, pp. 287–292, 2015.
- [7] G. Westerman, D. Bonnet, and A. McAfee, *Leading Digital: Turning Technology into Business Transformation*. Harvard Business Review Press, 2014.
- [8] J. Bogner and A. Zimmermann, “Towards Integrating Microservices with Adaptable Enterprise Architecture,” in *Proceedings - IEEE International Enterprise Distributed Object Computing Workshop, EDOCW*, 2016.
- [9] C. M. Aderaldo, N. C. Mendonça, C. Pahl, and P. Jamshidi, “Benchmark Requirements for Microservices Architecture Research,” in *Proceedings - 2017 IEEE/ACM 1st International Workshop on Establishing the Community-Wide Infrastructure for Architecture-Based Software Engineering, ECASE 2017*, 2017.
- [10] L. Zhu, D. Csiro, and G. Champlin-scharff, “DevOps and Its Practices,” pp. 32–34, 2016.
- [11] Z. Xiao, I. Wijegunaratne, and X. Qiang, “Reflections on SOA and Microservices,” in *Proceedings - 4th International Conference on Enterprise Systems: Advances in Enterprise Systems, ES 2016*, 2016.
- [12] O. Tinoco Gómez, P. P. Rosales López, and J. Salas Bacalla, “Criterios de selección de metodologías de desarrollo de software,” *Ind. Data*, vol. 13, no. 2, p. 070, 2014.
- [13] Y. D. Amaya Balaguera, “Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles,” *Rev. Tecnol.*, vol. 12 número, pp. 111–124, 2013.
- [14] A. Seehorn, “Métodos de investigación transversal,” 2018. [Online]. Available: <https://www.geniolandia.com/13142481/metodos-de-investigacion-transversal>. [Accessed: 11-Feb-2018].

- [15] D. Sunner, “Agile: Adapting to need of the hour: Understanding Agile methodology and Agile techniques,” *Proc. 2016 2nd Int. Conf. Appl. Theor. Comput. Commun. Technol. iCATccT 2016*, pp. 130–135, 2017.
- [16] A. M. M. Hamed and H. Abushama, “Popular agile approaches in software development: Review and analysis,” *Proc. - 2013 Int. Conf. Comput. Electr. Electron. Eng. 'Research Makes a Differ. ICCEEE 2013*, pp. 160–166, 2013.
- [17] IBM, “Arquitectura orientada a servicios (SOA) de IBM - España.” [Online]. Available: <https://www-01.ibm.com/software/es/solutions/soa/index.html>. [Accessed: 04-Jan-2018].
- [18] N. Dragoni *et al.*, “Microservices: yesterday, today, and tomorrow,” 2016.
- [19] S. Newman, *Building Microservices*. O'Reilly Media.
- [20] L. Johansson, “Part 1: RabbitMQ for beginners - What is RabbitMQ? - CloudAMQP,” 2015. [Online]. Available: <https://www.cloudamqp.com/blog/2015-05-18-part1-rabbitmq-for-beginners-what-is-rabbitmq.html>. [Accessed: 13-Jan-2018].
- [21] Pivotal Software, “RabbitMQ - RabbitMQ tutorial - Publish/Subscribe,” 2018. [Online]. Available: <https://www.rabbitmq.com/tutorials/tutorial-three-python.html>. [Accessed: 21-Jul-2018].
- [22] Lovisa Johansson, “Part 4: RabbitMQ Exchanges, routing keys and bindings - CloudAMQP,” 2015. [Online]. Available: <https://www.cloudamqp.com/blog/2015-09-03-part4-rabbitmq-for-beginners-exchanges-routing-keys-bindings.html>. [Accessed: 26-Jul-2018].
- [23] Pivotal Software, “RabbitMQ - AMQP 0-9-1 Model Explained,” 2018. [Online]. Available: <https://www.rabbitmq.com/tutorials/amqp-concepts.html>. [Accessed: 25-Jul-2018].
- [24] Docker, “What is Docker?” [Online]. Available: <https://www.docker.com/what-docker>. [Accessed: 10-Jan-2018].
- [25] T. Miura, J. L. Wijekoon, S. Prageeth, and H. Nishi, “Novel Infrastructure with Common API using Docker for Scaling the Degree of Platforms for Smart Community Services,” *Int. Conf. Ind. Informatics (IEEE-INDIN 2017)*, pp. 474–479, 2017.
- [26] L. Herrera-izquierdo and M. Grob, “A performance evaluation between Docker container and Virtual Machines in cloud computing architectures,” *Maskana*, pp. 127–133, 2017.
- [27] Docker, “Docker engine.” [Online]. Available: <https://docs.docker.com/engine/docker-overview/>. [Accessed: 10-Jan-2018].
- [28] L. Guminski, “Orchestrate Containers for Development | Codeship | via @codeship,” 2017. [Online]. Available: <https://blog.codeship.com/orchestrate-containers-for-development-with-docker-compose/>. [Accessed: 13-Jan-2018].
- [29] Docker, “Understanding Swarm clusters.” [Online]. Available: <https://docs.docker.com/get-started/part4/#understanding-swarm-clusters>. [Accessed: 12-Jan-2018].

- [30] N. Naik, "Docker container-based big data processing system in multiple clouds for everyone," *2017 IEEE Int. Symp. Syst. Eng. ISSE 2017 - Proc.*, 2017.
- [31] S. Haines, "Getting Started with Docker Swarm: Part 1 of Blog Series," 2016. [Online]. Available: <https://turbonomic.com/blog/on-technology/getting-started-with-docker-swarm-part-1/>. [Accessed: 13-Jan-2018].
- [32] Technet, "Active Directory Collection: Active Directory." [Online]. Available: [https://technet.microsoft.com/en-us/library/cc780036\(WS.10\).aspx](https://technet.microsoft.com/en-us/library/cc780036(WS.10).aspx). [Accessed: 13-Jan-2018].
- [33] MIT, "Protocolo ligero de acceso a directorios (LDAP)." [Online]. Available: <http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-ldap.html>. [Accessed: 16-Jan-2018].
- [34] M. Y. Afanasev, Y. V. Fedosov, A. A. Krylova, and S. A. Shorokhov, "An application of microservices architecture pattern to create a modular computer numerical control system," *Conf. Open Innov. Assoc. Fruct*, vol. 2017–April, pp. 10–19, 2017.
- [35] MongoDB, "Reinventando la gestión de datos | MongoDB." [Online]. Available: <https://www.mongodb.com/es>. [Accessed: 14-Jan-2018].
- [36] MongoDB, "MongoDB Architecture | MongoDB." [Online]. Available: <https://www.mongodb.com/mongodb-architecture>. [Accessed: 17-Jan-2018].
- [37] G. P. Abdul Hasseb, "A review on NoSQL: Applications and challenges," *Int. J. Adv. Res. Comput. Sci.*, vol. 8, no. 1, pp. 203–207, 2017.
- [38] Git, "Git - Acerca del control de versiones." [Online]. Available: <https://git-scm.com/book/es/v1/Empezando-Acerca-del-control-de-versiones>. [Accessed: 16-Jan-2018].
- [39] M. Vuorre and J. P. Curley, "Curating Research Assets in Behavioral Sciences : A Tutorial on the Git Version Control System," *Curating Res. Assets Behav. Sci.*, pp. 1–33.
- [40] K. Beck and M. Fowler, *Planning Extreme Programming Kent Beck Martin Fowler*. 2000.
- [41] V. Armenise, "Continuous Delivery with Jenkins: Jenkins Solutions to Implement Continuous Delivery," *2015 IEEE/ACM 3rd Int. Work. Release Eng.*, pp. 24–27, 2015.
- [42] O. L. ; B. B. ; M. K. ; G. Grynkevych, "services continuos integration based on modern free infraestructure," pp. 7–10, 2017.
- [43] ThoughtWorks, "Continuous integration." [Online]. Available: <https://www.thoughtworks.com/continuous-integration>. [Accessed: 18-Jan-2018].
- [44] Amazon, "¿Qué es la integración continua?" [Online]. Available: <https://aws.amazon.com/es/devops/continuous-integration/>. [Accessed: 18-Jan-2018].
- [45] ThoughtWorks, "Continuous deployment | Technology Radar." [Online]. Available: <https://www.thoughtworks.com/radar/techniques/continuous-deployment>. [Accessed: 18-Jan-2018].

- [46] J. Humble, “What is Continuous Delivery?” [Online]. Available: <https://continuousdelivery.com/>. [Accessed: 19-Jan-2018].
- [47] J. Humble and D. Farley, “Continuous Delivery,” *Contin. Deliv.*, p. 497, 2015.
- [48] W. B. Santos, J. A. O. G. Cunha, H. Moura, and T. Margaria, “Practical Implications from a Preliminary Theory of Simplicity in Agile Software Development Based on a Qualitative Study,” *2017 XLIII Lat. Am. Comput. Conf.*, no. September, pp. 1–10, 2017.
- [49] J. Rasmusson, *The Agile Samurai*. 2014.
- [50] J. Highsmith, *Adaptive Software Development: a collaborative approach to managing complex systems*. 2000.
- [51] K. Beck, *Extreme Programming Explained , Second Edition*. 2004.
- [52] K. Schwaber, *Agile Project Management with Scrum*, vol. 7, no. Cmm. 2004.
- [53] J. H. Canós, P. Letelier, C. Penadés, and D. P. De Valencia, “Métodologías Ágiles en el Desarrollo de Software,” *Development*, pp. 1–8, 2003.
- [54] Proyectos Ágiles, “Qué es SCRUM.” [Online]. Available: <https://proyectosagiles.org/que-es-scrum/>. [Accessed: 20-Jan-2018].
- [55] K. Schwaber and J. Sutherland, “La Guía de Scrum,” *Scrumguides.Org*, vol. 1, p. 21, 2013.
- [56] A. Menzinsky, G. López, and J. Palacio, *Scrum Manager: Guía de formación*. 2016.
- [57] P. Sharma and N. Hasteer, “Analysis of linear sequential and extreme programming development methodology for a gaming application,” *Int. Conf. Commun. Signal Process. ICCSP 2016*, pp. 1916–1920, 2016.
- [58] S. M. M. Valladarez, M. E. Gaitan, and N. N. P. Reyes, “Metodologia Ágil De Desarrollo De Software Programacion Extrema.,” vol. 1, p. 146, 2016.
- [59] Don Wells, “Extreme Programming: A Gentle Introduction.,” 2013. [Online]. Available: <http://www.extremeprogramming.org/>. [Accessed: 23-Jan-2018].
- [60] J. Martin, *Rapid Application Development*. Indianapolis, IN, USA: Macmillan Publishing Co., Inc., 1991.
- [61] N. M. N. Daud, N. A. A. A. Bakar, and H. M. Rusli, “Implementing Rapid Application Development (RAD) methodology in developing practical training application system,” *Proc. 2010 Int. Symp. Inf. Technol. - Syst. Dev. Appl. Knowl. Soc. ITSim’10*, vol. 3, pp. 1664–1667, 2010.
- [62] M. V. T. Echeverria, S. G. Cornejo, and F. A. Pontigo, “Development of an information platform for observation of seismic events in Chile using RAD methodology,” *2016 IEEE Int. Conf. Autom. ICA-ACCA 2016*, 2016.
- [63] Automated Architecture, “Rapid Application Development,” pp. 1–5, 2005.
- [64] J. P. Womack, D. T. Jones, and D. Roos, *The Machine That Changed the World*. 1990.

- [65] M. A. Razzak, "An empirical study on lean and agile methods in global software development," *Proc. - 11th IEEE Int. Conf. Glob. Softw. Eng. Companion Proceedings, ICGSEW 2016*, pp. 61–64, 2016.
- [66] A. Janes, "A guide to lean software development in action," *2015 IEEE 8th Int. Conf. Softw. Testing, Verif. Valid. Work. ICSTW 2015 - Proc.*, pp. 1–2, 2015.
- [67] M. Poppendieck, "Lean Software Development," *29th Int. Conf. Softw. Eng. (ICSE'07 Companion)*, pp. 165–166, 2007.
- [68] M. Poppendieck and M. A. Cusumano, "Lean software development: A tutorial," vol. 29, no. 5, pp. 26–32, 2012.
- [69] V. B. S. Silva, F. Schramm, and A. C. Damasceno, "A multicriteria approach for selection of agile methodologies in software development projects," *IEEE Int. Conf. Syst. Man, Cybern.*, pp. 2056–2060, 2016.
- [70] B. S. Akpolat and W. Slany, "Enhancing software engineering student team engagement in a high-intensity extreme programming course using gamification," *2014 IEEE 27th Conf. Softw. Eng. Educ. Training, CSEE T 2014 - Proc.*, pp. 149–153, 2014.
- [71] S. H. Adelyar and A. Norta, "Towards a Secure Agile Software Development Process," *Proc. 2016 10Th Int. Conf. Qual. Inf. Commun. Technol.*, pp. 101–106, 2016.
- [72] R. M. Chopade, "Agile Software Development : Positive and Negative User Stories," pp. 297–299, 2017.
- [73] L. Layman, L. Williams, and L. Cunningham, "Exploring extreme programming in context: An industrial case study," *Proc. Agil. Dev. Conf. ADC 2004*, pp. 32–41, 2004.
- [74] A. Srivastava, "SCRUM Model for Agile Methodology," pp. 864–869, 2017.

## Anexos

### Anexo 1 Preguntas de la entrevista realizada al jefe de TIC's

#### ENTREVISTA

1. ¿Cómo departamento de TIC's cuáles son los objetivos que esperan lograr actualmente?  
\_\_\_\_\_
2. ¿Cuáles considera usted que son los puntos fuertes del departamento de TIC's?  
\_\_\_\_\_
3. ¿Cuáles son las plataformas de desarrollo de software que se utilizan en la universidad?  
\_\_\_\_\_
4. ¿Cuáles son los aspectos que destacan en las aplicaciones desarrolladas por el departamento de TIC's?  
\_\_\_\_\_
5. ¿Tienen identificado como se relacionan los distintos sistemas de la sede?  
\_\_\_\_\_
6. ¿Se manejan procesos para la integración y despliegue de aplicaciones?  
\_\_\_\_\_
7. En vista de que se está haciendo una integración de las aplicaciones locales con la sede principal, ¿se manejan procesos para la migración?  
\_\_\_\_\_
8. ¿Cuáles son los frameworks que se utilizan para el desarrollo de software?  
\_\_\_\_\_
9. De los frameworks que se han utilizado ¿Cuál cree que es el mejor que se ha adaptado a la sede?  
\_\_\_\_\_
10. ¿Considera usted que las interfaces de las aplicaciones desarrolladas por el departamento de TIC's son intuitivas?  
\_\_\_\_\_
11. ¿Qué niveles de seguridad implementan en las aplicaciones para el acceso a información clasificada?  
\_\_\_\_\_
12. ¿En los sistemas desarrollados por el departamento de TIC's los implementan bajo arquitectura SOA?  
\_\_\_\_\_
13. ¿Cuál es el nivel de disponibilidad de las aplicaciones que administra el departamento de TIC's?  
\_\_\_\_\_
14. ¿Cuando un cliente solicita el desarrollo de un sistema explica de manera clara y concisa sus necesidades al departamento de TIC's?  
\_\_\_\_\_
15. ¿Las funcionalidades que solicitan los clientes en sus sistemas, se manejan bajo estándares de experiencia de usuario genéricas o son personalizados de acuerdo a la visión del cliente?  
\_\_\_\_\_

## Anexo 2 Encuesta realizada al departamento de TIC's

### ENCUESTA

Fecha: ...../...../2018

**Objetivo de la encuesta:** El objetivo de la presente encuesta es conocer la manera en la que el departamento de TIC's planifica el desarrollo de proyectos software y las herramientas que utiliza.

Responda con una X las respuestas seleccionadas.

1. ¿Qué problemas considera más comunes para los retrasos en los proyectos de software?

- Falta de información durante el levantamiento del requerimiento.
- Fallas por omisiones de detalles
- Falta de un adecuado cronograma de trabajo.
- Falta de comunicación entre los departamentos.
- Inconvenientes en la etapa de desarrollo.
- Constantes cambios por parte del usuario.

Otros: \_\_\_\_\_

2. ¿Considera importante utilizar frameworks o librerías durante la etapa de desarrollo?

- Si
- No

3. ¿Cuáles son los frameworks o librerías que utilizan en el desarrollo de las aplicaciones?

- Angular
- Django
- Zend
- .NET
- React
- JQuery
- Spring
- Bootstrap 3
- Yii
- Ruby on Rails

Otros: \_\_\_\_\_

4. En base a la respuesta anterior, ¿cuánto tiempo lleva utilizando el framework?

- Menos de un año.
- 1 a 3 años.
- 4 a 6 años.
- Más de 6 años.

5. En relación a la respuesta anterior, ¿Qué ventajas les ofrece el modelo de framework que utilizan?

- ( ) Facilidad de uso.
- ( ) Optimización de Tiempo.
- ( ) Soporte de comunidad, fabricante, libros.
- ( ) La reutilización de componentes.
- ( ) Uso de patrones de diseño.
- ( ) Estandarización.
- ( ) Organización de código

6. ¿Cuándo va a desarrollar un proyecto software se establecen estimaciones de esfuerzo, alcance y costo?

- ( ) Siempre o casi siempre
- ( ) Usualmente
- ( ) Alguna vez
- ( ) Rara vez o nunca

7. ¿Cuándo se va a desarrollar un proyecto software se hace un plan de proyecto?

- ( ) Siempre o casi siempre
- ( ) Usualmente
- ( ) Alguna vez
- ( ) Rara vez o nunca

8. ¿Durante el desarrollo de un proyecto software hay compromiso con el plan de parte del cliente?

- ( ) Siempre o casi siempre
- ( ) Usualmente
- ( ) Alguna vez
- ( ) Rara vez o nunca

9. ¿Durante el desarrollo de un proyecto software se evalúan objetivamente los procesos y los productos finales?

- ( ) Siempre o casi siempre
- ( ) Usualmente
- ( ) Alguna vez
- ( ) Rara vez o nunca

10. ¿Durante el desarrollo de un proyecto software se proporciona una visión objetiva?

- ( ) Siempre o casi siempre
- ( ) Usualmente
- ( ) Alguna vez
- ( ) Rara vez o nunca

11. ¿Cuándo se va a desarrollar un nuevo proyecto software se identifica todos los requisitos del cliente?

- Siempre o casi siempre
- Usualmente
- Alguna vez
- Rara vez o nunca

12. ¿Cuándo se va a desarrollar un nuevo proyecto software se identifica todos los requisitos del producto?

- Siempre o casi siempre
- Usualmente
- Alguna vez
- Rara vez o nunca

13. ¿Cuándo se va a desarrollar un nuevo proyecto software se analizan y validan los requisitos?

- Siempre o casi siempre
- Usualmente
- Alguna vez
- Rara vez o nunca

### Anexo 3 Historias de usuario módulo Evaluación Laboral

## Evaluación Laboral

# SPRINT	1
TOTAL PUNTOS	15

Dev 1	FullStack / DBA Senior
Dev 2	Frontend & Backend Senior
Dev 3	Frontend & Backend Junior

HISTORIA DE USUARIO			
Número:	1	Cliente:	RRHH
Nombre de historia:	[TT] Desarrollar vistas que permitan tener acceso a los datos personales de los trabajadores en la base de datos de RRHH		
Prioridad en negocio:	Alta	Riesgo en desarrollo:	Media
Estimación en puntos:	1	Iteración asignada:	1
Programador responsable:	Dev 1 / Dev 2		

<b>Descripción:</b>	
Desarrollar vistas que obtengan los datos personales de los trabajadores (identificación, nombres, apellidos, correo, fecha ingreso, cargo)	
<b>Validación / Pruebas de aceptación:</b>	
Con la cédula o pasaporte puedo obtener la información personal de un trabajador	

<b>HISTORIA DE USUARIO</b>			
<b>Número:</b>	2	<b>Cliente:</b>	RRHH
<b>Nombre de historia:</b>	[TT] Generar Key y Token en Microsoft Apps con la url de la aplicación backend para el inicio de sesión con AD		
<b>Prioridad en negocio:</b>	Alta	<b>Riesgo en desarrollo:</b>	Media
<b>Estimación en puntos:</b>	1	<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Dev 2 / Dev 3		
<b>Descripción:</b>			
Generar App en Microsoft Apps que permita iniciar sesión en el backend mediante el uso de Key y Token			
<b>Validación / Pruebas de aceptación:</b>			
La url registrada en la app debe ser la misma que este configurada en el backend			
La Key y el Token que proporciona la app de Microsoft deben estar registrados en el archivo de configuración del backend			

<b>HISTORIA DE USUARIO</b>			
<b>Número:</b>	3	<b>Cliente:</b>	RRHH
<b>Nombre de historia:</b>	[TT] Diseñar e implementar el esquema de la base de datos para la aplicación		
<b>Prioridad en negocio:</b>	Alta	<b>Riesgo en desarrollo:</b>	Alta
<b>Estimación en puntos:</b>	2	<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Dev 1 / Dev 2		
<b>Descripción:</b>			
Implementar la base de datos para almacenar toda la información sobre la evaluación laboral y configurar la aplicación			
<b>Validación / Pruebas de aceptación:</b>			
La BDD debe tener las tablas que permitan guardar la evaluación realizada.			
La BDD debe tener las tablas que permitan guardar fechas de apertura y cierre para realizar la evaluación			
La BDD debe tener las tabla para guardar la asignación manual de evaluación entre trabajadores			

La BDD debe tener una vista para obtener reportes individuales por trabajador de las evaluaciones
La BDD debe tener una vista para obtener reportes generales por trabajador de las evaluaciones

HISTORIA DE USUARIO			
<b>Número:</b>	4	<b>Cliente:</b>	RRHH
<b>Nombre de historia:</b>	[TT] Levantar la aplicación base de backend con inicio de sesión usando Office365		
<b>Prioridad en negocio:</b>	Alta	<b>Riesgo en desarrollo:</b>	Medio
<b>Estimación en puntos:</b>	2	<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Dev 2 / Dev 3		
<b>Descripción:</b>			
Implementar la aplicación base usando Yii2 Advanced y PHP 7			
<b>Validación / Pruebas de aceptación:</b>			
La aplicación debe correr sobre PHP 7			
La aplicación se debe conectar al AD de la empresa para obtener los usuarios			
La aplicación debe permitir iniciar sesión mediante el portal Office365 de Microsoft			

HISTORIA DE USUARIO			
<b>Número:</b>	5	<b>Cliente:</b>	RRHH
<b>Nombre de historia:</b>	[TT] Prepara ambiente de Integración continua y Despliegue continuo		
<b>Prioridad en negocio:</b>	Alta	<b>Riesgo en desarrollo:</b>	Alta
<b>Estimación en puntos:</b>	2	<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Dev 1 / Dev 2		
<b>Descripción:</b>			
Preparar ambientes de integración continua y despliegue continuo usando Bitbucket y Jenkins			
<b>Validación / Pruebas de aceptación:</b>			
En bitbucket debo tener 4 ramas: dev, qa, stage y prod			
En Jenkins debo tener las configuraciones necesarias para que automáticamente al hacer un push se realice la integración y despliegue continuo			

HISTORIA DE USUARIO			
<b>Número:</b>	6	<b>Cliente:</b>	RRHH
<b>Nombre de historia:</b>	[TT] Implementar en la aplicación backend el acceso a opciones generales y de administración basado en los roles que tenga el usuario asignado en el AD		

<b>Prioridad en negocio:</b>	Alta	<b>Riesgo en desarrollo:</b>	Media
<b>Estimación en puntos:</b>	2	<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Dev 2 / Dev 3		
<b>Descripción:</b>			
La aplicación debe mostrar las diferentes opciones al usuario basado en el rol que este posea dentro del AD			
<b>Validación / Pruebas de aceptación:</b>			
Las opciones de administración sólo se deben mostrar a los usuarios con el rol de administrador			
Las opciones generales se deben mostrar sólo a los usuarios que estén registrados en el AD y que estén en estado activo			

<b>HISTORIA DE USUARIO</b>			
<b>Número:</b>	7	<b>Cliente:</b>	RRHH
<b>Nombre de historia:</b>	[TT] Extraer información en la aplicación backend desde las vistas implementadas en las bases de datos		
<b>Prioridad en negocio:</b>	Alta	<b>Riesgo en desarrollo:</b>	Media
<b>Estimación en puntos:</b>	2	<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Dev 1 / Dev 2		
<b>Descripción:</b>			
Las información almacenada en las tablas y vistas de la BDD deben ser presentadas en la aplicación backend			
<b>Validación / Pruebas de aceptación:</b>			
Al dar click en la opción reportes individuales debo obtener la información de la vista reportes individuales			
Al dar click en la opción reportes generales debo obtener la información de la vista reportes generales			
Al dar click en la opción asignación manual de evaluación debo obtener la información de la tabla asignaciones manuales			
Al dar click en la opción configuración de fechas debo obtener la información de la tabla de configuración de fechas			

<b>HISTORIA DE USUARIO</b>			
<b>Número:</b>	8	<b>Cliente:</b>	RRHH
<b>Nombre de historia:</b>	[TT] Crear EndPoint para que se pueda tener acceso a las distintas funcionalidades desde el frontend		
<b>Prioridad en negocio:</b>	Alta	<b>Riesgo en desarrollo:</b>	Medio
<b>Estimación en puntos:</b>	3	<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Dev 2 / Dev 3		
<b>Descripción:</b>			

Crear los distintos endpoints para obtener e ingresar información en el backend desde el frontend mediante Token
<b>Validación / Pruebas de aceptación:</b>
Hay un endpoint que me permite almacenar la evaluación mediante una petición POST enviando cómo parámetros la identificación del evaluado, la identificación del evaluador, el id de la evaluación, el id de la pregunta y el puntaje
Hay un endpoint que me permite obtener todas las evaluaciones asignadas a un trabajador enviando cómo parámetro la identificación y mediante una petición GET
Hay un endpoint que me permite obtener todos los resultados de las evaluaciones mediante una petición GET
Hay un endpoint que me permite obtener los resultados individuales de un trabajador enviando cómo parámetro la identificación mediante una petición GET
Hay un endpoint que me permite obtener los resultados generales de un trabajador enviando cómo parámetro la identificación mediante una petición GET

<b>HISTORIA DE USUARIO</b>			
<b>Número:</b>	9	<b>Cliente:</b>	RRHH
<b>Nombre de historia:</b>	[TT] Implementar middleware de mensajería para el intercambio asíncrono de información entre los endpoints y el frontend		
<b>Prioridad en negocio:</b>	Alta	<b>Riesgo en desarrollo:</b>	Medio
<b>Estimación en puntos:</b>	2	<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Dev 1 / Dev 2		
<b>Descripción:</b>	Implementar middleware de mensajería RabbitMQ para el intercambio asíncrono de información entre los endpoints y el frontend		
<b>Validación / Pruebas de aceptación:</b>			
Hay un producer de tipo directo para el endpoint que permite almacenar la evaluación enviando cómo parámetros la identificación del evaluado, la identificación del evaluador, el id de la evaluación, el id de la pregunta y el puntaje			
Hay un producer de tipo directo para el endpoint que me permite obtener todas las evaluaciones asignadas a un trabajador enviando cómo parámetro la identificación y mediante una petición GET			
Hay un producer de tipo directo para el endpoint que me permite obtener todos los resultados de las evaluaciones mediante una petición GET			
Hay un producer de tipo directo para el endpoint que me permite obtener los resultados individuales de un trabajador enviando cómo parámetro la identificación mediante una petición GET			
Hay un producer de tipo directo para el endpoint que me permite obtener los resultados generales de un trabajador enviando cómo parámetro la identificación mediante una petición GET			

## Evaluación Laboral

# SPRINT	2
TOTAL PUNTOS	9

Dev 1	FullStack / DBA Senior
Dev 2	Frontend & Backend Senior
Dev 3	Frontend & Backend Junior

HISTORIA DE USUARIO			
<b>Número:</b>	1	<b>Cliente:</b>	RRHH
<b>Nombre de historia:</b>	[EL-001] Implementar interfaz para la administración de fechas de apertura y cierre para hacer la evaluación en el backend		
<b>Prioridad en negocio:</b>	Alta	<b>Riesgo en desarrollo:</b>	Alta
<b>Estimación en puntos:</b>	1	<b>Iteración asignada:</b>	2
<b>Programador responsable:</b>	Dev 2 / Dev 3		
<b>Descripción:</b>			
Se debe implementar la interfaz para administración de fechas de apertura y cierre de la aplicación evaluación laboral			
<b>Validación / Pruebas de aceptación:</b>			
Al dar click en la opción fechas aparece una interfaz que permite administrar las fechas de apertura y cierre para la aplicación			
La opción sólo está disponible para un usuario con rol de administrador			

HISTORIA DE USUARIO			
<b>Número:</b>	2	<b>Cliente:</b>	RRHH
<b>Nombre de historia:</b>	[EL-002] Implementar interfaz que permita asignar manualmente evaluación entre trabajadores en el backend		
<b>Prioridad en negocio:</b>	Alta	<b>Riesgo en desarrollo:</b>	Alta
<b>Estimación en puntos:</b>	2	<b>Iteración asignada:</b>	2
<b>Programador responsable:</b>	Dev 2 / Dev 3		
<b>Descripción:</b>			
Se debe implementar la interfaz para la asignación manual de evaluación entre trabajadores			
<b>Validación / Pruebas de aceptación:</b>			
Al dar click en la opción asignación de evaluación aparece una interfaz que permite asignar evaluación entre trabajadores			
La opción sólo está disponible para un usuario con rol de administrador			

<b>HISTORIA DE USUARIO</b>			
<b>Número:</b>	3	<b>Cliente:</b>	RRHH
<b>Nombre de historia:</b>	[EL-003] Implementar interfaz que permita obtener Reportes individuales de los trabajadores en el frontend		
<b>Prioridad en negocio:</b>	Alta	<b>Riesgo en desarrollo:</b>	Alta
<b>Estimación en puntos:</b>	3	<b>Iteración asignada:</b>	2
<b>Programador responsable:</b>	Dev 1 / Dev 2		
<b>Descripción:</b>			
Se debe implementar la interfaz que permita visualizar los reportes individuales de las evaluaciones de los trabajadores en el frontend			
<b>Validación / Pruebas de aceptación:</b>			
Al dar click en Evaluación Laboral / Reportes Individuales, aparece en pantalla un reporte con los detalles de la evaluación que obtuvo cada trabajador de sus distintos evaluadores			
La opción solo debe estar accesible para el usuario con el rol de Jefe de RRHH y Pro-Rector			

<b>HISTORIA DE USUARIO</b>			
<b>Número:</b>	4	<b>Cliente:</b>	RRHH
<b>Nombre de historia:</b>	[EL-004] Implementar interfaz que permita obtener Reportes generales de la evaluación de todos los trabajadores en el frontend		
<b>Prioridad en negocio:</b>	Alta	<b>Riesgo en desarrollo:</b>	Alta
<b>Estimación en puntos:</b>	3	<b>Iteración asignada:</b>	2
<b>Programador responsable:</b>	Dev 1 / Dev 2		
<b>Descripción:</b>			
Se debe implementar la interfaz que permita visualizar los reportes generales de las evaluaciones de los trabajadores en el frontend			
<b>Validación / Pruebas de aceptación:</b>			
Al dar click en Evaluación Laboral / Reportes generales, aparece en pantalla un reporte con el resumen de la evaluación que obtuvo cada trabajador			
La opción solo debe estar accesible para el usuario con el rol de Jefe de RRHH y Pro-Rector			

Anexo 4 Historias de usuario del módulo Biométrico

## Biométrico

# SPRINT	1
TOTAL PUNTOS	15

Dev 1	FullStack / DBA Senior
Dev 2	Frontend & Backend Senior
Dev 3	Frontend & Backend Junior

HISTORIA DE USUARIO			
Número:	1	Cliente:	RRHH
Nombre de historia:	[TT] Desarrollar vistas que permitan tener acceso a los datos personales de los trabajadores en la base de datos de RRHH		
Prioridad en negocio:	Alta	Riesgo en desarrollo:	Media
Estimación en puntos:	1	Iteración asignada:	1
Programador responsable:	Dev 1 / Dev 2		
Descripción:	Desarrollar vistas que obtengan los datos personales de los trabajadores (identificación, nombres, apellidos, correo, fecha ingreso, cargo)		
Validación / Pruebas de aceptación:	Con la cédula o pasaporte puedo obtener la información personal de un trabajador		

HISTORIA DE USUARIO			
Número:	2	Cliente:	RRHH
Nombre de historia:	[TT] Generar Key y Token en Microsoft Apps con la url de la aplicación backend para el inicio de sesión con AD		
Prioridad en negocio:	Alta	Riesgo en desarrollo:	Media
Estimación en puntos:	1	Iteración asignada:	1
Programador responsable:	Dev 2 / Dev 3		
Descripción:	Generar App en Microsoft Apps que permita iniciar sesión en el backend mediante el uso de Key y Token		
Validación / Pruebas de aceptación:	La url registrada en la app debe ser la misma que este configurada en el backend		
	La Key y el Token que proporciona la app de Microsoft deben estar registrados en el archivo de configuración del backend		

<b>HISTORIA DE USUARIO</b>			
<b>Número:</b>	3	<b>Cliente:</b>	RRHH
<b>Nombre de historia:</b>	[TT] Levantar la aplicación base de backend con inicio de sesión usando Office365		
<b>Prioridad en negocio:</b>	Alta	<b>Riesgo en desarrollo:</b>	Medio
<b>Estimación en puntos:</b>	2	<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Dev 2 / Dev 3		
<b>Descripción:</b>			
Implementar la aplicación base usando Yii2 Advanced y PHP 7			
<b>Validación / Pruebas de aceptación:</b>			
La aplicación debe correr sobre PHP 7			
La aplicación se debe conectar al AD de la empresa para obtener los usuarios			
La aplicación debe permitir iniciar sesión mediante el portal Office365 de Microsoft			

<b>HISTORIA DE USUARIO</b>			
<b>Número:</b>	4	<b>Cliente:</b>	RRHH
<b>Nombre de historia:</b>	[TT] Preparar ambiente de Integración continua y Despliegue continuo		
<b>Prioridad en negocio:</b>	Alta	<b>Riesgo en desarrollo:</b>	Alta
<b>Estimación en puntos:</b>	2	<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Dev 1 / Dev 2		
<b>Descripción:</b>			
Preparar ambientes de integración continua y despliegue continuo usando Bitbucket y Jenkins			
<b>Validación / Pruebas de aceptación:</b>			
En bitbucket debo tener 4 ramas: dev, qa, stage y prod			
En Jenkins debo tener las configuraciones necesarias para que automáticamente al hacer un push se realice la integración y despliegue continuo			

<b>HISTORIA DE USUARIO</b>			
<b>Número:</b>	5	<b>Cliente:</b>	RRHH
<b>Nombre de historia:</b>	[TT] Implementar en la aplicación backend el acceso a opciones generales y de administración basado en los roles que tenga el usuario asignado en el AD		
<b>Prioridad en negocio:</b>	Alta	<b>Riesgo en desarrollo:</b>	Media
<b>Estimación en puntos:</b>	2	<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Dev 2 / Dev 3		
<b>Descripción:</b>			
La aplicación debe mostrar las diferentes opciones al usuario basado en el rol que este posea dentro del AD			
<b>Validación / Pruebas de aceptación:</b>			
Las opciones de administración sólo se deben mostrar a los usuarios con el rol de administrador			
Las opciones generales se deben mostrar sólo a los usuarios que estén registrados en el AD y que estén en estado activo			

<b>HISTORIA DE USUARIO</b>			
<b>Número:</b>	6	<b>Cliente:</b>	RRHH
<b>Nombre de historia:</b>	[TT] Extraer información en la aplicación backend desde las vistas implementadas en las bases de datos		
<b>Prioridad en negocio:</b>	Alta	<b>Riesgo en desarrollo:</b>	Media
<b>Estimación en puntos:</b>	2	<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Dev 1 / Dev 2		
<b>Descripción:</b>			
Las información almacenada en las tablas y vistas de la BDD deben ser presentadas en la aplicación backend			
<b>Validación / Pruebas de aceptación:</b>			
Al dar click en la opción marcaciones debo obtener la información de la vista marcaciones por mes			

<b>HISTORIA DE USUARIO</b>			
<b>Número:</b>	7	<b>Cliente:</b>	RRHH
<b>Nombre de historia:</b>	[TT] Crear EndPoint para que se pueda tener acceso a las distintas funcionalidades desde el frontend		
<b>Prioridad en negocio:</b>	Alta	<b>Riesgo en desarrollo:</b>	Medio
<b>Estimación en puntos:</b>	3	<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Dev 2 / Dev 3		
<b>Descripción:</b>			
Crear los distintos EndPoints para obtener e ingresar información en el backend desde el frontend mediante Token			
<b>Validación / Pruebas de aceptación:</b>			
Hay un endpoint que me permite obtener las marcaciones de un trabajador enviando cómo parámetro la identificación mediante una petición GET			
Hay un endpoint que me permite obtener las marcaciones generales de todos los trabajadores enviando cómo parámetro el mes y el año mediante una petición GET			

<b>HISTORIA DE USUARIO</b>			
<b>Número:</b>	8	<b>Cliente:</b>	RRHH
<b>Nombre de historia:</b>	[TT] Implementar middleware de mensajería para el intercambio asíncrono de información entre los endpoints y el frontend		
<b>Prioridad en negocio:</b>	Alta	<b>Riesgo en desarrollo:</b>	Medio
<b>Estimación en puntos:</b>	2	<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Dev 1 / Dev 2		
<b>Descripción:</b>			
Implementar middleware de mensajería RabbitMQ para el intercambio asíncrono de información entre los endpoints y el frontend			
<b>Validación / Pruebas de aceptación:</b>			
Hay un producer de tipo directo para el endpoint que me permite obtener las marcaciones individuales de un trabajador enviando cómo parámetro la identificación mediante una petición GET			
Hay un producer de tipo directo para el endpoint que me permite obtener las marcaciones de todos los trabajadores enviando cómo parámetro el mes y el año mediante una petición GET			

## Biométrico

# SPRINT	2
TOTAL PUNTOS	6

Dev 1	FullStack / DBA Senior
Dev 2	Frontend & Backend Senior
Dev 3	Frontend & Backend Junior

HISTORIA DE USUARIO			
<b>Número:</b>	1	<b>Cliente:</b>	RRHH
<b>Nombre de historia:</b>	[SB-001] Implementar interfaz que permita obtener Reportes individuales de las marcaciones de los trabajadores en el frontend		
<b>Prioridad en negocio:</b>	Alta	<b>Riesgo en desarrollo:</b>	Alta
<b>Estimación en puntos:</b>	3	<b>Iteración asignada:</b>	2
<b>Programador responsable:</b>	Dev 1 / Dev 2		
<b>Descripción:</b>			
Se debe implementar la interfaz que permita visualizar los reportes individuales de las marcaciones de los trabajadores en el frontend, debe tener un filtro para la visualización por mes			
<b>Validación / Pruebas de aceptación:</b>			
Al dar click en Marcaciones Biométrico / Reportes, aparece en pantalla un reporte con los detalles de las marcaciones que realizó cada trabajador durante el mes			

HISTORIA DE USUARIO			
<b>Número:</b>	2	<b>Cliente:</b>	RRHH
<b>Nombre de historia:</b>	[SB-002] Implementar interfaz que permita obtener Reportes generales de las marcaciones de todos los trabajadores en el frontend		
<b>Prioridad en negocio:</b>	Alta	<b>Riesgo en desarrollo:</b>	Alta
<b>Estimación en puntos:</b>	3	<b>Iteración asignada:</b>	2
<b>Programador responsable:</b>	Dev 1 / Dev 2		
<b>Descripción:</b>			
Se debe implementar la interfaz que permita visualizar los reportes generales de las marcaciones de todos los trabajadores durante el mes en el frontend			
<b>Validación / Pruebas de aceptación:</b>			
Al dar click en Marcaciones Biométrico / Reportes, aparece en pantalla un reporte con el resumen de las marcaciones que realizaron todos los trabajadores durante el mes			
Esta opción solo debe estar accesible para el usuario con el rol de Jefe de RRHH y Pro-Rector			

## Reportes Aula Virtual

# SPRINT	1
TOTAL PUNTOS	16

Dev 1	FullStack / DBA Senior
Dev 2	Frontend & Backend Senior
Dev 3	Frontend & Backend Junior

HISTORIA DE USUARIO			
Número:	1	Cliente:	Dir. E.
Nombre de historia:	[TT] Desarrollar una vista en la base de datos MySQL para tener reportes del uso de la plataforma Moodle		
Prioridad en negocio:	Alta	Riesgo en desarrollo:	Media
Estimación en puntos:	1	Iteración asignada:	1
Programador responsable:	Dev 1 / Dev 2		
<b>Descripción:</b>			
Desarrollar vistas que obtengan reportes de las veces que un profesor ha enviado tareas por materia			
<b>Validación / Pruebas de aceptación:</b>			
Con el correo institucional del docente se puede acceder al reporte de las tareas por materia			
Los reportes son mostrados basados en el rol que posea el usuario ya sea Director de escuela o Director académico			

HISTORIA DE USUARIO			
Número:	2	Cliente:	Dir. E.
Nombre de historia:	[TT] Guardar información de la vista creada en la base de datos MySQL en una colección de MongoDB		
Prioridad en negocio:	Alta	Riesgo en desarrollo:	Media
Estimación en puntos:	1	Iteración asignada:	1
Programador responsable:	Dev 2 / Dev 3		
<b>Descripción:</b>			
Guardar la información obtenida de la vista creada en la base de datos MySQL en una colección de MongoDB			
<b>Validación / Pruebas de aceptación:</b>			
La información obtenida de la vista en MySQL debe coincidir con la información copiada en la colección de MongoDB			

HISTORIA DE USUARIO			
<b>Número:</b>	3	<b>Cliente:</b>	Dir. E.
<b>Nombre de historia:</b>	[TT] Generar Key y Token en Microsoft Apps con la url de la aplicación backend para el inicio de sesión con AD		
<b>Prioridad en negocio:</b>	Alta	<b>Riesgo en desarrollo:</b>	Media
<b>Estimación en puntos:</b>	1	<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Dev 2 / Dev 3		
<b>Descripción:</b>			
Generar App en Microsoft Apps que permita iniciar sesión en el backend mediante el uso de Key y Token			
<b>Validación / Pruebas de aceptación:</b>			
La url registrada en la app debe ser la misma que este configurada en el backend			
La Key y el Token que proporciona la app de Microsoft deben estar registrados en el archivo de configuración del backend			

HISTORIA DE USUARIO			
<b>Número:</b>	4	<b>Cliente:</b>	Dir. E.
<b>Nombre de historia:</b>	[TT] Levantar la aplicación base de backend con inicio de sesión usando Office365		
<b>Prioridad en negocio:</b>	Alta	<b>Riesgo en desarrollo:</b>	Medio
<b>Estimación en puntos:</b>	2	<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Dev 2 / Dev 3		
<b>Descripción:</b>			
Implementar la aplicación base usando Yii2 Advanced y PHP 7			
<b>Validación / Pruebas de aceptación:</b>			
La aplicación debe correr sobre PHP 7			
La aplicación se debe conectar al AD de la empresa para obtener los usuarios			
La aplicación debe permitir iniciar sesión mediante el portal Office365 de Microsoft			

HISTORIA DE USUARIO			
<b>Número:</b>	5	<b>Cliente:</b>	Dir. E.
<b>Nombre de historia:</b>	[TT] Prepara ambiente de Integración continua y Despliegue continuo		
<b>Prioridad en negocio:</b>	Alta	<b>Riesgo en desarrollo:</b>	Alta
<b>Estimación en puntos:</b>	2	<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Dev 1 / Dev 2		
<b>Descripción:</b>			
Preparar ambientes de integración continua y despliegue continuo usando Bitbucket y Jenkins			
<b>Validación / Pruebas de aceptación:</b>			
En bitbucket debo tener 4 ramas: dev, qa, stage y prod			
En Jenkins debo tener las configuraciones necesarias para que automáticamente al hacer un push se realice la integración y despliegue continuo			

<b>HISTORIA DE USUARIO</b>			
<b>Número:</b>	6	<b>Cliente:</b>	Dir. E.
<b>Nombre de historia:</b>	[TT] Implementar en la aplicación backend el acceso a reportes por escuela basado en los roles que tenga el usuario asignado en el AD		
<b>Prioridad en negocio:</b>	Alta	<b>Riesgo en desarrollo:</b>	Media
<b>Estimación en puntos:</b>	2	<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Dev 2 / Dev 3		
<b>Descripción:</b>			
La aplicación debe mostrar las diferentes opciones al usuario basado en el rol que este posea dentro del AD			
<b>Validación / Pruebas de aceptación:</b>			
Las opciones de ver todos los reportes de todas las escuelas sólo debe estar disponible para el usuario con rol de Dirección Académica			
Las opciones generales se deben mostrar sólo a los usuarios que tengan asignado el rol de Director de Escuela			

<b>HISTORIA DE USUARIO</b>			
<b>Número:</b>	7	<b>Cliente:</b>	Dir. E.
<b>Nombre de historia:</b>	[TT] Extraer información en la aplicación backend desde la colección creada en MongoDB		
<b>Prioridad en negocio:</b>	Alta	<b>Riesgo en desarrollo:</b>	Media
<b>Estimación en puntos:</b>	2	<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Dev 1 / Dev 2		
<b>Descripción:</b>			
Las información almacenada en la colección de MongoDB deben ser presentadas en la aplicación backend			
<b>Validación / Pruebas de aceptación:</b>			
Al dar click en la opción reportes Moodle debo obtener la información de la colección por semestre y año			

<b>HISTORIA DE USUARIO</b>			
<b>Número:</b>	8	<b>Cliente:</b>	Dir. E.
<b>Nombre de historia:</b>	[TT] Crear endpoint para que se pueda tener acceso a las distintas funcionalidades desde el frontend		
<b>Prioridad en negocio:</b>	Alta	<b>Riesgo en desarrollo:</b>	Medio
<b>Estimación en puntos:</b>	3	<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Dev 2 / Dev 3		
<b>Descripción:</b>			
Crear los distintos EndPoints para obtener información en el backend desde el frontend mediante Token			
<b>Validación / Pruebas de aceptación:</b>			
Hay un endpoint que me permite obtener los reportes de todas las escuelas enviando como parámetro el semestre y el año una petición GET			
Hay un endpoint que me permite obtener los reportes de la escuela a la que pertenece el director de escuela enviando como parámetro el semestre y el año una petición GET			

<b>HISTORIA DE USUARIO</b>			
<b>Número:</b>	9	<b>Cliente:</b>	Dir. E.
<b>Nombre de historia:</b>	[TT] Implementar middleware de mensajería para el intercambio asíncrono de información entre los endpoints y el frontend		
<b>Prioridad en negocio:</b>	Alta	<b>Riesgo en desarrollo:</b>	Medio
<b>Estimación en puntos:</b>	2	<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Dev 1 / Dev 2		
<b>Descripción:</b>			
Implementar middleware de mensajería RabbitMQ para el intercambio asíncrono de información entre los endpoints y el frontend			
<b>Validación / Pruebas de aceptación:</b>			
Hay un producer de tipo directo para el endpoint que permite obtener los reportes de todas las escuelas enviando como parámetro el semestre y el año mediante una petición GET			
Hay un producer de tipo directo para el endpoint que me permite obtener los reportes de la escuela a la que pertenece el director de escuela mediante una petición GET			

## Reportes Aula Virtual

# SPRINT	2
TOTAL PUNTOS	3

Dev 1	FullStack / DBA Senior
Dev 2	Frontend & Backend Senior
Dev 3	Frontend & Backend Junior

HISTORIA DE USUARIO			
<b>Número:</b>	1	<b>Cliente:</b>	Dir. E.
<b>Nombre de historia:</b>	[RM-001] Implementar interfaz que permita obtener Reportes por escuela del uso de la plataforma		
<b>Prioridad en negocio:</b>	Alta	<b>Riesgo en desarrollo:</b>	Alta
<b>Estimación en puntos:</b>	3	<b>Iteración asignada:</b>	2
<b>Programador responsable:</b>	Dev 1 / Dev 2		
<b>Descripción:</b>			
Se debe implementar la interfaz que permita visualizar los reportes individuales de las marcaciones de los trabajadores en el frontend, debe tener un filtro para la visualización por mes			
<b>Validación / Pruebas de aceptación:</b>			
Al dar click en Marcaciones Biométrico / Reportes, aparece en pantalla un reporte con los detalles de las marcaciones que realizó cada trabajador durante el mes			

Anexo 6 Encuesta para evaluar el nivel de satisfacción de los cliente respecto a la nueva metodología utilizada.

## ENCUESTA

Fecha: ...../...../2018

**Objetivo de la encuesta:** Medir el nivel de satisfacción del cliente respecto al módulo: \_\_\_\_\_, el cual pertenece al proyecto software **Intranet**, adicionalmente se espera captar sugerencias respecto a nuevas características a implementar en el módulo.

Responda con una X las respuestas seleccionadas.

1. Cumplimiento de las necesidades del cliente

- Muy satisfactorio
- Satisfactorio
- Aceptable
- Insatisfactorio
- Muy Insatisfactorio

2. Usabilidad de la aplicación

- Muy satisfactorio
- Satisfactorio
- Aceptable
- Insatisfactorio
- Muy Insatisfactorio

3. Tiempo de entrega del proyecto

- Muy satisfactorio
- Satisfactorio
- Aceptable
- Insatisfactorio
- Muy Insatisfactorio

4. Gestión de cambios o modificaciones

- Muy satisfactorio
- Satisfactorio
- Aceptable
- Insatisfactorio
- Muy Insatisfactorio

**¿Cuáles son los cambios o mejoras que, en su opinión, debería incorporar el nuevo servicio?**