

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

SISTEMAS DIFUSOS DE APOYO A LA TOMA DE DECISIONES

(CÓDIGO: 31105151)

**SISTEMAS DE APOYO PARA LA TOMA DE DECISIONES EN
SITUACIONES DE ESTRÉS**

ESTUDIANTE: LUIS ALBERTO SERENO SÁNCHEZ-ARÉVALO

DIRECTOR: PEDRO JAVIER HERRERA CARO

CURSO: 2019-2020

CONVOCATORIA: FEBRERO

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

SISTEMAS DIFUSOS DE APOYO A LA TOMA DE DECISIONES

(CÓDIGO: 31105151)

**SISTEMAS DE APOYO PARA LA TOMA DE DECISIONES EN
SITUACIONES DE ESTRÉS**

TRABAJO FIN DE MÁSTER

ESTUDIANTE: LUIS ALBERTO SERENO SÁNCHEZ-ARÉVALO

DIRECTOR: PEDRO JAVIER HERRERA CARO

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

Fecha: 06/02/2020

Quién suscribe:

Autor(a): Luis Alberto Sereno Sánchez-Arévalo
D.N.I./N.I.E./Pasaporte.: 08880819-J

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

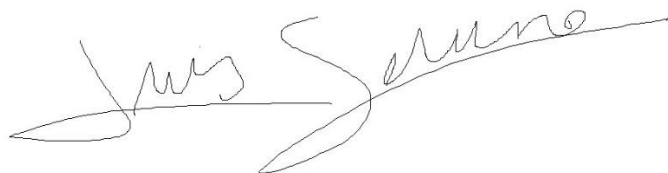
Título completo del trabajo.

SISTEMAS DE APOYO PARA LA TOMA DE DECISIONES EN SITUACIONES DE ESTRÉS

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. Luis Alberto Sereno Sánchez-Arévalo



IMPRESO TFDM05_AUTORPBL
AUTORIZACIÓN DE PUBLICACIÓN
CON FINES ACADÉMICOS

**Impreso TFDM05_AutorPbl. Autorización de publicación
y difusión del TFM para fines académicos**

Autorización

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

Firma del/los Autor/es

Juan del Rosal, 16
28040, Madrid

Tel: 91 398 89 10
Fax: 91 398 89 09

www.issi.uned.es

Resumen: El presente trabajo de fin de máster lleva a cabo, una investigación sobre la automatización de la toma de decisiones en situaciones de estrés en diferentes sistemas de flujo de personas. Esta automatización supondrá que los diferentes entornos de flujo por los que los usuarios pasan, se adapten dinámicamente en función del número de personas que están detectándose en el entorno en ese momento. Esto conllevará en un futuro cercano, a que los entornos sean totalmente maleables en función de los usuarios y a crear verdaderos sistemas de redes inteligentes, en los que no sean los usuarios los que tengan que adaptarse al medio, sino el medio a los usuarios.

Este trabajo aúna diferentes investigaciones llevadas a cabo en el mundo de los sistemas difusos de toma de decisiones y de la inteligencia artificial. Para realizar la toma de decisiones se obtienen datos de entrada de los usuarios en tiempo real, de manera invisible para el usuario, eliminando todo tipo de proactividad por parte de este.

En las siguientes páginas se desarrollará una aplicación que captará la información de todos los usuarios mediante diversos métodos no intrusivos y toma una serie de decisiones dependiendo del aprendizaje automático que realiza el sistema, en base a un modelo que será mejorado día a día con la información que proviene del entorno real.

El trabajo presenta dos resultados de la aplicación desarrollada, uno utilizando información del mundo real mediante los diferentes inputs que detectan sensores Bluetooth y WiFi y otro mediante un sistema de simulación de flujo de personas en una estación. Ambos resultados son complementarios, ya que uno ayuda a demostrar la integración correcta de la aplicación y el otro el rendimiento que la aplicación tendrá.

Durante el documento se desarrollará un entorno ligado a la practicidad de este, es decir, toda la investigación llevada a cabo y el sistema desarrollado está pensado para ser puesto en práctica en el mundo real por cualquier empresa que desee implementar un sistema de estas características. Es por ello, que existe un anexo que muestra al detalle la aplicación desarrollada y explica cómo funciona el proyecto a nivel técnico.

Palabras clave: Lógica difusa, inteligencia artificial, toma de decisiones, flujo de personas, tiempo real.

Contenido

1. Introducción	1
1.1. Motivación	1
1.2. Justificación	2
1.3. Descripción del problema.....	3
1.4. Objetivos.....	4
1.4.1. Facilidad de implantación.....	5
1.4.2. Bajo coste.....	6
1.4.3. Sistema sin licencia (open source).....	6
1.4.4. Alta resiliencia a fallos	6
1.4.5. Escalabilidad.....	6
1.4.6. Aséptico del sistema operativo	7
1.5. Organización del documento	7
2. Estado del arte.....	9
2.1. Categorización de imágenes de vídeo	9
2.2. Sistemas difusos basados en tecnología WiFi	10
2.3. Posicionamiento basado en tecnología Bluetooth	11
2.4. Aprendizaje Máquina como soporte a los sistemas difusos	11
2.5. Lógica difusa para la gestión del transporte de usuarios.....	12
2.6. Movilidad como Servicio (MaaS).....	13
3. Solución propuesta	15
3.1. Sistema central	17
3.1.1. Controlador	18
3.1.2. Aprendizaje automático	18
3.1.3. Sistema de respuesta	29
3.2. Sistema periférico	29
3.2.1. WiFi	30
3.2.2. Bluetooth	31
3.2.3. Cámara.....	32
3.3. Sistema de acción.....	33
3.3.1. Barreras.....	34
3.3.2. Móvil	34
4. Resultados obtenidos	36
4.1. Prueba en entorno real	37
4.2. Validaciones	43

4.3. Simulador	48
5. Conclusiones y Trabajo Futuro	55
6. Bibliografía.....	57
7. Lista de acrónimos.....	61
8. Anexo	62
8.1. Sistema central	62
8.1.1. Api Rest.....	63
8.1.2. Core.....	68
8.1.3. Firebase module	76
8.1.4. Schedule module e aprendizaje automático.....	81

Índice de Figuras

Figura 1 Esquema global del sistema	16
Figura 2 Gráfica pertenencia al conjunto difuso	20
Figura 3 Gráfica cuantil	22
Figura 4 Esquema sistema periférico	29
Figura 5 Arquitectura de capas CNN para clasificación de imágenes [39].....	32
Figura 6 Esquema sistema de acción	33
Figura 7 Detección de personas en tiempo real	36
Figura 8 Plano de edificio.....	38
Figura 9 Número de personas detectadas por dispositivo.....	39
Figura 10 Pronóstico empeoramiento	39
Figura 11 Pronóstico mejora	40
Figura 12 Cuadro de mando A.....	41
Figura 13 Cuadro de mando B.....	42
Figura 14 Validación cruzada [40].....	44
Figura 15 Zonas de validación	45
Figura 16 Fiabilidad	46
Figura 17 Precisión	46
Figura 18 Sensibilidad.....	47
Figura 19 Reporte de clasificación	47
Figura 20 Plano cenital estación	48
Figura 21 Sensores en estación.....	49
Figura 22 Simulador.....	50
Figura 23 Sensores detectando personas.....	51
Figura 24 Tornos cambiando	52
Figura 25 Tornos cambiando 2	52
Figura 26 Datos sensores Kibana	53
Figura 27 Datos entrenamiento Kibana.....	53
Figura 28 Datos algoritmo y resultado.....	54
Figura 29 Datos cuantil	54
Figura 30 Detalle Sistema Central.....	62
Figura 31 Documentación Swagger servicios	64
Figura 32 Comunicaciones de seguridad	68
Figura 33 Diagrama de clases Core.....	69
Figura 34 Esquema de clases DatabaseManager.....	71
Figura 35 Esquema de clases de la Base de Datos	71
Figura 36 Gráfica cuantiles	73
Figura 37 Diagrama de base de datos	75
Figura 38 Dashboard Firebase.....	76
Figura 39 Database Firebase.....	79
Figura 40 Clase FirebaseBaseCore	79
Figura 41 Diagrama de clases Schedule y Aprendizaje Automático.....	82
Figura 42 Diagrama de clase ScheduleModule	83
Figura 43 Diagrama de clases TrainAlgorithm	85
Figura 44 Diagrama de clases ConvertPredictiveData	89

Índice de Tablas

Tabla 1 Término conjunto difuso	19
Tabla 2 Tabla relación Niveles-Ocupación	21
Tabla 3 Tabla de dispositivos.....	27
Tabla 4 Tabla para regresión semiparamétrica	28
Tabla 5 Matriz de confusión	45
Tabla 6 Lista de acrónimos	61

1. Introducción

Actualmente, el mundo de la informática ha girado su paradigma desde la orientación a los servicios hacia la orientación a los usuarios.

En los inicios, esta nació para dar soporte, ampliar y hacer que los procesos recurrentes fuesen más rápidos. Ofreciendo, gracias a su capacidad computacional, una ganancia temporal y económica a las diferentes empresas.

Una vez conseguida la automatización de los procesos de las diferentes empresas, estas están volviendo su cara de sus procesos internos hacia las personas, preguntándose globalmente cómo pueden captar más usuarios y cuáles son las inquietudes de éstos.

Basándose en esta nueva preocupación, las empresas están intentando inmiscuirse cada vez menos en la vida de sus usuarios de manera directa, buscando nuevas soluciones que, de manera indirecta, consigan darle al usuario una experiencia personalizada a sus gustos, evitando por todos los medios hacer que el usuario tenga que interactuar en los sistemas empresariales. La mayoría de estas empresas, buscan seguir aumentando sus beneficios, que se habían estancado, gracias a la búsqueda de nuevos usuarios, y para ello necesitan saber qué necesitan estos y por qué estos optan por otros servicios que no son los suyos.

Por todo lo anterior y buscando sobre todo la no interacción y las menores molestias de los usuarios con los sistemas hardware empresariales (como tornos o validadores de usuarios) es necesario encontrar las mayores fuentes de datos posibles que no impliquen la colaboración activa por parte de los usuarios.

1.1. Motivación

La motivación fundamental para la realización de este trabajo de fin de máster, sobre la toma de decisiones en situaciones de estrés basándose en sistemas difusos, se basa sobre todo en la disposición para hacer la vida más fácil a las personas, logrando que el mundo se pueda adaptar en función a las distintas necesidades que estas requieran y garantizando a su vez una total facilidad en sus recorridos diarios.

Otro de los motivos, también muy importante, es mi conocimiento y experiencia laboral gracias al trabajo en equipos de I+D+i en el campo de los transportes, tanto públicos (autobuses, trenes, aviones, taxi...) como privados (coche compartido, Cabify, Uber), en el seno de la Unión Europea, que hace que tenga una mayor sensibilidad a la hora de ver las situaciones de estrés que se producen en muchos pasajeros en el momento de acceder a sus medios de transporte. Siendo, en muchas ocasiones, el principal motivo por el que las personas no utilizan cierto tipo

de transportes públicos, con los problemas que ello conlleva: contaminación, atascos, accidentes...

También es importante remarcar, que debido al conocimiento de la línea que quiere seguir la UE en materia de transportes, desarrollo este proyecto con especial interés. Dentro de toda la amplia reglamentación de la UE en materia de transporte, se está haciendo una fuerte inversión para que los ciudadanos que la componen puedan viajar de forma más segura y sostenible dentro de las fronteras de la UE, esto implica de manera clara, que se deben hacer los transportes públicos y compartidos más atractivos al usuario para poder aumentar el número de pasajeros y reducir la cantidad de medios privados dentro de ella.

Es debido a que se necesitan hacer los transportes públicos más atractivos, uno de los fundamentos de desarrollo de este trabajo, ya que pretende mejorar los accesos a diversos medios de transporte mejorando el conocimiento de las estaciones y las compañías de sus flujos de viajeros y sus comportamientos.

Se suman a todas estas motivaciones anteriores, el hecho de intentar disminuir la huella de carbono debida al movimiento de las personas en el mundo real. Esta motivación viene determinada por la preocupación del investigador sobre los problemas del cambio climático que están asolando ciertas áreas y por ello, este trabajo intenta ser su granito de arena y su pequeña aportación a la solución de este problema.

1.2. Justificación

La realización de este trabajo se intentará enmarcar dentro de lo que comúnmente se conoce como “estación viva”. Este término, viene a concretar el hecho de que la estación de acceso a los transportes funcione o se mueva en relación al flujo de personas o pasajeros que se mueven a través de ella.

En la actualidad, la mayoría de las estaciones y los sistemas de acceso a los lugares son sistemas estáticos, donde es el viajero o el cliente el que debe esperar y adecuarse a los diversos sistemas existentes, generando en multitud de ocasiones colas y retrasos innecesarios, por no hablar de problemas arquitectónicos que pueden generar diversas barreras para personas con problemas de movilidad. Esto es así, porque la estación, en muchas ocasiones, se diseña pensando en los diversos sistemas de validación que van a existir en ella, o al menos, debe adecuarse a estos sistemas de validación.

El sistema anteriormente descrito, puede generar un rechazo en los usuarios, haciendo que muchos se nieguen a usar estos medios por estrés o pérdidas de tiempo.

En los sistemas conocidos como “estación viva” la estación sería dinámica, y se comportaría de diferentes maneras dependiendo del flujo de personas que detectase a través de diferentes sensores. Estos sensores recogerían toda la información que proviene del mundo real y la

transformarían para poder aprender de ella, acabando este aprendizaje con diversos resultados que deberán resultar beneficiosos para las personas que se muevan a través de ella.

La estación dinámica conllevaría muchos beneficios a los pasajeros y las diferentes empresas por muchos y diferentes motivos. A los usuarios les facilitaría el tránsito por la estación reduciendo el tiempo de paso y disminuyendo el estrés que puede ocasionar, además la estación no tendría por qué estar acoplada a los diferentes tipos de validadores y es probable que se pudiesen eliminar algunas dificultades para los pasajeros con movilidad reducida.

Por otra parte, para las empresas supondría un ahorro ya que no deberían invertir tanto dinero en hardware y maquinaria pesada, además de optimizar sus servicios gracias a que puede conocer, mejorar y ampliar su flujo de viajeros, y, además, mejoraría la felicidad de sus clientes generando una dinámica positiva del uso de sus servicios, lo que puede conllevar un aumento en los beneficios empresariales.

1.3. Descripción del problema

Para llevar a cabo una relación pragmática con el mundo real, este trabajo tendrá muchas referencias al mundo de los transportes públicos, el cual muestra de forma clara la problemática explicada anteriormente y mejorarlos es el principal objetivo de este trabajo.

La movilidad y el transporte son uno de los motores de crecimiento económico de un país [1]. Desafortunadamente, este motor muestra signos de debilidad por los nuevos tiempos, debido principalmente a los problemas de congestión, accesibilidad y seguridad vial, derivados en su mayoría por la masificación de las grandes ciudades que concentran, en muchos casos, más del 70% de las personas que habitan un país. El campo de la innovación en los transportes, aunque es un campo que se está desarrollando a pasos agigantados, es un campo donde las capacidades son enormes y es relativamente nuevo, por ejemplo: el congreso internacional ITS (Intelligent Transport System) se viene celebrando desde hace 25 años que, si lo comparamos con la aparición de los primeros transportes públicos, se ve que es un congreso relativamente novedoso en su campo.

Los transportes, comúnmente se han caracterizado por intentar atraer a los usuarios mediante la mejora de sus propios servicios, mejorando la velocidad de las conexiones entre ellos, mejorando la comodidad e incluso mejorando los beneficios económicos que estaban asociados a ellos. Pero, aunque estas mejoras han sido de muy importante calado, el mundo del transporte privado ha seguido avanzando de manera exponencial en los últimos años, ya que no se ha logrado dar respuesta a muchas de las preocupaciones y molestias que sufre el usuario en su día a día. En muchas ocasiones, para muchos usuarios, el uso del transporte público es una auténtica agonía o es tomado como último recurso para llegar al objetivo deseado.

Debido al uso del transporte privado de forma masiva, la contaminación mundial ha aumentado sobremanera, siendo actualmente uno de los principales problemas que afronta la humanidad en su conjunto.

Este es el escenario donde la Unión Europea quiere aportar su claro apoyo mediante una serie de programas (como el programa Horizonte 2020) gracias a los cuales fomentar de manera clara el uso del transporte público dentro de la eurozona. Para ello, existen diferentes acercamientos gracias a los cuales los transportes públicos serán fundamentales para la comunicación del usuario con su destino. Algunos de ellos como Shift2Rail [2] o My-TRAC [3], buscan la facilidad y la sencillez para que el usuario pueda despreocuparse en sus trayectos y realizar un viaje desde su casa a su destino teniendo que interactuar lo menos posible con los diferentes operadores de transportes.

También existen iniciativas fomentadas por ciertos países del norte de Europa, como Finlandia y Noruega que buscan, mediante otras aproximaciones como tarifas planas de transporte, acercar los distintos medios de transporte públicos a los usuarios para que éstos no tengan que coger el transporte privado. Incluso existen ciudades que están intentando realizar su propia personalización de estaciones dependiendo del flujo de usuarios que en ellas se encuentren, como es Londres.

Es en este sentido donde cobra fuerza este trabajo, ya que uno de los motivos por los que un usuario suele rechazar utilizar el transporte público es por los continuos atascos y colas que pueden existir a ciertas horas puntuales dentro de las estaciones. Estas colas se producen, sobre todo, a horas clave para el usuario, como pueden ser la de entrada o la de salida del trabajo, o cuando se quiere ir a eventos como conciertos o partidos de fútbol. En estos ejemplos tan claros, queda patente la necesidad de un sistema automatizado que sea capaz de gestionar correctamente la apertura de puertas desde un lado hacia el otro, ya que son situaciones en las que los usuarios principalmente utilizan solo un flujo de salida o, de entrada, estando el otro flujo prácticamente sin utilizar.

Es evidente, que la recepción de señales de los múltiples usuarios mediante múltiples canales, al generar una gran multitud de datos en tiempo real, puede llevar en un futuro a la personalización absoluta de las redes y los sistemas de tránsito de personas, haciendo cada vez más sencillo y atractivo el uso de los transportes públicos a los usuarios.

1.4. Objetivos

Los objetivos de este proyecto son múltiples y variados, pero fundamentalmente el proyecto intenta llevar a cabo y conseguir resolver problemáticas del mundo real actual, aplicando las últimas tecnologías, e investigando como mejorar y unificar todos los resultados de ellas por separado, para intentar conseguir reducir el estrés ocasionado por los diversos sistemas de validación existentes en las estaciones, adelantándose a que estas situaciones ocurran.

Esta investigación será llevada a cabo para ser implantada en la actualidad o al menos en un futuro próximo, para que pueda dar soporte y mejorar la vida de las personas.

El objetivo general de este trabajo será asentar las bases para la realización de sistema de tránsito dinámico que ayude a los usuarios en sus rutinas de transporte y que pueda ser mejorado en siguientes estudios o trabajos.

Los objetivos específicos del problema propuesto de la toma de decisiones en situaciones de estrés mediante detección de personas se basarán en una serie de principios que serán los siguientes:

- Facilidad de implantación
- Bajo coste
- Sistemas sin licencia (*open source*)
- Alta resiliencia a fallos
- Escalabilidad
- Aséptico del sistema operativo

Como se puede observar por los principios anteriores, para cumplir con los objetivos para el desarrollo de la solución se ha tenido en cuenta el conocimiento adquirido en el mundo laboral. Gracias a este conocimiento se ha considerado no solo la parte investigadora, sino también la visión del cliente que podría ser el usuario del sistema, y que en este caso es una parte fundamental del proceso investigador, ya que si no se tiene en cuenta el pensamiento o la opinión del posible cliente se corre el peligro de que, aunque los proyectos de investigación sean muy ambiciosos, nunca serán llevados a cabo por las empresas, quedándose estos en un injusto olvido.

Por ello, ahora se pasará a explicar el motivo de la elección de cada principio de manera detallada centrándonos en los objetivos de este trabajo.

1.4.1. Facilidad de implantación

Cualquier sistema novedoso que se intente implantar y explicar a unos posibles usuarios o clientes, debe tener un punto de sencillez para ser instalado y manejado, ya que así podemos disminuir las posibles reticencias de los clientes o usuarios a aceptar este sistema [4].

Estas reticencias existen siempre por los miedos a lo desconocido y a la novedad. Son reticencias intrínsecas a la persona, que además pueden verse acrecentadas por una complejidad de uso de los sistemas o una falta de rendimiento en ellos, lo que hace que la curva de aprendizaje sea lineal en vez de logarítmica. Por lo tanto, aumentará los temores a salir de la zona de confort en la que están asentados los posibles usuarios del sistema [5].

1.4.2. Bajo coste

Que el precio sea bajo, permite que al sistema pueda acceder prácticamente cualquier empresa, lo que facilita entrar en el mercado y obtener referencias para asentarse y seguir mejorando en entornos reales. Tener un bajo precio, hace que cualquier proyecto de inversión pueda ser llevado a la fase de pruebas reales de manera más sencilla y que muchos clientes puedan beneficiarse de él.

Además, el proyecto es totalmente modularizable por lo que si algún hardware es más caro o no interesa puede eliminarse, o si se dispone de dispositivos que logren captar inputs del entorno (por ejemplo, cámaras de video) el sistema se puede adaptar perfectamente sin tener que realizar compra de hardware, ya que el sistema no está adherido a ningún hardware, sino que es totalmente aséptico a ellos.

1.4.3. Sistema sin licencia (open source)

Este es un punto fuerte del proyecto, ya que el software a utilizar estará totalmente libre de licencias comerciales, lo que abarataría su mantenimiento y daría libertad total para su uso [6].

1.4.4. Alta resiliencia a fallos

El sistema estará desarrollado bajo una arquitectura modular. Esta arquitectura modular hace que el sistema pueda trabajar con los datos de los que dispone en cada momento, de esta forma, aunque el sistema pueda tener caídos algunos sistemas hardware, con los que funcionan puede seguir emitiendo resultados.

Se hace evidente comentar, que, aunque puede seguir funcionando y emitiendo datos “con sentido” al perder sistemas hardware que envíen datos de lo que está pasando en el mundo real, el sistema perderá en cierto sentido la granularidad y el detalle con el que funcionaba al tener el sistema completo.

Además de la resistencia a los fallos de hardware, también tiene una enorme resistencia a los fallos de software e incluso de red, ya que cada módulo trabaja de manera independiente entre sí. Por lo tanto, si algunos de los módulos ofrecieran algún tipo de error por una mala configuración o por caídas en la red, los otros sistemas podrían seguir funcionando. Aun así, el sistema tiene una parte con una funcionalidad central, sin la cual no tendría sentido seguirlo teniendo en funcionamiento.

1.4.5. Escalabilidad

Al estar el sistema totalmente modularizado, es posible dar más potencia a ciertas partes de la aplicación e incluso replicar horizontalmente las funcionalidades para no saturar la aplicación.

Este beneficio hace que la aplicación esté viva y pueda adaptarse fácilmente al rendimiento que necesite el entorno. Entre más hardware e inputs tengamos en el sistema, se producirá un mayor aumento de requerimientos del sistema. Esta característica es muy importante en sistemas

reales que pueden ir ampliándose o disminuyendo sus inputs con el tiempo. También, un factor que afectará enormemente al sistema será el número de catas en el tiempo que se vayan haciendo para enviar datos a los módulos de cálculo centrales.

Esta característica, aunque se tiene en cuenta en el desarrollo, no será de importancia crítica dentro del alcance de este proyecto. Ya que en él se realizarán catas de 2 o 3 dispositivos hardware diferentes y cada 30 segundos.

1.4.6. Aséptico del sistema operativo

El sistema desarrollado será totalmente independiente del sistema operativo en el que se utilice. Esta característica es fundamental si se quiere llevar algún día el proyecto investigador al mundo real, ya que muchas empresas temen el pago de licencias de ciertos sistemas operativos, sobre todo, cuando estas licencias se cobran por hardware instalado. Tenemos que tener en cuenta, que un sistema como este puede contar con miles o millones de dispositivos hardware interconectados y que pagar licencia por cada uno de ellos puede hacer inviable la puesta en marcha de un sistema de estas características.

Por ello, y como parte de este estudio se llevará a cabo el desarrollo del sistema de toma de decisiones 3 de los sistemas operativos más conocidos que son: Windows, Linux y OS. Además, una parte visual y de comprobación de que el sistema está funcionando se llevará a cabo en un dispositivo Android, aunque el código también esté listo para funcionar en OS.

Por todo esto, se considera que el proyecto es totalmente aséptico del sistema operativo en que corre.

1.5. Organización del documento

La organización del documento se realiza en cinco capítulos.

En el primer capítulo se hace una introducción a todo el sistema que se va a explicar, detallando en él las motivaciones, justificaciones y objetivos que han llevado a tomar la decisión de emprender este trabajo basándose en los conocimientos aprendidos en la asignatura de Sistemas Difusos. En él también se encontrará explicada la organización del documento para una comprensión más fácil de este.

En el segundo capítulo se realiza un estudio pormenorizado del estado del arte en las diversas disciplinas asociadas a este trabajo. En él se detallará una revisión bibliográfica sobre las tecnologías y estudios existentes hasta la fecha en la materia tanto de sistemas difusos, como de transportes o inteligencia artificial. También se explicarán los fundamentos teóricos con los que se llevará a cabo la propuesta y se identificarán soluciones similares que permitan justificar el trabajo realizado por el alumno.

El tercer capítulo está destinado a explicar la propuesta realizada, detallando cada paso dado y justificándolo para demostrar que sus suposiciones son válidas, sus mejoras aportan un nuevo punto de vista al problema o que la fusión de diversas tecnologías aplicadas a este problema realmente produce resultados destacables en su campo.

El cuarto capítulo está destinado a los resultados obtenidos y en él se explicarán qué validez tienen los resultados, intentando demostrar que realmente el sistema funciona y también se explicará si se ha conseguido el objetivo propuesto al principio del trabajo. Tanto en caso de que se haya conseguido, como que no se haya conseguido, se mostrarán los problemas con los que se han encontrado y cómo se han podido o no solventar.

El quinto capítulo contendrá las conclusiones y el posible trabajo futuro que se puede realizar a partir de esta investigación.

En el anexo, por otra parte, se mostrarán explicaciones más detalladas sobre diversas partes del sistema de una manera más técnica, que consiga disipar las dudas que pudiesen existir sobre la resolución de la propuesta.

2. Estado del arte

En este capítulo se realizará una investigación que llevará a profundizar en los diferentes aspectos relacionados con los sistemas difusos dentro del ámbito de los transportes. Buscando para ello, tecnologías existentes utilizadas y, también, realizando comparativas entre sistemas similares al desarrollado para poder conocer las herramientas y las tecnologías utilizadas en otros estudios.

Actualmente los sistemas difusos para el apoyo de toma de decisiones están relacionados con multitud de diferentes campos laborales, van desde campos tan diversos como la industria hasta la sanidad.

En el trabajo actual nos centraremos en ver el estado actual del arte de los sistemas difusos para la toma de decisiones en ámbitos en los que el tiempo real es imprescindible para la mejora de los servicios para los usuarios. En esta cantidad de servicios se buscarán, sobre todo sistemas que tengan como fuentes de datos diversos sistemas que estén relacionados con los usuarios y que, por tanto, mediante ellos, se pueda conocer el número de usuarios que están utilizándolos en cada momento.

Algunos de ellos pueden ser:

- Vídeo: mediante la señal de vídeo, existen diferentes tipos de algoritmos que pueden reconocer patrones de personas en las imágenes.
- WiFi: gracias a diversos sistemas de escucha de señales WiFi, podemos descubrir el número de dispositivos que está buscando red en cada momento, esto daría una señal aproximada de los usuarios que puede haber en cada momento en un lugar.
- Bluetooth: gracias a diversos sistemas de obtención de señal Bluetooth, podemos descubrir el número de dispositivos que está buscando señal en cada momento, obteniendo una aproximación para conocer el volumen de usuarios que puede haber en cada momento en un lugar.
- Aprendizaje Máquina (*Machine Learning*): los sistemas difusos pueden ir aprendiendo continuamente mediante algoritmos de *Machine Learning* que definan y mejoren la granularidad de las categorizaciones que se realicen.

2.1. Categorización de imágenes de vídeo

Actualmente, la cantidad de fotos o videos que se comparten y utilizan a través de internet es gigantesca y cada vez más y más PetaBytes de información se acumulan sin control en los

servidores. Por ello, es muy importante la utilización de algoritmos genéticos y de lógica difusa para poder categorizar las imágenes que se vayan sucediendo y almacenar correctamente la información para poder realizar sobre ella distintas operativas, desde borrarlas con el paso del tiempo a aplicar un mayor formato de compresión.

Algunos trabajos han logrado mejorar la capacidad para categorizar logrando obtener cuadros que captan la información más relevante de cada *frame*, gracias sobre todo a los algoritmos genéticos que permite que el sistema aprenda con cada intento y así mejorar el sistema difuso con cada iteración [7]. Con el conjunto total de cuadros se puede obtener una buena categorización del video para poder etiquetarlo y realizar con él diversas tareas.

Por otra parte, dentro del dominio de las escenas de video, se han desarrollado también investigaciones en el ámbito de la seguridad, ya que esta es una parte importante dentro del uso del video en entornos reales. Utilizando lógica difusa se logra mejorar la detección y el seguimiento de las personas [8], y también, resuelve uno de los problemas fundamentales, como es el solapamiento de varios eventos dentro de una misma escena, logrando mejorar los rendimientos de los algoritmos existentes anteriormente prácticamente un 2% y rozando el 100%.

2.2. Sistemas difusos basados en tecnología WiFi

Gracias a que la mayoría de dispositivos, en la actualidad, poseen diferentes formas de interactuar con señales WiFi (*Wireless Fidelity*), estos pueden ser localizados en diversos sistemas *indoor*. En la actualidad, mediante el conocimiento de la potencia de la señal WiFi en diversos puntos de acceso, se pueden generar mapas que permitan a sistemas terceros generar sistemas de localizaciones *indoor*. Estos sistemas de localización, pueden ser sensibles a determinadas longitudes de onda que pueden introducir ruidos en el sistema, haciendo que el sistema obtenga ciertos errores a la hora de posicionar los objetos. Gracias al método Fuzzy Rule-based Classifier (FRBC) se logra reducir este ruido y conseguir un mejor posicionamiento de los elementos dentro de los edificios [9].

Aprovechando la disponibilidad de WiFi en los móviles, también se están consiguiendo crear entornos inteligentes, que logren interactuar con el usuario de manera automática gracias a la detección de este en el entorno [10].

Este paradigma lleva un paso más allá los estudios realizados sobre robots (mucho más sencillos de predecir) y hace que el entorno interactúe con elementos más dinámicos como son las personas. Este sistema, totalmente funcional, logra que el usuario se sienta más atraído al entorno, ya que hace que este se adapte de manera automática (después de unas configuraciones iniciales) a los requerimientos del usuario. Aspectos como la humedad, la temperatura o la música ambiente podrían con este sistema ser fácilmente automatizados para el entorno.

Otro punto importante en los que los sistemas WIFI son utilizados con sistemas difusos son Sistemas de Gestión de Energía en el Hogar (HEMS). Estos sistemas logran gracias a la aplicación de sistemas difusos de toma de decisiones, en conjunto con algoritmos genéticos, reducir el consumo de energía en casas y zonas residenciales hasta más de un 12% [11]. En la actualidad, cuando el planeta entero se está preocupando por el ahorro energético y la reducción de la contaminación, estos sistemas toman una relevancia mayor.

También son importantes los dispositivos WIFI para el envío de alarmas dependiendo de factores ambientales [12]. Por ejemplo, se pueden enviar señales de incendios dependiendo de la captación de diversas entradas en el entorno y aplicando a estos valores algoritmos de lógica difusa, entonces, si es requerido se envían alarmas mediante señales WIFI con procesos IFTTT.

2.3. Posicionamiento basado en tecnología Bluetooth

Los Smartphone y dispositivos Bluetooth están actualmente integrados en la vida cotidiana de las personas, ayudándoles en muchos aspectos de la vida. Uno de los principales aspectos en los que ayudan a las personas es en la localización en posiciones *indoor* [13]. Gracias a estos sistemas, y basándose en el uso de dispositivos Bluetooth de bajo consumo (*Bluetooth low energy*, BLE) se puede saber dónde se encuentra cada dispositivo en el momento, guiarle a través de una red de pasos o incluso realizar mapas de calor.

La utilización de estos sistemas pasa a ser totalmente relevante en el entorno de las ciudades inteligentes, ya que, mediante un bajo consumo de energía, un bajo coste y la utilización de una serie de algoritmos de lógica difusa en los que se utiliza como parámetros la intensidad y la cercanía a los distintos beacons se pueden localizar con exactitud la posición de los elementos del entorno, permitiendo así su guiado o la comprensión de lo que está pasando o puede suceder en el sistema.

2.4. Aprendizaje Máquina como soporte a los sistemas difusos

El desarrollo del aprendizaje automático o aprendizaje máquina (del inglés, *machine learning*) en la actualidad no ha estado asociado, por regla general, a la lógica difusa. En gran medida, porque la lógica difusa no ha sabido mostrarse, en muchas ocasiones, a las nuevas tecnologías y descubrimientos que se estaban haciendo en los diferentes campos existentes, entre ellos, el campo del *machine learning* [14]. Últimamente, se denota que cada vez más se van enlazando los caminos, y mientras los desarrollos y fundamentos del aprendizaje automático logran crear un aprendizaje en las máquinas, el campo de la lógica difusa logra, por su parte, dotarlas de una gran comprensión en la toma de decisiones. Consiguiendo de esta manera, potenciar ambas áreas de una manera simbiótica.

Algunas de las áreas importantes en las que han trabajado de manera conjunta son:

1. Identificación de casos de *bullying* en colegios [15]: En este caso se ha utilizado un sistema de *machine learning* para interpretar diversas tipos palabras en los textos de los alumnos y mediante sistemas difusos se toman las decisiones oportunas para categorizar el tipo de alumno que es.
2. Autogestión de redes [16]: Gracias a este sistema, se pueden automantener redes gracias a las decisiones tomadas por un motor de lógica difusa que es actualizado continuamente por un sistema de *machine learning* que permite dinámicamente mejorar en la toma de decisiones debido a sus algoritmos (*k-Means and k-Nearest Neighbour*).

2.5. Lógica difusa para la gestión del transporte de usuarios

Ahora se va a pasar a identificar soluciones similares a las propuestas o que estén relacionadas con el dominio principal del trabajo, en el caso de existir soluciones similares, se profundizará en las herramientas y tecnologías propuestas para obtener una base que nos permita comenzar y avanzar nuestra solución sobre una base sólida existente.

Los fundamentos teóricos sobre este tema son múltiples y variados, existiendo una gran bibliografía al respecto.

En [17] podemos ver como para la ciudad de Dublín se proponen aplicaciones para la concienciación sobre el uso del transporte público mediante análisis y la visualización en tiempo real grandes volúmenes de datos de tráfico procedentes de diversas fuentes. Para ello utilizan IBM InfoSphere Streams [18] que es una plataforma de análisis de datos en tiempo real con micro-latencia. En vez de recopilar los datos, guardarlos y luego analizarlos, InfoSphere Streams aplica el análisis a los datos en movimiento. El análisis puede disparar eventos o alertas, de manera que se pueda reaccionar en tiempo real. InfoSphere Streams está diseñado para ser más escalable y soportar flujos de datos a un mayor ritmo que las soluciones tradicionales de procesamiento complejo de eventos.

También, buscando en las diferentes exposiciones sobre ciudades inteligentes y transporte, se pueden encontrar trabajos que se asemejan en cierta medida al actual, desarrollados por algunos grandes del sector [19]. Algunos de estos, como la compañía alemana Deutsche Bahn o la Corporación América en unión con Transport for London están llevando a cabo trabajos para optimizar el mantenimiento de los servicios ferroviarios y mejorar sus sistemas o, por ejemplo, recolectar información mediante cámaras que permita saber en tiempo real la cantidad de personas que hay en las estaciones, de modo tal que el usuario pueda anticipar si se encontrará con mucha congestión al querer tomar el transporte. Además, también están llevando a cabo algunos desarrollos para avisar a personas invidentes de la llegada del transporte.

Por otro lado, también se están realizando investigaciones [20] para desarrollar modelos que simulen estaciones de pasajeros y que sirvan para resolver y anticipar los problemas sobre la

elección de otras líneas en caso de que el tren actual esté retrasado. En la búsqueda de una solución para estos problemas, se utilizan algoritmos difusos para determinar variables lingüísticas como sería “poco”, “medio”, “mucho” ... que no pueden ser llevadas al entorno matemático y por ello hay que tener en cuenta la lógica difusa. Este ejercicio se realiza con más de 100 variables, y con ellas realiza una reducción para obtener solo los valores de las más relevantes. Para todo este proceso se está basando en el sistema Mamdani y para obtener los datos utiliza el sistema GTN de la compañía de trenes de República Checa.

En relación a los algoritmos de regresión en los sistemas difusos y su uso en entornos de generación de modelos para el aprendizaje automático [21], podemos decir que usan los estudios de Mamdani y Sugeno para realizar una fuzzificación de los datos que entran en la base de datos y así tratar de transformarlos a lenguaje matemático para poder ejecutar funciones en base a los resultados obtenidos. La librería R del lenguaje de programación Python utiliza, por defecto, estos estudios para realizar sus funciones de regresión. En este sentido, existe también la librería Scikit-fuzzy [22], que hereda sus funciones de la librería Scikit-learn que es una de las librerías más utilizadas para los desarrolladores Python a la hora de ejecutar funcionalidades de aprendizaje automático o inteligencia artificial.

Para categorizar los distintos sistemas difusos existen diversos estudios y diferentes modos. En [23] para realizar las categorizaciones los diferentes cuartiles que se decidan de antemano. Estos cuartiles dividirán las gráficas dependiendo del número de resultados existentes en cada cuartil, así se puede determinar de manera sencilla en que rango de la escala se encuentra el valor determinado [24]. Este sistema de categorización es sencillo, potente y es utilizado en sistemas de regresión, como en árboles de decisión [25], para dotar de mayor velocidad y potencia al sistema, ya que consigue reducir el número de ramas o resultados ajustándolos a los distintos cuartiles y evitando que puedan existir un número no controlado de soluciones.

2.6. Movilidad como Servicio (MaaS)

El futuro del transporte está en lo que se denomina actualmente, Movilidad como servicio (MaaS) que es la integración de varias formas de servicios de transporte en un único servicio de movilidad accesible bajo demanda por el usuario. Para satisfacer la demanda de un cliente, un operador de MaaS facilita una amplia gama de opciones de transporte, ya sea transporte público, paseo, coche o bicicleta compartida, taxi o alquiler de coches, o una combinación de los mismos. Para el usuario, MaaS puede ofrecer un valor añadido a través del uso de una única aplicación para proporcionar acceso a la movilidad, con un único canal de pago en lugar de múltiples operaciones de ticketing y pago. Para sus usuarios, el MaaS debería ser la mejor propuesta de valor, ayudándoles a satisfacer sus necesidades de movilidad y a resolver las partes inconvenientes de los viajes individuales, así como todo el sistema de servicios de movilidad.

Un servicio de MaaS [26] correctamente implementado aportará nuevos modelos de negocio y formas de organizar y operar las diversas opciones de transporte, con ventajas para los operadores de transporte, incluyendo el acceso a una mejor información sobre los usuarios y la demanda y nuevas oportunidades para atender la demanda insatisfecha. El objetivo de MaaS es proporcionar una alternativa al uso del coche privado que pueda ser más conveniente, más sostenible, que ayude a reducir la congestión y las limitaciones de la capacidad de transporte, pudiendo ser incluso más barata.

No será necesario comprar billetes de viaje ni contratar cuentas de transporte separadas [27]: las cuentas de usuarios que MaaS ofrecerá a los pasajeros, a sus familias o a sus empresas darán la libertad de elegir la movilidad que necesiten, mediante una suscripción mensual o el pago por uso.

Teniendo en cuenta que los usuarios dispondrán de una sola cuenta para sus viajes a través de múltiples transportes, y que esa cuenta puede asociar distinta información del usuario como pueden ser:

- Tarjetas de transporte convencionales
- Tarjetas de crédito
- Información biométrica
- Información de dispositivos móviles o *wearables*.

Se podrá utilizar toda esta información para reconocer al viajero dentro de las estaciones, y es sencillo pensar que, utilizando la información que emiten los dispositivos que lleva consigo el usuario es posible saber dónde entra y dónde sale y así poder luego calcular la tarifa de viaje sin tener que molestar al viajero en ningún momento.

3. Solución propuesta

En el contexto de este estudio podemos preguntarnos hasta qué punto es posible automatizar la resolución de los problemas de transporte centrándonos en sus flujos de pasajeros y qué métodos necesitamos para lograr resultados. Un cerebro humano sabe cómo trabajar con términos vagos intuitivamente y también puede tomar a menudo decisión altamente cualificada sobre la base de incertidumbre información, que no está cuantificada exactamente. El cerebro conoce y trabaja normalmente con los términos “bajo”, “medio” o “alto” que podrían obtenerse de un flujo de viajeros en una estación, pero las matemáticas no conocen esos términos ni saben cómo ejecutar ecuaciones sobre ellos.

Los métodos matemáticos clásicos, en principio, no nos permiten hacer nada apropiado en este momento. Siempre existe el dilema que surge de un hecho que, cuando una complejidad de cualquier sistema y una cantidad de información sobre él aumentan, nuestra capacidad de procesar esta información, de describir y de predecir por medio de este sistema de información, disminuye. Es en este entorno de duda donde se desenvuelve realmente bien la lógica difusa, ya que puede llevar estos sistemas de múltiples entradas con términos vagos a entornos matemáticos y entonces, poder llevar a cabo operaciones matemáticas con ellos, y que estas sirvan para tomar las decisiones correspondientes.

Teniendo en cuenta los problemas anteriores, se ha creado un sistema que cumple con los objetivos propuestos y cuya arquitectura a alto nivel se describe en la Figura 1. El sistema se compone de tres partes bien diferenciadas. La parte principal y más importante es el sistema central, que es el núcleo de la aplicación, donde se aplican en diversos momentos los algoritmos de toma de decisiones y de aprendizaje automático, aunque también se encuentran en él las bases de datos en las cuales se almacena toda la información que debe ser procesada por el sistema. Por otro lado, la parte secundaria, pero no por ello menos importante, es la parte de los dispositivos periféricos. Estos dispositivos obtienen datos del mundo real de distintas maneras: mediante WiFi, señal de imagen, Bluetooth o mediante la información de barreras. Esta información es enviada de manera individual por cada dispositivo al sistema central para que la almacene y opere con ella. La información de estos dispositivos es realmente importante ya que sin esta información el sistema estaría prácticamente ciego, sobre todo en las fases iniciales donde el aprendizaje de los algoritmos es fundamental. Aquí cabe remarcar, que los sistemas de reconocimiento de personas en imágenes, utilizan unos algoritmos ya entrenados que pueden reconocer personas de manera fácil mediante videos.

Los dispositivos existentes en el sistema son etiquetados de dos formas diferentes dependiendo de su función en el sistema:

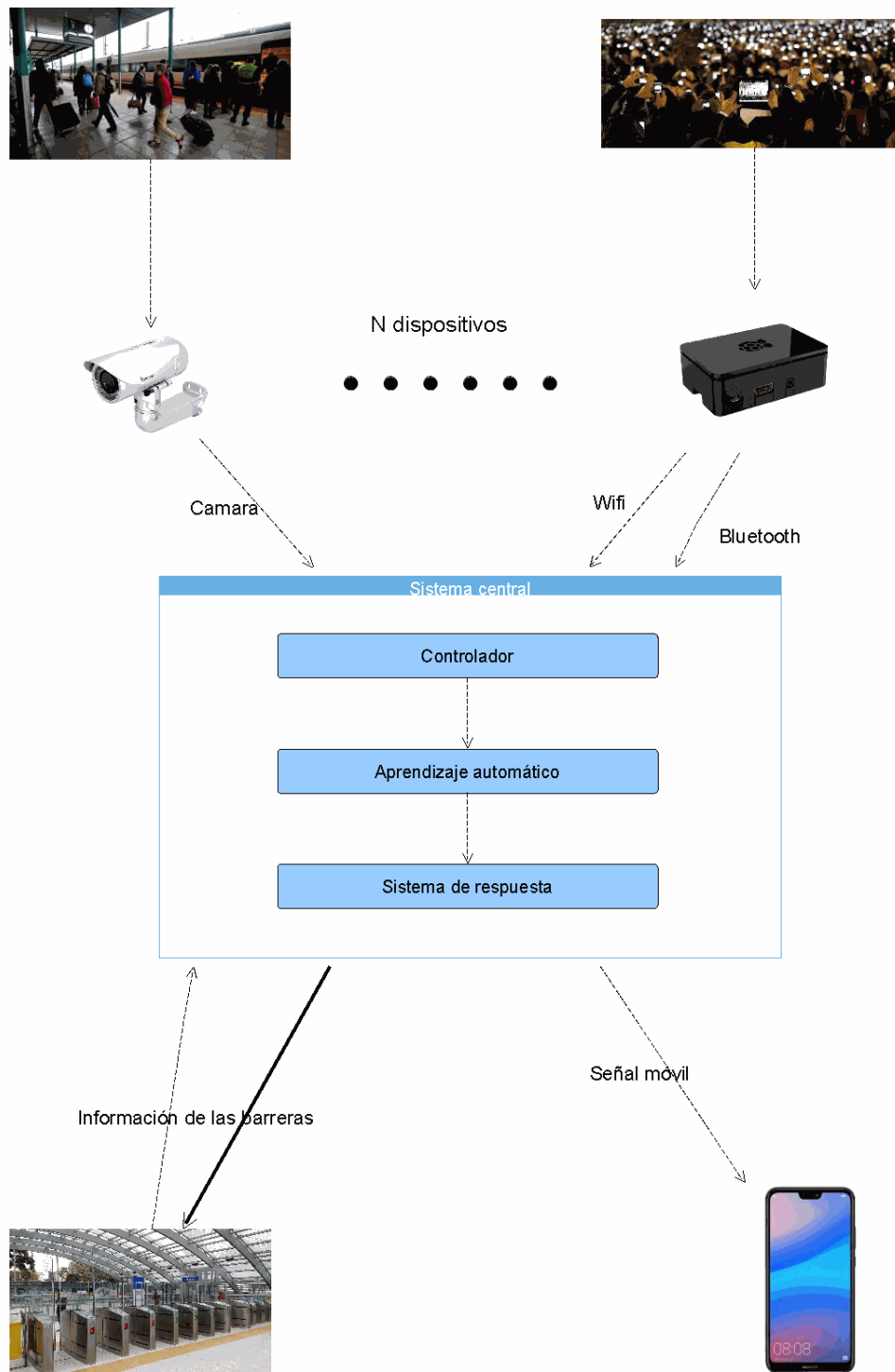


Figura 1 Esquema global del sistema

- Objetivo: Los dispositivos objetivos son aquellos sobre los que se realiza el aprendizaje automático mediante el uso de la lógica difusa. Estos dispositivos son de vital importancia en el sistema, ya que mediante ellos calculamos la ocupación real, puesto que el sistema confía en el resultado de la detección que estos dispositivos envían. Gracias a la confianza en estos dispositivos obtenemos un valor confiable que determina la ocupación real en el sistema. La funcionalidad final del sistema, y su entrenamiento, es adelantarse a los resultados que estos dispositivos van a encontrarse

en un futuro que puede ser más cercano o lejano en el tiempo, y en última instancia, eliminar estos dispositivos ya que el sistema está lo suficientemente bien entrenado para no necesitar obtener datos de estos dispositivos, sino que pueden hacerse funcionar de manera automática en función de las detecciones de los demás. Estos dispositivos, además, pueden realizar funcionalidades determinadas dependiendo de las reglas existentes en el sistema.

En el caso actual, se determina solamente un dispositivo objetivo que serán las barreras existentes en las estaciones. Las barreras de las estaciones serán un dispositivo claro sobre el cual determinar la ocupación del sistema, ya que las personas deben de salir siempre de las estaciones, por lo tanto, serán una medida suficientemente confiable y, además, uno de los objetivos es eliminarlas con el tiempo o al menos disminuir su influencia en el paso de los viajeros.

- No objetivo: Los dispositivos no objetivo tienen como función suministrar los datos de personas que estén detectando en tiempo real para conocer lo que está pasando en su zona de influencia. Los datos de estos dispositivos por sí solos no aportan gran valor al sistema, ya que se nutre de multitud de dispositivos no objetivos y cuyos datos se cruzan con los de los dispositivos objetivos correspondientes. Gracias a los datos de cada dispositivo no objetivo, junto con los datos de los objetivos objetivo se podrá realizar una matriz que ejecuta un modelo que predice, dependiendo del número de personas detectadas por los dispositivos, qué va a suceder en el dispositivo objetivo.

Por último, tendremos una tercera parte, que es la fase de acción de todo el sistema. Esta tercera fase consiste en que el sistema central, después de realizar todos los cálculos debidos con la información que recibe, envía la señal correspondiente al dispositivo que corresponda. En este sentido, el dispositivo ejecuta la función para la que ha sido programado según la respuesta recibida.

La respuesta podrá ser singular, que es el foco de este proyecto, o también múltiple, lo que hace que varios dispositivos puedan recibir su respuesta correspondiente y que no tiene que ser igual para cada uno de ellos.

En este trabajo se intentarán utilizar pasos reales de estaciones, pero por si no fuese posible disponer de dicho hardware, se ha desarrollado una aplicación móvil que informa del estado de la ocupación en el lugar donde el sistema esté instalado.

Ahora, se pasará a explicar más en detalle el sistema desarrollado en cada parte:

3.1. Sistema central

El sistema central es la zona donde se reciben todos los datos provenientes de las fuentes existentes. Este sistema obtiene, de esta manera, todos los datos en tiempo real de los dispositivos periféricos y realiza las operaciones correspondientes.

Para realizar lo anterior, el sistema central cuenta con tres sistemas diferenciados:

3.1.1. Controlador

El controlador es la parte del sistema que se encarga de recibir y almacenar todas las peticiones que son enviadas por los dispositivos fuente. Estos datos enviados por los dispositivos son almacenados para su posterior tratamiento al realizar el aprendizaje automático.

El controlador contiene además un sistema de suscripción, por lo que los dispositivos pueden ser añadidos al sistema de manera asíncrona en cualquier momento. Si en el futuro se desea tener nuevos dispositivos, se pueden dar de alta de manera fácil y sencilla, suscribiéndolo al sistema. Por otro lado, también pueden ser borrados del sistema cuando ya no se utilicen y entonces no se tienen en cuenta a la hora de realizar el aprendizaje automático.

Este sistema también se encarga de activar el sistema de inteligencia artificial para realizar pronósticos, y de detenerlo en caso de que sea necesario.

3.1.2. Aprendizaje automático

En los últimos años la inteligencia artificial [28] ha ido aumentando su nivel en cuanto a investigación se refiere, los sistemas difusos se han venido consolidando como una herramienta útil para modelar sistemas complejos y no lineales como el desarrollado. Las técnicas de aprendizaje automático se han convertido en una herramienta fundamental para abordar problemas complejos, también en el área de control automático.

El sistema de aprendizaje automático es la base del sistema difuso desarrollado. Este sistema de aprendizaje automático sirve para obtener, en base a la ocupación del sistema detectada por los dispositivos, una respuesta que determine la acción a llevar a cabo.

Los flujos de personas en el sistema están determinados por la ocupación que se detecta en cada dispositivo. Según el dispositivo objetivo se ha dividido la ocupación en: Baja, Medio-Baja, Medio-Alta y Alta. Estas categorizaciones son muy vagas e imprecisas, ya que pueden depender de la persona que lo esté viendo en cada momento y habrá en ciertas ocasiones en las que el número de viajeros esté entre dos categorías sin saber exactamente en cual. Además, el sistema se complica ya que se pueden tener multitud de dispositivos que estén detectando personas en cada momento, por lo que la dificultad del sistema puede aumentar enormemente dependiendo del número de dispositivos y de las categorías que existan en el sistema.

Es por ello que las matemáticas clásicas no pueden ayudarnos a predecir qué está pasando en el sistema y se hacen necesario los modelos difusos para que nos ayuden a resolver y a eliminar de forma clara las imprecisiones existentes.

A diferencia de la lógica tradicional [29], que solo utiliza dos valores de verdadero o falso, la lógica difusa permite a los sistemas definir valores intermedios en un intento por aplicar un

modo de pensamiento similar al del ser humano. En esta situación los sistemas expertos tienen mucho que ver con lo que significa inferir conocimiento, utilizando las famosas reglas de inferencia o también conocidas como reglas de producción. Dentro de la lógica difusa se utiliza el método de inferencia de Mamdani [30] que hace uso de las reglas Si X Entonces Y, si premisa entonces conclusión.

El sistema de aprendizaje automático, se compone de dos partes fundamentales: categorización y generación del modelo:

3.1.2.1. Categorización

La primera parte es la realización de la categorización para poder ejecutar funciones dependiendo de la ocupación que se encuentre en el sistema. Un conjunto difuso A se caracteriza por una función de pertenencia:

$$\mu_{\tilde{A}}: U \rightarrow [0,1] \quad (1)$$

Que asocia a cada elemento x de U un número $\mu_{\tilde{A}}(x)$ del intervalo [0,1], que representa el grado de pertenencia de x al conjunto difuso \tilde{A} . A U se le llama universo de discurso. En nuestro caso, el término ocupado puede definirse mediante el conjunto difuso siguiente:

Ocupación	Grado de pertenencia
Baja	1
Media-Baja	0.75
Media-Alta	0.50
Alta	0.25

Tabla 1 Término conjunto difuso

Es decir, la función de pertenencia al conjunto difuso ocupación viene dada por:

$$\tilde{A} := \{x \in U : \mu_{\tilde{A}}(x) \geq \alpha\} \quad (2)$$

Que se puede representar en la gráfica mostrada en la Figura 2:

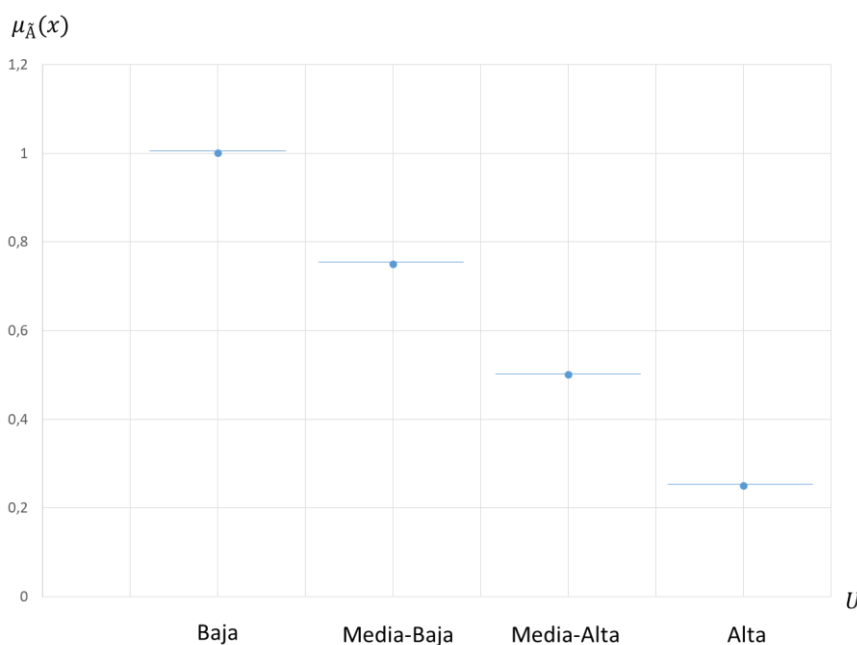


Figura 2 Gráfica pertenencia al conjunto difuso

En general, si una función de pertenencia se da especificando los valores correspondientes a un conjunto discreto de elementos del universo de discurso, el valor asociado al resto de los elementos se obtiene por interpolación (utilizando la ecuación de la recta que une los dos puntos).

Una vez visto todo lo anterior, y representado en las ecuaciones (1) y (2), la Tabla 1 y la Figura 2, aplicamos todos estos conocimientos al sistema actual. La ocupación del sistema depende totalmente de las personas (flujo de viajeros) que estén siendo detectadas por el dispositivo objetivo. En el sistema actual, hemos determinado la ocupación solo para el dispositivo objetivo y no para todos los dispositivos, ya que el sistema de aprendizaje automático, junto con la categorización por cuartiles, es capaz de relacionar las personas detectadas en cada momento en el sistema con la futura ocupación que vaya a tener el dispositivo objetivo. De tal modo, que la ocupación estará dentro de los límites definidos por la ecuación (3).

$$X_{Bajo}^{Medio} \leq Ocupación actual \leq Y_{Medio}^{Alto} \quad (3)$$

Donde X será el nivel de ocupación inferior e Y será el nivel de ocupación superior. De este modo, podremos granular la ocupación en tantos niveles como se desee.

Hay que tener en cuenta, que en la ecuación (3) están dentro los valores mínimos y máximos, es decir, los que se denominan vagamente como “Bajo” y “Alto”, por lo tanto, la ocupación actual podrá ser tanto “Baja” como “Alta” en algunas ocasiones. Aunque en un sistema productivo no deberían ser estados que se mantuviesen demasiado tiempo, ya que indicarán una sobresaturación del sistema o una infrautilización de los recursos. De todos modos, el

sistema es capaz de autobalancearse para que estos estados no sean indefinidos en el tiempo, ya que al final de cada día se realiza una ponderación de la ocupación total del sistema, haciendo que el número de personas detectadas para cada categoría pueda fluctuar. Esta fluctuación será mayor al principio, ya que el sistema dispondrá de pocos datos y debe de aprender, pero será casi inamovible en cuanto el sistema suficientes valores con los que trabajar.

Para obtener la categorización de la ocupación siempre necesitaremos unos niveles que nos indiquen las personas detectadas por los dispositivos objetivos que nos determinen la ocupación que va a haber en el sistema. Estos niveles de personas siempre cumplirán la norma definida por la ecuación (4).

$$\text{Ocupación} = \text{Niveles} + 1 \quad (4)$$

De tal modo que la ocupación existente en el sistema siempre será mayor en una unidad a los niveles existentes. En el sistema de trabajo actual se están determinando tres niveles, por lo que existen 4 categorizaciones. Los niveles los determinaremos mediante las palabras “Alto”, “Medio”, “Bajo” y dependiendo de estos niveles encontraremos las ocupaciones “Alto”, “Medio-Alto”, “Medio-Bajo”, “Bajo”, como se muestra en la Tabla 2.

Niveles	Ocupación
<i>Si número de personas < Bajo</i>	<i>Ocupación = Baja</i>
<i>Si Bajo < Número de personas < Medio</i>	<i>Ocupación = Media – Baja</i>
<i>Si Medio < Número de personas < Alta</i>	<i>Ocupación = Media – Alta</i>
<i>Si número de personas > Alta</i>	<i>Ocupación = Alta</i>

Tabla 2 Tabla relación Niveles-Ocupación

La Tabla 2 nos muestra las equivalencias existentes entre los niveles y la ocupación detectada por el dispositivo objetivo. Dependiendo de cada nivel de ocupación se ejecutan diferentes funcionalidades que llevan al sistema a obtener el dinamismo deseado. En el caso actual, solo dispondremos de 3 niveles, pero puede ser que en otros casos interese tener otro número diferente de niveles, aunque siempre existirá una relación similar a la anterior.

El número de personas detectadas por el dispositivo objetivo interfiere directamente con el comportamiento del sistema, ya que estas personas detectadas serán las que sobrepasen o no los distintos niveles existentes en el sistema, y nos indiquen la ocupación existente en él. Por lo tanto, hay una relación directa entre el número de personas detectadas y el número de personas que determina el nivel correspondiente.

Para determinar el número de personas que reflejan el mínimo de cada nivel, ejecutaremos la función cuantil. En probabilidad la función cuantil de una distribución de probabilidad es la inversa de la función de distribución, tal y como se define en la ecuación (5).

$$F^{-1}(p) = \inf\{x \in R : p \leq F(x)\} \quad (5)$$

Para una probabilidad $0 < p < 1$, devolviendo la función cuantil el valor mínimo de x para el cual se mantiene la probabilidad anterior.

Al existir 3 niveles, se han determinado 3 cuantiles que serán: 0,25, 0,5 y 0,75. Los cuantiles 0,25; 0,50 y 0,75 de la distribución normal, más conocidos como los cuantiles Q_1 , Q_2 y Q_3 , que dividen la distribución en cuatro bloques, cada uno de los cuales contiene el 25% de los datos.

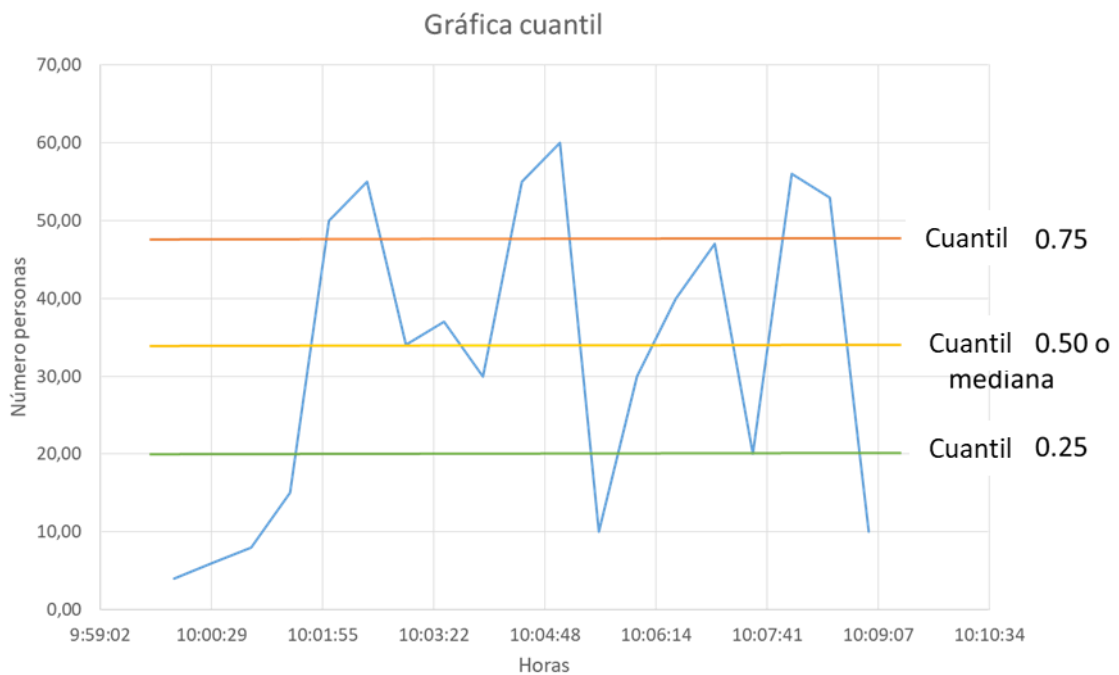


Figura 3 Gráfica cuantil

Estos cuantiles nos sirven para indicar cuál es el valor que mejor se adapta a cada nivel. Los cuantiles se usan con frecuencia en los datos de ventas y encuestas para dividir las poblaciones en grupos. Por ejemplo, usar la función cuantil para determinar el 25 por ciento de ingresos más altos en una población. De esta manera, se puede determinar fácilmente los niveles de ocupación de un dispositivo, ya que al utilizar el valor de los cuantiles como mínimo de cada categoría, se puede ver si el valor actual sobrepasa la cantidad de personas determinadas por el cuantil o no.

Los cuantiles suelen usarse por grupos que dividen la distribución en partes iguales; entendidas estas como intervalos que comprenden la misma proporción de valores. Dependiendo del número de cuantiles de los que dispongamos, tendremos el número de cuartiles, como vamos a tener 3 cuantiles (que serán 0,25, 0,50 y 0,75) tendremos 4 cuartiles que corresponde a los estados de ocupación que existen en el sistema. Es importante apreciar, que, aunque en este sistema se han considerado que estos cuantiles son suficientes para alertar de la ocupación en el sistema, no sería ningún problema aumentarlos si fuese necesario. Algunos otros cuantiles que se podrían tener serían:

- Quintiles: dividen a la distribución en cinco partes (corresponden a los cuantiles 0,20; 0,40; 0,60 y 0,80);
- Deciles: dividen a la distribución en diez partes;
- Percentiles: dividen a la distribución en cien partes.

En el cálculo de cuantiles con distribuciones de variable discreta (como es el caso que ocupa en este sistema) debemos conformarnos con que estas partes sean aproximadamente iguales. Para realizar esta operación matemática de aproximación al valor que divide la gráfica en partes iguales se ha tomado la decisión de utilizar una interpolación lineal ya que para distribuciones como las que nos ocupa, no existe ningún método que pueda determinar partes 100% iguales.

En el método lineal se presupone que el valor deseado se encuentra entre dos puntos de datos $i < j$ [31]:

- lineal: como se muestra en la ecuación (6), fracción es la parte fraccionaria del índice rodeada de i y j .

$$i + (j - i) * \text{fracción} \quad (6)$$

Se ha seleccionado esta interpolación porque los resultados son correctos para los casos de uso de prueba en laboratorio. Gracias a este sistema, podemos fácilmente determinar los 4 estados de ocupación (Alta, Medio-Alta, Medio-Baja, Baja). La ocupación del sistema nos indica entre qué rango de cuartiles nos estamos moviendo. Así, todos los pasajeros que se detecten en el rango entre el cuartil X y el cuartil $X+1$ estarán dentro de la ocupación Y correspondiente. Una vez determinada la ocupación del sistema se pasan a ejecutar las funciones, que en el caso del sistema serán envíos de señales al dispositivo objetivo.

3.1.2.2. Generación del modelo

Para la segunda parte, generación del modelo, se ha utilizado un algoritmo que nos permite realizar un aprendizaje automático de lo que ocurre en el sistema, y, además, es capaz de predecir lo que puede ocurrir en el sistema. Para ello, tiene en cuenta todos los datos que existan

almacenados en el sistema y a partir de ellos genera un modelo. Una vez el modelo es creado y se ha comprobado su fiabilidad a la hora de predecir resultados, se insertan los datos actuales que están detectando los dispositivos del sistema. En función de los valores detectados y los que se detectaron anteriormente, ya inmersos en el modelo, el sistema es capaz de predecir lo que va a ocurrir.

Para realizar este pronóstico se tienen en cuenta algoritmos de regresión logística semiparamétricos. Estos algoritmos nos ayudan a predecir, dependiendo del número de personas detectadas por los dispositivos no objetivos, cuál va a ser la ocupación de los dispositivos objetivos en un futuro próximo.

Los métodos de regresión se dividen principalmente en dos clases generales: los llamados métodos paramétricos y no paramétricos [32].

Los modelos de regresión paramétrica son útiles para describir la relación entre la respuesta y las variables dependientes. Tanto teórica como prácticamente, ha habido una fuerte y generalizada objeción al uso arbitrario de la regresión por cuantiles. La regresión cuantitativa está destinada a estimar y realizar inferencias sobre las funciones condicionales de los cuantiles. En comparación con las técnicas de regresión lineal estándar, la regresión por cuantiles es capaz de proporcionar un análisis estadístico más completo de las relaciones estocásticas entre predictores y respuestas, es decir, la regresión por cuantiles nos permite estudiar el impacto de los predictores en diferentes cuantiles de la distribución de la respuesta; por lo tanto, proporciona una imagen completa de la relación entre predictores y respuestas. La regresión cuantitativa se beneficia de algunas ventajas sobre las regresiones de mínimos cuadrados (absolutas); por ejemplo:

- Las regresiones de mínimos cuadrados (absolutas) pueden ser ineficaces en casos con errores altamente no normales, mientras que la regresión por cuantiles es más robusta ante errores no normales y valores atípicos.
- La regresión cuantitativa proporciona una caracterización de datos más rica que nos permite considerar el impacto de una covariable en toda la distribución de la respuesta, no sólo en su valor medio condicional.
- La regresión cuantitativa es invariable a las transformaciones monótonas.
- La regresión de cuantiles es más flexible que las regresiones de cuadrados (absolutos) mínimos para modelar datos con distribuciones condicionales heterogéneas.

Sin embargo, a veces se pueden aplicar métodos de regresión paramétrica y tener el riesgo de introducir sesgos de modelización.

Los modelos de regresión no paramétrica por otra parte, no asumen ninguna estructura de modelo anterior y pueden proporcionar información útil para un ajuste paramétrico posterior. Se puede ver que, al tratarse de un enfoque totalmente no paramétrico, puede causar algunos inconvenientes, como el problema de la dimensionalidad, la dificultad de interpretación y la falta de capacidad de extrapolación.

Por otro lado, utilizando el modelo de regresión semiparamétrica parcialmente lineal es más ventajoso y flexible que el modelo de regresión lineal correspondiente [33]. La razón es que tanto los componentes paramétricos como los no paramétricos pueden estar presentes simultáneamente en el modelo. En particular, el modelo paramétrico puede permitir una inferencia precisa, mientras que el modelo no paramétrico se asocia tanto con una mayor robustez como con una menor precisión. Una regresión semiparamétrica parcialmente lineal puede escribirse como:

$$y_i = x_i^t \beta + f(t_i) + \epsilon_i \quad (7)$$

Donde t_i es una variable observable introducida en una ecuación de forma no lineal de acuerdo con una función de valor real delimitada definida en $[0,1]$, y donde ϵ_i son errores aleatorios inobservables. El parámetro dimensional finito β es la parte paramétrica del modelo, y la función desconocida f representa la parte no paramétrica del modelo.

Este modelo ha sido ampliamente utilizado en econometría, finanzas, biología y sociología. Sin embargo, en muchas aplicaciones de sistemas expertos y de inteligencia artificial, como es el caso del sistema desarrollado en este trabajo, la relación entre un conjunto de variables de predicción y de respuesta no puede medirse con precisión debido a algunas situaciones inesperadas. Para tales casos, los métodos de regresión difusa se han aplicado con éxito como un método experto de predicción de la toma de decisiones, en el que algunos elementos del modelo están representados por datos difusos. En el sistema desarrollado, los datos para los dispositivos objetivo están basados en elementos de lógica difusa semiparametrizados, que se basan en cuantiles, para obtener los valores del dominio que es el de la ocupación.

Se ha elegido un modelo de regresión utilizando cuantiles para la parametrización porque es más flexible que los modelos de regresión de error menos cuadrado o menos absoluto. Esta técnica de regresión es muy usada en muchas industrias debido a su escalabilidad y por su facilidad a la hora de ser explicada. En este trabajo, se va a llevar este modelo al entorno difuso, utilizando para ello un método semiparamétrico basado en predictores no difusos, respuestas difusas y una función de suavidad difusa lineal como se puede observar en la ecuación (7).

El aprendizaje automático es el punto intermedio donde las habilidades computacionales y algorítmicas de la ciencia de datos satisfacen el pensamiento estadístico, y el resultado es una colección de enfoques de inferencia y exploración de datos, que no son tanto de teoría efectiva como de computación efectiva.

El aprendizaje automático se clasifica a menudo como un subcampo de la inteligencia artificial, pero esto puede ser engañoso. Es cierto que el aprendizaje automático surgió de la investigación en este contexto, pero en la aplicación de los métodos de aprendizaje automático en la ciencia de los datos, es más útil pensar en el aprendizaje automático como un medio para construir modelos de datos.

En última instancia, el aprendizaje automático implica la construcción de modelos matemáticos para ayudar a entender los datos. Este aprendizaje entra en juego cuando damos a estos modelos parámetros afinables, que pueden ser adaptados a los datos observados; de esta manera se puede considerar que el sistema aprende a partir de los datos. Una vez que estos modelos han sido ajustados a los datos previamente vistos, pueden ser usados para predecir y entender aspectos de los nuevos datos observados. Dependiendo del tipo de aprendizaje, este se puede categorizar en [34]:

- *Aprendizaje supervisado*: modela la relación entre las características medidas de los datos y algunas etiquetas asociadas con los datos; Una vez que este modelo es determinado, puede ser usado para aplicar etiquetas a datos nuevos y desconocidos. Esto se subdivide en tareas de clasificación y tareas de regresión. En la clasificación, las etiquetas son categorías discretas, mientras que, en la regresión, las etiquetas son cantidades continuas.
- *Aprendizaje sin supervisión*: modela las características de un conjunto de datos sin referencia a ninguna etiqueta, y a menudo se describe como “que el conjunto de datos hable por sí solo”. Estos modelos incluyen tareas como la agrupación y la reducción de la dimensionalidad. Los algoritmos de agrupamiento identifican distintos grupos de datos, mientras que los algoritmos de reducción de la dimensionalidad buscan representaciones más sucintas de los datos
- *Aprendizaje por refuerzo*: Estos algoritmos de aprendizaje definen modelos y funciones enfocadas a maximizar unas medidas por las que obtendrá “recompensas” y que están basadas en “acciones” y también en el ambiente en el que el agente inteligente se desempeñará. Este algoritmo es muy similar a la psicología conductista en los humanos, ya que es un modelo acción-recompensa, que busca que el modelo se ajuste a la mejor “recompensa” dada por el ambiente, y sus acciones por tomar estén sujetas a las recompensas. Este tipo de métodos pueden usarse para hacer que los robots aprendan a realizar tareas diferentes.

El sistema en concreto sobre el que se ha desarrollado el trabajo, se basa en un sistema de aprendizaje supervisado de regresión semiparamétrica de respuesta discreta. Esto es así, porque se utilizan los datos de las personas que se detecten en el sistema, las cuales pueden ser un número completo (mayor o igual que 0) de cualquier magnitud como fuente y entrada de datos, pero, sin embargo, como salida solo mostraremos cuatro posibilidades que serán las de la ocupación ya descrita anteriormente (ALTA, MEDIA-ALTA, MEDIA-BAJA, BAJA).

Aunque, como podemos ver, la entrada de datos muestra un conjunto de datos de regresión complejo, gracias a la salida discreta, conseguimos que nuestro sistema difuso pueda realizar las acciones necesarias en función de las diversas categorías en que se encuentre.

Para ejecutar todo el sistema de regresión semiparamétrica de respuesta discreta se han evaluado una serie de librerías sobre la ciencia de datos en Python [35], encontrando que, principalmente, la librería Scikit-learn de Python es la más utilizada y, por lo tanto, se ha desarrollado el sistema siguiendo esta librería, que trabaja en términos de tabla. Por este motivo, se tratan todos los datos recibidos por los distintos dispositivos mediante un array para generar, de manera sencilla, tablas con las que trabajar. Para la librería, cada lista de datos de cada dispositivo será una columna y a cada columna se la denominará característica, y dependiendo del número de columnas tendremos n-características en el sistema.

	Ref. Disp. 1	Ref. Disp. 2	Ref. Disp. 3	Ref. Disp. 4	Ref. Disp. n
Nº personas tiempo X	12	1	23	4	5
Nº personas tiempo X+1	0	123	1	323	23

Tabla 3 Tabla de dispositivos

De la Tabla 3 se deduce que cada dispositivo es una característica para el sistema de aprendizaje automático.

Además de la matriz de n-características “X”, también trabajamos generalmente con una matriz de etiquetas o de objetivos, a la que por defecto se llama “Y” y será lo que estadísticamente se conoce como variable dependiente.

La matriz de objetivos suele ser unidimensional, con longitud n_muestras (en el sistema las mismas muestras que existen para la tabla “X” existirán para la tabla “Y”). La matriz de destino puede tener valores numéricos continuos, pero en nuestro sistema tendrá categorías/etiquetas discretas (es por esta razón, que la salida de la predicción será discreta) ya que es de mayor utilidad para ejecutar acciones dependiendo del resultado. Además, el sistema de regresión semiparamétrico de la librería puede manejar múltiples valores discretos en la forma de una matriz de objetivos bidimensional[n_muestras, n_objetivos], esta característica permitirá una potencialidad al sistema enorme, ya que abre la puerta a tener diversos dispositivos objetivos y poder intentar predecir qué ocurrirá para cada uno de ellos a la misma vez, por lo que se podría tener una auténtica zona “viva” que reacciona con diversos elementos a la vez dependiendo de las personas que se detecten en que cada lugar de esa zona.

En el sistema actual, se pone el foco en que esta matriz sea unidimensional, por lo que se deja a siguientes trabajos aumentar la potencialidad del sistema. La Tabla 4, muestra de manera simplificada como sería la matriz que se introduce en el sistema de aprendizaje automático, ya teniendo en cuenta el vector objetivo, en nuestro caso llamado dispositivo objetivo.

El uso básico y más general para realizar estimaciones de lo que va a suceder, o sea, para realizar pronósticos dentro del sistema desarrollado es el siguiente:

1. Elige una clase o modelo en un formato apropiado para ser importado en el algoritmo.
2. Elige los parámetros de la librería con los que el modelo puede ejecutarse de manera correcta, aunque esto puede ajustarse en posteriores ejecuciones hasta lograr el comportamiento deseado, ya que la librería contiene una preselección de parámetros.
3. Organiza los datos para crear una matriz de características y un vector objetivo de forma correcta para entrenar el algoritmo.
4. Entrena el algoritmo con los datos insertados llamando a la función *fit* de la librería.
5. Con el modelo ya entrenado, puedes empezar a aplicar nuevos valores para ver qué sucedería. Para ello, se puede aplicar la función *predict* tanto para realizar un aprendizaje supervisado como, no supervisado. O la función *transform* para aprendizaje no supervisado.

		Características		Vector Objetivo
		ref_dispositivo_1	ref_dispositivo_2	ref_dispositivo_objetivo
n° muestras		12	10	23
		22	20	42
		12	2	14

Tabla 4 Tabla para regresión semiparamétrica

Pero, aunque el sistema de regresión lineal semiparamétrico parcialmente lineal es un sistema con muchas bondades, puede que el modelo que se genere a partir de él, no sea todo lo correcto que debería ser. Si un modelo no ha sido creado de manera correcta, bien sea por problemas de configuración o bien sea por falta de datos para poder dar predicciones, entonces el modelo no se debe de tener en consideración, ya que puede devolver respuestas que no sean válidas y llevar al sistema y a los usuarios a errores que dificulten aún más todo el flujo de personas.

Por lo tanto, todo lo anterior se puede resumir en que existe un sistema de categorización que se encarga de fijar un valor determinado a las distintas ocupaciones que existen en los dispositivos objetivo, para ello se utilizan cuantiles y cuartiles. Los valores resultantes de la categorización se utilizan para alimentar, junto con los números de personas detectados por los dispositivos no objetivo, el modelo predictivo que está basado en una función de regresión logística semiparamétrica, y que se encarga de advertir al sistema de lo que va a ocurrir en el futuro.

3.1.3. Sistema de respuesta

El sistema de respuesta es la parte del sistema encargado de enviar la información calculada por el modelo de pronosis a los distintos dispositivos objetivo. Este sistema se compone de una base de datos que se sincroniza con todos los objetivos en tiempo real, obteniendo así de los dispositivos toda la información que necesiten para ejecutar sus diversas operaciones. Este sistema envía información desde dispositivos como tornos y barreras hasta aplicaciones móviles o web que puedan permitir a los usuarios tomar sus propias decisiones.

3.2. Sistema periférico

El sistema periférico se compone de todos aquellos dispositivos que van a detectar el número de personas existentes en sus proximidades. Estos dispositivos se encargan de hacer los cálculos y las ecuaciones necesarias para determinar qué usuarios hay a su alrededor y enviar esta información al sistema central para que los pueda tratar. Cada dispositivo de este sistema trabaja como un ente unitario y, por lo tanto, no tendrá conocimiento de la existencia de otros dispositivos, obteniendo así el desacoplamiento deseado.

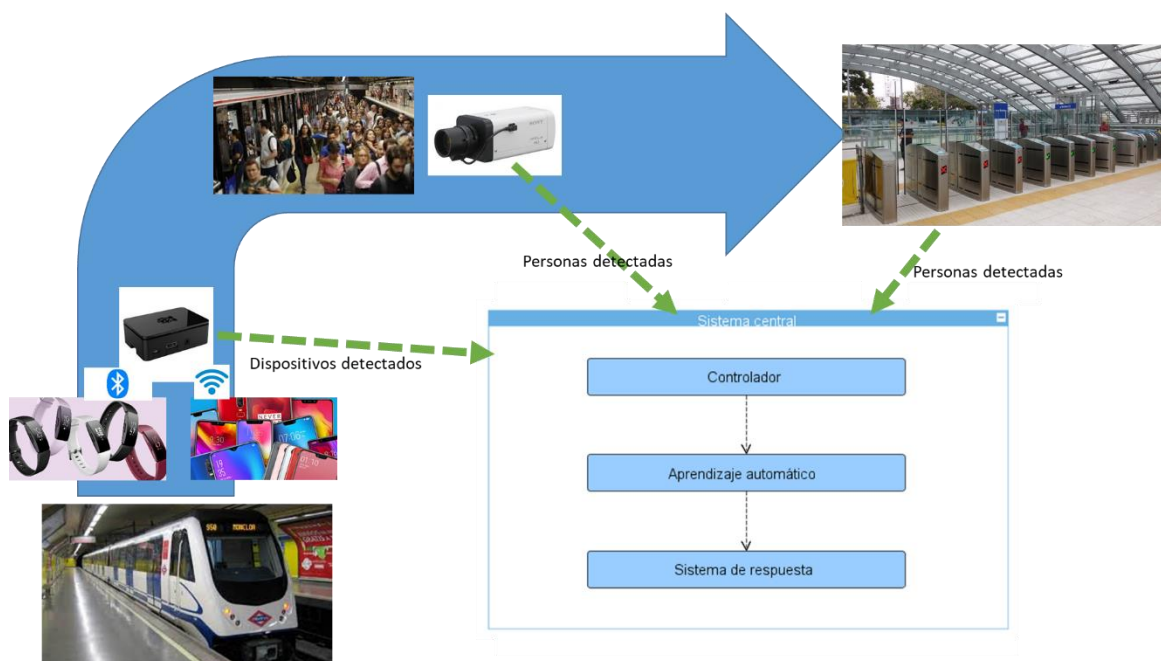


Figura 4 Esquema sistema periférico

Para realizar lo anterior, el sistema periférico cuenta con tres sistemas diferentes:

3.2.1. WiFi

De cara a ayudar a la toma de datos para conocer el número de personas detectadas, se captan las direcciones MAC (sniffing) de los dispositivos personales (móviles, tablets, ordenadores) de los pasajeros durante el trayecto que realicen [36], para así establecer los lugares por los que están pasando. Para ello, se pueden utilizar estos dos procedimientos:

- Configurar el módulo WiFi del dispositivo en modo punto de acceso: se reciben todas las solicitudes de búsqueda de puntos de acceso de los dispositivos personales de los pasajeros (esta solicitud incluye la MAC del solicitante)
- Configurar el módulo WiFi del dispositivo en modo monitor: de esta manera se escuchan todas las transmisiones WiFi de su entorno, incluyendo las solicitudes de búsqueda de puntos de acceso, que es la manera más sencilla de obtener las direcciones MACs de los dispositivos existentes en el entorno

Dependiendo del lugar donde esté localizado el dispositivo se aconsejará implementarlo de una manera o de otra.

En el caso de que el dispositivo estuviese embarcado en un transporte se aconseja implementarlo de la primera manera, dado que el tráfico que es necesario filtrar según el segundo método es enorme y es posible que se deriven problemas de rendimiento y saturación en el equipo embarcado.

Por otro lado, en el caso que se está asumiendo en este trabajo, se activa el dispositivo en modo Monitor, ya que el dispositivo no se va a mover porque estará en un lugar fijo y mediante los algoritmos de categorización, utilizando cuantiles, se eliminan las detecciones de dispositivos que no pertenezcan a usuarios, reduciendo así el “ruido” del entorno.

Para realizar todo lo anterior, se dispondrá en el proyecto de un terminal inalámbrico (Raspberry Pi) que está provisto internamente de una tarjeta de red inalámbrica con doble banda de frecuencia de 2.4GHz y 5.8GHz y se accede a una red a través de WiFi. El método comprende los siguientes pasos:

1. El terminal inalámbrico está preparado para funcionar en modo Monitor.
2. Los paquetes de datos de comunicación inalámbrica WiFi son capturados periódicamente en cada canal de frecuencia de 2.4GHz y 5.8GHz.
3. Se analiza una cabecera de trama de protocolo 802.11 de cada paquete de datos, y se extrae una dirección MAC de un campo BSSID, un campo de dirección de origen y un campo de dirección de destino de la cabecera de trama.
4. Se realiza un conteo de todas las MAC detectadas y el número de personas detectadas es enviado al sistema central.

De este modo, los terminales inalámbricos circundantes pueden ser descubiertos en tiempo real a través de este método, por lo que los titulares de los terminales inalámbricos pueden ser buscados sin omisiones. Esta búsqueda es muy útil ya que se considera de forma fehaciente que, en el mundo actual, sobre todo en lo que se denomina primer mundo, todas las personas suelen contar con un dispositivo móvil, y que según la cuota de mercado actual *Android* es usado por el 75% de los usuarios, *iPhone* por el 23% y el resto utilizan sistemas operativos varios. En los dispositivos *Android*, la WiFi es difícilmente desactivable ya que, aunque se deshabilite de forma activa, siempre estará utilizándose en segundo plano por el sistema de localización, que puede ser deshabilitado de forma activa por el usuario, pero utilizado en segundo plano por las aplicaciones que el usuario tiene instaladas. En *iPhone* el sistema es más simple, y si el usuario desactiva la WiFi en primer plano, ya no se utilizará en segundo plano. Estos datos hacen que consideremos que una gran mayoría de los dispositivos móviles existentes en el planeta, están enviando señales WiFi de manera constante, por lo que gracias a ellos podemos saber dónde se encuentran los usuarios.

Durante la captación de datos de los usuarios, los problemas que se pueden detectar son:

- Detección de MACs - WiFi de forma simultánea por más de un dispositivo del sistema por la proximidad entre ambos.
- Detección discontinúa o intermitente de MACs.
- Detección de MACs espurias en estaciones, es decir, detecciones de dispositivos que no estén asociadas a usuarios, sino que sean dispositivos fijos en la estación.
- Detección de MACs en personas que lleven más de un dispositivo móvil.
- Como el sistema no almacena los datos de las MACs de los usuarios, sino que solo los utiliza para saber el número de personas cercanas, podemos obviar el potencial problema legal de registro de MACs [37].

3.2.2. Bluetooth

Del mismo modo que la detección de dispositivos WiFi es utilizada para reconocer las personas existentes en el entorno, son utilizados los dispositivos Bluetooth. Estos dispositivos de detección son capaces de detectar todos los dispositivos Bluetooth existentes en el entorno y contabilizarán su número para ser enviado al sistema central, que será el que los procese.

La capacidad de configurar un concentrador y detectar dispositivos Bluetooth resuelve el problema de identificar cuándo las personas ingresan, abandonan y, con un poco de lógica, dónde se encuentran las personas en un espacio, simplemente mediante los dispositivos Bluetooth que las estas portan. Con el dispositivo concentrador de Bluetooth podemos detectar muchas personas que usan Fitbits, auriculares inalámbricos y dejan el Bluetooth en el teléfono todo el tiempo. Esa es una excelente manera de saber dónde están las personas, proporcionando un puente entre el mundo físico y lo digital.

3.2.3. Cámara

Reconocer a personas en imágenes es una práctica que aporta un valor con una alta probabilidad de acierto para conocer el número de personas que están pasando por un lugar concreto en un momento determinado. Una vez la cámara reconoce a las personas que están apareciendo en su pantalla, esta información es enviada al sistema central para que sea unificada junto con todas las demás personas localizadas por los demás dispositivos.

Para ello se utilizan redes neuronales convolucionales (también denominadas CNN o ConvNet) que son una clase de redes neuronales profundas que se han adoptado ampliamente en aplicaciones de visión artificial e imágenes visuales.

Las redes neuronales convolucionales constan de múltiples capas diseñadas para requerir un preprocesamiento relativamente pequeño en comparación con otros algoritmos de clasificación de imágenes.

Un famoso caso de aplicación CNN fue detallado en [38] por un equipo de investigación de Stanford en el que demostraron la clasificación de las lesiones cutáneas usando una sola CNN. La red neuronal fue entrenada a partir de imágenes utilizando solo píxeles y etiquetas de enfermedades como entradas.

Los algoritmos aprenden usando filtros y aplicándolos a las imágenes. El algoritmo toma un cuadrado pequeño o ventana, y comienza a aplicarlo sobre la imagen. Cada filtro permite que la CNN identifique ciertos patrones en la imagen. La CNN busca partes de la imagen donde un filtro coincida con el contenido de la imagen. La Figura 9 muestra el esquema de la arquitectura de capas de esta red.

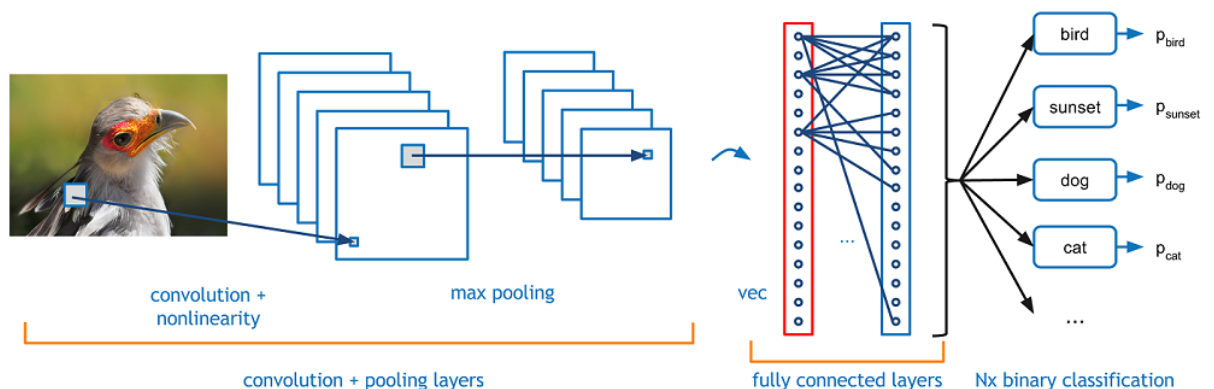


Figura 5 Arquitectura de capas CNN para clasificación de imágenes [39]

Estas imágenes vienen de entornos en tiempo real, es decir, videos que estén siendo capturados en el momento por la cámara correspondiente. El reconocimiento de patrones en imágenes necesita ser entrenado previamente y preconfigurado, este entrenamiento y esta preconfiguración permiten que el sistema pueda ser implantando en multitud de cámaras de

video diferentes, desde cámaras convencionales, pasando por cámaras de seguridad, llegando incluso a cámaras en dispositivos móviles.

Para realizar todo lo anterior se utiliza el software open-source Tensorflow para *machine learning*. TensorFlow es una biblioteca de código abierto para aprendizaje automático a través de un rango de tareas, y desarrollado por Google para satisfacer sus necesidades de sistemas capaces de construir y entrenar redes neuronales para detectar y descifrar patrones y correlaciones, análogos al aprendizaje y razonamiento usados por los humanos.

3.3. Sistema de acción

El sistema de acción se compone de todos aquellos dispositivos que dan respuesta a los usuarios existentes en la estación para facilitar su paso a través de ella. Estos dispositivos se encargan de hacer las operaciones necesarias con la información recibida desde el sistema central. Esta información recibida contiene el pronóstico sobre qué va a suceder en un tiempo futuro, enviando para ello el valor de la categorización acordada, y con ese valor el sistema de acción determina lo que debe o no debe hacer según su programación.

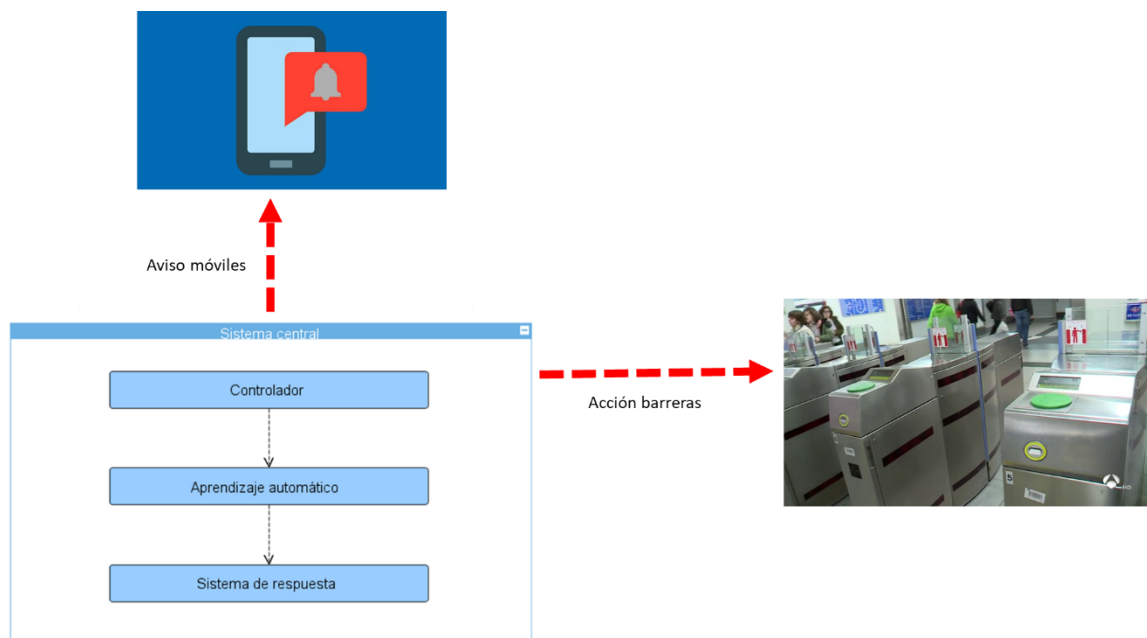


Figura 6 Esquema sistema de acción

En el caso del sistema de acción, todos los dispositivos pueden trabajar como entidades unitarias o como un sistema coaligado para realizar una función, ya que el sistema central devuelve los siguientes valores:

- Número de personas detectadas por cada dispositivo en el momento actual.
- Pronóstico de la ocupación de cada dispositivo en un momento futuro.

Para realizar lo anterior, el sistema de acción contará con dos sistemas diferenciados:

3.3.1. Barreras

Mejorar el sistema de barreras (pasos) es el objetivo del sistema, ya que los pasos en muchas estaciones, suponen todo un lastre para el tránsito de los viajeros, que, además, se acentúa sobre todo en momentos de estrés como horas punta, haciendo que se formen colas innecesarias en muchas de las estaciones.

Si clasificamos los pasos por el tipo de puertas que tienen, habría los siguientes grupos:

- Torniquetes (ejemplo metro Madrid)
- Puertas deslizantes (ejemplo RENFE cercanías)
- Puertas batientes (ejemplo Riyadh)

También los podemos clasificar por el sentido en que permiten cruzar a los pasajeros:

- Bidireccional (permite a los pasajeros validar sus títulos y cruzar en ambos sentidos)
- Unidireccional (el equipo solo trabaja en un sentido y permite al pasajero entrar en la estación o salir)

Mediante el sistema pronóstico de flujo de pasajeros se puede cambiar la polaridad de los pasos, para que cambien el modo entrada/salida y faciliten el flujo en el sentido más congestionado o abrirse directamente para que las acumulaciones desapareciesen de manera automatizada e incluso, adelantándose a que estas ocurriesen.

Es evidente, que muchos sistemas de salida conllevan validaciones de tickets a la salida, para confirmar que el usuario ha validado su ticket o saber cuánto habría que cobrarle en cada momento. Por lo que este sistema, podría generar cierta pérdida monetaria a la empresa que lo implantase, pero sin embargo por otro lado, la felicidad de los clientes al facilitarles las salidas y las entradas a las estaciones haría que el aumento de viajeros en ella pudiese ejercer una atracción de viajeros de los que antes no se disponía. Aumentando así, el número de personas que viajan en el transporte público y, por lo tanto, reduciendo la contaminación al cambiar la alternativa de transporte privada por la pública y suponiendo un posible incremento de los ingresos de la compañía de transportes al disponer de más usuarios.

Por otra parte, este sistema no tiene por qué tener pérdidas para el operador de transportes debido a la nueva aproximación de los transportes a los entornos de Movilidad como Servicio.

3.3.2. Móvil

Además de la parte de los pasos, que estaría en línea con la “estación viva” y de hacer que el hardware modifique su comportamiento en función de decisiones difusas mediante software,

también existe la posibilidad de informar al viajero de forma instantánea para que conozca el estado de la red de transportes y pueda tomar sus decisiones mientras camina por la propia estación.

Esta funcionalidad es informativa y puede ayudar a los pasajeros a cosas tan diferentes como:

- Tomar el ascensor en vez de la escalera mecánica o tomar la escalera estática.
- Decidir por cual salida salir, ya que hay algunas que pueden tener menos gente y pueden ahorrar tiempos de paso al usuario.
- Elegir el vagón con menos ocupación, y de esta manera evitar tener unos llenos y otros vacíos. Además de minimizar los movimientos y cruces de personas en los andenes.

Para ello, dentro de este estudio se ha realizado una aplicación móvil híbrida en tecnología Angular, que muestra en tiempo real la ocupación actual de las zonas y también el pronóstico de la ocupación futura que va a existir en el dispositivo objetivo.

Esta aplicación se ha realizado con fines meramente informativos y poder tener un seguimiento en tiempo real de los acontecimientos detectados por el sistema.

4. Resultados obtenidos

Como resultado del consiguiente trabajo, se han obtenido unos resultados que serán mostrados a continuación y que, además, demuestran la capacidad y posibilidad de que el sistema desarrollado pueda ser llevado a cabo en entornos reales con unos flujos mayores de personas.

Como consideraciones previas a los resultados obtenidos, en comparación con lo que se buscaba desde el inicio, se indican los siguientes puntos:

- Prueba en un entorno fuera del ámbito del transporte: No se ha podido llegar a un acuerdo por el cual probar el sistema en ninguno de los transportes públicos existentes en la comunidad de Madrid. Por ello, se ha probado en un entorno abierto de flujo de personas como lo es la entrada al edificio en el que trabajo actualmente.
- Imposibilidad de obtener imágenes reales de cámaras de video: Debido a la ley de protección de datos, a que el edificio en el que trabajo sigue protocolos OTAN de seguridad (que impide la grabación de personas en el edificio) y al recelo de las empresas que se dedican a la video vigilancia. En las pruebas no existe ningún dispositivo real que sea fuente de datos que registre información a través de cámaras de video, aunque el algoritmo haya sido probado a través de videos públicos en internet de personas paseando en las calles. Sin embargo, el algoritmo de video ha sido probado y su validez contrastada en laboratorio, pero no se ha tenido en cuenta para la generación del algoritmo de pronóstico y, por lo tanto, no se ha tenido en cuenta como resultado del sistema difuso, ya que sus datos no estarían nunca en consonancia con el entorno real, generando así errores e incongruencias.

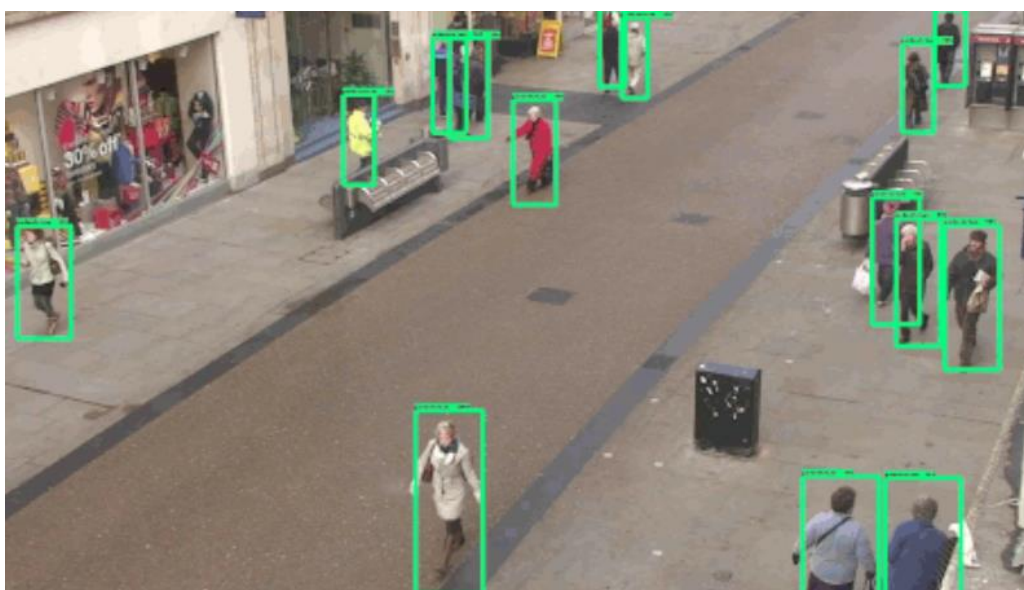


Figura 7 Detección de personas en tiempo real

La figura 7 nos muestra un ejemplo funcional del algoritmo de detección de personas en tiempo real sobre un video público existente en la web. El sistema desarrollado realiza el sumatorio de las personas detectadas cada X segundos y envía esa información al sistema central para que tenga en cuenta esa información a la hora de realizar el pronóstico.

- Sin conexión con barreras: Debido a la imposibilidad de trabajar de manera oficial con ningún laboratorio de una empresa de transporte, no se ha podido utilizar ninguna barrera real. Aunque este escollo, ha sido salvado gracias a una aplicación móvil que muestra en tiempo real los cambios que se daría en el sistema. Por lo tanto, lo único que sería necesario hacer en el futuro es programar el paso para que actúe de cierta manera en función de los resultados que se obtienen del sistema.

A pesar de los inconvenientes anteriores los resultados obtenidos se han obtenido de diversas fuentes de información como son:

- Raspberry Pi 3b+ y Raspberry 4: gracias a estos dispositivos de precio asequible se han obtenido los datos de los dispositivos WiFi y Bluetooth que existen en un radio cercano al dispositivo.

4.1. Prueba en entorno real

Para realizar las pruebas en un entorno real, se ha utilizado el edificio en el que actualmente trabajo durante una sola mañana, ya que no se ha podido disponer de más disponibilidad para ello. En la Figura 7 se observa un plano simple de la planta baja del edificio donde se ha realizado la prueba. En ella, se puede observar que se situó una Raspberry con los sensores WiFi y Bluetooth cerca de los tornos de entrada/salida para poder tomar muestras lo más similares a lo que está sucediendo en los tornos de entrada/salida, es la zona que denominaremos “previa”. Para la zona “objetivo” se utilizó otra Raspberry en la cual también se extraerán los datos de dispositivos WiFi y Bluetooth, haciendo que los datos de la WiFi sean los datos objetivos. Debido a que el sensor de captación de WiFi está en la Raspberry y que está dispone también de un sensor Bluetooth, se aprovechó para utilizar también ese sensor como captación de información en el mismo lugar, este caso es un caso especial donde la distancia entre los dos sensores es, evidentemente, cero.

El objetivo de esta prueba, es determinar el flujo de personas que accede al edificio por los tornos (paso obligatorio) y que luego pasa a través del pasillo que lleva a los siguientes edificios. De esta manera, se puede comprobar el flujo de personas que acceden al edificio 2 por las mañanas y hacer en él las actuaciones que fuesen necesarias (desde dejar la puerta del pasillo abierta, a utilizar esta información para aumentar o disminuir el flujo de aire de la calefacción del edificio).

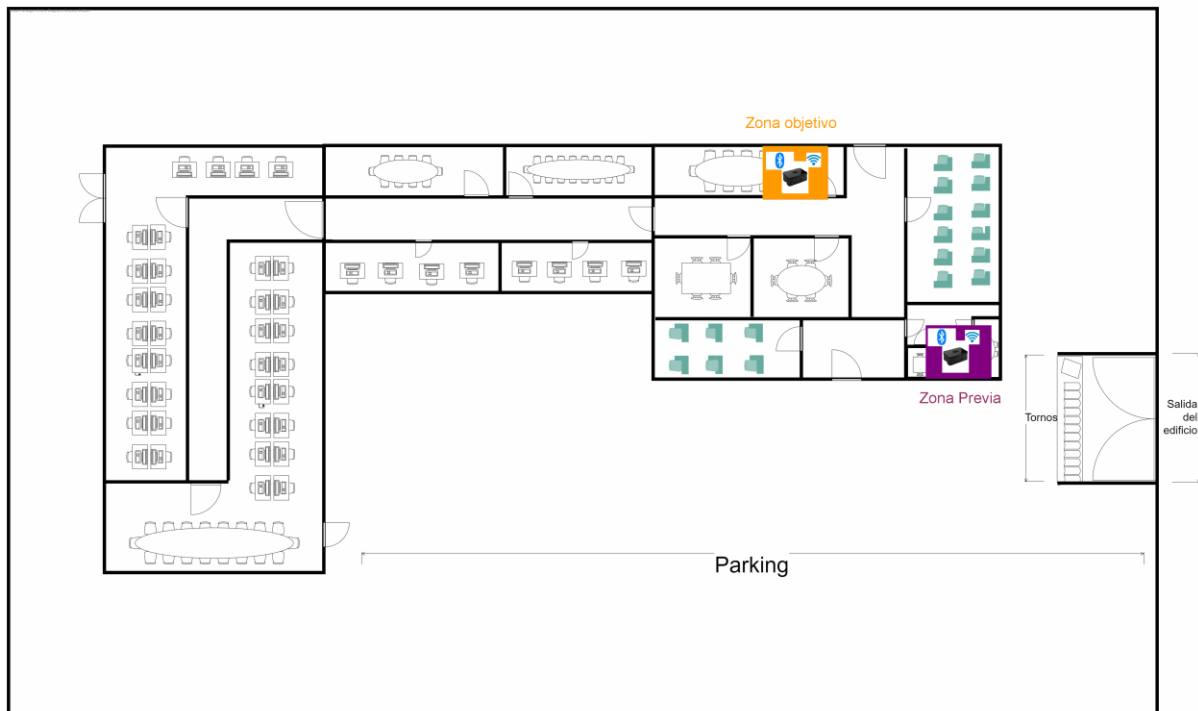
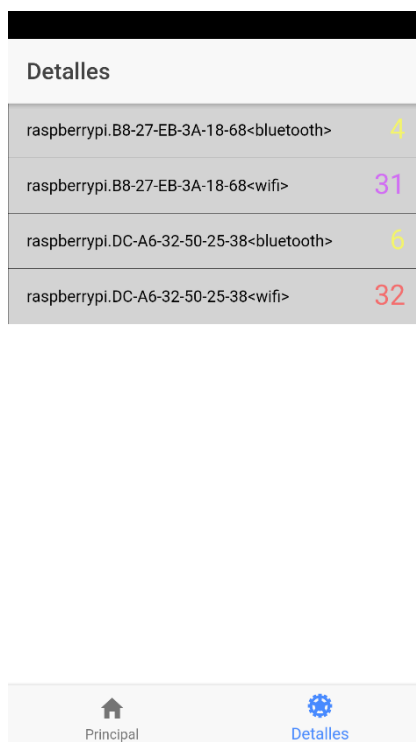


Figura 8 Plano de edificio

Con los datos obtenidos de los dos dispositivos anteriores se ha podido conocer el flujo de personas que pasan cerca de las zonas de las que se quiere obtener la información para realizar el cálculo del modelo que servirá para activar y poner en funcionamiento el sistema difuso. Que en este caso se basa en avisar a los usuarios, mediante una aplicación móvil, del “atasco” que se pueden encontrar más adelante en su ruta.

En la aplicación móvil se puede encontrar en tiempo real, desde el número de personas detectadas por cada dispositivo (pantalla de carácter informativo para controlar el sistema) Figura 9, hasta el resultado actual de la ocupación y su pronóstico para cuando el usuario llegue (con un código de colores de menor a mayor: verde-amarillo-rojo-morado) al siguiente paso en el flujo de su viaje: la parte izquierda es la ocupación actual en el dispositivo objetivo, la parte derecha es la predicción que se hace de esa ocupación cuando el usuario llegue a la altura del dispositivo objetivo.



Detalles	
raspberrypi.B8-27-EB-3A-18-68<bluetooth>	4
raspberrypi.B8-27-EB-3A-18-68<wifi>	31
raspberrypi.DC-A6-32-50-25-38<bluetooth>	6
raspberrypi.DC-A6-32-50-25-38<wifi>	32

Principal Detalles

Figura 9 Número de personas detectadas por dispositivo

La Figura 10 nos muestra como la ocupación se estima que empeore a que se pasa del Rojo (ocupación media-alta) al morado (ocupación alta).

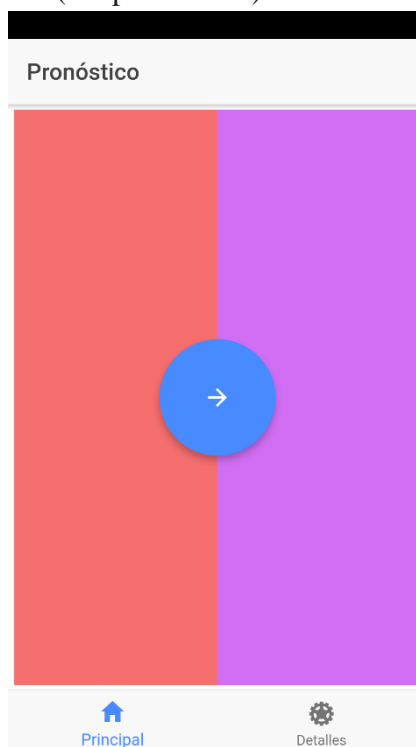


Figura 10 Pronóstico empeoramiento

Sin embargo, la Figura 11 nos muestra como el sistema tiende a mejorar ya que se pasa del morado (ocupación alta) al verde (ocupación baja)

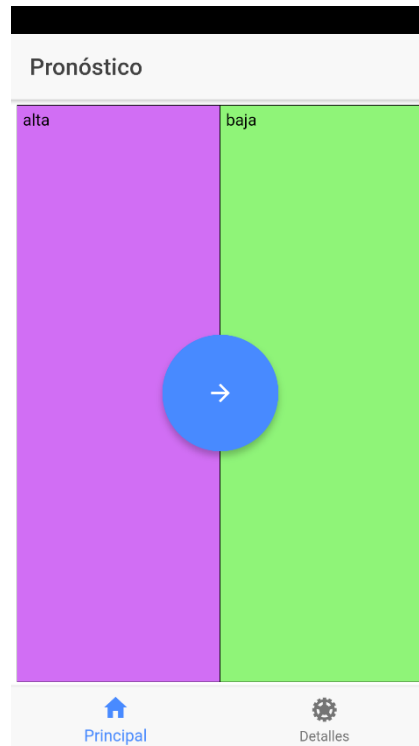


Figura 11 Pronóstico mejora

Para llevar un control fácil y en tiempo real del sistema, a parte de la aplicación móvil se ha utilizado una plataforma de *Business Intelligence* (BI) que centralizará todos los datos interesantes que se produzcan en el sistema central y en los dispositivos periféricos. Mostrando de esta manera en tiempo real, lo que está sucediendo en la aplicación y dando una idea, a los usuarios con permisos, de por qué está pasando.

La plataforma elegida para mostrar la información ha sido *Elastic Search* que es una plataforma Open Source (uno de los principios de este TFM) y tiene una gran comunidad de desarrollo. Gracias a esta plataforma se obtiene el cuadro de mando detallado en las Figuras 12 y 13, que permite conocer el estado de la aplicación en tiempo real:

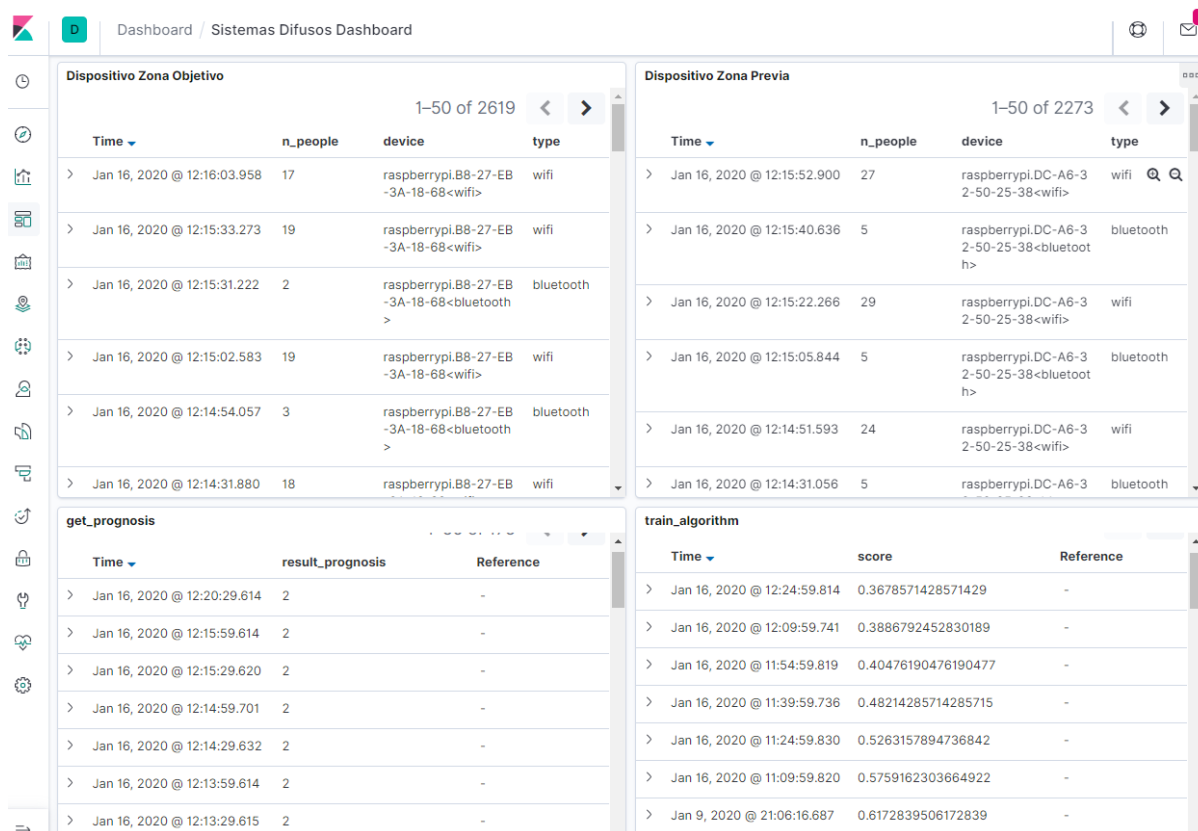


Figura 12 Cuadro de mando A

En la Figura 12 se puede observar la primera parte del cuadro de mando que muestra las personas que están siendo detectadas por cada sensor del dispositivo, además del resultado de hacer el pronóstico sobre lo que va a suceder y la calidad del modelo de predicción.

Los dos primeros cuadros de la Figura 12 muestran los dos dispositivos de los que dispone nuestro sistema actualmente, en ellos se muestra el número de personas detectadas y el dispositivo que la detecta (Bluetooth o WiFi).

El cuadro con el título *get_prognosis* de la Figura 12 enseña el valor del pronóstico en ese mismo momento, que coincide con el valor que se verá en la aplicación móvil en cada momento.

El cuadro con el título *train_algorithm* de la Figura 12 muestra la calidad general del modelo entrenado cada vez que este es entrenado. Esta calidad mostrada es fundamental para que el sistema funcione correctamente, ya que un modelo de baja calidad no produciría un pronóstico correcto y podría producir más problemas que beneficios. Si la calidad es inferior a un mínimo determinado entonces no se producirían pronósticos por parte del sistema, evitando así enviar información poco fiable al usuario.

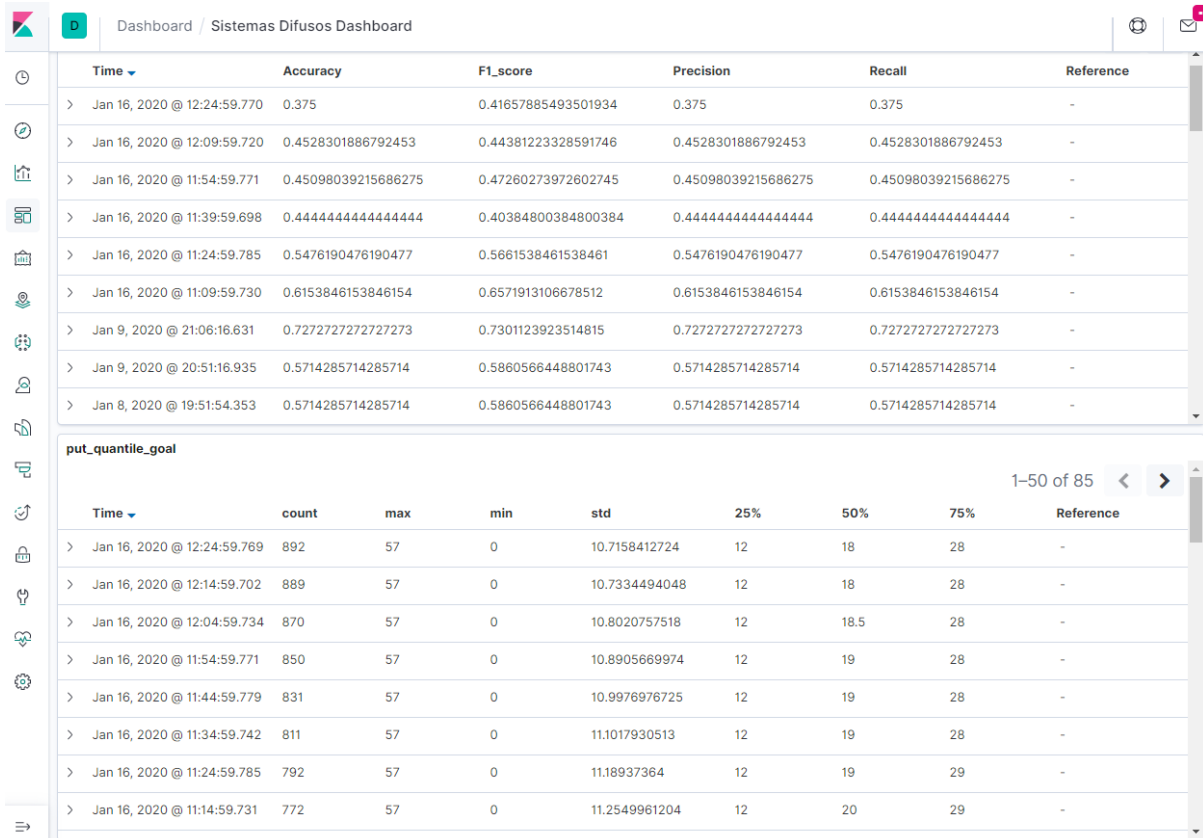


Figura 13 Cuadro de mando B

Por otro lado, en la Figura 13 se muestra la segunda parte del cuadro de mando, destinada principalmente a conocer las variables internas del sistema y los resultados tanto de la validación del modelo como los valores de los cuartiles utilizados.

El cuadro con el título *validation* de la Figura 13 enseña el valor de las distintas validaciones del modelo de regresión generado para predecir las futuras ocupaciones. En cada fila se puede observar que cada columna representa un valor de cada tipo de validación a la que se ha sometido el modelo. Estos valores serán importantes para conocer la calidad y la fiabilidad del modelo.

El cuadro con el título *put_quantile_goal* de la Figura 13 muestra la evolución de los valores de los cuartiles asociados al sistema central. Con un rápido vistazo de este cuadro se puede conocer la evolución de los cuartiles en tiempo real, y, además, se puede observar cómo los valores de los cuartiles se van ajustando a medida que el sistema va recibiendo más y más datos según pasa el tiempo. Este cuadro contiene los valores 25%, 50% y 75% correspondientes a los cuartiles 0.25, 0.5, 0.75.

Como se puede observar en la Figura 13, se han obtenido durante la mañana 892 registros de datos a través de los sensores, lo que, debido a la gran variabilidad en el flujo de entrada de las personas, hace que el algoritmo de inteligencia artificial no de unos resultados los suficientemente satisfactorios como para ser fiables. Esto se ve en las Figuras 12 y 13, que

muestran el resultado del entrenamiento del algoritmo, y cuya media de acierto nos indica que, en el mejor de los casos, fue de un 72% lo que no nos garantiza un resultado suficientemente confiable. Para empezar a confiar en el algoritmo sería interesante disponer una media superior al 85% de aciertos, es por ello, que el algoritmo necesitará más información para poder realizar un buen entrenamiento que nos devuelva un modelo válido. Es importante tener en cuenta, que tampoco interesa que la media sea muy cercana a 1, ya que sería un indicador de que el modelo puede estar sobreadaptado, y por lo tanto sería un mal modelo.

4.2. Validaciones

Por otro lado, como se ha comentado anteriormente, las validaciones son una parte fundamental del sistema, porque ayudan a obtener un modelo de calidad y fiable, que evite los errores y que realmente ayude a los usuarios a que su tránsito sea más fácil.

Por este motivo, se han llevado a cabo cierto tipo de validaciones que aseguran al sistema que el modelo de regresión generado es lo suficientemente bueno y preciso para poder obtener una confianza sobre él. Ya que, en caso de que el sistema haga que el entorno funcione peor o que incluso ponga en peligro el uso de ciertos sistemas o ciertos usuarios, se generaría una desconfianza sobre él que puede generar un rechazo a este tipo de sistemas. Es por ello que la validación es parte crucial y que, sin unos buenos resultados en esta, el sistema es solamente pasivo ante lo que ocurra, es decir, espera obteniendo datos e intentando generar con ellos nuevos modelos, sin emitir respuestas de pronósticos, hasta que el modelo generado sea válido y confiable, y entonces a partir de ahí, empezar a enviar información a los sistemas objetivo.

La validación del modelo en profundidad garantiza la integridad y la calidad de los datos. El sistema, además, muestra información al administrador mediante una serie de logs centralizados en Elastic Search que demuestran la calidad y los resultados de las predicciones que se van a realizar con este modelo. Con estos datos se puede rebalancear el algoritmo y así descubrir y optimizar mejores combinaciones para conseguir un resultado más realista o que más se aproxime a la realidad o a las necesidades del sistema en cada caso.

Para comprobar el sistema de regresión, se ha aplicado el método de validación cruzada [40] que enseña que entrenar una función de predicción y probarla con los mismos datos es un error metodológico: un modelo que simplemente repitiera las etiquetas de las muestras que acaba de ver tendría una puntuación perfecta pero no podría predecir nada útil en datos aún no vistos. A esta situación se la denomina sobreadaptación o sobreajuste. Un modelo va a estar sobreajustado cuando vemos que se desempeña bien con los datos de entrenamiento, pero su precisión es notablemente más baja con los datos de evaluación; esto se debe a que el modelo ha memorizado los datos que ha visto y no pudo generalizar las reglas para predecir los datos que no ha visto. El sobreajuste está relacionado con la complejidad del modelo, mientras más complejidad le agreguemos, mayor va a ser la tendencia a sobreajustarse a los datos, ya que va a contar con mayor flexibilidad para realizar las predicciones y puede ser que los patrones que

encuentre estén relacionados con el ruido (pequeños errores aleatorios) en los datos y no con la verdadera señal o relación subyacente.

Para evitarlo, es práctica común cuando se realiza un experimento de aprendizaje automático (supervisado) mantener parte de los datos disponibles como un conjunto de pruebas x_{test} , y_{test} . Es por ello que se realiza la división del conjunto de datos entre un conjunto de entrenamiento y un conjunto de prueba. Se utilizará alrededor de un 80% de datos para entrenar el modelo y un 20% como datos de prueba. En la Figura 14 se muestra un esquema típico de una validación cruzada.

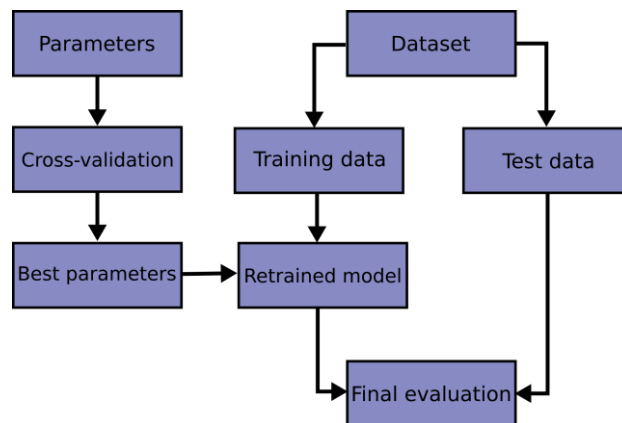


Figura 14 Validación cruzada [40]

Una vez se ha realizado esta validación cruzada y esta devuelve una precisión media por encima del mínimo esperado, se procede a predecir los datos con el modelo actual y las matrices obtenidas de la separación primera, y comprobamos los datos que devuelve. Gracias a una serie de métricas de rendimiento que se aplican se determinará la calidad y la precisión del modelo, y se obtendrán datos en mayor detalle que nos permiten saber cuántos falsos positivos y falsos negativos obtenemos con el modelo [41].

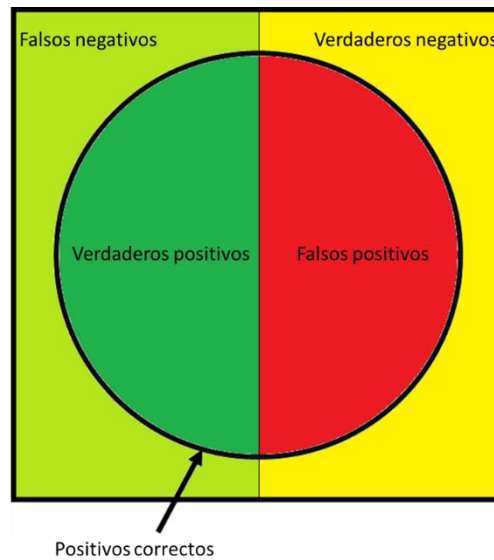


Figura 15 Zonas de validación

En el esquema anterior se pueden observar los valores:

- *Verdadero positivo*: Cuando el algoritmo predice algo de manera positiva y el valor es positivo.
- *Verdadero negativo*: Cuando el algoritmo predice algo negativo y en realidad es negativo.
- *Falso positivo*: Cuando el algoritmo predice de manera positiva y el resultado real es negativo.
- *Falso negativo*: cuando el algoritmo predice algo de manera negativa cuando sin embargo el resultado es positivo.

Las anteriores posibilidades se pueden observar en la siguiente tabla:

Matriz de confusión		Predicción	
		Negativo	Positivo
Actual	Negativo	Verdadero negativo	Falso positivo
	Positivo	Falso negativo	Verdadero positivo

Tabla 5 Matriz de confusión

Las funciones aplicadas serán las siguientes:

- *Matriz de confusión*: Esta función nos muestra la matriz de confusión, que indica el número de falsos positivos y negativos, junto con los verdaderos positivos y negativos, que ha predicho el modelo con los datos que se le han insertado. Esta matriz está representada en la imagen anterior.
- *Fiabilidad*: La métrica más común para la clasificación es la fiabilidad, que es la fracción de las muestras predicha correctamente. Se puede observar fácilmente en la siguiente imagen:

$$\begin{aligned} & \text{Fiabilidad (Fracción predicha correctamente)} & (8) \\ & \frac{VP + VN}{VP + VP + FP + FN} \end{aligned}$$

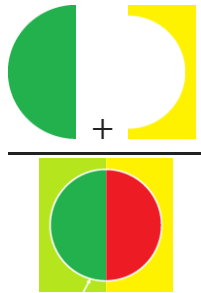


Figura 16 Fiabilidad

Dependiendo del sistema en que se quiera aplicar, esta métrica puede no ser la más fiable para determinar la calidad del modelo, ya que en el caso de que se quieran encontrar eventos que suceden en muy pocas ocasiones, aunque se tenga una gran fiabilidad, puede ocurrir que se falle justamente en los eventos que se pretenden encontrar, haciendo que el modelo no sea correcto para el caso de uso.

- *Precisión*: La precisión es la fracción de los eventos positivos previstos que son realmente positivos, se muestra en la siguiente imagen:

$$\begin{aligned} & \text{Precisión (fracción de los predichos positivos que realmente)} & (9) \\ & \text{son positivos} \\ & = \\ & \frac{VP}{VP + FP} \end{aligned}$$



Figura 17 Precisión

- *Sensibilidad (recall)*: Es la fracción de eventos positivos que se predijeron correctamente. En la imagen se puede observar:

$$\begin{aligned} \text{Sensibilidad}(\text{recall}) & \text{ Fracción de positivos predichos correctamente} & (10) \\ & = \frac{VP}{VP + FP} \end{aligned}$$

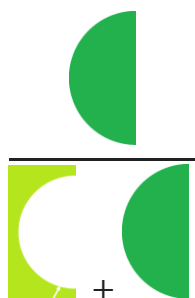


Figura 18 Sensibilidad

Se puede aumentar la sensibilidad del sistema disminuyendo el umbral para detectar positivos o aumentando el valor de positivos en el sistema, aunque esto aumentará también el número de falsos positivos.

- *F1*: La puntuación *F1* es la media armónica de la sensibilidad (*recall*) y la precisión, con una puntuación más alta como mejor modelo. Para obtener la nota, utilizaremos la siguiente ecuación:

$$F1 = \frac{2}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}} = \frac{2 * (\text{precision} * \text{recall})}{\text{precision} + \text{recall}} \quad (11)$$

- *Reporte de clasificación*: Por último, utilizaremos el método de reporte de la clasificación para obtener en los *logs* la clasificación en términos generales del modelo, estos datos nos permiten ver las métricas con un solo vistazo, y así poder rebalancear el algoritmo o al menos, saber qué está pasando e intentar intuir por qué sucede. El resultado sería como se muestra en la Figura 19.

Clasification report:				
	precision	recall	f1-score	support
1	1.00	0.76	0.86	71
2	1.00	0.84	0.91	43
3	1.00	0.74	0.85	89
4	0.98	0.95	0.96	288
5	0.87	1.00	0.93	367
avg / total	0.94	0.93	0.93	858

Figura 19 Reporte de clasificación

4.3. Simulador

Aunque el sistema se ha demostrado funcional en las pruebas que se han realizado en el mundo real, mostrando todos los componentes integrados y captando datos en tiempo real. La realidad es que la cantidad de datos y la calidad del algoritmo han sido demasiado bajas como para poder ser consideradas unas pruebas de rendimiento válidas. Es por ello, que una vez pasadas las pruebas de integración de todos los componentes del sistema, se va a proceder a inyectar cantidades mayores de datos en el sistema para realizar unas pruebas de rendimiento de este.

Debido a que el sistema va aprendiendo en función de los datos que recibe, es necesario obtener una cantidad ingente de datos para que el sistema se comporte de manera estable y eficiente. Con una gran cantidad de datos, el sistema podrá hacer predicciones cada vez mejores.

Es por lo anterior que se ha construido un simulador de estaciones con flujos de pasajeros, gracias al cual se consigue obtener un conjunto de datos de una amplitud considerable. Además, gracias al simulador se puede ver como el sistema está desarrollado y es totalmente funcional para el cálculo de predicciones de varios objetivos, consiguiendo de esta forma cubrir estaciones más complejas y que dispongan de varias salidas.

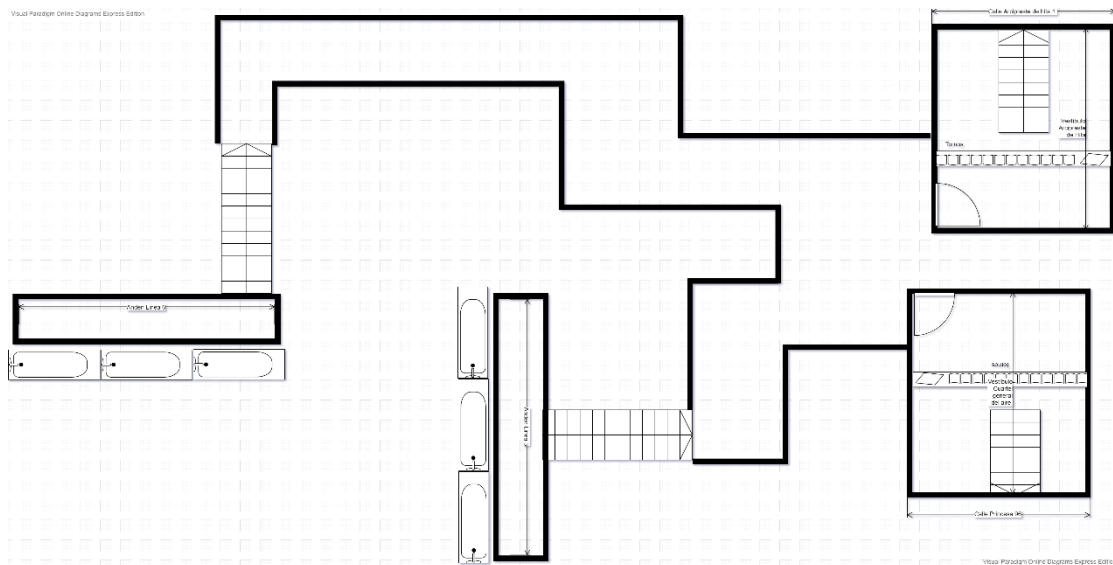


Figura 20 Plano cenital estación

Para la realización del simulador, se ha generado un mapa simple (Figura 20) con vista cenital de una estación de metro basado en la estación de Moncloa, en la cual existen dos líneas, Línea 3 y Línea 6, con sus correspondientes andenes y escaleras mecánicas y que dispone también de dos salidas, una a la Calle Arcipreste de Hita y otra a la calle Princesa. En el mapa se pueden encontrar también la posición de los tornos en los dos vestuarios.

Sobre el mapa de la Figura 20 se han posicionado los diversos sensores que podrían existir en esta estación para captar el flujo de pasajeros. En la Figura 21 existe una proposición de colocación de los diversos sensores.

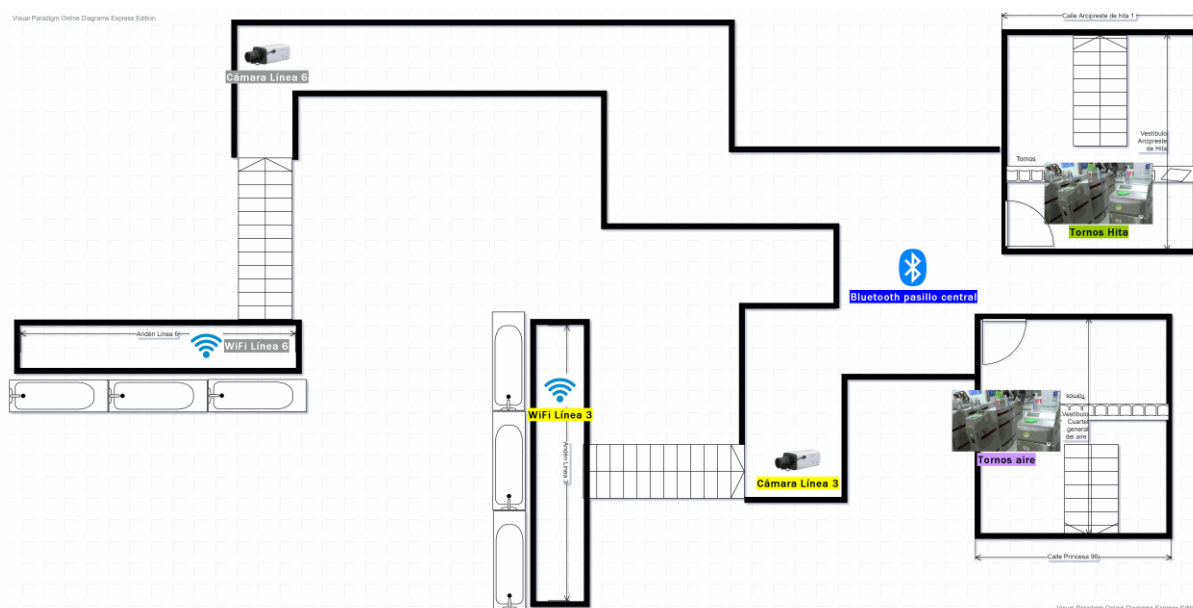


Figura 21 Sensores en estación

En la Figura 21 se pueden encontrar los distintos sensores que podríamos tener en una estación de cualquier lugar, en ella se pueden observar los siguientes sensores:

- WiFi: estarán posicionados en los andenes y detectarán las personas que bajan del transporte. Existirán 2 sensores, uno por cada andén.
- Cámaras: estarán posicionadas en la subida de las escaleras mecánicas, estas cámaras detectarán el número de personas que pasan por las escaleras. Existirán dos cámaras, una por cada andén.
- Bluetooth: estará situado en el área común entre los dos pasillos para acceder al andén.
- Tornos: Sabrán exactamente cuántas personas están saliendo de la estación por cada lugar. Existirán 2 sistemas de tornos, uno por cada vestíbulo de la estación.

Estos sensores serán más que suficiente para una estación, aunque entre más sensores existan más precisión tendrá el modelo y mejor podrá advertir a los objetivos de lo que va a suceder en el futuro.

Sobre estos planos se ha generado el simulador que mostrará cómo se está comportando la estación en cada momento.

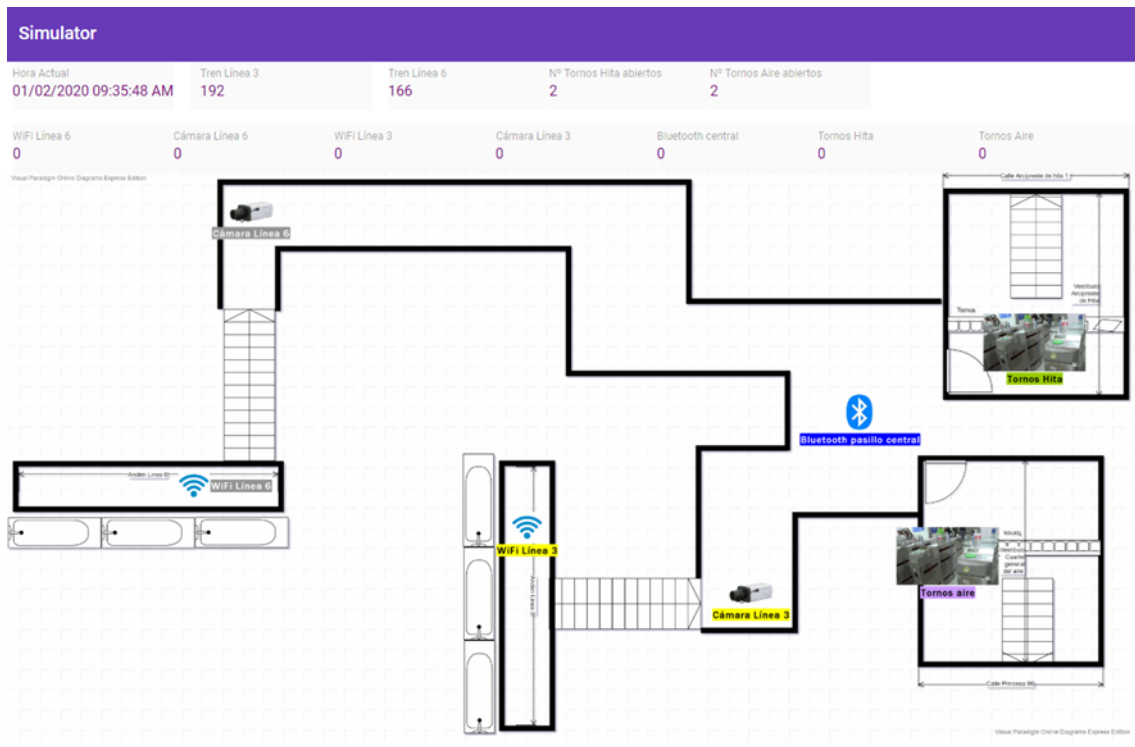


Figura 22 Simulador

En la Figura 22 se muestra que el simulador enseña, además del plano de la estación, las personas que se están detectando en cada momento. En la imagen, se pueden dividir los datos de la columna superior en:

- Hora: Muestra la hora actual que es en el sistema.
- Datos de trenes: Muestran los pasajeros que van a descender de cada tren.
- Tornos abiertos: Muestran los tornos de cada vestíbulo que estarán abiertos en cada momento.

Según la hora en la que se encuentre el sistema a lo largo del día, los horarios de los trenes y el número de personas que van viajando en ellos irán variando. De esta forma, existen tres tipos de horario: Alta, Media, Baja. Cuando el horario es de tipo Alta, los trenes pasarán con una frecuencia de 3 minutos con una ocupación alta (alrededor de 180 personas). Cuando el horario es de tipo Media, los trenes pasarán con una frecuencia de 5 minutos con una ocupación media (alrededor de 100 personas). Cuando el horario es de tipo Baja, los trenes pasarán con una frecuencia de 10 minutos con una ocupación baja (alrededor de 50 personas). Los tipos de horario se dispersan en el día haciendo que en los horarios donde la gente sale o entra a trabajar sean horarios de tipo Alta (07:00-10:00, 17:00-20:00) y en los horarios más extremos la ocupación sea Baja (20:00-02:00). Esta configuración es solo una de las posibles y puede modificarse en caso de que sea necesario.

El número de tornos abiertos será indicado por el sistema de pronóstico, que nos dirá cuántos tornos deben abrirse en función de la gente que el sistema espera que llegue a los tornos, ya

que por el camino las personas pueden cambiarse de un andén al otro, y no tener por qué salir de la estación a través de los tornos. Para la comunicación con los tornos, se ha utilizado el mismo sistema que muestra en el móvil la ocupación que va a ocurrir en el futuro, es decir, el sistema de comunicaciones de datos en tiempo real Firebase.

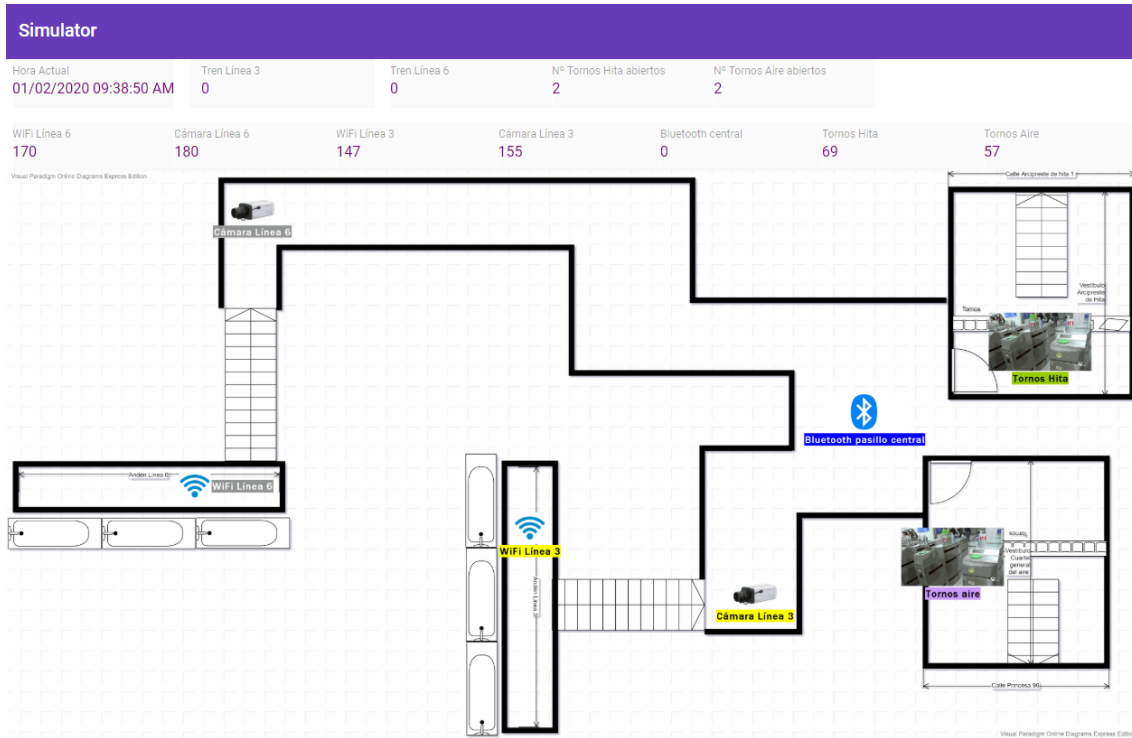


Figura 23 Sensores detectando personas

En la Figura 23 se observa como el simulador está detectando flujos de personas a través de la estación, pero todavía el sistema no ve necesario abrir más tornos para que la gente tenga que salir.

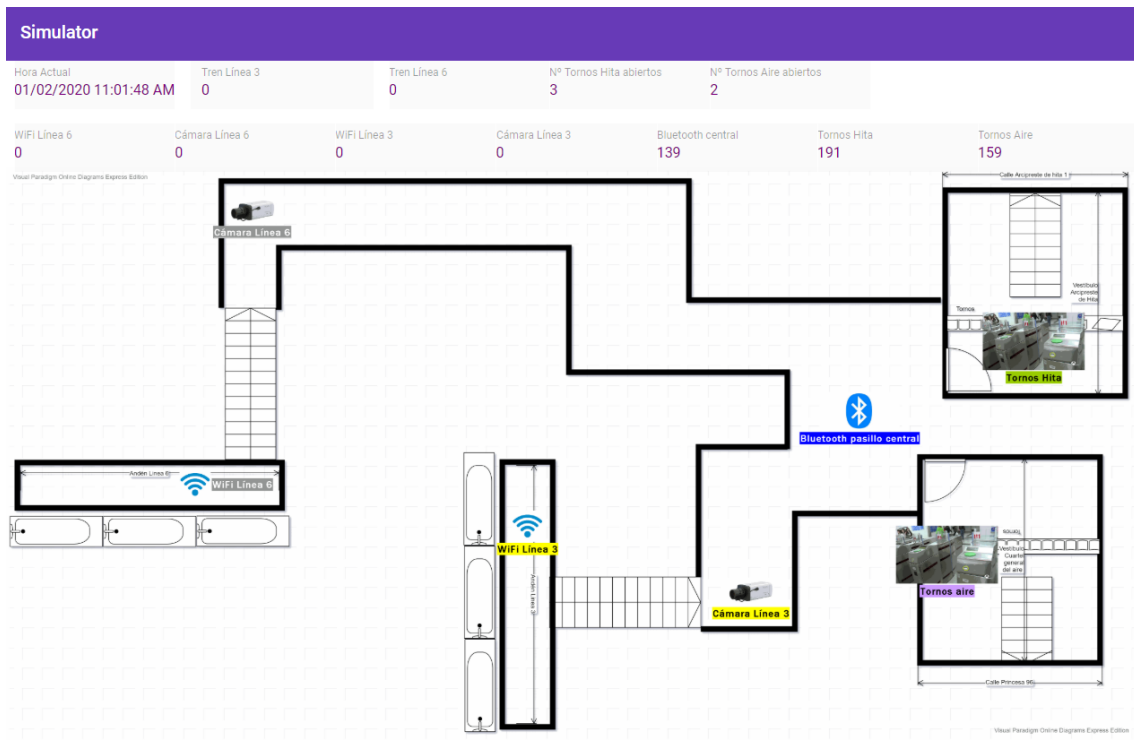


Figura 24 Tornos cambiando



Figura 25 Tornos cambiando 2

En las Figuras 24 y 25 se puede ver como el número de tornos cambia dependiendo del vestíbulo y de la gente que se detecta en cada zona. Actualmente en el torno del vestíbulo de la calle Arcipreste de Hita se está abriendo un torno más que en el torno del vestíbulo que da a la calle Princesa (denominado vestíbulo Ejército del Aire).

El simulador enviará todos los datos al sistema central que registra todos los datos en la plataforma de *Business Intelligence* (BI) que centralizará todos los datos interesantes que se produzcan en el sistema central y en los dispositivos periféricos. De esta manera, podemos ver mediante un cuadro de mando todos los datos en tiempo real que se están produciendo en el sistema.

sensor_wifi_linea_3				sensor_wifi_linea_6			
Time	id_device	n_people	type	Time	id_device	n_people	type
> Feb 1, 2020 @ 10:26:51.035	wifi_linea_3	173	wifi	> Feb 1, 2020 @ 10:26:50.914	wifi_linea_6	188	wifi
> Feb 1, 2020 @ 10:23:50.716	wifi_linea_3	167	wifi	> Feb 1, 2020 @ 10:23:50.686	wifi_linea_6	173	wifi
> Feb 1, 2020 @ 10:20:50.804	wifi_linea_3	176	wifi	> Feb 1, 2020 @ 10:20:50.689	wifi_linea_6	162	wifi
> Feb 1, 2020 @ 10:17:53.602	wifi_linea_3	151	wifi	> Feb 1, 2020 @ 10:17:53.430	wifi_linea_6	181	wifi
> Feb 1, 2020 @ 10:14:50.745	wifi_linea_3	147	wifi	> Feb 1, 2020 @ 10:14:50.719	wifi_linea_6	171	wifi
> Feb 1, 2020 @ 10:11:52.034	wifi_linea_3	158	wifi	> Feb 1, 2020 @ 10:11:52.017	wifi_linea_6	163	wifi
> Feb 1, 2020 @ 10:08:51.060	wifi_linea_3	155	wifi	> Feb 1, 2020 @ 10:08:51.027	wifi_linea_6	155	wifi
> Feb 1, 2020 @ 10:05:51.160	wifi_linea_3	181	wifi	> Feb 1, 2020 @ 10:05:51.020	wifi_linea_6	162	wifi

sensor_camara_linea_3				sensor_camara_linea_6			
Time	id_device	n_people	type	Time	id_device	n_people	type
> Feb 1, 2020 @ 10:26:51.041	camara_linea_3	182	camara	> Feb 1, 2020 @ 10:27:14.013	camara_linea_6	199	camara
> Feb 1, 2020 @ 10:23:50.818	camara_linea_3	176	camara	> Feb 1, 2020 @ 10:26:51.016	camara_linea_6	199	camara
> Feb 1, 2020 @ 10:20:50.799	camara_linea_3	186	camara	> Feb 1, 2020 @ 10:24:13.977	camara_linea_6	182	camara

Figura 26 Datos sensores Kibana

sensor_bluetooth_central				sensor_torno_aire			
Time	id_device	type	n_people	Time	id_device	n_people	type
> Feb 1, 2020 @ 19:38:36.434	bluetooth_central	bluetooth	119	> Feb 1, 2020 @ 19:38:54.478	torno_aire	71	torno
> Feb 1, 2020 @ 19:38:06.438	bluetooth_central	bluetooth	251	> Feb 1, 2020 @ 19:38:12.419	torno_aire	140	torno
> Feb 1, 2020 @ 19:35:36.445	bluetooth_central	bluetooth	136	> Feb 1, 2020 @ 19:36:06.443	torno_aire	77	torno
> Feb 1, 2020 @ 19:35:06.447	bluetooth_central	bluetooth	272	> Feb 1, 2020 @ 19:35:24.455	torno_aire	172	torno
> Feb 1, 2020 @ 19:32:35.609	bluetooth_central	bluetooth	118	> Feb 1, 2020 @ 19:33:18.435	torno_aire	77	torno
> Feb 1, 2020 @ 19:32:05.639	bluetooth_central	bluetooth	244	> Feb 1, 2020 @ 19:32:38.628	torno_aire	166	torno
> Feb 1, 2020 @ 19:29:35.641	bluetooth_central	bluetooth	146	> Feb 1, 2020 @ 19:29:50.630	torno_aire	93	torno
> Feb 1, 2020 @ 19:29:05.670	bluetooth_central	bluetooth	285	> Feb 1, 2020 @ 19:29:08.680	torno_aire	174	torno
> Feb 1, 2020 @ 19:26:35.654	bluetooth_central	bluetooth	132	> Feb 1, 2020 @ 19:27:02.666	torno_aire	72	torno

sensor_torno_hita				train_algorithm	
Time	id_device	n_people	type		
> Feb 1, 2020 @ 19:38:54.470	torno_hita	50	torno	> Feb 1, 2020 @ 18:25:20.355	0.8
> Feb 1, 2020 @ 19:38:12.421	torno_hita	167	torno	> Feb 1, 2020 @ 18:19:21.220	0.7902439024390244
> Feb 1, 2020 @ 19:36:06.435	torno_hita	85	torno	> Feb 1, 2020 @ 18:19:20.449	0.7990196078431373
> Feb 1, 2020 @ 19:35:24.458	torno_hita	193	torno	> Feb 1, 2020 @ 18:13:21.408	0.7941176470588235
> Feb 1, 2020 @ 19:33:18.444	torno_hita	64	torno	> Feb 1, 2020 @ 18:13:20.563	0.8029556650246306
				> Feb 1, 2020 @ 18:07:21.266	0.7970297029702971

Figura 27 Datos entrenamiento Kibana

Time	Accuracy	F1_score	Precision	Recall
> Feb 1, 2020 @ 18:07:21.003	0.8292682926829268	0.8932773109243699	0.8292682926829268	0.8292682926829268
> Feb 1, 2020 @ 18:07:20.179	0.7804878048780488	0.7727272727272728	0.7804878048780488	0.7804878048780488
> Feb 1, 2020 @ 18:01:21.008	0.8536585365853658	0.8491704374057315	0.8536585365853658	0.8536585365853658
> Feb 1, 2020 @ 18:01:20.138	0.8	0.8185644661054498	0.8	0.8
> Feb 1, 2020 @ 17:55:21.013	0.825	0.8903743315508021	0.825	0.825
> Feb 1, 2020 @ 17:55:20.170	0.8	0.8006393490264457	0.8	0.8
> Feb 1, 2020 @ 17:49:20.970	0.825	0.8929110105580693	0.825	0.825
> Feb 1, 2020 @ 17:49:20.181	0.775	0.760823875577974	0.775	0.775

get_prognosis		1-50 of 1217	
Time	result_prognosis		
> Feb 1, 2020 @ 18:07:20.122	4		
> Feb 1, 2020 @ 18:07:19.846	4		
> Feb 1, 2020 @ 18:05:30.108	3		
> Feb 1, 2020 @ 18:05:29.823	4		
> Feb 1, 2020 @ 18:05:20.113	3		
> Feb 1, 2020 @ 18:05:19.833	3		
> Feb 1, 2020 @ 18:05:00.085	3		
> Feb 1, 2020 @ 18:04:59.825	3		

Figura 28 Datos algoritmo y resultado

put_quantile_goal								1-50 of 730	
Time	count	max	min	std	25%	50%	75%		
> Feb 1, 2020 @ 18:07:20.205	464	531	1	98.7355876874	54	88	164.25		
> Feb 1, 2020 @ 18:07:19.907	464	520	3	97.2924288315	56	86	167		
> Feb 1, 2020 @ 18:03:20.176	462	531	1	98.8969236914	54	88	164.75		
> Feb 1, 2020 @ 18:03:19.854	462	520	3	97.4508852857	56	86	166.75		
> Feb 1, 2020 @ 17:59:20.821	458	531	1	99.1695364922	54	88	164		
> Feb 1, 2020 @ 17:59:20.331	458	520	3	97.755972775	56	85.5	166.75		
> Feb 1, 2020 @ 17:55:20.315	456	531	1	99.311544838	53.75	88	164		
> Feb 1, 2020 @ 17:55:19.979	456	520	3	97.9436490845	55.75	85	167		

Figura 29 Datos cuantil

En las Figuras 26, 27, 28 y 29 se observa como con una cantidad de datos de 928 datos los resultados del algoritmo mejoran y los datos de Accuracy y F1_Score superan el 85%. Esto significa que el algoritmo va aprendiendo cada vez más y podrá dar unos resultados cada vez mejores.

5. Conclusiones y Trabajo Futuro

El trabajo realizado ha servido para demostrar la utilidad de los sistemas difusos para evitar situaciones de estrés en el dominio de los transportes públicos. Como se ha podido observar durante todo el trabajo, actualmente el dominio de los transportes, aunque es un sector que crece y mejora a pasos agigantados, todavía no dispone de suficiente material innovador en la mejora de la experiencia del usuario en lo tocante al estrés que sufre al moverse por las estaciones.

Es por ello que este trabajo sirve y servirá como punta de lanza en la investigación en este sentido, y es debido a esto, por lo que el trabajo se ha desarrollado siguiendo un enfoque pragmático que permita que en un futuro próximo empresas de cualquier tamaño pudiesen implementarlo como parte de su *suite* de productos de cara a mejorar la experiencia del usuario.

Un enfoque como el realizado en el trabajo, pone de manifiesto que, si bien el campo de los sistemas difusos no es un campo nuevo dentro de la informática, sí que sigue ofreciendo mucha potencia y facilidad de entendimiento en muchos de los aspectos en los que la vida cotidiana es introducida en el mundo de los computadores. Esto ha sido mostrado claramente en el trabajo realizado, al generar un sistema modular que utiliza las últimas tecnologías para detectar a seres humanos y unir esos datos mediante modelos inteligencia artificial a los que se le aplica lógica difusa para predecir comportamiento y, de esta manera, hacer que el hardware alrededor de los seres humanos pueda comportarse de manera dinámica, facilitándole la vida a estos. El caso desarrollado, como comentaba anteriormente, muestra la utilidad que siguen teniendo a día de hoy los sistemas difusos ya que, unidos a las tecnologías más avanzadas y novedosas del mercado, son capaces de obtener resultados complejos que ayudan y mejoran la vida de las personas eliminando algunos problemas en su rutina diaria.

No es posible olvidar, que, aunque los resultados han sido satisfactorios en entornos pequeños o de laboratorio, ha habido múltiples condiciones que han sesgado pruebas complejas y contundentes (legislaciones y contactos con entidades de transporte) y, serán estas condiciones, las que deberán resolverse en el futuro para encontrar errores en el sistema y mejorarlo en la medida que la tecnología lo permita.

Aun así, los siguientes pasos a dar para mejorar el sistema deberían ser:

- Añadir más dispositivos al entorno, así como, añadir más dispositivos objetivo que aumenten la complejidad de cálculo del modelo generado mediante aprendizaje automático.
- Probar los algoritmos que permiten añadir cámaras de video al sistema en entornos reales.
- Conseguir conectar el sistema con barreras o pasos, que permitan mostrar como el hardware es modificado a través del sistema.

Y por último y más importante, conseguir que el sistema sea capaz de seguir y prever flujos en múltiples direcciones, dependiendo de fechas y de horas, ya que este paso hará que el sistema sea totalmente abierto y aplicable a muchos tipos diferentes de dominios.

6. Bibliografía

- [1] M. Vanderschuren, "The benefits of intelligent transport systems: modelling the effects of different ITS systems.," 2003.
- [2] Unión Europe, "Shift2Rail," Unión Europea, 14 07 2014. [Online]. Available: <https://shift2rail.org/>. [Accessed 11 01 2020].
- [3] Unión Europea, "My Travel Companion," Unión Europea, 12 09 2017. [Online]. Available: <https://cordis.europa.eu/project/id/777640>. [Accessed 11 01 2020].
- [4] H. P. Fani, "Human resistance to the use of information technology in construction companies," *Civil Engineering Dimension*, vol. 7, no. 2, p. 68 – 74, 2005.
- [5] hbr.org, "How to Deal With Resistance to Change," hbr.org, [Online]. Available: <https://hbr.org/1969/01/how-to-deal-with-resistance-to-change>. [Accessed 10 11 2019].
- [6] J. Feller and B. Fitzgerald, *Understanding open source software development*, London: Addison-Wesley, 2002.
- [7] D. T. Meitei, T. Khelchandra, K. Manglem Singh and S. Roy, "A Genetic Algorithm and Fuzzy Logic Approach for Video Shot Boundary Detection," *Hindawi Publishing Corporation, Computational Intelligence and Neuroscience*, no. 8469428, p. 11, 2016.
- [8] E. Elbaşı, "Fuzzy Logic-Based Scenario Recognition from Video Sequences," *Journal of Applied Research and Technology*, vol. 11, no. 5, pp. 702-707, 2013.
- [9] J. M. Alonso, N. Hernández, F. Herranz, Á. Llamazares, M. A. Sotelo, L. M. Bergasa and L. Magdalena, "Enhanced WiFi localization system based on SoftComputing techniques to deal with small-scale variations in wireless sensors," *Applied Soft Computing*, vol. 11, no. 8, pp. 4677-4691, 2011.
- [10] T. Garcia-Valverde, A. García-Sola, H. Hagraas, J. Dooley, V. Callaghan and J. Botía, "A Fuzzy Logic-Based System for Indoor Localization Using WiFi in Ambient Intelligent Environments," *Fuzzy Systems*, no. 21, pp. 702-718, 2013.
- [11] A. Soetedjo, Y. Nakhoda and C. Saleh, "Embedded Fuzzy Logic Controller and Wireless Communication for Home Energy Management Systems," *Electronics*, 2018.
- [12] Y. Anekwong, T. Yooyativong and R. Chaisricharoen, "The wifi multi-sensor network for fire detection mechanism using fuzzy logic with IFTTT process based on cloud.," in *14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2017.
- [13] F. Orujov, R. Maskeliūnas, R. Damaševičius, W. Wei and Y. Li, "Smartphone based intelligent indoor positioning using fuzzy logic.," *Future Generation Comp.*, vol. 89, pp. 335-348, 2018.
- [14] E. Hüllermeier, "Does machine learning need fuzzy logic?," *Fuzzy Sets and Systems*, no. 281, pp. 292-299, 20015.
- [15] C. R. Sedano, E. L. Ursini and P. S. Martins, "A Bullying-Severity Identifier Framework Based on Machine Learning and Fuzzy Logic.," *ICAISC*, 2017.
- [16] P. Magdalinos, K. Apostolos, S. Panagiotis, G. Katsikas and N. Alonistioti, "Enhancing a Fuzzy Logic Inference Engine through Machine Learning for a Self- Managed Network.," *Mobile Networks and Applications*, no. 16, pp. 475-789, 2011.

- [17] E. Boulliet, L. Gasparini and O. Verscheure, "Towards A Real Time Public Transport Awareness System: Case Study In Dublin," in *Proceedings of the 19th International Conference on Multimedia 2011*, Scottsdale, 2011.
- [18] A. A. Rivas, "Introducción a IBM InfoSphere Streams y SPL," IBM, 21 08 2012. [Online]. Available: <https://www.ibm.com/developerworks/ssa/data/library/techarticle/introduccion-streams/index.html>. [Accessed 17 11 2019].
- [19] D. Jaimovich, "Estaciones inteligentes: las propuestas para modernizar el subte porteño," Infobae, 26 04 2019. [Online]. Available: <https://www.infobae.com/tendencias/innovacion/2019/04/26/estaciones-inteligentes-las-propuestas-para-modernizar-el-subte-porteno/>. [Accessed 17 11 2019].
- [20] V. Záhorová, "The application of fuzzy logic to a simulation of railway station," in *5th International Scientific Conference "Theoretical and Practical Issues in Transport"*, Pardubice, Czech Republic, 2010.
- [21] L. Riza, C. Bergmeir, F. Herrera and J. Benítez, "Fuzzy Rule-Based Systems for Classification and Regression in R," *Journal of Statistical Software*, vol. 65, 2015.
- [22] Scikit, The scikit-fuzzy Documentation, The scikit-image team, 2016.
- [23] C. Jiménez Ramirez and H. Álvarez Zapata, "Minería de datos basada en lógica difusa para la interpretación de consultas vagas dependientes del contexto lingüístico," *Dyna*, 2012.
- [24] M. Elkano, M. Uriz, H. Bustince and M. Galar, "Aplicando la transformada integral de la probabilidad para reducir la complejidad de los árboles de decisión difusos multi-vía en problemas de clasificación Big Data," *Workshop en Big Data y Analisis de Datos Escalable*, vol. II, 2016.
- [25] S. Media Hurtado and G. Paniagua Gómez, "Modelo de inferencia difuso para estudio de crédito," *Dyna*, no. 154, pp. 215-229, 2007.
- [26] A. Durand, L. Harms, S. Hoogedoorn-Lanser and T. Zijlstra, "Mobility-as-a-Service and changes in travel preferences and travel behaviour: a literature review," in *KiM Netherlands Institute for Transport Policy Analysis*, The Hague, Ministry of Infrastructure and Water Management, 2018.
- [27] MaaS Alliance, "White Paper," *Guidelines & Recommendations to create the foundations for a thriving MaaS Ecosystem*, 2017.
- [28] E. Peñafiel, A. Arellano and G. Vallejo, "Control de Velocidad de un auto empleando Sistemas Difusos," *Revista Científica y Tecnológica UPSE*, vol. 4, 2017.
- [29] C. Álvarez and W. Castro, "Módulo para sistemas basados en técnicas de inteligencia artificial: aplicación al péndulo invertido.," 2019.
- [30] E. Mamdani and S. A. , "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1-13, 1975.
- [31] Numpy, "Manual librería quantile," Numpy, [Online]. Available: <https://numpy.org/devdocs/reference/generated/numpy.quantile.html>. [Accessed 10 11 2019].
- [32] M. G. Akbari and G. Hesamiana, "Fuzzy Quantile Linear Regression Model Adopted with a Semi-Parametric Technique based on Fuzzy Predictors and Fuzzy Responses," *Expert Systems With Applications*, no. 118, p. 585–597, 2019.

- [33] P. M. Robinson, "Root-N-Consistent Semiparametric Regression," *Econometrica*, vol. 56, no. 4, pp. 931-954, 1988.
- [34] P. Lisbon, "An introduction to machine learning.," *Language Technology Group (LTG)*, vol. 35, no. 1, p. 13, 2015.
- [35] J. VanderPlas, "Python Data Science Handbook.," in *Essential Tools for Working with Data*, 2016, pp. Cap- 3-5.
- [36] C. Dapi and W. Chenglong, "Wireless terminal MAC address acquisition method and device," 09 04 2014. [Online]. Available: <https://worldwide.espacenet.com/publicationDetails/biblio?FT=D&date=20140409&DB=EPODOC&locale=&CC=CN&NR=103716777A>. [Accessed 01 12 2019].
- [37] PrivazyPlan, "Artículo 4 UE RGDP," EU general data protection regulation 2016/679 (GDPR), 25 5 2018. [Online]. Available: <http://www.privacy-regulation.eu/es/4.htm>. [Accessed 01 12 2019].
- [38] A. Esteva, B. Kuprel, J. Ko, S. M. Swetter, R. A. Novoa, H. M. Blau and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, no. 542, p. 115–118, 2017.
- [39] H. Bendemra, "Construye tu primer clasificador de Deep Learning con TensorFlow: Ejemplo de razas de perros," <https://medium.com/>, 30 4 2018. [Online]. Available: </datos-y-ciencia/construye-tu-primer-clasificador-de-deep-learning-con-tensorflow-ejemplo-de-razas-de-perros-ed218bb4df89>. [Accessed 1 20 2019].
- [40] Scikit-learn, "Cross-validation: evaluating estimator performance," Scikit-learn, [Online]. Available: https://scikit-learn.org/stable/modules/cross_validation.html#cross-validation. [Accessed 10 11 2019].
- [41] A. Long, "Understanding Data Science Classification Metrics in Scikit-Learn in Python," towardsdatascience, 2018 08 2018. [Online]. Available: <https://towardsdatascience.com/understanding-data-science-classification-metrics-in-scikit-learn-in-python-3bc336865019>. [Accessed 10 11 2019].
- [42] developers.sw.com, "¿SOAP o REST?," developers.sw.com, [Online]. Available: <https://developers.sw.com.mx/knowledge-base/soap-o-rest/>. [Accessed 10 11 2019].
- [43] codeday.me/, "Servicios web – ¿Por qué preferir REST sobre SOAP?," codeday.me/, [Online]. Available: <https://codeday.me/es/qa/20190321/339647.html>. [Accessed 10 11 2019].
- [44] Swagger, "Swagger Official Page," Swagger, [Online]. Available: <https://swagger.io/>. [Accessed 10 11 2019].
- [45] Swagger, "Cómo documentar nuestras APIs con Swagger," Swagger, [Online]. Available: <https://medium.com/@lfernopalacio/c%C3%B3mo-documentar-nuestras-apis-con-swagger-1eca6d5524a8>. [Accessed 10 11 2019].
- [46] codeclimate, "5 Reasons to Use Protocol Buffers Instead of JSON for Your Next Service," codeclimate, [Online]. Available: <https://codeclimate.com/blog/choose-protocol-buffers/>. [Accessed 10 11 2019].
- [47] Medium, "Session vs Token Based Authentication," Medium, [Online]. Available: <https://medium.com/@sherryhsu/session-vs-token-based-authentication-11a6c5ac45e4>. [Accessed 10 11 2019].

- [48] Wikipedia, "Singleton," [Online]. Available: <https://es.wikipedia.org/wiki/Singleton>. [Accessed 10 11 2019].
- [49] SQLAlchemy, "SQLAlchemy Official Page," SQLAlchemy, [Online]. Available: <https://docs.sqlalchemy.org/en/13/index.html>. [Accessed 10 11 2019].
- [50] J. Mockor, "Fuzzy logic interpretations in categories of fuzzy objects.," *Journal of Fuzzy Set Valued Analysis*, pp. 63-74, 2016.
- [51] A. Amo, J. Montero, G. Biging and V. Cutelloc, "Fuzzy classification systems," *Computing, Artificial Intelligence and Information Technology*, vol. 156, no. 2, pp. 495-507, 2004.
- [52] P. Holeček, J. Talasová and S. Talasová, "Fuzzy classification systems and their applications.," 2011.
- [53] SQLite, "Appropriate Uses For SQLite," SQLite, [Online]. Available: <https://www.sqlite.org/whentouse.html>. [Accessed 10 11 2019].
- [54] Firebase, "Firebase página oficial," Firebase, [Online]. Available: <https://firebase.google.com/?hl=es-419>. [Accessed 10 11 2019].
- [55] Firebase, "Firebase Realtime Database," Firebase, [Online]. Available: <https://firebase.google.com/docs/database?hl=es-419>. [Accessed 10 11 2019].
- [56] Firebase, "Firebase página de precios," Firebase, [Online]. Available: <https://firebase.google.com/pricing?authuser=0>. [Accessed 10 11 2019].
- [57] Q. Nguyen, *Mastering Concurrency in Python: Create faster programs using concurrency*, Birmingham: Packt, 2018.
- [58] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. Vanderplas, A. Joly, B. Holt and G. Varoquaux, "API design for machine learning software: Experiences from the scikit-learn project," 2013.

7. Lista de acrónimos

WiFi	Wireless Fidelity
I+D+i	Investigación, desarrollo e innovación
UE	Unión Europea
ITS	Intelligent Transportation Systems
FRBC	Fuzzy Rule-Based Classifier
IFTTT	If This Then That
HEMS	Sistemas de Gestión de Energía en el Hogar
BLE	Bluetooth Low Energy
MaaS	Mobility as a Service
MAC	Media Access Control
CNN Convolucionales	Convolutional Neural Network
BI	Business Intelligence
HTTPS	Hypertext Transfer Protocol Secur
ORM	Object-Relational Mapping
NoSQL	No solo SQL o No Relacional
JSON	Notación de objeto de JavaScript
API	Interfaz de Programación de Aplicaciones
URI	Identificador de Recursos Uniforme

Tabla 6 Lista de acrónimos

8. Anexo

Dado que los desarrollos llevados a cabo en este trabajo han sido complejos de desarrollar, en el anexo se pasará a explicar el funcionamiento en detalle del sistema central, que es realmente el sistema que desarrolla toda la lógica de los sistemas difusos que se han desarrollado.

8.1. Sistema central

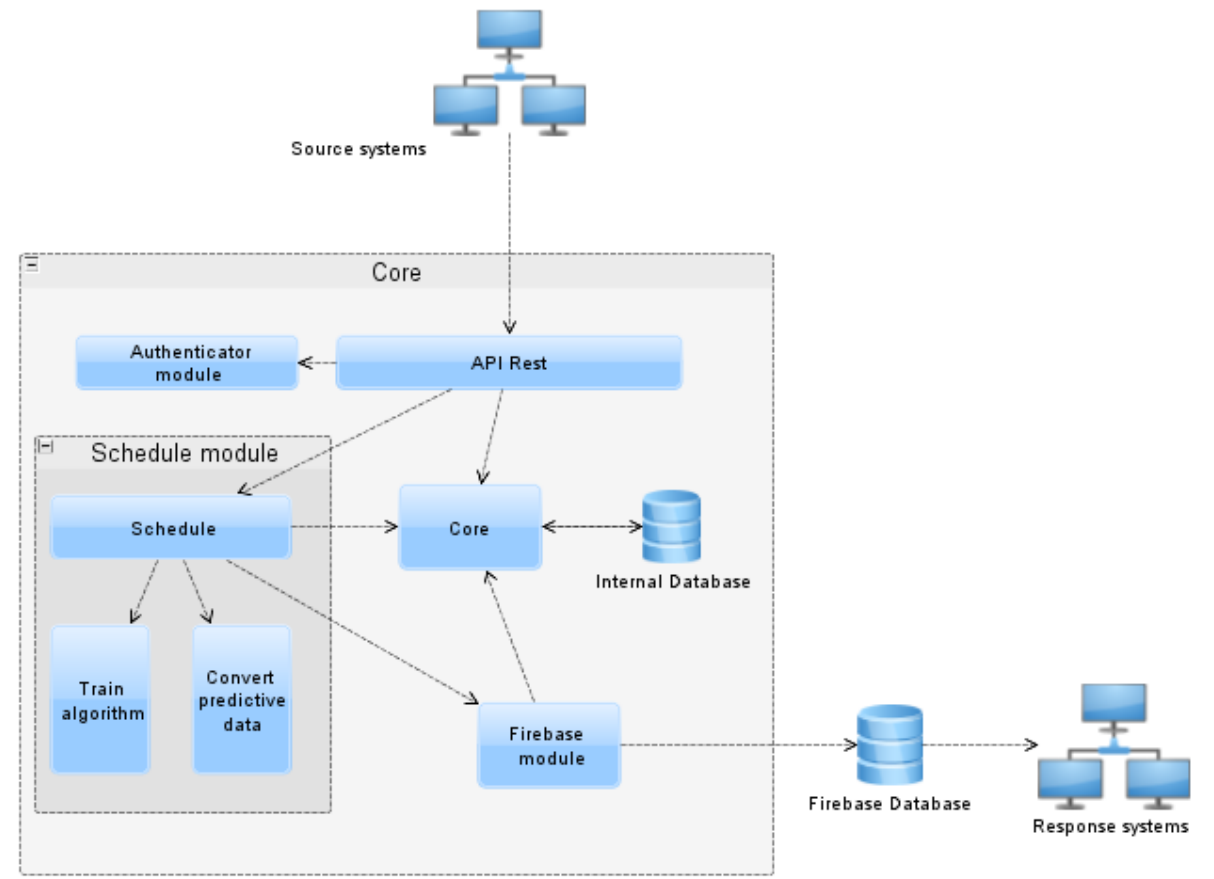


Figura 30 Detalle Sistema Central

El sistema central es el core de la aplicación, en él están desarrollados tanto los sistemas de comunicación como los sistemas de predicción y almacenado. Cada módulo está desarrollado de forma que en caso de error no bloquee los demás servicios.

La función del sistema central es recibir los datos desde las diversas y distintas fuentes de datos que existan en el sistema, almacenar y calcular los valores necesarios correspondientes y enviar la información de las predicciones a los sistemas de respuesta, los cuales utilizan un sistema intermedio (Firebase) de comunicación en tiempo real. De esta manera, los sistemas fuentes están separados del sistema central y se pueden poner o quitar a nuestro antojo, teniendo en cuenta que añadir o eliminar fuentes de datos en tiempo real puede hacer que los resultados del algoritmo puedan permanecer inestables algún tiempo y que la mejor opción es intentar

mantenerlos todo el tiempo que se desee tener el sistema. También pasa lo mismo con el sistema de respuesta, que está separado del sistema central por un sistema de comunicación intermedio que se encarga de enviar las diversas señales a todos los dispositivos que sea necesario, este sistema puede cambiar por otro si fuese necesario ya que es totalmente externo al sistema.

Ahora se va a pasar a explicar los módulos más relevantes del core de manera individualizada:

8.1.1. Api Rest

La Api es el punto de entrada al sistema tanto para los dispositivos como para el administrador que desee arrancar o detener el sistema, desde ella se realizan tres tareas fundamentales:

- Actualizar datos de los dispositivos de manera individualizada: este servicio permite a los dispositivos periféricos actualizar frecuentemente sus datos. Estos datos serán almacenados en la base de datos para su posterior tratamiento.
- Obtener los datos que existen en el sistema, para poder comprobar en tiempo real qué está sucediendo: este servicio permite en cualquier momento al usuario conocer cuáles son los resultados de las operaciones que se están llevando a cabo en tiempo real, y también conocer el valor específico de las personas que están siendo detectadas por cada dispositivo.
- Arrancar y detener el sistema para que no siga entrenando al algoritmo ni siga emitiendo datos a los dispositivos de respuesta: este servicio dispone de dos posibilidades, una es “start” y otra “stop”. Mediante estos dos estados, el usuario le indicará al sistema que debe comenzar a realizar el tratamiento de los datos de los periféricos y a calcular en tiempo real que debe de ocurrir en el sistema mediante el sistema de pronóstico del que dispone. Si el administrador del sistema no activa este servicio, entonces no se enviará nada a los sistemas de respuesta. El administrador puede detener el sistema en el momento que desee sin que los datos se pierdan en ningún momento, gracias a la base de datos que tiene incorporada el sistema.

Para la API se han utilizado servicios Rest en formato json, además de por su sencillez, también porque son más livianos a la hora de realizar comunicaciones [42]. Además, estos servicios al ser más fáciles de modificar pueden ser actualizados de manera más sencilla, por lo tanto, da al sistema una gran facilidad para ser mejorado en el futuro [43]. También es destacable, que los servicios Rest son más rápidos de desarrollar (al eliminar los contratos intermedios existentes en las comunicaciones SOA) y pueden desarrollar una documentación de una gran calidad utilizando diversos framework como, en este caso es Swagger [44].

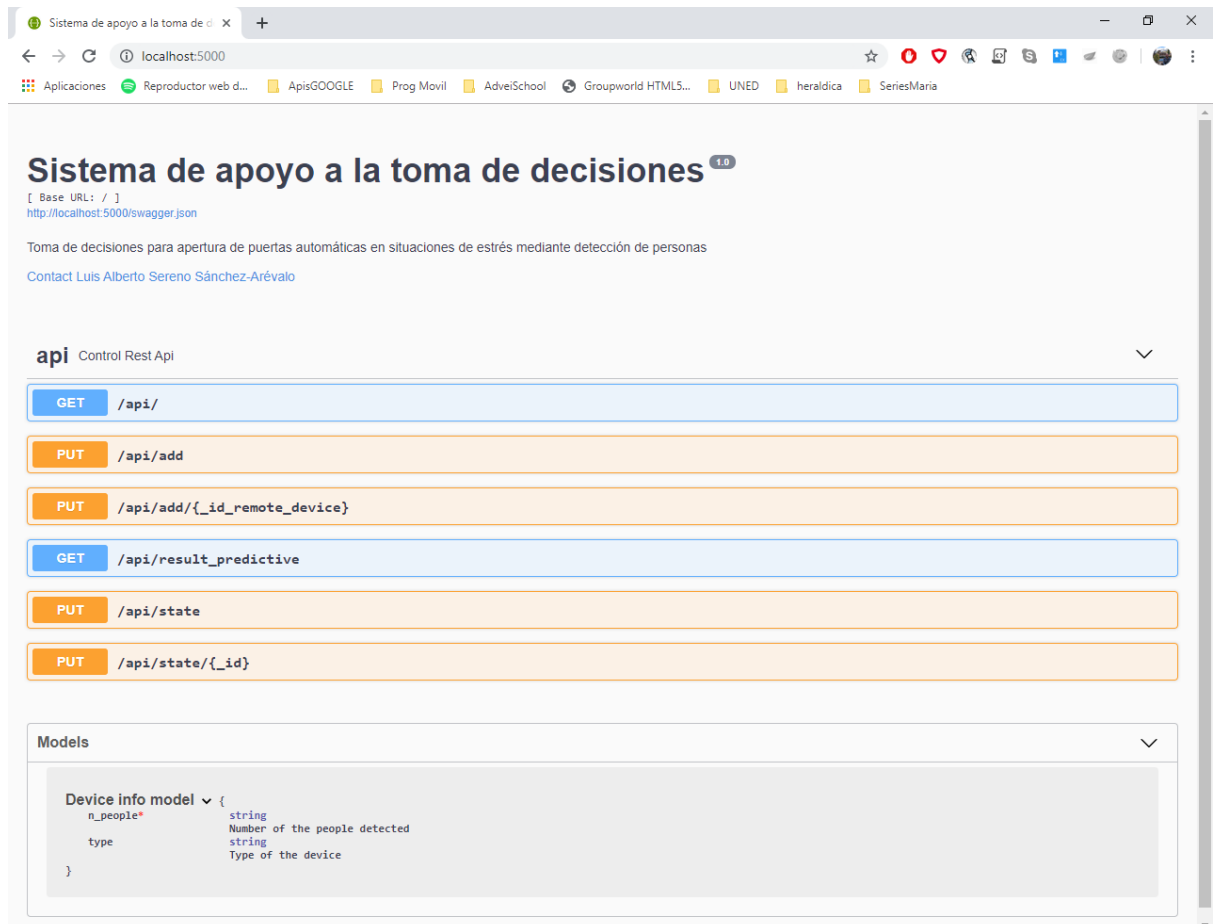


Figura 31 Documentación Swagger servicios

Swagger es un framework open-source respaldado por un gran ecosistema de herramientas que ayuda a los desarrolladores a diseñar, construir, documentar y consumir servicios web RESTful. Además, también ofrece un sistema de reglas que se está empezando a convertir en un estándar entre la comunidad de desarrollo [45]. En este sentido también hay otras herramientas en el mercado con las cuales documentar las APIS como pueden ser Blueprint o RAML, pero se eligió Swagger por el tamaño de la comunidad de desarrolladores, la gran documentación existente y la experiencia en el uso de la que se dispone de ella.

En el sistema es muy importante que el tráfico de datos sea el menor posible, ya que las comunicaciones son muchas y continuas, por lo tanto, hay que buscar un formato liviano y el cual proporciona un tránsito de datos lo menor posible, pero siempre manteniendo la misma potencia, es por ello que se ha elegido el formato JSON para este tipo de comunicaciones, ya que ahorra bastante tamaño y además es fácilmente entendible para el ojo humano. También se valoró el uso de un buffer de bytes del tipo “protobuf” [46] pero se descartó por la dificultad que añadía para la comprensión lectora al ojo humano, pero sin duda en un entorno productivo se optaría por este formato.

El código Swagger de la API es el siguiente:

```
{
```



```
"swagger": "2.0",
"basePath": "/",
"paths": {
  "/api/": {
    "get": {
      "responses": {
        "200": {
          "description": "OK"
        }
      },
      "description": "Llamada de prueba para confirmar que el servidor está levantado",
      "operationId": "get_help",
      "tags": [
        "api"
      ]
    },
    "put": {
      "responses": {
        "200": {
          "description": "OK"
        }
      },
      "description": "Mensaje de error si se llama sin el identificador o sin el cuerpo",
      "operationId": "put_update_info_none",
      "tags": [
        "api"
      ]
    }
  },
  "/api/add/{_id_remote_device}": {
    "put": {
      "responses": {
        "200": {
          "description": "OK"
        }
      },
      "description": "Servicio que inserta en el sistema datos de los dispositivos fuente",
      "operationId": "put_update_info_num_passenger",
      "parameters": [
        {
          "name": "_id_remote_device",
          "in": "path",
          "required": true,
          "type": "string",
          "description": "Specify the _id_remote_device associated with the device"
        },
        {
          "name": "payload",
          "required": true,
          "in": "body",
          "schema": {
            "$ref": "#/definitions/Device info model"
          }
        }
      ],
      "tags": [
        "api"
      ]
    }
  }
}
```

```

    ]
  }
},
"/api/result_predictive": {
  "get": {
    "responses": {
      "200": {
        "description": "OK"
      }
    },
    "description": "Servicio que inserta devuelve el estado actual del servicio, tanto la predicción
como los últimos datos de cada dispositivo",
    "operationId": "get_predictive",
    "tags": [
      "api"
    ]
  }
},
"/api/state": {
  "put": {
    "responses": {
      "200": {
        "description": "OK"
      }
    },
    "description": "Mensaje de error si se llama sin el código de estado",
    "operationId": "put_update_state_none",
    "tags": [
      "api"
    ]
  }
},
"/api/state/{_id}": {
  "parameters": [
    {
      "name": "_id",
      "in": "path",
      "required": true,
      "type": "string"
    }
  ],
  "put": {
    "responses": {
      "200": {
        "description": "OK"
      }
    },
    "description": "Servicio que arranca el sistema interno para empezar a emitir resultados",
    "operationId": "put_update_state",
    "parameters": [
      {
        "description": "Specify the Id associated with the status of the system \"start\" or \"stop\" ",
        "name": "id",
        "type": "string",
        "in": "query"
      }
    ]
  },
  "tags": [
    "api"
  ]
}

```

```

    ]
  }
}
},
"info": {
  "title": "Sistema de apoyo a la toma de decisiones",
  "version": "1.0",
  "description": "Toma de decisiones para apertura de puertas automáticas en situaciones de estrés mediante detección de personas",
  "contact": {
    "name": "Luis Alberto Sereno Sánchez-Arévalo",
    "email": "lsereno1@uned.es",
    "url": null
  }
},
"produces": [
  "application/json"
],
"consumes": [
  "application/json"
],
"tags": [
  {
    "name": "api",
    "description": "Control Rest Api"
  }
],
"definitions": {
  "Device info model": {
    "required": [
      "n_people"
    ],
    "properties": {
      "n_people": {
        "type": "string",
        "description": "Number of the people detected"
      },
      "type": {
        "type": "string",
        "description": "Type of the device"
      }
    },
    "type": "object"
  }
},
"responses": {
  "ParseError": {
    "description": "When a mask can't be parsed"
  },
  "MaskError": {
    "description": "When any error occurs on mask"
  }
}
}
}

```

Este código puede insertarse en diversos editores online cómo “<https://editor.swagger.io/>” que nos mostrarán el formato Swagger de una manera simple y cómoda de entender.

La API dispone de un servicio que le aporta un cierto nivel de seguridad, a un nivel básico, que ha sido implementado para demostrar que es posible también darle al sistema la seguridad que necesite. Este módulo es denominado Authenticator. En él, lo que se comprueba es que cada petición que llegue a cualquier servicio tenga en la cabecera el parámetro Authorization con el usuario y contraseña cifrados con *Basic Authentication*. Este modo de seguridad se denomina basado en token, y es llamado así porque una vez que el usuario se registra con su usuario y password, este recibirá un token aprobado por el servidor y con el que se realizarán las siguientes conexiones [47].

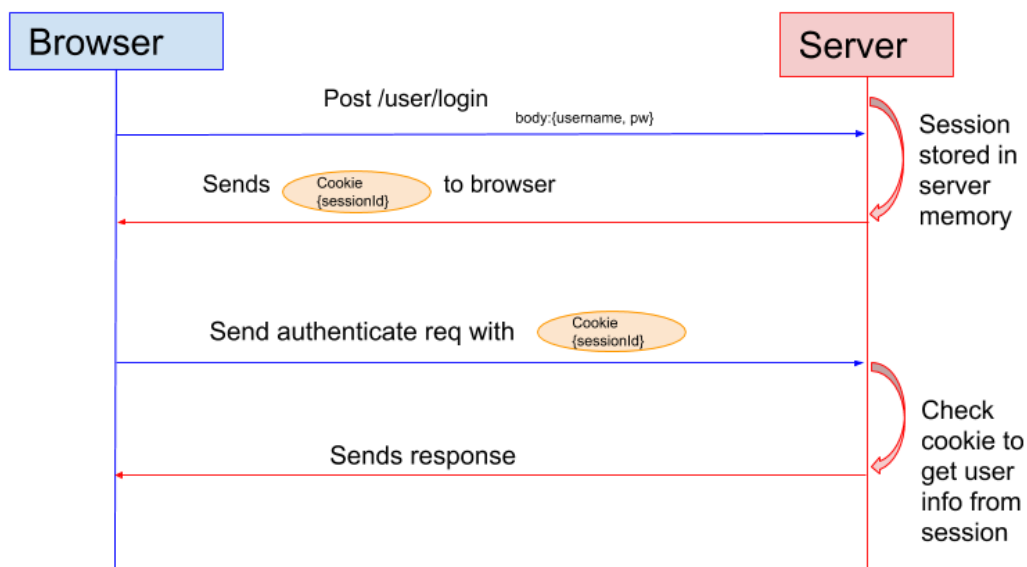


Figura 32 Comunicaciones de seguridad

Si el sistema fuese a un entorno productivo tendría que mejorar esta seguridad y también realizar todas las comunicaciones cifradas con un certificado para que se realizará mediante HTTPS y fuesen aún más seguras. Para el ámbito de este proyecto, con este sistema de validación se considera suficiente seguridad.

8.1.2.Core

El núcleo del sistema difuso se compone de una serie de métodos y clases que son utilizadas por las demás funcionalidades y que, además, se comunican con la base de datos. Estas clases contenidas en el core de la aplicación, aunque no realizan funcionalidades extremadamente complejas, sirven de unión en todo el sistema y hacen que la integración entre los distintos componentes sea más sencilla.

En este módulo, se realizan las conexiones iniciales a la base de datos que serán utilizadas por toda la aplicación y además también se realiza la preconfiguración del sistema en caso de que sea su primera inicialización.

También dispondrá este módulo del método principal para determinar la ocupación del sistema en sí. La función `select_occupancy` se ocupa directamente de este trabajo. Para ello, obtiene los valores de las personas que están siendo detectadas por el dispositivo y los cruza con los valores mínimos de cada categoría para ese dispositivo (en el sistema los denominamos barreras, que están asociadas a los cuantiles), una vez encontrado el cuartil en que se encuentra el número de personas detectadas, se devuelve la ocupación para ese dispositivo.

Ahora se mostrará un diagrama de clases que será explicado a continuación:

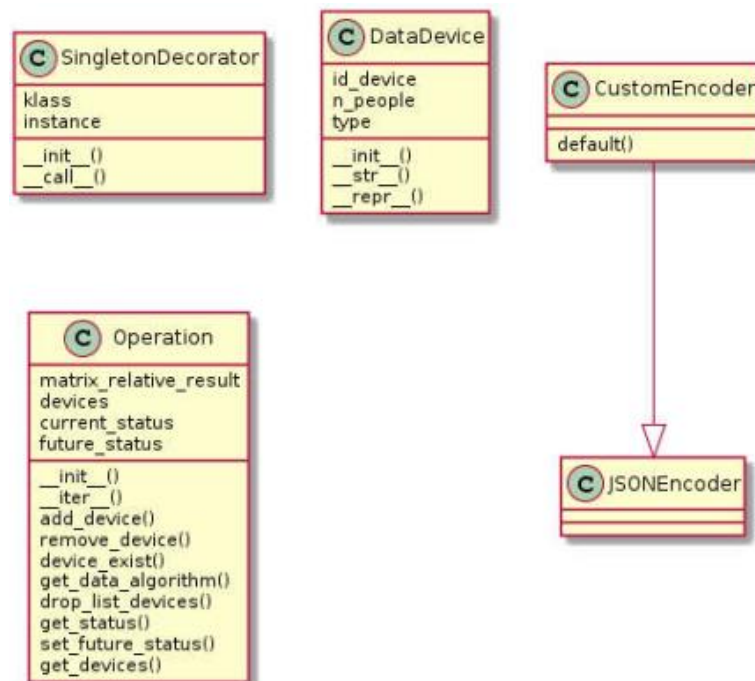


Figura 33 Diagrama de clases Core

El diagrama de clases anterior muestra el total de las clases existentes en el módulo de core (exceptuando las funciones generales que han sido detalladas anteriormente). Las clases son las siguientes:

- **SingletonDecorator:** La función de esta clase es proveer al sistema de un decorador para poder realizar clases mediante el patrón Singleton [48]. Este decorador nos permite tener una única instancia de una clase para toda la aplicación, por lo tanto, siempre que se quiera instanciar esta clase desde otra, la clase llamante obtendrá la única instancia de esa clase del sistema con los valores que posea. Este patrón es muy interesante para asegurar la integridad de los datos con los que se trabaja en la aplicación, reduce el espacio en memoria y además evita llamadas innecesarias a base de datos.
- **DataDevice:** Esta clase tiene como objetivo principal ser la interfaz de entrada, desde el controlador, para la actualización de los datos de los periféricos fuente del sistema.

Es sobretodo utilizado por el servicio “update_info_num_passenger” para trabajar con la clase “Operation”.

- CustomEncoder: La única función de esta clase es realizar una conversión desde objetos complejos en Python a JSON para enviar estos datos a través de los servicios de comunicación. Para ello esta función hereda la funcionalidad en la clase JSONEncoder propia de Python y que tiene un rendimiento excelente por ser parte de su código fuente.
- Operation: La principal función de esta clase es mantener en memoria todos los datos importantes del sistema, sus funciones van desde añadir o actualizar los datos de nuevos dispositivos en tiempo real y obtener esos datos, hasta obtener el estado actual de la aplicación. Esta función es una de las más importantes dentro del módulo Core ya que además se conecta con la base de datos para actualizar toda la información que sea necesaria en la base de datos u obtenerla en el caso necesario. Esta función utiliza el patrón Singleton mencionado anteriormente para eliminar posibles incongruencias en el sistema y de paso ahorrar memoria (ya que solo existe una instancia en el sistema) y aportar velocidad a los procesamientos.

Dentro del módulo Core del sistema, también se encuentra la clase que hace de capa intermedia con la base de datos. Esta capa es muy importante, pues se encarga de dar una interfaz y una separación entre la base de datos y el sistema en sí mismo. De este modo, podríamos tener cualquier tipo de base de datos y no necesitar realizar prácticamente ningún cambio en el código. Esta ventaja es muy importante en sistemas de investigación, porque te permite realizar evoluciones y aumentar la potencia de la herramienta de manera sencilla y ahorrando tiempos en los desarrollos futuros, y también, en posibles implantaciones en sistemas productivos.

El módulo de conexión con la base de datos “DatabaseManager” contiene además de la clase con los métodos para interactuar con la base de datos, distintas clases que sirven para autogenerar los esquemas de base de datos, para ello utiliza el framework ORM “SqlAlchemy” [49] que es el ORM más potente en la actualidad para Python y que facilita la vida del desarrollador enormemente. Dentro de este módulo de conexión existe un método estático que se encarga de generar la sesión y la conexión con la base de datos, y además de hacer que las operaciones sean transaccionales, lo que aporta una estabilidad muy importante a la base de datos.

A continuación se muestran las clases existentes (Figura 34):

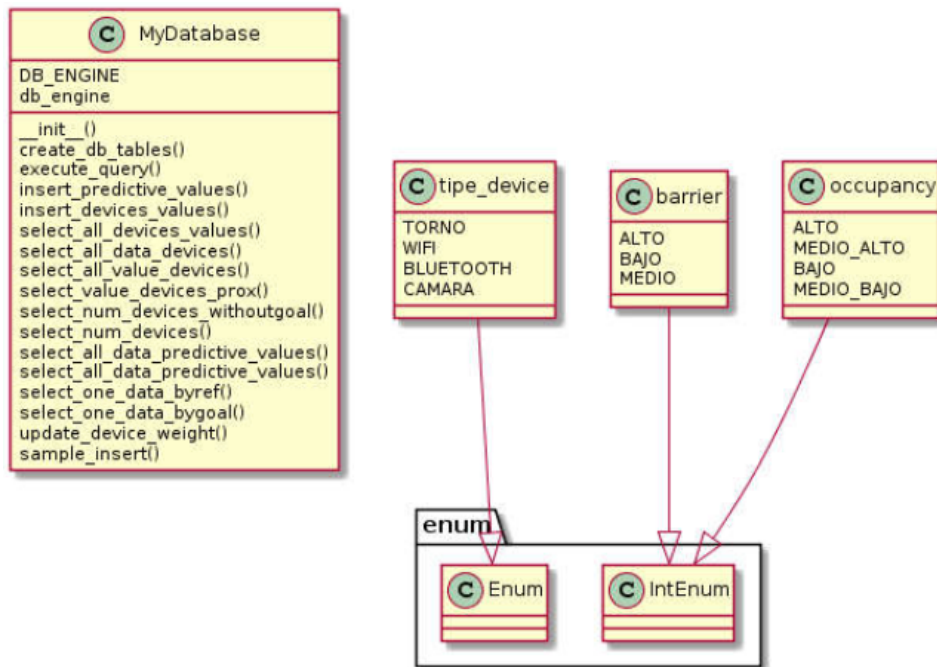


Figura 34 Esquema de clases DatabaseManager

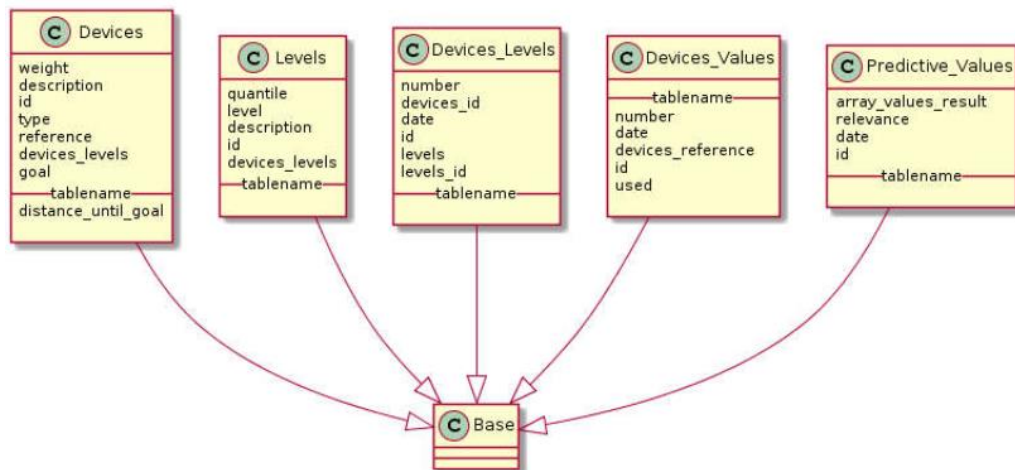


Figura 35 Esquema de clases de la Base de Datos

Las clases del diagrama de la Figura 35 muestra el total de las clases existentes en el módulo de DatabaseManager. Las clases son las siguientes:

- **DatabaseManager:** Esta es la clase principal de este módulo. Sirve para conectarse y realizar operaciones con la base de datos. Esta clase será accedida por todas las demás clases que necesiten acceder a la base de datos, realizando de esta manera la función de capa intermedia entre los servicios y la base de datos. Al utilizar un ORM, no es necesario conocer ni utilizar queries, que podrían correr el riesgo de ser fuertemente dependientes del tipo de base de datos que se fuese a utilizar.

Esta clase cuenta con el método de inserción de valores predeterminados en la base de datos, la primera vez que se ejecute el sistema se ejecutará este método y se creará una base de datos con unos valores iniciales para los periféricos y sus distintos niveles de ocupación, estos datos serán actualizados en cuanto el sistema comience a operar con datos reales.

- **TypeDevice:** Esta clase genera un tipo enumerado que nos permitirá trabajar de manera fácil con los tipos de dispositivos que existan en el sistema. Los tipos en el sistema desarrollado serán: WIFI, BLUETOOTH, CAMERA, TORNO.
- **Barrier:** Esta clase genera un tipo enumerado que servirá para indicar los mínimos de cada estado de ocupación. Estos índices mínimos son llamados en el sistema barreras (cuantiles), y cada vez que la ocupación sobrepase un nivel, será que ha evolucionado a un nuevo estado. Siempre existirán un número de niveles igual a ESTADOS-1, de tal manera que, como en el sistema que se va a desarrollar hay 4 estados, entonces dispondremos de 3 niveles que serán: HIGH, MEDIUM, LOW.
- **Occupancy:** Esta clase genera un tipo enumerado que servirá para indicar los estados de ocupación existentes en el sistema. Estos estados servirán para indicar la ocupación existente y dar una categorización al número de personas que están siendo detectadas en cada dispositivo objetivo.

De esta manera, se trabajará de manera fácil y sencilla la ocupación existente en el sistema mediante un número explícito y esperado de categorizaciones, en vez de por un formato numérico que pueda disponer de cualquier valor. En este sentido, la categorización es fundamental, ya que será la que determine las acciones a llevar a cabo en función del estado de ocupación en que se encuentre el sistema [50]. Si no existiese esta categorización, no sería posible realizar las acciones a llevar a cabo en función de la ocupación ya que no podríamos interconectar las funciones con las relaciones existentes, pues la categorización nos ayuda a trasladar la ingente cantidad de datos del mundo real a un sistema que permita tomar decisiones [51].

La ocupación será determinada para cada dispositivo [52] objetivo. Siempre existirán un número de estados igual a NIVELES+1, de tal manera que, como en el sistema que se va a desarrollar hay 3 niveles, entonces dispondremos de 4 estados que serán: HIGH, MEDIUM-HIGH, MEDIUM-LOW, LOW.

- **Devices:** Esta clase sirve para la generación y conexión del sistema central con la base de datos. Representa, mediante una serie de campos a un objeto de tipo Device. Existirá una fila en la tabla del sistema por cada dispositivo o sensor existente en él. En esta clase los parámetros más importantes son goal y distance_until_goal.

El primer parámetro indica al sistema que ese dispositivo será el que utilicemos como objetivo para entrenar al algoritmo de aprendizaje automático, es decir, que el estado

de ocupación del sistema se tomará en base a la ocupación que demuestren los otros dispositivos en relación al dispositivo objetivo, de esta manera conseguiremos que en el futuro, podamos adelantarnos al comportamiento solo con saber que está pasando en los otros dispositivos y que además, el dispositivo objetivo pueda ser eliminado en cualquier momento (estaciones sin barrera o ascensores que bajan en el momento oportuno son acciones que podrían llevarse a cabo, por ejemplo).

El segundo parámetro indicará la distancia, en segundos, que existe desde el dispositivo hasta el dispositivo objetivo. Este parámetro de distancia nos servirá para realizar distintos cálculos a la hora de normalizar la fecha y la hora a la que se estima que se toma la cata de los datos del dispositivo.

- **Levels:** Esta clase sirve para la generación y conexión del sistema central con la base de datos. Representa, mediante una serie de campos a un objeto de tipo Levels. Este objeto servirá para indicar los niveles de ocupación de cada dispositivo en función de los cuantiles que se determinen en el momento inicial. Al existir 3 niveles, se han determinado 3 cuantiles que serán: 0,25, 0,5, 0,75. Los cuantiles 0,25; 0,50 y 0,75 de la distribución normal -más conocidos como los cuantiles Q_1, Q_2 y Q_3-, dividen la distribución en cuatro bloques, cada uno de los cuales contiene el 25% de los datos.

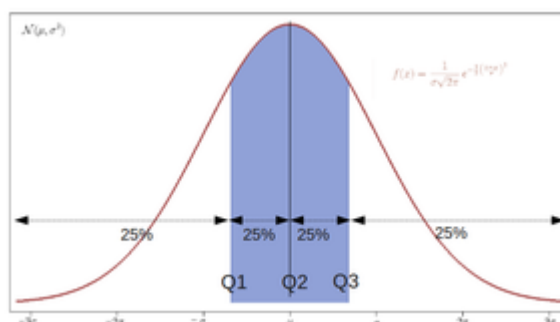


Figura 36 Gráfica cuantiles

Estos cuantiles nos servirán para indicar cuál es el valor que mejor se adapta a cada nivel. De esta manera, se puede determinar fácilmente los niveles de ocupación de un dispositivo, ya que al utilizar el valor de los cuantiles como mínimo de cada categoría, se puede ver si el valor actual sobrepasa la cantidad de personas determinadas por el cuantil o no. Gracias a este sistema, podemos fácilmente determinar los 4 estados de ocupación (HIGH, MEDIUM-HIGH, MEDIUM-LOW, LOW).

- **Devices_Levels:** Esta clase sirve para la generación y conexión del sistema central con la base de datos. Representa, mediante una serie de campos a un objeto de tipo Devices_Levels. El principal objetivo de esta clase es realizar una correlación entre los niveles del sistema y el número de personas que representaría ese nivel por dispositivo objetivo. Es decir, esta clase sirve para dar un número de personas concreto a los cuantiles determinados en el sistema para cada dispositivo, y así crear los mínimos que

luego servirán para determinar si las personas que están siendo detectadas por el dispositivo pertenecen a una ocupación u otra.

- **Devices_Values:** Esta clase sirve para la generación y conexión del sistema CORE con la base de datos. Representa, mediante una serie de campos a un objeto de tipo `Devices_Values`. Este objeto servirá para manejar los datos que actualmente están siendo enviados por los dispositivos fuente y almacenarlos en el sistema. Todas las peticiones con la ocupación correspondiente por parte de cada dispositivo serán almacenadas en la tabla correspondiente a esta clase. Los campos más relevantes de esta clase serán `number` y `used`.
 - `number`: será el campo en el que se almacenará el valor del número de personas que están siendo detectadas por el dispositivo. Este campo se utilizará para calcular la ocupación del sistema y también para realizar y mejorar el conocimiento del sistema.
 - `used`: será el campo que nos dirá si las ocupaciones guardadas ya han sido tenidas en cuenta para ser almacenadas en la tabla `Predictive_Values` y así realizar las operativas que nos van a indicar la ocupación del sistema. Gracias a este campo, el sistema será más rápido a la hora de calcular los valores a tener en cuenta para realizar los algoritmos ya que se evitará tener en cuenta resultados que ya han sido utilizados en el pasado reciente.
- **Predictive_Values:** Esta clase sirve para la generación y conexión del sistema central con la base de datos. Representa, mediante una serie de campos a un objeto de tipo `Predictive_Values`. Esta clase es fundamental dentro del sistema, ya que se encarga de operar con la tabla del mismo nombre y que se utiliza para el cálculo y el entrenamiento de los algoritmos de aprendizaje automático. En esta tabla se almacenarán los datos de manera no relacional, ya que es necesario tenerlos todos siempre en cuenta a la hora de realizar los cálculos para las predicciones y el entrenamiento, y por lo tanto la forma más rápida de extraer los datos siempre será con este modelo al evitar realizar consultas cruzadas entre diferentes tablas.

El campo más importante de esta clase será el campo `“array_values_result”` que almacenará en formato json todos los valores de la ocupación para cada dispositivo. Todos estos json serán utilizado por el sistema de inteligencia artificial para aprender sobre el sistema real sobre el que está operando y así realizar predicciones sobre lo que va a pasar dependiendo de estos valores.

Una vez que han sido explicadas todas las clases relacionadas con el sistema ORM utilizado, se va a pasar a ver el esquema generado de base de datos para comprender mejor las relaciones existentes entre las diferentes clases.

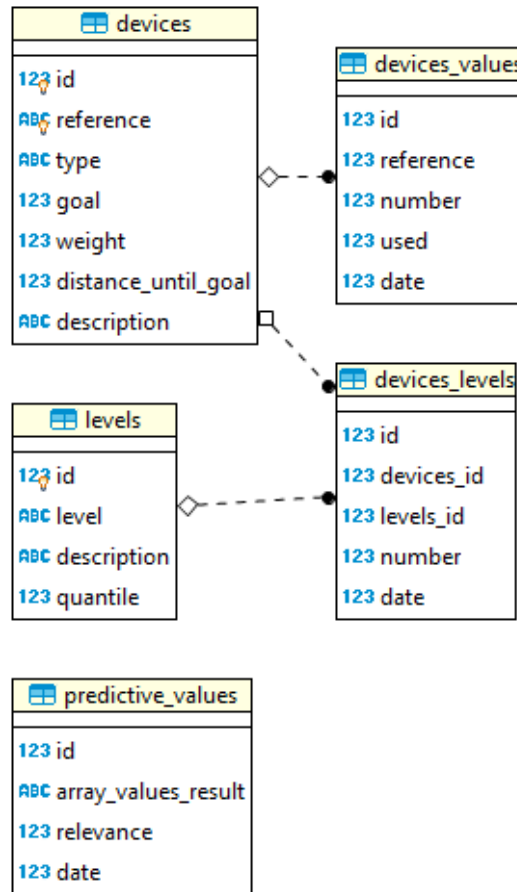


Figura 37 Diagrama de base de datos

Como se puede apreciar en el esquema entidad relación de la base de datos anterior, todas las tablas están desarrolladas bajo un esquema de relación menos la tabla de valores predictivos, que funciona de manera individual, sin tener en cuenta las relaciones anteriores.

Como se tiene un esquema, que se podría llamar mixto ya que existe una tabla que no tiene ninguna relación, se ha elegido como RDBMS (sistema de gestión de bases de datos relacionales) la base de datos SQLITE [53] que trabaja realmente bien en entornos con poca concurrencia de peticiones y también en entornos de análisis de datos, siendo muy flexible y que además, es una base de datos liviana y con una velocidad de inserción y extracción elevada. El problema de esta base de datos es que está almacenada directamente sobre un fichero de texto, y por lo tanto, en caso de un gran crecimiento de la base de datos (como sería en un entorno productivo) su rendimiento caería en picado teniendo que utilizar otras opciones, pero se ha determinado el uso de esta herramienta porque el sistema actualmente desarrollado está diseñado para ser ejecutado en un entorno controlado “de laboratorio” y por lo tanto se considera que flujo de datos será de pequeño tamaño.

En caso de un sistema productivo, se recomienda utilizar dos bases de datos diferentes. La primera base de datos sería una base de datos relacional para almacenar los valores de los dispositivos en tiempo real, que admita alta velocidad en las inserciones de datos, que es la

principal funcionalidad que se lleva a cabo en el sistema relacional. Y una segunda base de datos, NoSQL que almacenará los valores predictivos y que necesita un gran rendimiento en la extracción de datos para llevar a cabo las funcionalidades de los algoritmos predictivos y de aprendizaje automático.

La división del sistema en dos bases de datos diferentes dará más autonomía al sistema, ya que se podría aumentar la capacidad de cada una de ellas de manera individual realizando un manejo más exhaustivo de cada una de ellas. Además, gracias a que el sistema utiliza un ORM, el sistema es totalmente abstracto de las bases de datos que se van a utilizar, siendo ésta una característica fundamental en este tipo de framework.

8.1.3. Firebase module

El módulo de Firebase se encarga de conectar al sistema con el sistema de base de datos en la nube de Firebase, que se encargará de realizar las comunicaciones necesarias con los dispositivos de respuesta que se configuren.

Firebase [54] es un sistema de desarrollo móvil y web desarrollado en la nube de Google. Se trata de una plataforma de desarrollo disponible en diversas plataformas (entre ellas Android, iOS y web). Esta plataforma surgió para resolver el problema de la sincronización y el guardado de los datos en las plataformas Cloud, por ello Firebase provee una API para guardar y sincronizar datos en la nube en tiempo real.

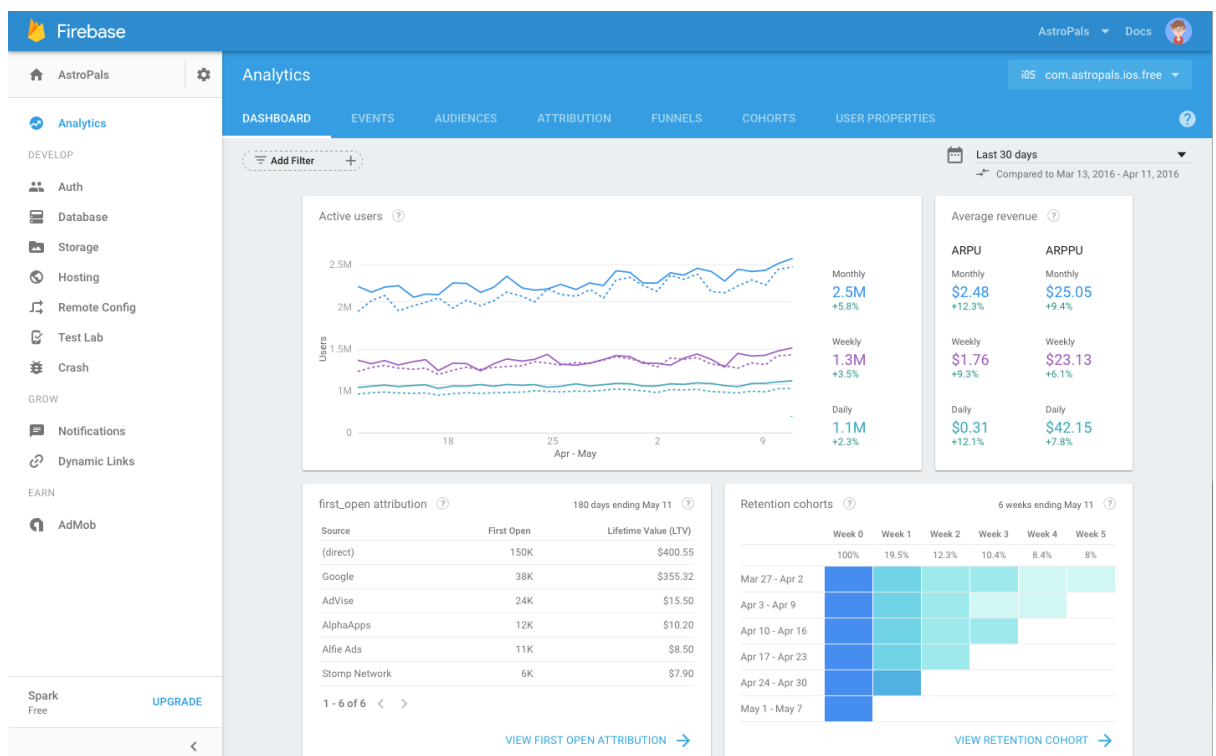


Figura 38 Dashboard Firebase

Gracias a que todo en la plataforma se realiza mediante una API bien documentada, las conexiones con distintos dispositivos son realmente fáciles, más cuando se trata de la sincronización de datos, que siempre ha sido una problemática en cuanto a los desarrollos de sistemas en tiempo real, debido en muchos casos a las pérdidas de rendimiento y seguridad que se obtienen al abrir buffers o canales de comunicación durante largos periodos. Sus principales características son:

- **Analíticas:** Provee al sistema de una solución gratuita para tener todo tipo de medidas (hasta 500 tipos de eventos), gestionando todo desde un único panel.
- **Desarrollo:** Permite la construcción de apps con mayor calidad, permite delegar determinadas operaciones en Firebase, para poder ahorrar tiempo, evitar bugs y obtener un aceptable nivel de calidad. Entre sus características destacan el almacenamiento, testeo, configuración remota, mensajería en la nube o autenticación, entre otras.
- **Crecimiento:** Permite gestionar los usuarios de las aplicaciones, pudiendo además captar nuevos. Para ello dispondremos de funcionalidades como las de invitaciones, indexación o notificaciones.
- **Monetización:** Permite ganar dinero gracias a su extensión AdMob.
- **Documentación detallada:** La calidad de su documentación y el detalle de ésta en todos y cada uno de sus servicios, hace que el uso de toda la plataforma sea simple y sencilla. Además, la comunidad de desarrolladores (<https://developers-latam.googleblog.com/2018/05/novedades-de-firebase-en-io-2018.html>) es amplia y siempre se puede encontrar alguna ayuda en la web (cuentan con más 1,2 millones de apps).
- **No es necesario administrar la infraestructura:** Toda la plataforma es autogestionada por google y no es necesario montar nada, todo lo dan montado ellos.
- **Plataforma integral:** contiene una gran cantidad de servicios y que dispone de una gran variedad de extensiones.
- **Pago por uso:** dispone de varios tipos de licenciamiento de pago por uso, entre ellos uno gratuito con diversos límites que en muchos casos puede ser de gran ayuda para comenzar y desarrollar aplicaciones en laboratorio.

Para el sistema desarrollado se ha utilizado una mínima potencia de todo el sistema pero que, sin embargo, es suficiente para cumplir con el objetivo marcado. Dentro de toda la plataforma de herramientas ofrecida por Firebase solo se ha utilizado el sistema de base de datos en tiempo real. Este sistema nos permite que todos los dispositivos de respuesta tengan comunicación directa y actualizada de los datos que se están almacenando en tiempo real.

Firestore Realtime Database [55] es una base de datos que está alojada en la nube. Los datos son almacenados en formato JSON y son sincronizados en tiempo real con todos los clientes que estén conectados. Cuando se compilan las aplicaciones multiplataforma con los SDK de iOS, Android, C++, Unity y JavaScript, todos los clientes compartirán la misma instancia de Realtime Database y recibirán las actualizaciones automáticamente con los datos en tiempo real actualizados. Firestore Database Realtime sirve para almacenar y sincronizar los datos de una base de datos NoSQL en la nube. Los datos se sincronizan con todos los clientes en tiempo real y se mantienen incluso cuando las aplicaciones no tienen conexión. Sus principales funciones son:

- **Tiempo real:** las solicitudes a la API se realizan mediante solicitudes HTTP. Cada vez que se inserta un nuevo JSON en la base de datos, los dispositivos conectados son avisados y se les envían los nuevos datos.
- **Sin conexión:** las aplicaciones conectadas continúan funcionando en modo offline. Por lo tanto, los usuarios pueden seguir teniendo una experiencia colaborativa agradable, cuando la conexión vuelva al dispositivo los datos serán actualizados con el estado actual del servidor. Cuando el dispositivo vuelve a conectarse, Realtime Database sincroniza los cambios de los datos locales con las actualizaciones remotas que ocurrieron mientras el cliente estuvo sin conexión, lo que combina los conflictos de forma automática.
- **Acceso desde dispositivos cliente:** Los datos pueden ser accedidos desde diversos dispositivos cliente, no hace falta disponer de ningún servidor de aplicaciones. La seguridad y validación de acceso se realiza mediante la configuración de reglas en el entorno Firestore.
- **Escalabilidad:** En caso de que el crecimiento de la base de datos aumentase se podría cambiar de licenciamiento y generar diversas instancias de la base de datos dentro del mismo proyecto. Así se podría seguir satisfaciendo las necesidades del proyecto de manera fácil y sencilla, sin la necesidad de tener que desarrollar nada nuevo.

El sistema de tiempo real utilizado es totalmente gratuito [56] mientras que los datos almacenados no superen 1 GB de almacenamiento, las operaciones de escritura de documentos no sean más de 20.000 al día, las de lectura 50.000 al día y las de eliminación sean 20.000 al día. En el caso del sistema actual, estos valores no se aproximan ni remotamente a la cantidad necesaria para tener una cuenta de pago, por lo que la gratuidad del sistema es algo que se ha tenido en cuenta para la elección de un sistema tan potente y fácil de utilizar.

De todos modos, el sistema está desarrollado para no utilizar demasiados datos en este ámbito, ya que se utilizan mucho más las bases de datos en local que la base de datos en la nube. Para conseguir este ahorro de datos, se ha conseguido que la base de datos en la nube solo tenga los datos necesarios para los sistemas periféricos de respuesta, es decir, el valor actual del sistema y un resumen del estado del proyecto en sí. Esta información es almacenada directamente sobre

dos únicos registros que serán actualizados en conveniencia cada vez que el sistema cambie. Este uso de la base de datos queda perfectamente demostrado en la imagen anterior, donde se muestran los dos únicos registros existentes en la base de datos y que el sistema irá actualizando.

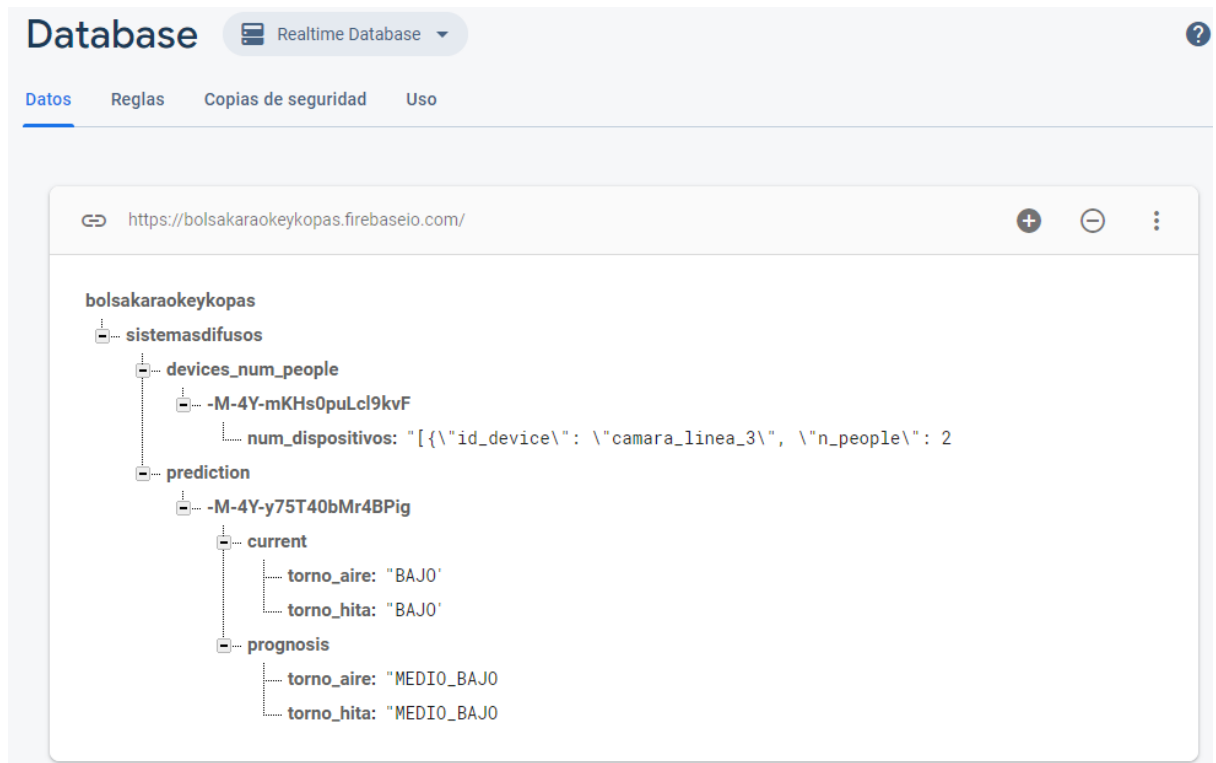


Figura 39 Database Firebase

Una vez explicada la potencialidad y el uso de la plataforma de Firebase y de su base de datos en tiempo real, se va a explicar de manera pormenorizada su uso dentro del sistema que desarrolla el Sistema Difuso en cuestión.

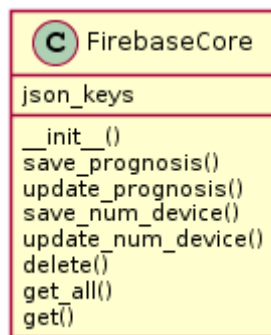


Figura 40 Clase FirebaseBaseCore

La clase `FirestoreCore` es la clase que va a comunicar el sistema con la base de datos Firebase. Esta clase utiliza el patrón Singleton para facilitar su uso dentro de la aplicación y también para ahorrar memoria, ya que con una sola instancia podemos encargarnos de realizar cualquier conexión y petición a la base de datos en la nube. Esta clase utiliza la librería desarrollada por

Firestore para permitir la conexión desde entornos Python a su plataforma. Es por ello que tiene como variable del módulo completo una instancia de Firestore que apunta a una URI que será la URI base en la que se encontrará el proyecto Firestore utilizado. La clase en cuestión contiene los siguientes métodos:

- `save_prognosis`: Éste método almacena mediante el método `post` de Firestore datos en la base de datos. Estos datos serán almacenados en la URI `"/sistemasdifusos/prediction"` ya que indicarán cual es el estado actual de ocupación del sistema. Este método será llamado la primera vez que se inserten datos en la base de datos desde la instancia de Firestore. Es importante anotar que si el sistema se detiene y se vuelve a arrancar, la instancia de Firestore cambiará y entonces almacenará otro registro en la base de datos de Firestore y será sobre este sobre el que trabaje la instancia.
- `update_prognosis`: Éste método actualiza mediante el método `put_async` de Firestore datos en la base de datos. Estos datos serán actualizados en la URI `"/sistemasdifusos/prediction"` ya que indicarán cual es el estado actual de ocupación del sistema y cuál será la predicción de lo que va a suceder en los momentos siguientes. Se utiliza el método asíncrono ya que gracias a él se consigue un mejor rendimiento en el sistema, ya que se tiene en cuenta que se pueden admitir pérdidas en alguna conexión y que la respuesta de la llamada no aporta ningún dato válido de peso que deba ser tenido en cuenta. Por lo tanto, se obtiene una considerable mejora en la velocidad de procesamiento a cambio de una pérdida del control sin importancia en el sistema.
- `save_num_device`: Éste método almacena mediante el método `post` de Firestore datos en la base de datos. Estos datos serán almacenados en la URI `"/sistemasdifusos/devices_num_people"` ya que indicarán cual es el estado actual del número de personas de cada dispositivo fuente. Este método será llamado la primera vez que se inserten datos en la base de datos desde la instancia de Firestore. Es importante anotar que si el sistema se detiene y se vuelve a arrancar, la instancia de Firestore cambiará y entonces almacenará otro registro en la base de datos de Firestore y será sobre este sobre el que trabaje la instancia.
- `update_num_device`: Éste método actualiza mediante el método `put_async` de Firestore datos en la base de datos. Estos datos serán actualizados en la URI `"/sistemasdifusos/devices_num_people"` ya que indicarán cual es el estado actual del número de personas de cada dispositivo fuente. Se utiliza el método asíncrono ya que gracias a él se consigue un mejor rendimiento en el sistema, ya que se tiene en cuenta que se pueden admitir pérdidas en alguna conexión y que la respuesta de la llamada no aporta ningún dato válido de peso que deba ser tenido en cuenta. Por lo tanto, se obtiene una considerable mejora en la velocidad de procesamiento a cambio de una pérdida del control sin importancia en el sistema.

- delete: Éste método borraría los datos existentes en la base de datos Firebase relacionados con las personas que están siendo detectadas por los dispositivos actualmente. Por lo tanto, estos datos dejarían de enviarse a los dispositivos periféricos de respuesta.
- get_all: Éste método obtendría todos los datos existentes en la base de datos Firebase relacionados con las personas que están siendo detectadas por los dispositivos actualmente
- get: Éste método obtendría un json dependiendo de su clave con los datos existentes en la base de datos Firebase relacionados con las personas que están siendo detectadas por los dispositivos actualmente

La clase de conexión con Firebase se comunica principalmente con el Schedule module, ya que este es el motor que irá actualizando los datos en los dispositivos de respuesta dependiendo del tiempo en el que se configuren sus acciones. Para ello, se aprovechará la funcionalidad de realizar el pronóstico de ocupación global en el sistema, para enviar el valor resultante de esta y también los valores del sistema en ese momento a la base de datos en la nube, que actualizará en ese mismo instante los valores en los dispositivos de respuesta que estén conectados a Firebase.

8.1.4. Schedule module e aprendizaje automático.

Este módulo realiza el grueso de las operaciones que se llevan a cabo en el sistema. Es un módulo tan complejo como fundamental en el sistema, ya que su labor es realizar los continuos chequeos que actualizan los valores en el sistema. Por lo tanto, se ocupa tanto de saber la ocupación real del sistema en el momento, como de realizar la ponderación en cuartiles de los datos asociados a cada dispositivo, y también, de realizar el proceso de aprendizaje en el sistema.

Es por ello, que las operaciones realizadas en este módulo, aunque son dispares entre sí, sirven para darle un sentido total al sistema y, de esta forma, poder mostrar resultados coherentes mediante las estimaciones realizadas.

Teniendo en cuenta la complejidad de estos algoritmos, se hace necesario estipular una programación de ejecución de cada flujo para que las operaciones respeten un orden y no se mezclen entre sí, problema que puede originar cierta inestabilidad en los datos ofrecidos. Esta inestabilidad es debida, principalmente, a la complejidad de la algoritmia que se está ejecutando y a la imposibilidad de realizarla de manera constante, ya que en ese caso, ralentizará el sistema de manera crítica, pudiendo saturar la memoria del dispositivo por la cantidad de cálculos numéricos que debe realizar el sistema de aprendizaje automático. Por todo esto, en las próximas páginas se explicarán cuándo y por qué se recomiendan la ejecución de los distintos flujos dentro del sistema, si bien son solo una serie de recomendaciones, ya que el sistema

puede proveerse de una máquina mucho más potente que resuelva los cálculos en menor tiempo, pero entonces, estaríamos rompiendo con una de las máximas del proyecto, que es la de minimizar el coste económico de su implantación.

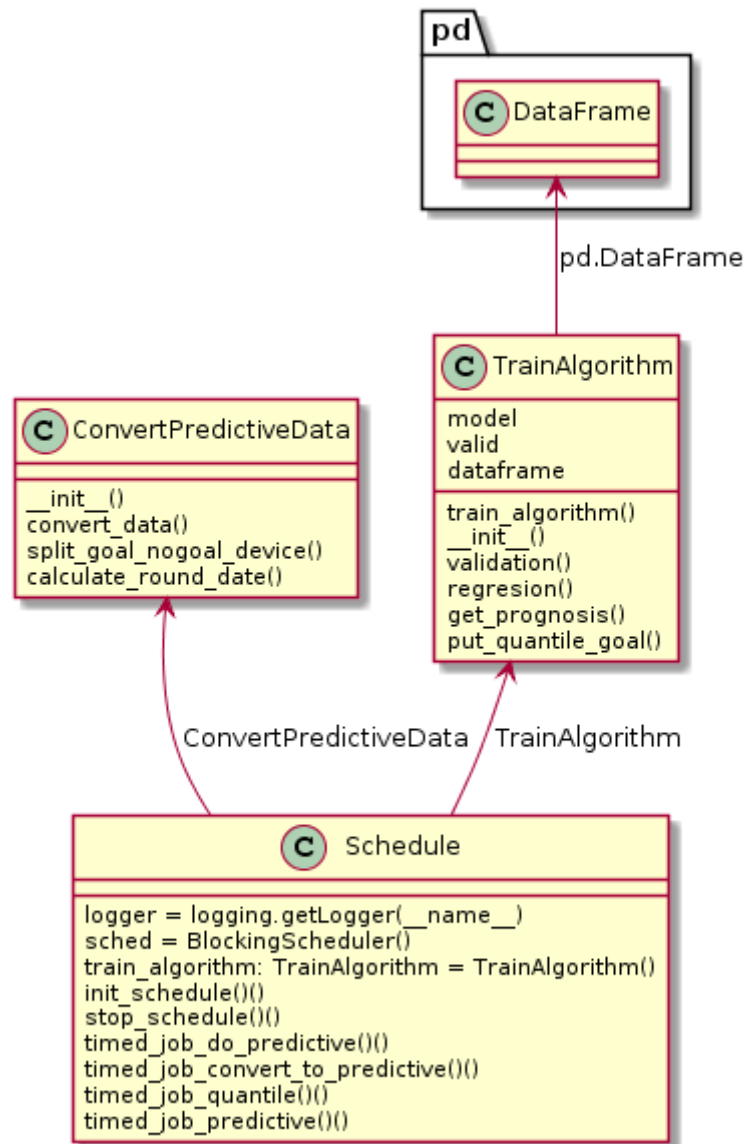


Figura 41 Diagrama de clases Schedule y Aprendizaje Automático

En la imagen superior podemos ver las relaciones que existen en el módulo Schedule. Este módulo se compone principalmente de otros dos submódulos que serán llamados por las tareas que se ejecutan en el módulo principal y realizarán los cálculos que necesiten. Estos módulos debido a su complejidad serán explicados individualmente y en mayor profundidad en las siguientes secciones.

Schedule module

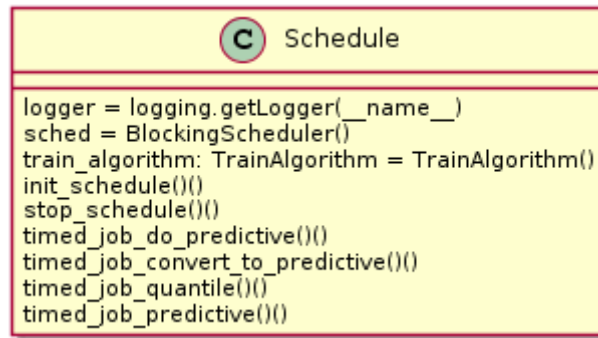


Figura 42 Diagrama de clase ScheduleModule

La Figura 42 muestra el módulo Schedule basado en la librería BlockingScheduler [57] cuya complejidad en el desarrollo es baja, ya que utiliza decoradores de clase y estos son fáciles de utilizar, además aporta concurrencia a las acciones que está ejecutando y que realiza una serie de tareas que son programadas en función del aporte y la complejidad de éstas al sistema por el desarrollador, pero que en caso de ser necesarias, pueden ser cambiadas para ajustarse al sistema real. Los métodos de esta clase son:

- `init_schedule`: este método es utilizado para iniciar los procesos cronológicos, por defecto, cuando el sistema arranca, los procesos cronológicos no están iniciados y no es hasta que no se realice la llamada correspondiente a la API `"/state/start"` cuando los sistemas cronológicos empezarán a ejecutarse en el orden correspondiente.
- `stop_schedule`: este método es utilizado para detener los procesos cronológicos cuando se realice la llamada correspondiente a la API `"/state/stop"`. Una vez se realice la llamada, los sistemas cronológicos dejarán de ejecutarse.
- `timed_job_do_predictive`: esta función será la función encargada de determinar qué es lo que va a ocurrir en el sistema dependiendo de los valores que existan en la base de datos y que provienen de los dispositivos que proveen los datos de las personas que están detectando en el momento. Se ejecutará en intervalos de 30 segundos, aunque realmente puede aumentarse el tiempo o disminuirse dependiendo de la configuración de catas de ocupación de los distintos periféricos fuente que existan en el sistema. No tendría sentido que los periféricos envíen su información cada 2 minutos y que el sistema intentase actualizar sus valores cada 30 segundos, ya que solo se podrían ver resultados diferentes en 1 de cada 4 ocupaciones.

Este algoritmo obtiene los valores de todos los dispositivos que no son del tipo "goal" (objetivo). De todos los valores obtenidos se obtiene su fecha y la distancia que existen entre ellos y el dispositivo objetivo, para así normalizar las horas y hacer una extrapolación para saber a qué hora llegaron esas personas hasta el dispositivo objetivo. Una vez se ha realizado esta predicción, se lleva a cabo un filtro que nos permite recoger los últimos valores de cada dispositivo asociados a una fecha en el tiempo

correspondiente (podría ser la actual, o una fecha en un futuro inmediato). Una vez obtenidos los últimos valores de cada dispositivo más próximos a una fecha se genera una estructura de datos que será enviada al motor de aprendizaje automático para que pronostique cual será el valor correspondiente en el dispositivo objetivo.

Una vez acabado todo el proceso anterior, los datos del pronóstico serán enviados mediante el módulo de Firebase a la base de datos en el cloud, junto con los valores de las personas detectadas por cada dispositivo y el valor actual de las personas detectadas por el dispositivo objetivo. Con estos tres valores, el dispositivo de respuesta puede, además de mostrar los valores, mostrar si el flujo de personas va a aumentar o disminuir en un futuro cercano.

- `timed_job_convert_to_predictive`: esta función se encargará de convertir los datos de la ocupación en cada dispositivo en unos datos válidos para poder ser almacenados en la base de datos que utilizará el sistema de aprendizaje automático para aprender. La característica principal de estos datos es que tienen que estar en formato json, además su clave tiene que ser el identificador del dispositivo y su valor debe de ser el número de personas que ha detectado el dispositivo en ese momento. Con este formato, podremos insertar directamente información al sistema de aprendizaje automático para que realizando ciertas operaciones matemáticas pueda ofrecer un pronóstico adecuado. Además, una vez utilizada la información de los dispositivos esta información se marca en estado “usada”, ya que así tenemos un filtro simple, para hacer las queries más veloces, ya que no se buscarán en la base de datos valores que ya han sido tenidos en cuenta para ser insertados en la tabla que se usa para realizar los cálculos de aprendizaje automático.

Se ejecutará en intervalos de 30 minutos, aunque realmente puede aumentarse el tiempo o disminuirse dependiendo de la configuración de catas de ocupación de los distintos periféricos fuente que existan en el sistema. No tendría sentido que los periféricos envíen su información cada 2 minutos y que el sistema intentase actualizar sus valores cada 30 segundos, ya que solo se añadirían datos a la tabla `Devices_Values` en 1 de cada 4 ocupaciones, por lo que cuando se ejecutase esta funcionalidad existirían muy pocos datos para validar. Lo ideal es encontrar una buena combinación entre el número de datos nuevos insertados y la ejecución del algoritmo para ganar eficiencia.

- `timed_job_quantile`: Esta función realiza el trabajo de actualizar los niveles mínimos para pasar de una categoría de ocupación a otra dependiendo de los números de personas que existan para cada dispositivo objetivo en la base de datos, con estos valores, realiza un algoritmo matemático que ayuda al sistema a categorizar los datos existentes para ese dispositivo de manera fácil y sencilla.

La función para determinar los cuantiles se ejecutará una vez al día, aunque realmente puede aumentarse el tiempo o disminuirse dependiendo de la configuración de catas de ocupación de los distintos periféricos fuente que existan en el sistema. Este sistema

utiliza todos los valores de la tabla `Devices_Values` por ello puede llegar a ser un proceso lento, por lo que se recomienda no ejecutarlo en momentos de alta carga en el entorno, sino en momentos donde su uso esté siendo menor (como puede ser el horario nocturno).

- `timed_job_predictive`: Esta función realiza el entrenamiento del algoritmo predictivo. Cada vez que es llamada recoge todos los datos disponibles en la tabla `Predictive_Values` y los utiliza para aprender sobre el sistema y así poder realizar predicciones sobre lo que va a pasar en base a lo que pasó en el futuro. Esta función no devolverá ningún resultado, solo entrenará al algoritmo con más datos cada vez que se ejecute para que las demás funciones puedan utilizar ese aprendizaje para mostrar sus resultados a los dispositivos de respuesta.

Se ejecutará una vez al día, aunque realmente puede aumentarse el tiempo o disminuirse dependiendo de la configuración y de la potencia del sistema en el que se vaya a ejecutar. Este sistema utiliza todos los valores de la tabla `Predictive_Values` realizando sobre ella procesos matemáticos complejos por ello puede llegar a ser un proceso lento, por lo que se recomienda no ejecutarlo en momentos de alta carga en el entorno, sino en momentos donde su uso esté siendo menor (como puede ser el horario nocturno en cierto tipo de empresas).

Train Algorithm module

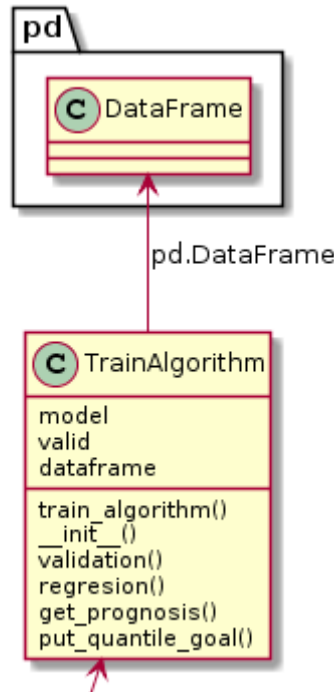


Figura 43 Diagrama de clases `TrainAlgorithm`

La imagen anterior muestra el módulo `TranAlgorithm` que es el lugar donde se realizan los algoritmos más pesados y que contiene todo el sistema de aprendizaje automático. El aprendizaje automático es donde las habilidades computacionales y algorítmicas de la ciencia

de datos satisfacen el pensamiento estadístico de la ciencia de datos, y el resultado es una colección de enfoques de inferencia y exploración de datos que no son tanto de teoría efectiva como de computación efectiva.

Para poder utilizar sistemas de aprendizaje automático en Python, se han utilizado algunas de las librerías más potentes que existen en este momento, que son [35] DataFrame de Pandas y Scikit-Learn las cuales son las librerías por excelencia dentro del lenguaje de programación Python para desarrollos científicos.

Pandas

En el nivel más básico, los objetos Pandas pueden ser considerados como versiones mejoradas de la librería NumPy, matrices estructuradas en las que las filas y columnas se identifican con etiquetas en lugar de simples índices enteros. Pandas proporciona una gran cantidad de herramientas útiles, métodos y funcionalidades además de las básicas estructuras de datos. Principalmente, de todas las posibilidades de la librería, se utiliza DataFrame, que es el objeto con el cual generamos las tablas correspondientes a partir de los json generados y almacenados en la tabla Predictive_Values.

DataFrame puede ser considerado como una generalización de una matriz NumPy, o como una especialización de un diccionario Python. Un DataFrame es un análogo de una matriz bidimensional con índices de filas flexibles y nombres de columnas flexibles. Por lo tanto, se puede pensar en una matriz bidimensional como una secuencia ordenada de columnas unidimensionales alineadas, se puede pensar en un DataFrame como una secuencia de objetos alineados en serie. Aquí, por "alineados" queremos decir que comparten el mismo índice. Gracias a esta clase, conseguimos tratar de manera fácil y sencilla el array de json con las referencias de los dispositivos y la ocupación de éstos para generar una matriz con el que podemos trabajar fácilmente.

Scikit-learn

Existen varias librerías en Python que proporcionan una gran gama de implementaciones para resolver algoritmos de aprendizaje automático. Probablemente la más famosa sea Scikit-Learn [58], un paquete que proporciona versiones eficientes de un gran número de algoritmos comunes. Scikit-Learn se caracteriza por una API fácil, sencilla de entender y racionalizada, así como por una documentación en línea muy útil y completa. Una de las ventajas de esta facilidad es que una vez que una persona con un conocimiento básico en programación y matemáticas quiere desarrollar algoritmos de aprendizaje, puede mediante el uso básico y la sintaxis de Scikit-Learn generar y entrenar un modelo. Algunas de las principales características de esta librería son:

- Coherencia: Todos los objetos comparten una interfaz común extraída de un conjunto limitado de métodos, con una documentación coherente.

- Inspección: Todos los valores de parámetros especificados se exponen como atributos públicos.
- Jerarquía de objetos: Sólo los algoritmos están representados por las clases Python; los conjuntos de datos están representados en formatos estándar (matrices NumPy, Pandas DataFrames, matrices dispersas de SciPy) y los nombres de los parámetros utilizan sintaxis Python estándar.
- Composición: Muchas tareas de aprendizaje de la máquina pueden ser expresadas como secuencias de algoritmos más fundamentales, y Scikit-Learn hace uso de esto siempre que es posible.
- Valores predeterminados razonables: Cuando los modelos requieren parámetros especificados por el usuario, la biblioteca define un valor por defecto apropiado. Por lo que, de una forma estándar se logra un resultado razonable que puede ser depurado en momentos posteriores.

Lo más importante, y una de las características principales de la librería es que todo debe desarrollarse en términos de tabla. Por este motivo, la base de datos Predictive_Values almacena los datos en formato json, para que mediante un array de éstos se puedan generar, de manera sencilla, tablas con las que trabajar. La librería al cargar el conjunto de datos hará que, cada clave del json sea una columna y el valor el resultado para esa columna. Cada columna se denominará característica, y dependiendo del número de columnas tendremos n-características en el sistema. Es para esta transformación para la que se utiliza la librería Panda.

Una vez explicadas las librerías utilizadas para realizar todo el sistema de aprendizaje automático, solo queda decir como resumen que la regresión logística es una técnica de aprendizaje supervisado para clasificación, que es muy usada en muchas industrias debido a su escalabilidad y por su facilidad a la hora de ser explicada.

Los métodos de la clase TrainAlgorithm son:

- `train_algorithm`: este método sirve para entrenar el modelo de regresión que será utilizado para realizar predicciones sobre lo que va a ocurrir con los nuevos datos que se inserten. El método seleccionará de la tabla Predictive_Values todas las filas que se encuentren en ella y a partir de ahí generará una matriz de datos y un array de datos (la primera con los datos de los dispositivos no objetivo y el segundo será el array de destino con los datos de los dispositivos objetivos) con la cual se entrenará el modelo que servirá de base para las predicciones. Es importante conocer, que este método tiene en cuenta la calidad del modelo para ser tenido en cuenta o no a la hora de realizar predicciones. Esto es así, porque no tendría sentido tener un modelo con una cantidad de datos mínima y con alta probabilidad de dar una respuesta errónea, y sin embargo, estar enviando señales de predicción a los dispositivos de salida que se encuentren en el sistema.

- **regresion:** la función principal de este método es realizar el modelo a partir de la matriz que ha sido generada anteriormente con los datos de los dispositivos fuente no objetivo y del array de destino generado a partir de los datos de los dispositivos objetivos. Con estos datos, se realiza el entrenamiento del modelo mediante la función “fit”.

Una vez generado el modelo, y para comprobar su certeza, se le realizan una serie de comprobaciones que se basan en insertar al modelo una parte de los mismos valores con los que se entrenó y ver el resultado que estos nos devuelven a la hora de realizar la predicción. Si la puntuación está por encima de la barrera que se configure (85% o más a ser posible), consideraremos que el algoritmo en una primera fase es válido, pero para estar seguro de su utilidad se le aplicarán otra serie de operaciones matemáticas que nos asegura que, además de válido, el modelo es óptimo y de calidad, por ello utilizaremos una serie de funciones que se describirán en el método de “validation”.

Este método además ofrece, mediante logs, los coeficientes que el modelo ha asignado a cada característica del sistema, es decir, la importancia de cada dispositivo a la hora de realizar el modelo. Esto es muy importante a la hora de mejorar el sistema en el mundo real, porque permitirá detectar dispositivos que no están aportando ninguna información válida en el lugar donde están posicionados, y puedan ser eliminados o reposicionados para permitir obtener mejores valores, en definitiva permite realizar una optimización del sistema muy útil.

- **validation:** la función principal de este método es realizar una validación del modelo en profundidad para garantizar la integridad y la calidad de los datos. Éste método además, muestra información al sistema mediante una serie de logs que muestran la calidad y los resultados de las predicciones que se van a realizar con este modelo. Con estos datos, y algunos de los que se han descrito anteriormente se puede rebalancear el algoritmo y así descubrir y optimizar mejores combinaciones para conseguir un resultado más realista o que más se aproxime a la realidad o a las necesidades del sistema en cada caso.

Lo primero que se realizará será la división del conjunto de datos entre un conjunto de entrenamiento y un conjunto de prueba. Se utilizará alrededor de un 80% de datos para entrenar el modelo y un 20% como datos de prueba. Utilizando estos datos, se realiza una validación cruzada teniendo en cuenta el modelo actual.

- **put_quantile_goal:** este método es el encargado de calcular niveles mínimos de ocupación para saber qué número de personas corresponde a cada categoría. Para ello, utiliza un objeto Dataframe con todos los datos de la ocupación del dispositivo objetivo y realiza la función “quantile” con el cuantil asignado al dispositivo en la base de datos. La función “quantile” devolverá un valor que será el almacenado para ese dispositivo y para ese cuantil en la base de datos.

- `get_prognosis`: Este método, es el encargado de realizar la prognosis en el sistema. Gracias a él, se obtendrá un pronóstico de lo que va a ocurrir en el sistema dependiendo de los datos que le enviemos. Para realizar la predicción, le enviaremos los datos de todos los dispositivos no objetivo y, con el modelo de aprendizaje automático activo y validado por las funciones anteriores, se obtendrá el valor de lo que va a pasar en el sistema. Si se tuviese un sistema multiobjetivo, se obtendría la predicción para cada objetivo, lo cual, como se comentó anteriormente, puede aportar al sistema una complejidad y una potencialidad muy interesante para continuar el trabajo realizado por este sistema.

Convert Predictive module

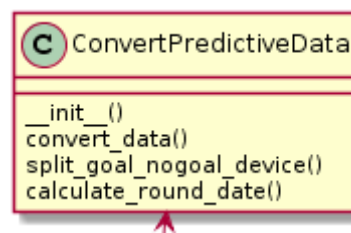


Figura 44 Diagrama de clases *ConvertPredictiveData*

La Figura 44 muestra el módulo `ConvertPredictiveData` que es el lugar donde se realizan los algoritmos para transformar los datos de los dispositivos en tiempo real del modelo relacional a json para ser insertados en la tabla que mantendrá la información que servirá para enseñar al modelo predictivo de inteligencia artificial.

El módulo obtendrá todos los datos en tiempo real de los dispositivos y los relacionará en función del dispositivo objetivo más cercano que haya emitido datos en el tiempo, para ello normalizará las fechas de los dispositivos no objetivo y calculará cuales están en el rango de influencia del dispositivo objetivo correspondiente.

Los métodos de la clase `ConvertPredictiveData` son:

- `convert_data`: este método es el método principal de la clase. En él, se obtienen todos los datos que no se hayan transformado ya en predictivos y se dividen los datos en dos matrices diferentes, una para los dispositivos objetivos y otro para los no objetivos. Una vez que tenemos los datos divididos en dos matrices y los valores no objetivo normalizados, se procede a buscar las relaciones entre cada dispositivo objetivo los dispositivos objetivos que están en su zona de actuación horaria. Lo ideal es que el tiempo entre dispositivos no objetivo y objetivo, una vez normalizadas las horas, fuese 0, pero se admiten unos pequeños retrasos para el caso en que no encontrase dispositivos en la zona horaria. Este proceso culminará con la matriz de dispositivos objetivo rellena de dispositivos objetivos relacionados con los datos de cada dispositivo no objetivo cercano. Una vez tenemos esta matriz, se procede a pasar los datos a json y

a insertarlos en la tabla de predicción para que los algoritmos de inteligencia artificial puedan mejorar su aprendizaje gracias a los nuevos datos insertados.

- `split_goal_nogoal_device`: esta función es la encargada de separar los datos que provienen de la tabla de valores reales de los dispositivos entre dispositivos objetivo y no objetivo. En caso de que los dispositivos sean no objetivo, entonces se obtiene el valor en el que se insertó ese dato en el sistema y se le suma el tiempo que se le asignó de distancia al objetivo. Este paso es el que denominaremos normalización, ya que lo que hará es llevar la fecha en la que se insertó el dato real a la fecha en la que las personas detectadas por el dispositivo llegarán al dispositivo objetivo correspondiente. Una vez obtenida esta normalización, entonces se podrá comparar qué personas se detectaron por el dispositivo no objetivo y cuáles fueron las detectadas por el dispositivo objetivo. Al final de esta función se obtendrán dos matrices diferentes, una con los datos de los dispositivos objetivo y otra con los datos de los dispositivos no objetivos.
- `calculate_round_date`: la función principal de esta función es unir los datos de la matriz de dispositivos objetivo con la matriz de dispositivos no objetivos. Para ello, seleccionará cada dispositivo no objetivo y mirará los dispositivos no objetivo que más se aproximen a su fecha de inserción en el sistema (que ya ha sido normalizada), entonces, si el resultado de la comparación de fechas es menor de X segundos, se considerará que los datos del dispositivo no objetivo seleccionado están relacionados con los datos del dispositivo objetivo en ese momento. Una vez que el dispositivo objetivo tiene un dato de cada tipo de sensor del sistema, se pasará al siguiente dispositivo objetivo, y en caso de que no se lleguen a completar los datos de los dispositivos no objetivo para el dispositivo objetivo entonces ese resultado no se tendrá en cuenta.