

# Memoria del proyecto

*Sistema para el control de aforo seguro en  
establecimientos.*

Trabajo de Fin de Grado

INGENIERÍA INFORMÁTICA



**VNIVERSIDAD  
D SALAMANCA**

CAMPUS DE EXCELENCIA INTERNACIONAL

**Agosto de 2021**

**Autor:**

*Alfonso Hernández Noriega*

**Tutores:**

*Gabriel Villarrubia González*

*André Sales Mendes*

*Juan Francisco de Paz Santana*

## Memoria del proyecto

## Certificado de los tutores

---

D. Gabriel Villarrubia González, D. André Filipe Sales Mendes y D. Juan Francisco De Paz Santana, profesores del Departamento de Informática y Automática de la Universidad de Salamanca.

HACEN CONSTAR:

Que el trabajo titulado “*Sistema para el control de aforo seguro en establecimientos*” ha sido realizado por D. Alfonso Hernández Noriega, con el número de documento 70881689-J y constituye la memoria del trabajo realizado para la superación de la asignatura Trabajo de Fin de Grado de la Titulación Grado de Ingeniería Informática de esta Universidad.

Y para que así conste a todos los efectos oportunos.

En Salamanca, a 4 de septiembre de 2021

## Memoria del proyecto

### Resumen

---

El proyecto realizado como Trabajo de Fin de Grado consiste en el diseño de un sistema para controlar el aforo de clientes en establecimientos. Este proyecto se lleva a cabo contextualizado en la actual situación de pandemia, provocada por la COVID-19 que ha precisado establecer una serie de nuevas normas en las interacciones sociales, de entre las que destaca el distanciamiento social. Para garantizar este distanciamiento social en los establecimientos cerrados es preciso controlar el aforo de dicho establecimiento, tomándose como referentes en este ámbito la combinación de los sistemas de control de aforo previos de que se dispone y las estrategias para la detección de las mascarillas.

Dentro de este marco conceptual, ha sido posible establecer el siguiente objetivo de trabajo: “diseñar y construir una aplicación que monitorice, mediante cámaras, los clientes que entran y salen de un establecimiento y, de esta manera, poder llevar un control de aforo. También debe detectar cuáles de los clientes llevan mascarilla puesta y cuáles no. Por último, el sistema debe ofrecer un portal donde los usuarios puedan gestionar y ver el estado de sus edificios, junto con su información relacionada”.

Para ello, se definen aquellas técnicas y herramientas necesarias tanto desde el enfoque de la aplicación web diseñada, como en lo que respecta al servidor cámara y a las herramientas CASE utilizadas, así como a las herramientas de documentación que permiten recuperar la información necesaria para presentar este informe. Además del manual del sistema que se presenta en esta memoria, y la reflexión acerca de los puntos más relevantes del desarrollo, las limitaciones y vías de trabajo futuro, en este documento se exponen aquellas conclusiones al proyecto y se anexa la información relativa a la planificación temporal del proyecto, la especificación de requisitos, el análisis de requisitos, el diseño del sistema de software, la documentación técnica y el manual de usuario.

**Palabras clave:** control aforo, aforo seguro, COVID-19, pandemia, reconocimiento facial, mascarilla.

## Summary

---

The project made as a Final Degree Project consists of the design of a system to control the capacity of customers in establishments. This project is carried out in the context of the current pandemic situation, caused by COVID-19, which has made it necessary to establish a series of new rules in social interactions, among which social distancing stands out. In order to guarantee this social distancing in closed establishments, it is necessary to control the capacity of the establishment, taking as references in this field the combination of the previous capacity control systems available and the strategies for the detection of masks.

Within this conceptual framework, it has been possible to establish the following work objective: "to design and build an application that monitors, by means of cameras, the customers entering and leaving an establishment and, in this way, to be able to carry out a capacity control. It must also detect which customers are wearing masks and which are not. Finally, the system should provide a portal where users can manage and view the status of their buildings and related information".

For this purpose, the necessary techniques and tools are defined both from the approach of the designed web application, the camera server and the CASE tools used, as well as the documentation tools that allow retrieving the necessary information to present this report. In addition to the system manual presented in this report, and the reflection on the most relevant points of the development, the limitations and ways of future work, this document presents the conclusions of the project and annexes the information related to the project time planning, the requirements specification, the requirements analysis, the design of the software system, the technical documentation and the user manual.

**Key words:** capacity control, secure capacity, COVID-19, pandemic, facial recognition, face mask.

# Tabla de contenido

---

1.	Introducción	1
2.	Estado del arte	3
2.1	Sistemas de control de aforo	3
2.2	Sistemas de detección de mascarillas	5
3.	Objetivos	7
3.1	Objetivos del sistema	7
3.2	Objetivos personales	8
4.	Conceptos teóricos	9
5.	Técnicas y herramientas	11
5.1	Aplicación Web	11
5.1.1	HTML y CSS	11
5.1.2	JavaScript	12
5.1.3	Vue	12
5.1.4	Google Maps Platform	13
5.1.5	PHP	13
5.1.6	Laravel Framework	14
5.1.7	Plantilla AdminLTE	17
5.1.8	Apache	17
5.1.9	MySQL	17
5.1.10	PHPStorm	18
5.2	Servidor Cámara	19
5.2.1	Python	19
5.2.2	OpenCV	20
5.2.3	Tensorflow	21
5.2.4	Keras	22
5.2.5	PyCharm	22
5.3	Herramientas CASE	23
5.3.1	Visual Paradigm	23
5.3.2	Microsoft Project	23
5.4	Herramientas de documentación	23
6.	Aspectos relevantes del desarrollo	25
6.1	Marco de trabajo	25
6.2	Planificación temporal	26

## Memoria del proyecto

<b>6.3</b>	<b>Especificación de requisitos</b>	<b>29</b>
6.3.1	Participantes	29
6.3.2	Objetivos del sistema	30
6.3.3	Requisitos de información	30
6.3.4	Requisitos funcionales	31
6.3.5	Requisitos no funcionales	33
<b>6.4</b>	<b>Análisis de requisitos</b>	<b>34</b>
6.4.1	Modelo de dominio	34
6.4.2	Realización de casos de uso - Análisis	35
6.4.3	Clases de análisis	35
6.4.4	Vista arquitectónica	36
<b>6.5</b>	<b>Diseño del sistema</b>	<b>36</b>
6.5.1	Patrones arquitectónicos	36
6.5.2	Subsistemas de diseño	37
6.5.3	Clases de diseño	38
6.5.4	Vista arquitectónica del modelo de diseño	39
6.5.5	Realización de casos de uso diseño	40
6.5.6	Diseño de la base de datos	41
6.5.7	Modelo de despliegue	42
<b>6.6</b>	<b>Implementación</b>	<b>43</b>
<b>6.7</b>	<b>Pruebas</b>	<b>44</b>
<b>6.8</b>	<b>Funcionalidad del sistema</b>	<b>45</b>
6.8.1	Usuario No Identificado	45
6.8.1.1	Acceder a la página pública	45
6.8.1.2	Iniciar sesión	46
6.8.2	Usuario Identificado	47
6.8.2.1	Página privada de bienvenida	48
6.8.2.2	Ver perfil	48
6.8.2.3	Listado de edificios	49
6.8.2.4	Añadir un edificio	50
6.8.2.5	Comparar edificios	53
6.8.2.6	Ver estado de un edificio	54
6.8.2.7	Editar edificio	55
6.8.2.8	Listado de puertas	56
6.8.2.9	Listado de incidencias	57
6.8.2.10	Edición del semáforo	58



## Memoria del proyecto

<b>6.8.3</b>	<b>Usuario Administrador</b>	59
<b>6.8.3.1</b>	<b>Modo Administrador</b>	59
<b>6.8.3.2</b>	<b>Listado de usuarios</b>	60
<b>6.8.3.3</b>	<b>Editar usuario</b>	60
<b>7.</b>	<b>Limitaciones del prototipo</b>	61
<b>8.</b>	<b>Conclusiones y líneas de trabajo futuras</b>	62
<b>8.1</b>	<b>Conclusiones</b>	62
<b>8.2</b>	<b>Líneas de trabajo futuras</b>	63
<b>9.</b>	<b>Referencias</b>	64

## Índice de figuras

---

Figura 1. Propuesta para el control del aforo en un establecimiento. Fuente: Aforoseguro.	4
Figura 2: Ejemplo de documentación de una función DoxyGen	24
Figura 3: Fases e iteraciones del Proceso Unificado	26
Figura 4: Diagrama de Gantt I	27
Figura 5: Diagrama de Gantt II	28
Figura 6: Diagrama de paquetes	31
Figura 7: Ejemplo de diagrama de casos de uso: Gestión de Incidencias	32
Figura 8: Diagrama de clases Modelo de Dominio	34
Figura 9: Ejemplo diagrama de secuencia.	35
Figura 10: Diagrama de comunicación del paquete Gestión de Incidencias	35
Figura 11: Modelo de diseño	37
Figura 12: Capa controlador de la aplicación web	38
Figura 13: Vista arquitectónica	39
Figura 14: Ejemplo de diagrama de secuencia diseño	40
Figura 15: Base de datos	41
Figura 16: Modelo de despliegue	42
Figura 17: Página pública	45
Figura 18: Información detallada	46
Figura 19: Iniciar sesión	47
Figura 20: Página privada de bienvenida	48
Figura 21: Ver perfil	49
Figura 22: Listado de edificios	50
Figura 23: Añadir edificio	51
Figura 24: Añadir edificio, último paso	52
Figura 25: Añadir edificio, puerta única	52
Figura 26: Comparar edificios	53
Figura 27: Ver edificio	54
Figura 28: Ver fotografías	54
Figura 29: Editar edificio	55
Figura 30: Listado de puertas	56
Figura 31: Listado de incidencias	57
Figura 32: Ver detalles del semáforo	58
Figura 33: Vista previa del semáforo	58
Figura 34: Página privada del administrador	59
Figura 35: Listado de usuarios	60
Figura 36: Editar usuario	60

## Índice de tablas

---

<i>Tabla 1: Participante Hernández Noriega, Alfonso</i>	<i>29</i>
<i>Tabla 2: Ejemplo de requisito de información</i>	<i>31</i>
<i>Tabla 3: Ejemplo especificación caso de uso</i>	<i>33</i>

### 1. Introducción

---

El presente documento recoge la memoria del Trabajo de Fin de Grado titulado “Sistema para el control de aforo seguro en establecimientos”. Este proyecto se desarrolla en el contexto de una pandemia, que ha requerido que la población mantenga cierta distancia social.

En el contexto de establecimientos de cara al público, puede controlarse esta distancia social mediante el uso de las nuevas tecnologías. Conforme a ello se desarrolla el presente proyecto. Diseña una aplicación para controlar el aforo gracias al uso de cámaras y algoritmos de reconocimiento facial.

En este documento se explicarán los aspectos más importantes en el desarrollo del proyecto. La estructura del documento es la siguiente:

- **Estado del arte:** estudio de algunos de los sistemas competidores disponibles en el mercado.
- **Objetivos:** se especifican los objetivos a cumplir en el proyecto.
- **Conceptos teóricos:** se desarrollan los conceptos considerados necesarios para la comprensión de este trabajo.
- **Técnicas y herramientas:** se expone la documentación de las técnicas y herramientas usadas en el proyecto.
- **Aspectos relevantes del desarrollo:** recoge los puntos destacables del desarrollo del proyecto.
- **Limitaciones del prototipo:** se habla sobre las limitaciones técnicas encontradas al final del prototipo.
- **Conclusiones y líneas de trabajo futuras:** se exponen las conclusiones obtenidas de la realización del proyecto.
- **Bibliografía**

## Memoria del proyecto

Este documento está completado con los siguientes anexos:

- **Anexo I – Planificación temporal:** estudio de los aspectos relacionados con la planificación temporal de las tareas que conforman el proyecto.
- **Anexo II – Especificación de requisitos:** recoge la especificación de los requisitos del sistema.
- **Anexo III – Análisis de requisitos:** documentación sobre la fase de análisis de los requisitos.
- **Anexo IV – Diseño del sistema software:** recoge la documentación de la fase de diseño del sistema.
- **Anexo V – Documentación técnica:** tiene como objetivo facilitar al programador la comprensión del código.
- **Anexo VI – Manual de usuario:** detalla en un documento todas las funcionalidades que ofrece el sistema a cada usuario.

## 2. Estado del arte

---

### 2.1 Sistemas de control de aforo

---

La masificación de las grandes ciudades tras la revolución industrial y el progresivo crecimiento poblacional de las sociedades ha influido en la consiguiente masificación de determinados espacios de uso cotidiano, tales como los establecimientos. Este aumento de usuarios que demandan los servicios de consumo ha supuesto una serie de problemas reflejados, tradicionalmente, en situaciones de emergencia, que habitualmente se solventaban a través de un estudio de prevención de riesgos laborales y, más específicamente, en el área de seguridad en el trabajo (Ortega et al., 2017).

Sin embargo, la situación actual de pandemia sobrevenida como consecuencia del COVID-19 ha sumado otros problemas a tener en cuenta en lo que respecta a dichas masificaciones, que exigen el control de los aforos de los lugares cerrados. Así, tras haberse identificado como una de las vías de contagio los aerosoles (que alcanzan de un individuo a otro a través del contacto cercano, a menos de un metro de distancia y se introducen por las mucosas al sistema respiratorio) (OMS, 2020a) se constató la importancia del distanciamiento social que, en lo que respecta al aforo de un establecimiento, se tradujo en una reducción de entre el 30% (en las etapas más dramáticas de la pandemia) al 50-75% en la consiguiente evolución y progresiva restauración de la normalidad (Consejo interterritorial, 2020).

No obstante, esta reducción del aforo debe llevar pareja una serie de estrategias que garanticen que realmente se está aplicando satisfactoriamente dicha reducción del aforo, para lo cual se han diseñado diversas herramientas que permiten controlar el aforo. Este interés en el control del aforo se ha definido, por ejemplo, en la página [aforoseguro.com](http://aforoseguro.com) (Aforoseguro, s.f.), a través de la cual se ofrece al usuario una estrategia de monitorización y control de aforo de los negocios, que utiliza la tecnología y la video inteligencia para establecer los niveles de aforo de un establecimiento dado, así como de sus partes o áreas permitiendo enviar alertas en caso de que se supere el aforo o que alcance

## Memoria del proyecto

niveles no seguros. En la siguiente figura podemos ver un esquema del funcionamiento.



Figura 1. Propuesta para el control del aforo en un establecimiento. Fuente: Aforoseguro.

Esta plataforma utiliza un código de colores a modo de semáforo para identificar los distintos estados de aforo, utilizándose también esta estrategia en otras aproximaciones de control de aforo ya no solo en establecimientos, sino, por ejemplo, en distintos espacios de la ciudad, como propone GeoSek (2021). GeoSek plantea una identificación en tres niveles según el aforo: en verde (entre 0 y 25% de aforo), en amarillo (entre 26 y 29% de aforo) y en rojo (más del 30% de aforo).

Sea como fuere, estas estrategias comparten que permiten controlar los aforos de distintos espacios conforme a las recomendaciones a nivel estatal, que se reflejan en el artículo publicado por Rojas (2020). Este autor advierte de la gran relevancia que actualmente tienen los controles de aforo, como estrategia de control de la distancia social en establecimientos comerciales. Conforme a ello, es posible identificar distintas estrategias para el control de aforo que van desde el conteo manual realizado por un trabajador del centro hasta la utilización de cámaras y sensores que lleven a cabo dicho conteo de manera automática,

## **Memoria del proyecto**

más eficiente que el control manual y que permite optimizar los recursos humanos de una empresa.

En el artículo publicado por Rojas (2020) se identifican las distintas fases de selección e implementación de un sistema de conteo a través de vídeo en un establecimiento, valorándose tanto las ventajas, como los aspectos que deben tenerse en cuenta en dicha elección. Además, este autor proporciona una serie de marcas de seguridad electrónica que han desarrollado sistemas de control de aforo, que se mencionan a continuación: Axis, quien ha desarrollado un control de aforo que incluye alertas sonoras y estima la ocupación de un entorno dado; Bosch, quien propone un análisis del aforo a través de cámaras IP; Dahua Technology, quien plantea este control de aforo teniendo en cuenta las posibles entradas y salidas de un espacio; Hikvision, con una elevada precisión en el conteo de personas; Scati, que facilita el número exacto de individuos en un momento dado o Vivotek son algunas de las propuestas planteadas. La mayor parte de estos proyectos se centran en el contaje de individuos y algunos de ellos desarrollan planteamientos más globales en los que se incluyen la identificación de mascarillas o la medición de la temperatura corporal como estrategias de medición de normalidad/enfermedad y, por tanto, proporcionan una atención más global para controlar la situación de pandemia y contagio.

En esta línea, el proyecto que se presenta combina el contaje de individuos que acceden o abandonan un establecimiento con la utilización de mascarilla, enmarcándose de este modo como una alternativa que ofrece un control más global de la situación del establecimiento.

## **2.2 Sistemas de detección de mascarillas**

---

Pese a que actualmente se asocia el uso de la mascarilla a la pandemia originada por el COVID-19, lo cierto es que se trata de un dispositivo de uso habitual para proteger a quien lo porta de inhalaciones atmosféricas peligrosas que pueden incluir desde humos, hasta gases, vapores, partículas en suspensión o microorganismos. En algunas ciudades con elevada contaminación, el uso de las mascarillas es muy habitual, mejorando en cierta



## Memoria del proyecto

medida la calidad de vida y el bienestar de quienes la portan, al minimizar la inhalación de los contaminantes que se encuentran suspendidos en la atmósfera (Oltra y Sala, 2015). Otro de los entornos en los que la mascarilla es un utensilio de uso habitual son los quirófanos y otros espacios hospitalarios, donde se utiliza como medida de protección tanto de los profesionales como de los propios pacientes (Del Río, 2013; Rubio, 2014). Además, en determinadas culturas, está normalizado su uso cuando una persona está enferma para reducir las posibilidades de contagiar a su entorno. Esta última utilización se ha universalizado como consecuencia de la pandemia provocada por el COVID-19 (OMS, 2020b).

La utilización de mascarilla se ha definido como una de las estrategias para la reducción de las posibilidades de contagio en la pandemia de COVID-19, por lo que constituye un beneficio desde el punto de vista sanitario que debe ser tenido en cuenta y ante el cual es necesario concienciar a la población y controlar su uso en diversos espacios, siendo especialmente oportuno en aquellos espacios cerrados. Ejemplo de este interés ha sido la utilización de las tecnologías de reconocimiento facial para detectar la mascarilla, a través de estrategias de análisis de la población a tiempo real mediante las cuales garantizar que las personas que acceden a un espacio cerrado la portan. Juanes (2020) advierte de la innovación que supone utilizar la inteligencia artificial para apoyar en el control de estas reglas de uso de las mascarillas, profundizando en la aplicación de detección de mascarillas diseñada por Hanwha Techwin como respuesta a esta nueva necesidad identificada en las sociedades. Esta autora expone que la aplicación nombrada como *Face Mask Detection* puede ejecutarse a través de cámaras de la gama Wisenet X de plataforma abierta y analiza los vídeos de que dispone para identificar si una persona que accede a un espacio cerrado no está llevando mascarilla, tras lo cual puede reproducir un mensaje recordatorio de la importancia de utilizar este dispositivo. En base a esta propuesta se contextualiza el diseño que se presenta en esta memoria, cuya filosofía también aborda las posibilidades de control y gestión de los individuos que acceden al establecimiento dado.

### 3. Objetivos

---

En este apartado se exponen los objetivos que debe cumplir el sistema para considerar completado este Trabajo de Fin de Grado.

#### 3.1 Objetivos del sistema

---

El objetivo principal del sistema es diseñar y construir una aplicación que monitorice, mediante cámaras, los clientes que entran y salen de un establecimiento y, de esta manera, poder llevar un control de aforo. También debe detectar cuáles de los clientes llevan mascarilla puesta y cuáles no. Por último, el sistema debe ofrecer un portal donde los usuarios puedan gestionar y ver el estado de sus edificios, junto con su información relacionada.

En el siguiente listado se detallará cada objetivo del sistema:

- **Detectar los clientes y el uso de mascarilla:** deberá ser capaz de detectar cada vez que alguien entre o salga del establecimiento. También debe ser capaz de detectar si un cliente lleva o no mascarilla puesta. Se guardará una fotografía de cada registro.
- **Controlar el aforo:** deberá ser capaz de generar y mostrar estadísticas diarias de cada edificio, informando sobre su ocupación y sobre el porcentaje de uso de mascarillas entre los clientes.
- **Gestionar los usuarios:** deberá permitir el registro y eliminación de usuarios, así como la modificación y visualización de los datos de su perfil. Además, debe ser capaz de gestionar diferentes roles de usuario.
- **Gestionar los edificios:** deberá permitir a un usuario identificado añadir, eliminar, ver, modificar y listar sus edificios. Cada edificio tendrá

## Memoria del proyecto

asociadas al menos una puerta de entrada y otra de salida o una puerta única.

- **Gestionar las puertas:** deberá permitir dar de alta una puerta en un edificio, eliminar la puerta y editar su información. Cada puerta lleva asociada una cámara. Si es tipo de puerta única tendrá asociadas dos cámaras, una para la entrada y otra para la salida.
- **Controlar las incidencias:** deberá llevar un registro, por cada edificio, de las posibles incidencias.
- **Gestionar los semáforos:** deberá ser capaz de gestionar un semáforo por cada edificio. Se colocará una Tablet en cada entrada del edificio en la que se mostrará la información en tiempo real del mismo.

### 3.2 Objetivos personales

---

A continuación, se describen los objetivos personales que me han llevado a realizar este proyecto.

- 1) Facilitar el control del aforo de los edificios, negocios y establecimientos que deseen implementar el proyecto, de manera que puedan reducirse las tasas de contagio del COVID-19, poniendo las TIC a disposición de la salud de las personas.
- 2) Participar de manera activa en el control y manejo del COVID-19, a través del sector del pequeño y mediano comercio (PYMES), proporcionando una herramienta útil, versátil, fácil de utilizar, intuitiva y, sobre todo, económica, que constituya una alternativa eficaz para que el empresario gestione su negocio bajo las actuales directrices de la nueva normalidad.
- 3) Reflexionar acerca del potencial de las TIC para mejorar la realidad y la calidad de vida de las personas mediante el ejercicio de puesta en práctica de los conocimientos adquiridos en un ejemplo de diseño de una aplicación.

### 4. Conceptos teóricos

---

En este apartado se exponen los conceptos teóricos que es recomendable detallar para facilitar la comprensión del proyecto.

Para el reconocimiento facial se ha utilizado una red neuronal convolucional (o Convolutional Neural Networks), que previamente ha sido entrenada con un conjunto de datos formado por cientos de imágenes de personas con y sin mascarilla. Este tipo de red neuronal artificial se caracteriza por un aprendizaje supervisado que procesa sus capas imitando al cortex visual del ojo humano, con el objetivo de identificar las características en las entradas. Es una variación de una red perceptrón multicapa, la cual realiza los cálculos con matrices bidimensionales, por lo que son muy efectivas para tareas de visión artificial, como en la clasificación y segmentación de imágenes.

Toda red neuronal artificial lleva asociada una función de activación, que es análoga a la tasa de potencial de acción existente en el cerebro humano. Proporciona la salida que será generada por una neurona en función del conjunto de entradas. Para la red neuronal convolucional de este proyecto se ha utilizado la función de activación ReLu (del inglés Rectified Linear Unit o Unidad Lineal Rectificada). Permite un entrenamiento más rápido en RNA profundas y proporciona una gran mejora respecto a problemas que presentan otras funciones, como la sigmoide o la tangente hiperbólica. Su fórmula es  $R(x) = \max(0, x)$ , si  $x < 0$ , entonces  $R(x) = 0$  y si  $x > 0$  entonces  $R(x) = x$ . Entre sus ventajas podemos destacar:

- Sus activaciones no requieren ningún cálculo exponencial, lo que asegura un entrenamiento más rápido que las funciones sigmoideas, debido a la menor complejidad del cálculo numérico.
- No tienen ningún problema al calcular el termino de error de una desactivación gradual, como ocurre en las funciones sigmoideas o de tangente hiperbólica.

## Memoria del proyecto

Tiene un inconveniente importante, cuando las entradas se acercan al valor cero o son negativas, entonces el gradiente se vuelve cero y no podría realizar un aprendizaje hacia atrás, lo que conlleva una pérdida de aprendizaje. Las neuronas que entran en esta situación se dice que están muertas y no intervienen en la red.

Los datos de entrada para el entrenamiento serán los píxeles de cada una de las imágenes. Su tamaño será el ancho por el alto y por un valor tres de profundidad, ya que son imágenes en color (RGB) rojo, verde y azul. La capa de convolución procesará la salida de las neuronas que están conectadas en regiones locales de entrada (es decir, píxeles cercanos), calculando el producto escalar entre sus pesos (valor de píxel) y una pequeña región determinada por el volumen de entrada. Se determina el uso de 32 filtros, que determinará la región del volumen de salida. El resultado de esta operación será utilizado en sucesivas convoluciones, creando un número exponencial de neuronas en las capas ocultas.

Por último, se debe unificar la última convolución en una capa oculta tradicional y determinar las salidas con una función exponencial normalizada, en nuestro caso con dos resultados: si un rostro lleva puesto mascarilla o no.

## 5. Técnicas y herramientas

---

En este apartado se detallan las técnicas, herramientas, frameworks y librerías que se han utilizado en el desarrollo del software. Se dividirá en tres grupos, según su uso:

- Desarrollo de la aplicación web
- Desarrollo del servidor de la cámara
- Herramientas CASE

### 5.1 Aplicación Web

---

#### 5.1.1 HTML y CSS

---

HTML (o HyperText Markup Language), es un lenguaje de marcas de hipertexto, usado para la elaboración de páginas web. "Hipertexto" hace referencia a los enlaces que conectan páginas web entre sí, ya sea dentro de un único sitio web o entre sitios web. Además, utiliza marcas para etiquetar texto, imágenes y otro contenido para mostrarlo en un navegador Web (Álvarez y Álvarez, 2018).

HTML generalmente se utiliza junto con otra tecnología para describir la apariencia de una página web, llamada CSS (o Cascading Style Sheets). Determina como será la presentación de un documento estructurado en HTML. CSS identifica las posibles hojas de estilo en cascada y estructura dentro de la página la presentación de los títulos, apartados, etc., mediante la identificación de tipografía y estilo de las letras, así como la definición de los colores que esta presenta. Estas hojas de estilo CSS permiten separar por una parte la estructura de la página y por otra parte la presentación, de manera que facilita la modificación de dicho estilo sin tener que trabajar con la información que esta contiene. Esta ventaja se refleja en las posibilidades de edición de la página web (Eguiluz, 2009).

### 5.1.2 JavaScript

---

JavaScript pertenece a los lenguajes de programación interpretados que se caracterizan por utilizar el código fuente sin compilar, por lo que requiere un intérprete que lea y ejecute dicho código. De manera habitual, dicho intérprete se incluye en el propio navegador de internet, variando su versión en función del navegador que se utilice (Gauchat, 2012).

JavaScript es utilizado en el diseño de páginas web y aplicaciones permitiendo la especificación y considerándose, generalmente, como un complemento para el desarrollo de múltiples propósitos.

La técnica más sencilla para introducir JavaScript en el sistema implica el aprovechamiento de los atributos disponibles en elementos de HTML, específicamente aquellos cuya función es la de manejador de eventos, ejecutando los códigos según las acciones del usuario.

JavaScript complementa a los lenguajes HTML y CSS de acuerdo con la siguiente distribución de la gestión de información: HTML controla la estructura; CSS gestiona el diseño y JavaScript determina la funcionalidad de la página.

### 5.1.3 Vue

---

Se define como un framework de código abierto diseñado conforme a la arquitectura MVVM (Model View-View Model) de JavaScript. A través de este framework pueden crearse de manera rápida, sencilla y estructurada páginas web dinámicas. Se utiliza como gestor de paquetes NPM en el que se cargan diversas librerías y herramientas para desarrollar la aplicación web de manera más cómoda (sitio web oficial de Vue, 2021).

De acuerdo con la complementación de lenguajes HTML, CSS y JavaScript descritos previamente, el proyecto Vue jerarquiza los componentes y los agrupa conforme a tres partes:

`<template></template>`: representa el lenguaje HTML y refleja la estructura de la aplicación web.

## Memoria del proyecto

`<script></script>`: representa el lenguaje JavaScript y refleja la funcionalidad de la aplicación web.

`<style></style>`: representa el lenguaje CSS y refleja el diseño de la aplicación web.

### 5.1.4 Google Maps Platform

---

Google Maps Platform es un conjunto de API y SDK que los desarrolladores pueden usar a fin de incorporar Google Maps en apps para dispositivos móviles y páginas web. En concreto se ha utilizado la versión Javascript de Google Maps, que presenta cuatro tipos de mapas básicos (mapa de carreteras, satélite, híbrido y terreno) que se pueden modificar mediante capas y estilos, controles y eventos, y diversos servicios y bibliotecas. Además permite personalizar los mapas con su propio contenido, imágenes y marcadores

### 5.1.5 PHP

---

PHP es un lenguaje de código abierto de gran popularidad y generalmente utilizado en el desarrollo web debido a que permite crear contenidos dinámicos y funciona en el propio servidor.

Este lenguaje también permite manejar, aparte del texto y la presentación en el diseño de la web, la inclusión de imágenes, ficheros PDF o cualquier tipo de documento XML. De acuerdo con estas múltiples aplicaciones pueden desprenderse de la utilización de PHP se comprende el potencial de utilización de este lenguaje y sus usos bastante variados (Cobo et al., 2005).

PHP se utiliza como base del Laravel Framework acerca del cual se profundiza en el apartado siguiente.



### 5.1.6 Laravel Framework

---

Laravel es una framework multiplataforma definida como de código abierto que se utiliza para desarrollar tanto aplicaciones como páginas web utilizando el lenguaje PHP como base (Kinsta, 2021). Esta framework se desarrolla bajo la filosofía de ofrecer un código PHP simple y elegante, que evite el denominado “código espagueti” y toma como referencia en su diseño a otras frameworks disponibles ya en el ámbito del diseño de aplicaciones y páginas web tales como “Sinatra” o “Ruby on Rails”.

Laravel se estructura en dependencias, por lo que el propio desarrollo de este framework depende del desarrollo de cada una de las dependencias que la conforman. Esta estructura, junto con el funcionamiento de Laravel permiten simplificar el trabajo a través de tareas comunes de autenticación, enrutamiento, gestión de sesiones o almacenamiento de caché (Haro et al., 2016).

Laravel aporta las siguientes ventajas, que la caracterizan y propician que sea una herramienta de elección en el desarrollo de aplicaciones y páginas web:

- 1) Utiliza un patrón modelo vista controlador, a través del cual se separa y modulariza el código. Esta estrategia mejora la experiencia de trabajo en equipo, la reutilización del código y el desarrollo claro y de fácil mantenimiento.
- 2) Integra Eloquent, que se define como un sistema de mapeado de datos relacional, una técnica de programación que convierte datos en un lenguaje orientado a objetos a una base de datos como motor de persistencia. A través de este sistema puede accederse de manera directa a los campos de un objeto con el fin de leerlos, modificarlos o insertarlos; ofrece también la construcción de consultas directas a la base de datos a través de Query Builder.
- 3) Facilita la manipulación y gestión de tablas desde código, guardando un sistema de versiones a través del sistema de migraciones de que dispone.
- 4) Se apoya en el sistema de plantillas Blade que proporciona una mayor velocidad al utilizar la caché y permite la creación de vistas a través de layouts, herencia y sesiones.

## Memoria del proyecto

- 5) Permite la inclusión de paquetes o librerías externas a través de las cuales facilitar el desarrollo de la aplicación y ahorrar tiempo de programación, extendiéndose la funcionalidad del sistema.
- 6) Incorpora Artisan, un intérprete de línea de comandos que facilita la realización de tareas rutinarias en el uso del framework tales como crear componentes de código, trabajar con la base de datos, realizar migraciones o gestionar las rutas, entre otros.

De manera general, todo proyecto creado en Laravel muestra una serie de características en lo que respecta a su estructuración de carpetas y ficheros, en la que se define su organización de código.

- App: guarda el código principal de la aplicación, así como el conjunto de controladores, modelo de datos y middlewares que se precisan. A su vez, lo conforman dos subcarpetas: *app/Http/Controllers*, en la que se encuentran los controladores a través de los cuales se interactúa tanto con los modelos como con las vistas y mediante los que se maneja la lógica de la aplicación; y *app/Http/Middleware*, donde se ubican aquellos filtros cuya función es la de validar permisos antes o después de la ejecución de una ruta.
- Config: contiene los archivos de la configuración de la aplicación.
- Database: en dicha carpeta se encuentra todo lo relacionado con la definición de la base del proyecto.
- Public: contiene la información visible en el servidor web, debe ser la única de carácter público del proyecto y guarda tanto los archivos CSS como los correspondientes a JavaScript y aquella información gráfica que se desee que forme parte de la aplicación o página web.
- Resources: conformada a su vez por tres carpetas: *assets*, *views* y *Lang*. Dentro de ellas, destaca la carpeta *views* por ser el directorio de las vistas de la aplicación que generalmente se presenta como plantilla HTML, y la carpeta *Lang*, que contiene los archivos PHP sobre los cuales se lleva a cabo la traducción automática.
- Routes: guarda los archivos de enrutamiento de la aplicación o página web.
- Bootstrap: en ella se incluye el código que procesa cada llamada del proyecto.

## Memoria del proyecto

- Storage: incluye la información interna que Laravel precisa para ejecutar en la web la aplicación diseñada, dentro de la cual se encuentran los archivos de sección, caché, compilación de vistas y logs del sistema.
- Tests: guarda los ficheros de pruebas automatizadas.
- Vendor: aloja aquellas librerías y dependencias anteriormente mencionadas que conforman el framework de Laravel.

Otros elementos que deben ser tenidos en cuenta para comprender el funcionamiento de Laravel se exponen a continuación, siendo los siguientes los archivos más importantes:

- .env: se define como el fichero de almacenaje de los valores de configuración que caracterizan a la instalación, permitiéndose editar dichos valores o establecer distintas opciones en función del sistema en producción.
- Composer.json: permite la instalación de Laravel, así como de sus librerías o paquetes externos a través de los cuales se añaden funcionalidad.

A modo de resumen, se expone a continuación el funcionamiento básico de Laravel: en primer lugar, las peticiones web entran a la URL a través del archivo index.php contenido en la carpeta public, que comprueba la validez de la URL de acuerdo con la información contenida en la carpeta routes.php; tras lo cual se ejecuta el método de controlador a la que se asigna la petición solicitada. A continuación, el método controlador puede acceder a los modelos que permitan obtener, modificar o eliminar los datos y, posteriormente, proporciona al usuario los datos solicitados a través de la presentación o vista de los datos que se asocian a dicha solicitud.

Con todo ello, puede comprenderse la importancia e interés de Laravel, siendo uno de los framework más demandado en el modelo empresarial actual y con una gran adaptación al proyecto que se presenta, facilitando el desarrollo del proyecto y minimizando los errores e inseguridades en la generación del código.

### 5.1.7 Plantilla AdminLTE

---

Se trata de una herramienta que sirve como patrón de diseño de interfaces a través de la cual se facilita la usabilidad de una aplicación y su adaptación a distintos dispositivos móviles; tales como tablets o smartphones, lo que facilita su implementación en el sistema empresarial para el que se diseña el proyecto (AdminLTE, 2021).

Esta plantilla de interfaz web ofrece un patrón de diseño intuitivo y de fácil utilización, que facilita la presentación de la información al usuario que maneja la aplicación. La plantilla propicia una adecuada accesibilidad, debido a sus características en las que se presenta el menú superior al entrar al sistema y se ofrece aquella información que resulte relevante para la gestión del aforo en el establecimiento. Dicha plantilla puede modificarse para adaptarse a las necesidades del proyecto.

### 5.1.8 Apache

---

Se define como un servidor web HTTP multiplataforma y de código abierto, que actualmente es el más utilizado a nivel mundial. Cuenta con un elevado número de instalaciones y su carácter gratuito ha facilitado que sea el servidor de elección utilizado también en el desarrollo de este proyecto, utilizándose como servidor remoto para llevar a cabo las tareas de programación.

### 5.1.9 MySQL

---

Se define como un sistema de bases de datos relacionales que se integra en la licencia dual GPL/Licencia comercial de Oracle Corporation, erigiéndose como una de las bases de datos open source más populares y demandadas en la actualidad, compitiendo de manera directa con Oracle y Microsoft SQL Server en lo que respecta a su utilización. Este servicio administrado de bases de datos facilita un desarrollo rápido de aplicaciones (Oracle, 2021).

## Memoria del proyecto

Esta base de datos relacional se caracteriza por incluir diversas tablas a través de las cuales se almacena y organiza la información, al mismo tiempo que permite integrar los lenguajes de programación más utilizados (dentro de los que destaca, de acuerdo con el trabajo realizado, PHP) y distintos sistemas operativos.

MySQL garantiza la persistencia de datos en el servidor facilitando la gestión de los usuarios y la información contenida acerca de los foros de cada establecimiento que se incluye en el proyecto.

### 5.1.10 PHPStorm

---

PHPStorm es uno de los principales soportes para frameworks que se utiliza en la actualidad, debido a las posibilidades que ofrece en lo que respecta a las distintas frameworks que admite, entre ellas, Laravel, pero también Symfony, Drupal, Magento, WordPress, Joomla! o CakePHP, entre otros.

Se trata de un editor de PHP que entiende tanto el código como su estructura, facilitando una mejor finalización del código y previniendo errores sobre la marcha. Además, este soporte permite valorar de manera inmediata aquellos cambios que se realicen en el código del proyecto, tanto en lo que respecta a HTML, como CSS o JavaScript, depurándose aquellos errores que se identifiquen a través de las pruebas realizadas. Por otro lado, integra un Sistema de Control de Versiones, así como bases de datos/SQL y otras herramientas que permite llevar a cabo tareas rutinarias facilitando a los desarrolladores el desarrollo de la aplicación web; por lo que se describe como un soporte eficiente, seguro y rápido (PHPStorm, 2021).

### 5.2 Servidor Cámara

---

#### 5.2.1 Python

---

Python es un lenguaje de programación interpretado que se centra en la legibilidad del código, cuya filosofía es que sea legible por cualquier persona que disponga de conocimientos básicos en programación, y se define como un lenguaje multiparadigma, al integrar y soportar tanto la orientación a objetos como la programación imperativa y funcional (Python, 2021; Visus, 2020).

Se trata de un lenguaje de escritura rápido, escalable, robusto y de código abierto que lo ubican como un apoyo de gran interés en lo que respecta a la inteligencia artificial al facilitar la expresión de ideas complejas en escasas líneas de código (Visus, 2020). Python es uno de los lenguajes de programación interpretado más habituales debido a una serie de particularidades o características que proporcionan múltiples ventajas para los desarrolladores, las cuales se exponen a continuación:

- 1) Se trata de un lenguaje de programación gratuito, o de código abierto, por lo que no es preciso pagar una licencia para utilizarlo.
- 2) Dispone de gran cantidad de librerías y aplicaciones que se encuentran en constante desarrollo y ampliación, debido a su carácter gratuito, es elegido por una gran comunidad de programadores, quienes a su vez aportan mejoras al sistema. Esto se convierte en una fortaleza para el lenguaje de programación y multiplica el interés de los programadores, quienes además utilizan los foros para resolver dudas, creándose una comunidad progresivamente más amplia, formada e implicada en la utilización de Python.
- 3) Python se describe como un lenguaje multiparadigma, propiciándose una amplia flexibilidad y aplicabilidad del mismo al combinarse diferentes paradigmas de programación. Por otra parte, resulta fácil su aprendizaje.
- 4) Esta flexibilidad también permite que sea aplicado en contextos muy diferentes, lo que a su vez garantiza una difusión exponencial a distintas

## Memoria del proyecto

áreas de trabajo y, por ello, una retroalimentación positiva según la cual se amplía la red de profesionales que la utilizan.

- 5) Es un lenguaje apto y adaptado a diversas plataformas, simplemente mediante la utilización del intérprete correspondiente.

Dentro de las limitaciones que pueden asociarse a Python, cabe destacar que se trata de un lenguaje más lento que otros disponibles, debido a que es interpretado, en lugar de compilado; no obstante, no se trata de un obstáculo que implique un gran peso en la elección de este lenguaje, debido a que las diferencias en velocidad no resultan sustanciales y a que el cuello de botella que se identifica en el desarrollo de proyectos no se encuentra en la CPU, pudiendo subsanarse esta mayor lentitud a través de la programación y computación en la nube.

Tal como recoge Visus (2020), Python dispone de un conjunto de características que resultan coherentes con la mentalidad de éxito actual, que se exponen en los cinco puntos siguientes:

- Explícito es mejor que implícito.
- Simple es mejor que complejo.
- Complejo es mejor que complicado.
- Si la implementación es difícil de explicar, es una mala idea.
- Si la implementación es fácil de explicar, podría ser buena idea.

### 5.2.2 OpenCV

---

OpenCV se corresponde con la abreviatura de Open Computer Vision (Visión Artificial Abierta) y se describe como una biblioteca libre de visión artificial originada en 1999, como software libre y de código abierto que utiliza como lenguajes de programación C++ y C, erigiéndose actualmente en una de las bibliotecas de visión artificial más populares. Permite la detección de objetos, el reconocimiento de movimientos e imágenes, la reconstrucción 3D de imágenes, entre otras funciones (OpenCV, 2021; Svitla, 2021). En este caso, se utiliza con el fin de reconocer los rostros para el control del aforo a través de las personas

## Memoria del proyecto

que entran y salen en el establecimiento, siendo esta una de las aplicaciones identificadas como áreas de aplicación de la biblioteca (reconocimiento facial).

Algunos de los motivos por los que esta biblioteca ha resultado tan exitosa y popular se exponen a continuación:

- Es gratuita, por lo que son numerosos los investigadores y desarrolladores que la utilizan tanto con fines de investigación como comerciales. Dispone de licencia BSD por lo que puede utilizarse libremente.
- Es multiplataforma, esto es, permite su utilización tanto en los sistemas operativos de Windows, Linux, Mac OS X o Android.
- Actualizada y documentada, por los propios organizadores de la biblioteca, lo que facilita que quienes la usan dispongan de documentos completos acerca de las funciones, ejemplos de uso y tutoriales que puedan utilizarse por cualquier usuario con mínimos conocimientos de visión artificial, además, la organización muestra un interés activo en formar, divulgar y producir libros que faciliten la utilización de esta biblioteca.

### 5.2.3 Tensorflow

---

Se trata de la biblioteca de aprendizaje automático y profundo que más popular es actualmente, lanzada por Google en 2015, gran parte de su popularidad reside en su simplicidad de la API y a los lenguajes que utiliza; en concreto, está escrita en Python y dispone de puerto de JavaScript, dada la popularidad de este último (Svitla, 2021).

TensorFlow se describe como una biblioteca de código abierto y gratuita que utiliza flujos de datos y programación diferencial que puede aplicarse en redes neuronales para el aprendizaje automático de las aplicaciones diseñadas. En su aprendizaje utiliza la programación diferencial y las matemáticas simbólicas para identificar diferencialmente determinados elementos, depurando el modelo y simplificando progresivamente dicha identificación. De este modo, se entrena en el reconocimiento de imágenes (en este caso, rostros) ofreciendo una solución de alto rendimiento.



### 5.2.4 Keras

---

Se define como una biblioteca de aprendizaje profundo en Python (Svitla, 2021) en la que se combinan funciones de distintas bibliotecas, dentro de las cuales destaca Tensorflow, también utilizada en este proyecto, constituyendo esto una ventaja sobre otras bibliotecas de aprendizaje profundo debido a que se ejecuta sobre Tensorflow, pudiéndose ejecutar además sobre otras bibliotecas de aprendizaje automático tales como Microsoft Cognitive Toolkit, Theano o PlaidML. Kera se caracteriza por la experimentación rápida con redes neuronales profundas, siendo sus particularidades definitorias la usabilidad, modularidad y extensibilidad y habiéndose diseñado como una API para seres humanos en lugar de para máquinas, reduciéndose en el proceso la carga cognitiva al minimizar el número de acciones que deben realizarse por parte del usuario.

### 5.2.5 PyCharm

---

PyCharm es uno de los entornos de desarrollo más completos para utilizar en Python, facilitando una finalización del código inteligente, inspecciones, identificación de errores sobre la marcha y arreglos rápidos que proporcionan una experiencia de navegación completa y funcional (PyCharm, 2021).

Se trata de un entorno compatible con Python, JavaScript o CSS entre otros, valiéndose y apoyándose en los lenguajes de programación y las plantillas más populares a nivel mundial. Además, se completa con diversas herramientas preconfiguradas que facilitan el desarrollo del proyecto, la depuración, realización de pruebas y generación de perfiles. Otra de las ventajas de que dispone PyCharm es su carácter multiplataforma, pudiendo usarse tanto en Windows, como en Linux o MacOS donde resulta cómodo y sencillo personalizar el interfaz de usuario de acuerdo con las preferencias del programador en cuanto a esquemas de color y atajos de teclado, así como en el uso de complementos.

## **5.3 Herramientas CASE**

---

### **5.3.1 Visual Paradigm**

---

Se trata de una herramienta UML Case diseñada con el objetivo de soportar el proceso completo de desarrollo del software y en la que se representan todo tipo de diagramas UML.

Se ha utilizado en el presente trabajo como referente a lo largo de la fase de especificación de requisitos permitiendo visualizar los análisis de requisitos y diseño del sistema que se presentan en forma de diagramas en los anexos de la memoria. Dentro de estos diagramas se encuentran ejemplos de diagramas de casos de uso, de paquetes, secuencia y estado.

### **5.3.2 Microsoft Project**

---

Se trata de un software de administración de proyectos que presta asistencia a los administradores de proyectos en el transcurso del desarrollo de planes, asignación de recursos o en el seguimiento al progreso, administración y análisis del trabajo. Este programa se ha utilizado en la fase de planificación del proyecto, facilitando la temporalización del mismo.

## **5.4 Herramientas de documentación**

---

Como herramientas de generación de documentos se ha utilizado Doxygen y DocsUtil. Se definen como generadores de documentación técnica en distintos lenguajes de programación, dentro de los cuales se encuentra PHP, Java, Python o C++, entre otros. Estas herramientas utilizan el código fuente para generar la documentación que se presenta en el documento Anexo V – *Documentación técnica*.

## Memoria del proyecto

DoxyGen se trata de una herramienta bastante versátil al utilizar distintos lenguajes de programación y también fácilmente adaptable para su utilización tanto en Windows como en Mac OS X, por lo que se define como multiplataforma.

De manera global, se identifican tres ventajas principales de uso por la que se selecciona esta herramienta (Marco de Desarrollo de la Junta de Andalucía, s.f.):

- 1) Puede generar documentación on-line (en HTML) y también un manual de referencia off-line (en latex) partiendo de una serie de ficheros fuente documentados. También proporciona documentos de salida en RTF, PDF, o HTML comprimido, entre otros, extrayéndose la información de manera directa de las fuentes, lo que facilita la creación de manuales y su posterior documentación.
- 2) Puede configurarse esta herramienta para extraer la estructura de un código que no esté documentado, facilitando al mismo tiempo la visualización de relaciones entre elementos.
- 3) Puede utilizarse para crear documentación normal, sirviendo como herramienta global en el desarrollo de un proyecto.

En la siguiente figura puede verse una muestra del resultado obtenido al aplicar la herramienta a nuestro código.

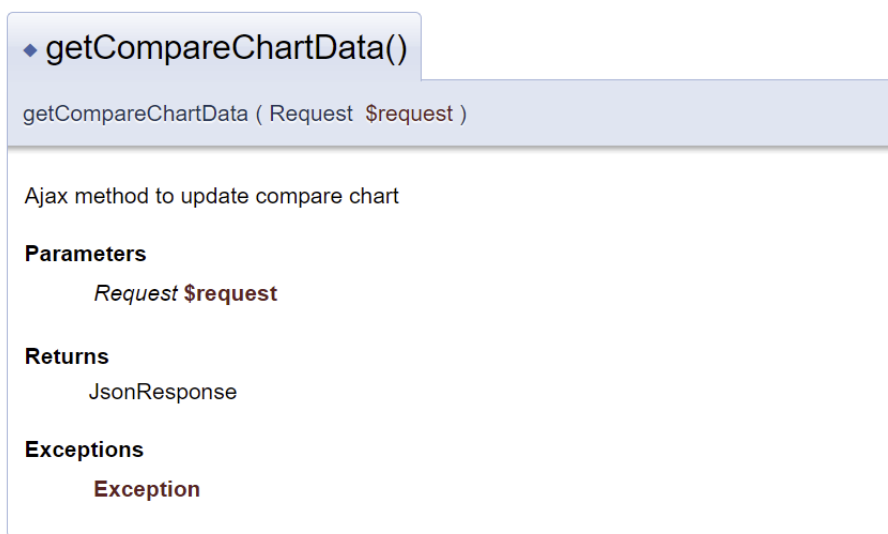


Figura 2: Ejemplo de documentación de una función DoxyGen

## 6. Aspectos relevantes del desarrollo

---

En este apartado se expondrán las fases más importantes y representativas del desarrollo del proyecto.

### 6.1 Marco de trabajo

---

En la realización del proyecto se ha utilizado el Proceso Unificado como marco de trabajo. Se caracteriza por:

- Está centrado en la arquitectura: se parte de una arquitectura inicial, formada por múltiples modelos y vistas, que se irá mejorando y refinando hasta obtener el resultado final.
- Está dirigido por los casos de uso: se utilizan para analizar los requisitos funcionales. También guiarán cada iteración, es decir, se tomará un conjunto de casos de uso en cada iteración que se desarrollará en las fases de Diseño, Implementación y Prueba.
- Es un proceso iterativo e incremental: está compuesto por cuatro fases llamadas Inicio, Elaboración, Construcción y Transición, que están divididas a su vez en iteraciones. Cada una de estas iteraciones da como resultado un incremento del producto que añade o mejora funcionalidades del sistema desarrollado.

Estas cuatro fases del Proceso Unificado tienen como objetivo:

- **Inicio:** definir el negocio, los deseos del usuario, presentar un modelo, unos plazos, costes y viabilidad.
- **Elaboración:** obtener una visión refinada del proyecto, donde aparecerán nuevos requisitos y se recalcularán las estimaciones.
- **Construcción:** desarrollar el proyecto.
- **Transición:** última fase, en la que el proyecto debe estar listo para ser probado, entregado y utilizado sin problemas.

## Memoria del proyecto

Al finalizar una fase se realiza un hito principal. Estos permiten revisar el progreso del proyecto. A su vez, cada fase puede estar compuesta por varias iteraciones secundarias, que finalizan con un hito secundario.

En la siguiente figura se pueden ver las fases e iteraciones del Proceso Unificado. También se aprecian las diferentes disciplinas que permiten agrupar las tareas que tienen características comunes

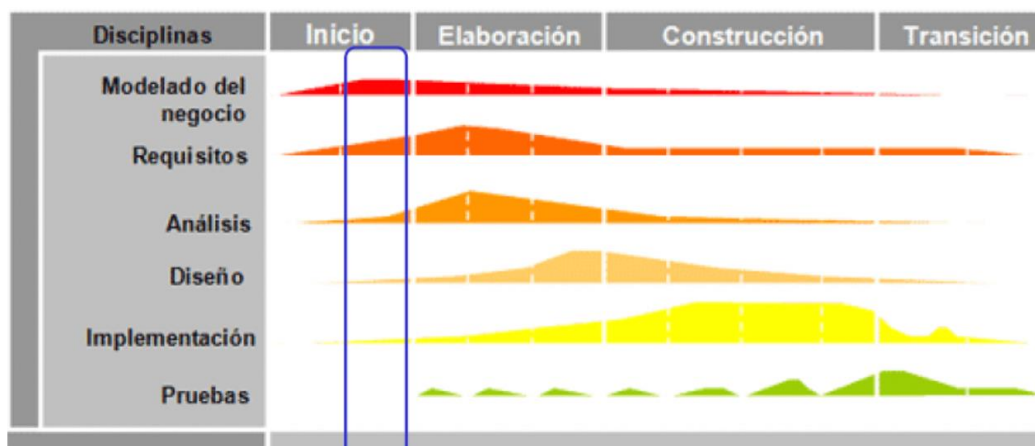


Figura 3: Fases e iteraciones del Proceso Unificado

## 6.2 Planificación temporal

---

Gracias a la planificación temporal es posible conocer todas las tareas necesarias para llevar a cabo el proyecto, así como el tiempo que se empleará en cada una de ellas. Con esta información podemos estimar la fecha de finalización del proyecto.

Para obtener más información sobre la planificación temporal se puede acceder al documento *Anexo I – Plan de proyecto* que acompaña esta memoria.

En las siguientes figuras se detallan las tareas del presente proyecto, así como su diagrama de Gantt.

# Memoria del proyecto

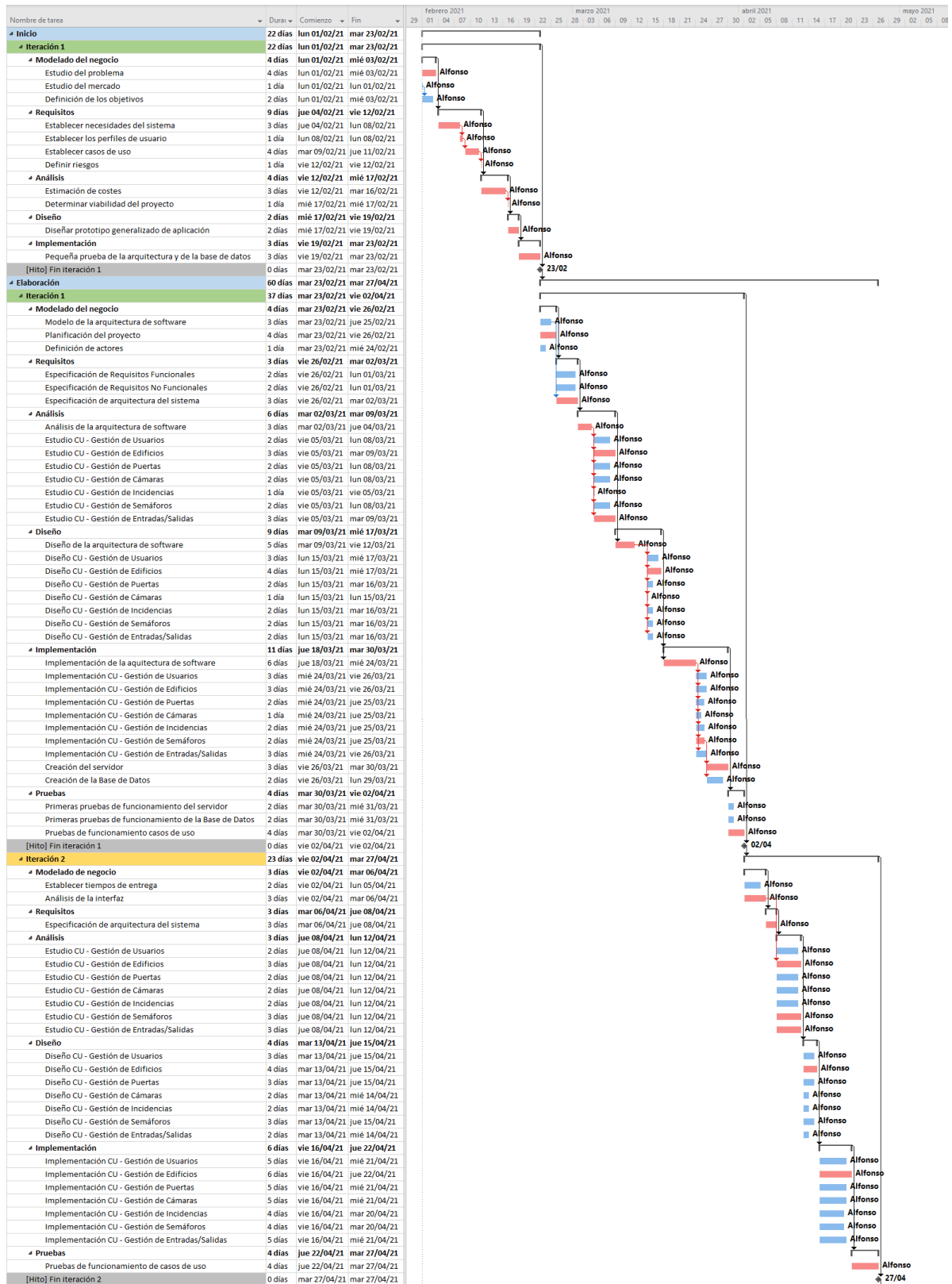


Figura 4: Diagrama de Gantt I

# Memoria del proyecto

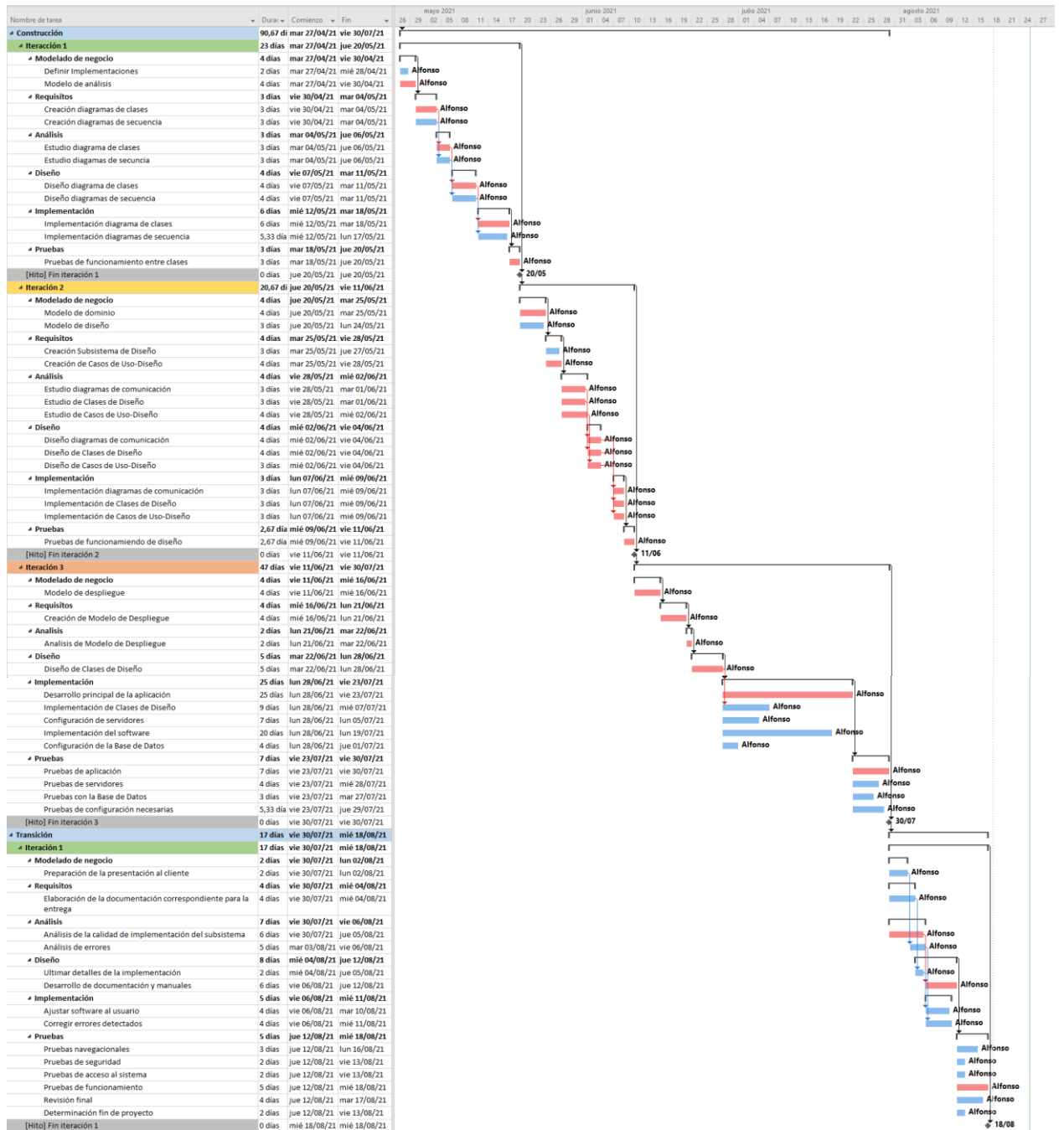


Figura 5: Diagrama de Gantt II

### 6.3 Especificación de requisitos

---

En esta fase se recogen los objetivos que se desean alcanzar en el proyecto, lo que servirá de base para las siguientes fases. Para consultar más información sobre la especificación de requisitos del sistema se puede consultar el documento *Anexo II – Especificación de requisitos software* que acompaña esta memoria. A continuación, podemos ver un ejemplo de cada apartado de la especificación de requisitos.

#### 6.3.1 Participantes

---

Los participantes del proyecto son:

- Hernández Noriega, Alfonso
- Villarrubia González, Gabriel
- Sales Mendes, André
- De Paz Santana, Juan Francisco

En la siguiente tabla se muestra un ejemplo de especificación de participantes.

<b>Participante</b>	<b>Hernández Noriega, Alfonso</b>
<b>Organización</b>	<a href="#">Universidad de Salamanca</a>
<b>Rol</b>	Desarrollador
<b>Es desarrollador</b>	Sí
<b>Es cliente</b>	No
<b>Es usuario</b>	No
<b>Comentarios</b>	Ninguno

Tabla 1: Participante Hernández Noriega, Alfonso



### 6.3.2 Objetivos del sistema

---

Los objetivos que se han definido para el sistema son:

- Detección de clientes y uso de mascarilla
- Control de aforo
- Gestión de usuarios
- Gestión de edificios
- Gestión de puertas
- Control de incidencias
- Gestión de semáforos

### 6.3.3 Requisitos de información

---

Estos requisitos establecen que información se va a almacenar en el sistema. Los definidos para este proyecto son:

- Información sobre usuarios
- Información sobre edificios
- Información sobre puertas
- Información sobre cámaras
- Información sobre registros de entrada o salida
- Información sobre incidencias
- Información sobre semáforos

En la siguiente tabla se muestra un ejemplo de especificación de requisito de información.

<b>IRQ-0001</b>	<b>Información sobre usuarios</b>
<b>Versión</b>	1.0
<b>Autores</b>	<a href="#">Alfonso Hernández Noriega</a>
<b>Fuentes</b>	<a href="#">Villarrubia González, Gabriel</a> <a href="#">Sales Mendes, André</a> <a href="#">De Paz Santana, Juan Francisco</a>
<b>Dependencias</b>	<a href="#">[OBJ-0003] Gestión de usuarios</a>
<b>Descripción</b>	El sistema deberá almacenar la información correspondiente a los usuarios registrados en el sistema. En concreto:

IRQ-0001	Información sobre usuarios	
Datos específicos	<ul style="list-style-type: none"> <li>• Identificador (Único)</li> <li>• Correo electrónico (Único)</li> <li>• Nombre</li> <li>• Apellido</li> <li>• Contraseña codificada</li> <li>• Foto de perfil</li> <li>• Rol</li> <li>• Fecha de creación</li> <li>• Fecha de actualización</li> </ul>	
Tiempo de vida	Medio	Máximo
Ocurrencias simultáneas	Medio	Máximo
Importancia	Importante	
Urgencia	Inmediatamente	
Estado	Validado	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 2: Ejemplo de requisito de información

### 6.3.4 Requisitos funcionales

Estos requisitos definen cómo se comporta el sistema en las diferentes situaciones. Primero se ha realizado una división del sistema en paquetes y la relación entre ellos, que se muestra en la siguiente figura.

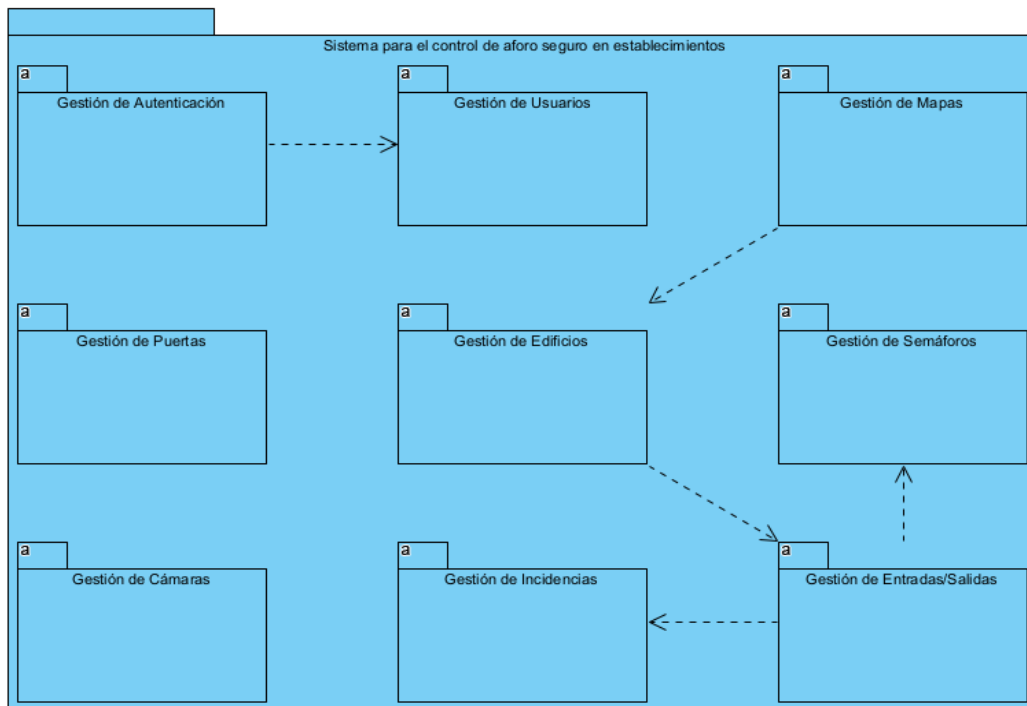


Figura 6: Diagrama de paquetes

## Memoria del proyecto

A continuación, se especifican los actores del sistema, que son los siguientes:

- Usuario No Identificado
- Usuario Identificado
- Usuario Administrador
- Usuario Sistema.

Una vez establecidos los actores se determinan los casos de uso del sistema. Estos se dividen entre los diferentes paquetes y se representan en diagramas de casos de uso. En la siguiente figura podemos ver un ejemplo de diagrama de caso de uso.

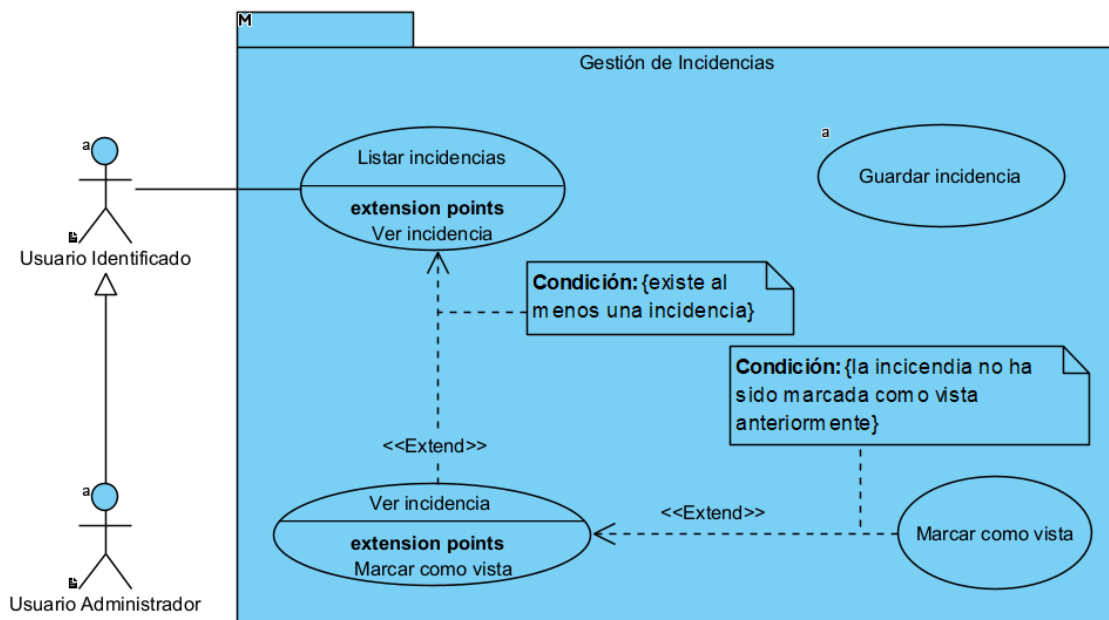


Figura 7: Ejemplo de diagrama de casos de uso: Gestión de Incidencias

También se describe en una tabla la especificación de cada caso de uso. En la siguiente tabla podemos ver un ejemplo de la especificación de un caso de uso.

<b>UC-0029</b>	<b>Añadir puerta</b>
<b>Versión</b>	1.0
<b>Autores</b>	<a href="#">Alfonso Hernández Noriega</a>

<b>UC-0029</b>	<b>Añadir puerta</b>	
<b>Fuentes</b>	<a href="#">Villarrubia González, Gabriel</a> <a href="#">Sales Mendes, André</a> <a href="#">De Paz Santana, Juan Francisco</a>	
<b>Dependencias</b>	<a href="#">[OBJ-0005] Gestión de puertas</a> <a href="#">[IRQ-0003] Información sobre puertas</a>	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización de los siguientes casos de uso: <a href="#">Listar puertas (UC-0026)</a>	
<b>Precondición</b>	El usuario tiene al menos un edificio	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor <a href="#">Usuario Identificado (ACT-0002)</a> solicita <i>añadir una nueva puerta</i> .
	2	El sistema <i>muestra el formulario en una nueva página</i> .
	3	El actor <a href="#">Usuario Identificado (ACT-0002)</a> <i>rellena los campos de las puertas deseadas, sus cámaras asociadas, selecciona un edificio y confirma la operación</i> .
	4	El sistema <i>comprueba la información y añade la(s) puerta(s) al sistema, así como su(s) cámara(s) asignadas</i> .
<b>Postcondición</b>	La puerta y toda su información quedan almacenados en el sistema.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	4	Si <i>alguno de los campos está vacío</i> , el sistema <i>cancela la operación e informa al usuario</i> , a continuación, el caso de uso <i>queda sin efecto</i> .
<b>Rendimiento</b>	<b>Paso</b>	<b>Tiempo máximo</b>
	-	-
<b>Frecuencia esperada</b>	2 veces por mes(es)	
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Inmediatamente	
<b>Estado</b>	Validado	
<b>Estabilidad</b>	Alta	
<b>Comentarios</b>	Ninguno	

Tabla 3: Ejemplo especificación caso de uso

### 6.3.5 Requisitos no funcionales

Estos requisitos se definen para imponer alguna restricción de diseño o estándar de calidad. A continuación, se establecen los requisitos no funcionales establecidos para este proyecto. Cada uno de ellos se desarrolla en una tabla como la siguiente.

- Usabilidad
- Compatibilidad
- Hacer uso de un ordenador por cámara

## 6.4 Análisis de requisitos

En la fase de análisis del sistema se desarrolla cada uno de los requisitos especificados en la fase anterior. Como resultado se obtendrá una comprensión más precisa del sistema y su estructura.

Para ver la información más detallada se puede consultar el *Anexo III – Análisis de requisitos* que acompaña esta memoria.

### 6.4.1 Modelo de dominio

El modelo de dominio permite identificar las clases más significativas en el dominio del problema. Además, recoge las necesidades de almacenamiento en la base de datos. En la siguiente figura se puede ver el modelo de dominio de este proyecto.

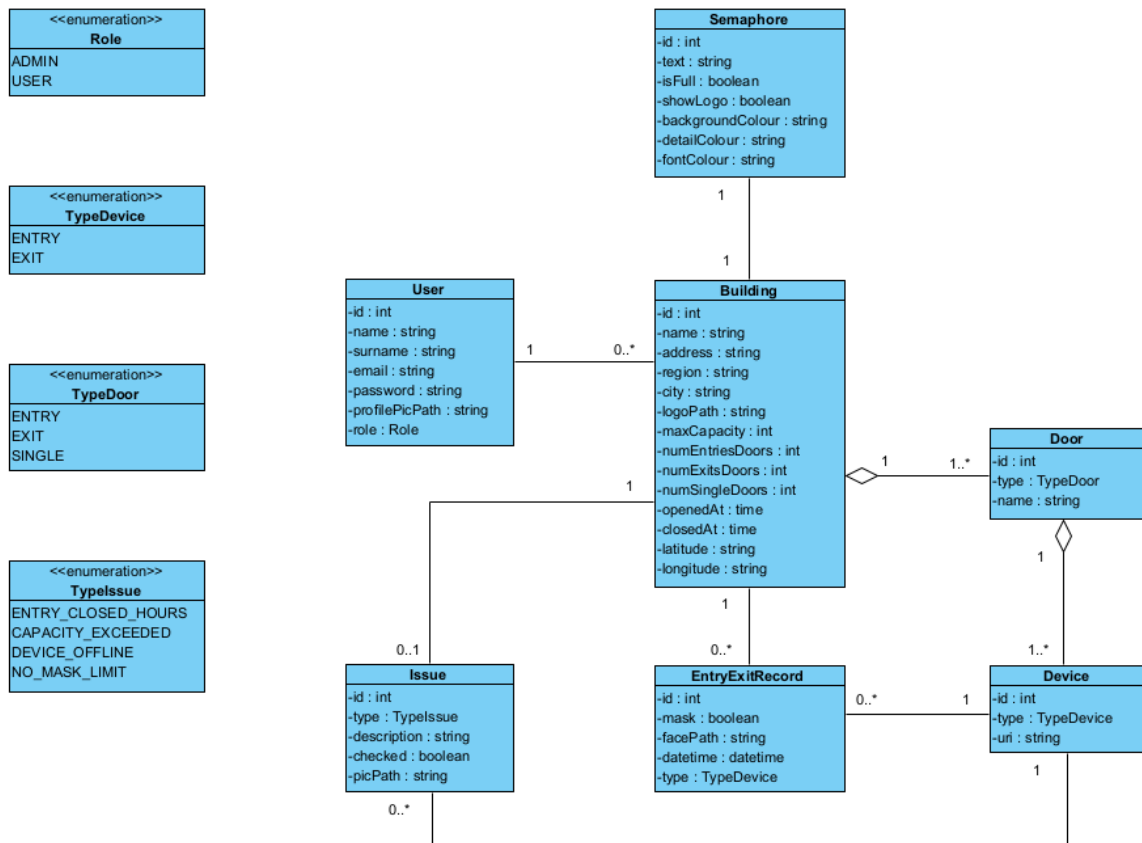


Figura 8: Diagrama de clases Modelo de Dominio

## 6.4.2 Realización de casos de uso - Análisis

En ese apartado se detallarán los casos de uso de los distintos paquetes en diagramas de secuencia. En ellos podemos ver las comunicaciones entre los elementos del sistema. En la siguiente figura podemos ver un ejemplo de diagrama de secuencia.

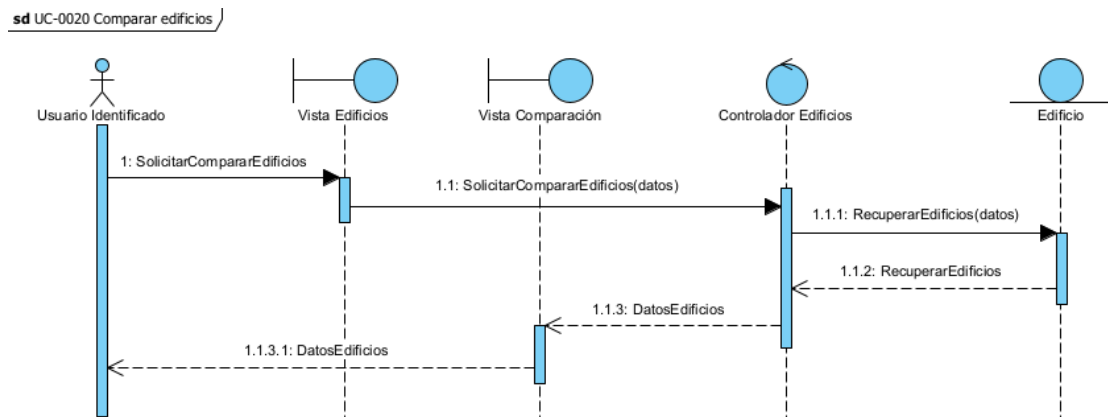


Figura 9: Ejemplo diagrama de secuencia.

## 6.4.3 Clases de análisis

En este apartado se han realizado diagramas de comunicación, que representan la relación entre las clases planteadas en los diagramas de los casos de uso. En la siguiente figura podemos ver un ejemplo de diagrama de comunicación de clases de análisis.

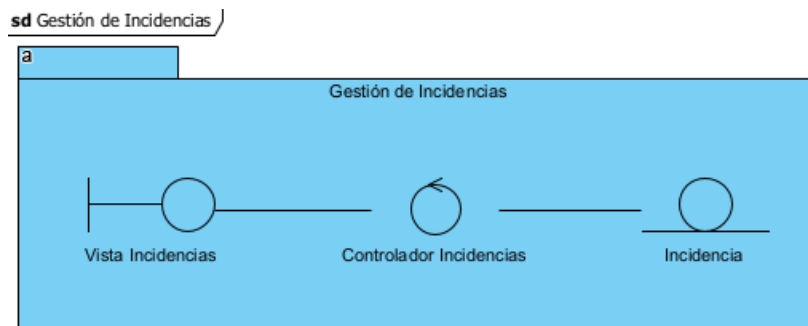


Figura 10: Diagrama de comunicación del paquete Gestión de Incidencias

### 6.4.4 Vista arquitectónica

---

En este apartado se distribuyen todas las clases de los apartados anteriores mediante el patrón Modelo-Vista-Controlador. La arquitectura obtenida será una aproximación, que se refinará en siguientes pasos.

## 6.5 Diseño del sistema

---

Esta fase se centra en el dominio de la solución, con la que se obtiene una aproximación muy cercana a la implementación final.

Para ver la información más detallada se puede consultar el *Anexo IV – Diseño del sistema software* que acompaña esta memoria.

### 6.5.1 Patrones arquitectónicos

---

El patrón principal usado en el proyecto es el patrón arquitectónico Modelo-Vista-Controlador. Además, se ha utilizado el patrón Publicador – Suscriptor, para mantener actualizada la información de las tablets (o semáforos) asociadas a los edificios.

El patrón Modelo-Vista-Controlador separa el modelo de datos, la interfaz del usuario y la lógica de control, de tal manera que sean independientes entre sí. Está compuesto por:

- **Modelo:** representa los datos del sistema, junto con su lógica de negocio.
- **Vista:** representa la información a través de las interfaces de usuario.
- **Controlador:** actúa como intermediario entre el Modelo y la Vista, dando la funcionalidad necesaria a la aplicación.

El patrón Publicador – Suscriptor, o *Publisher-Subscriber*, es un patrón basado en el envío de mensajes por parte de los publicadores, que reciben los suscriptores que deseen. Para que un suscriptor pueda recibir un mensaje

## Memoria del proyecto

primero debe suscribirse a un canal. A través de estos canales el publicador transmite la diferente información. Es una comunicación indirecta, en la que no hay que conocer la existencia entre los dos interlocutores. Esto evita que las entidades queden a la espera o bloqueen el canal de comunicación hasta que les llega un mensaje.

### 6.5.2 Subsistemas de diseño

---

En este apartado se recogen los diagramas que representan las distintas partes del sistema y sus relaciones. En la siguiente figura se pueden ver los subsistemas del proyecto.

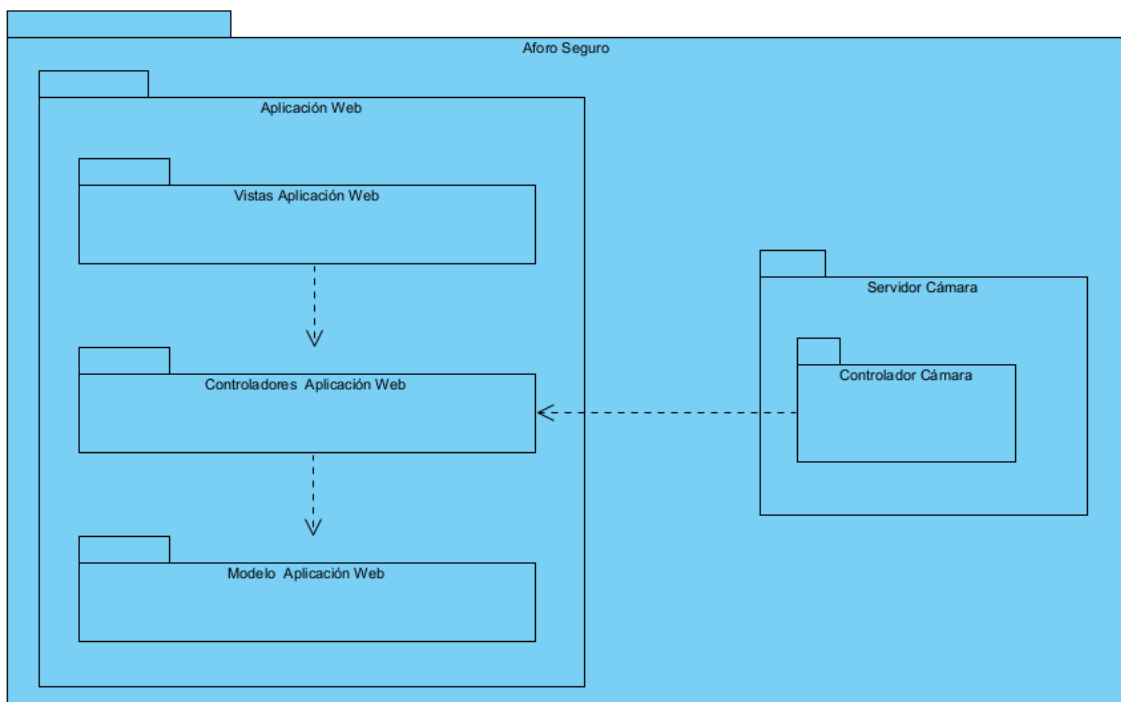


Figura 11: Modelo de diseño



### 6.5.3 Clases de diseño

Este apartado de clases de diseño se ha dividido en tres partes, una por cada capa del patrón Modelo-Vista-Controlador. Se especifica el nombre de cada clase, así como los métodos de la misma.

En la siguiente figura aparece un ejemplo que corresponde con la capa controladora de la aplicación web.

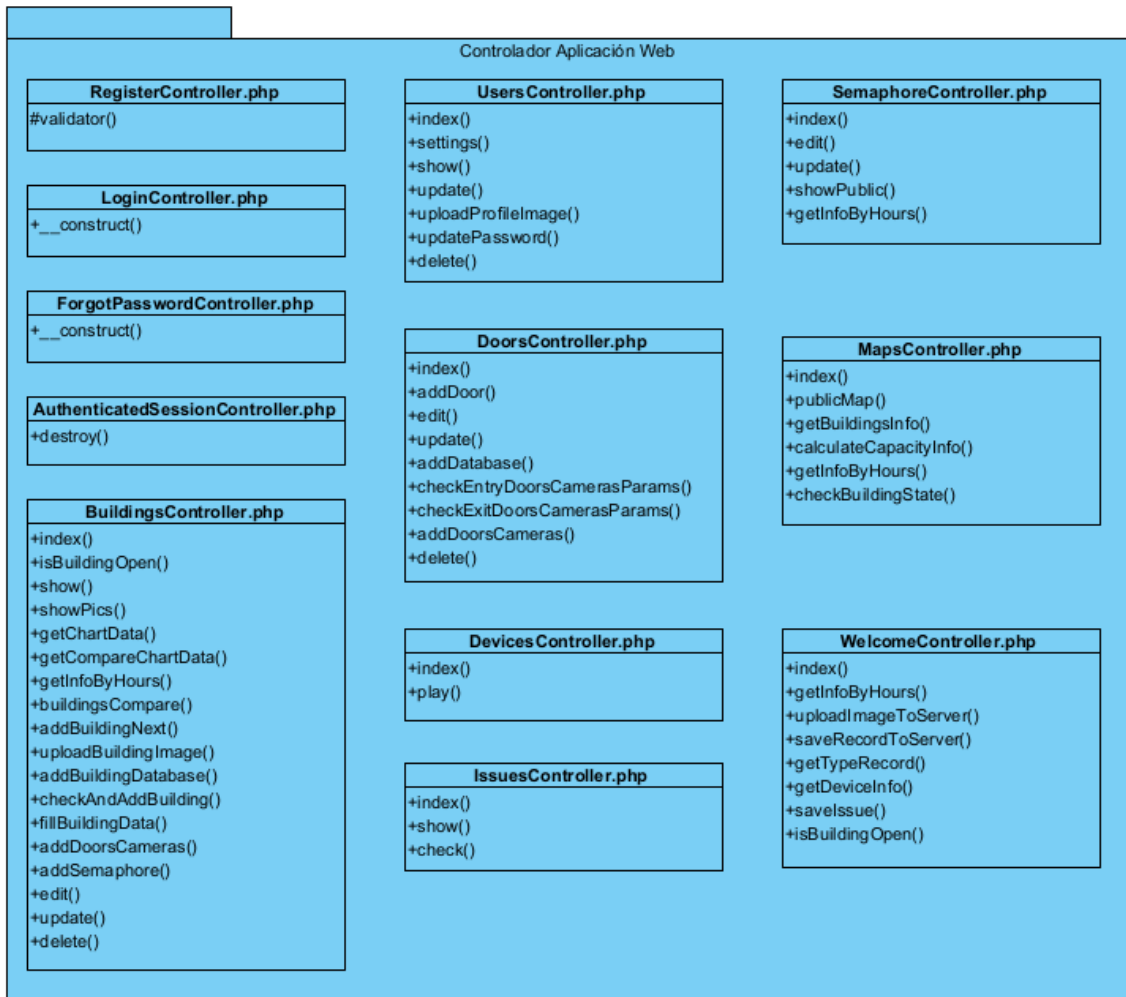


Figura 12: Capa controlador de la aplicación web

### 6.5.4 Vista arquitectónica del modelo de diseño

---

En este apartado los subpaquetes del sistema se ordenan por la capa del patrón arquitectónico a la que pertenece. En la siguiente figura se muestra el reparto en este proyecto.

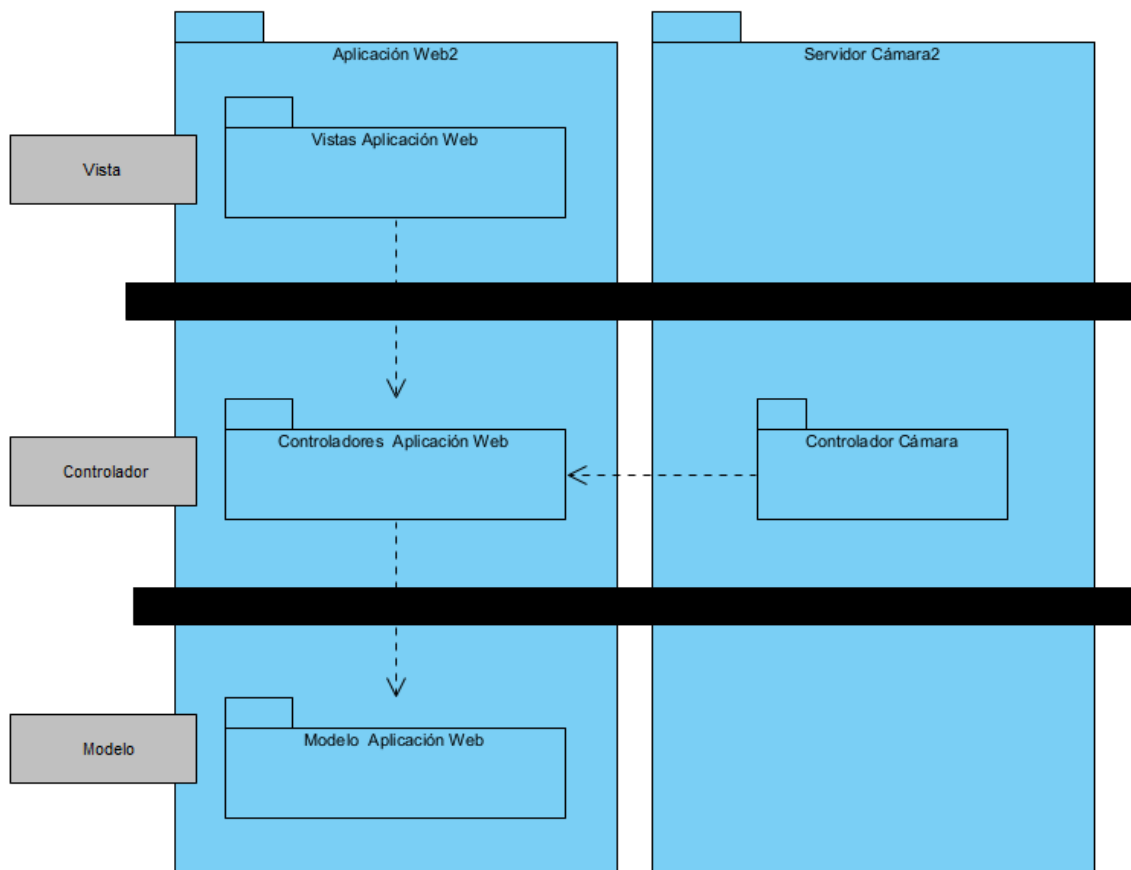


Figura 13: Vista arquitectónica

## 6.5.5 Realización de casos de uso diseño

En este apartado se desarrollan los diagramas de secuencia, en los que podemos ver los nombres y métodos de las clases que se van a usar en la implementación. En la siguiente figura podemos ver un ejemplo de estos diagramas.

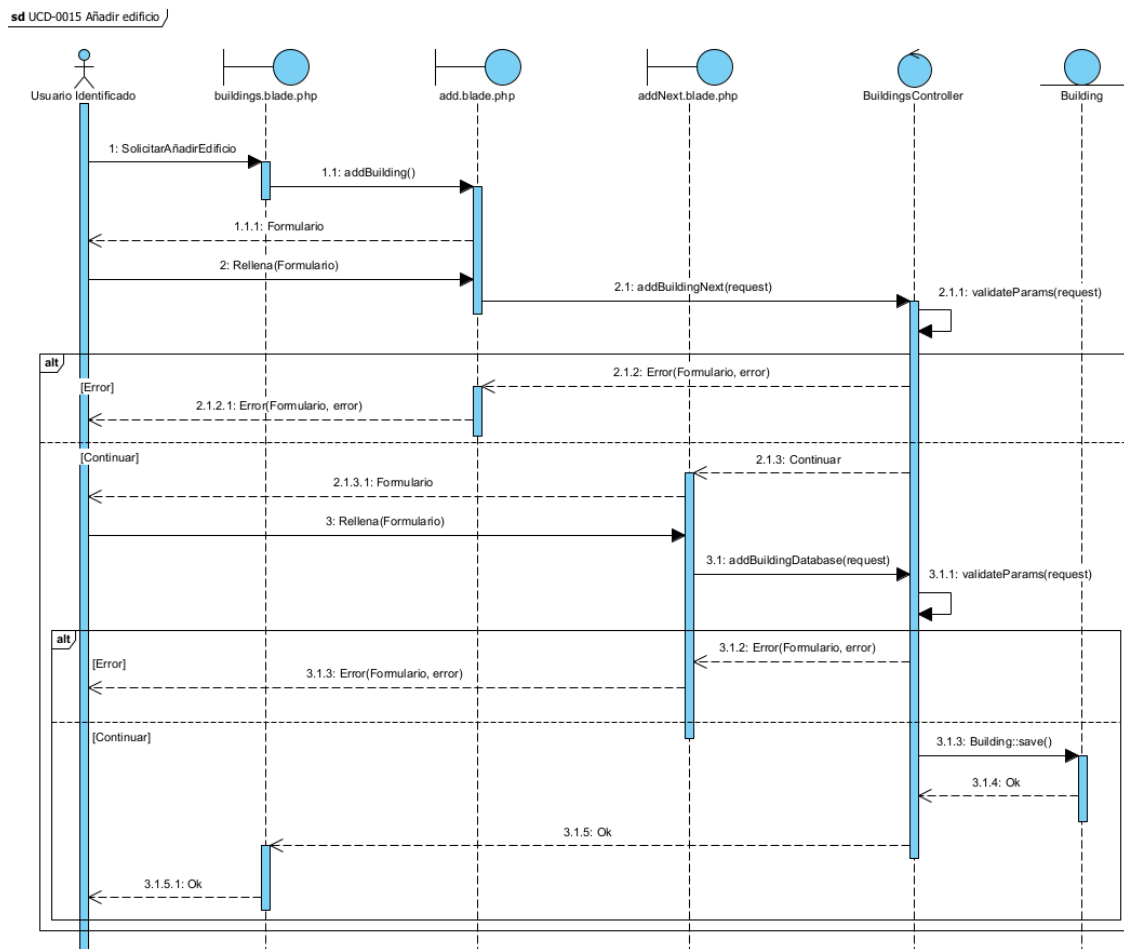


Figura 14: Ejemplo de diagrama de secuencia diseño

## 6.5.6 Diseño de la base de datos

Este apartado se detalla el diseño de la base de datos que el sistema usará para almacenar toda la información necesaria para su funcionamiento. El sistema utiliza una base de datos MySQL para almacenar y gestionar toda la información necesaria para su funcionamiento. En la siguiente figura se muestran las relaciones de dicha base de datos.

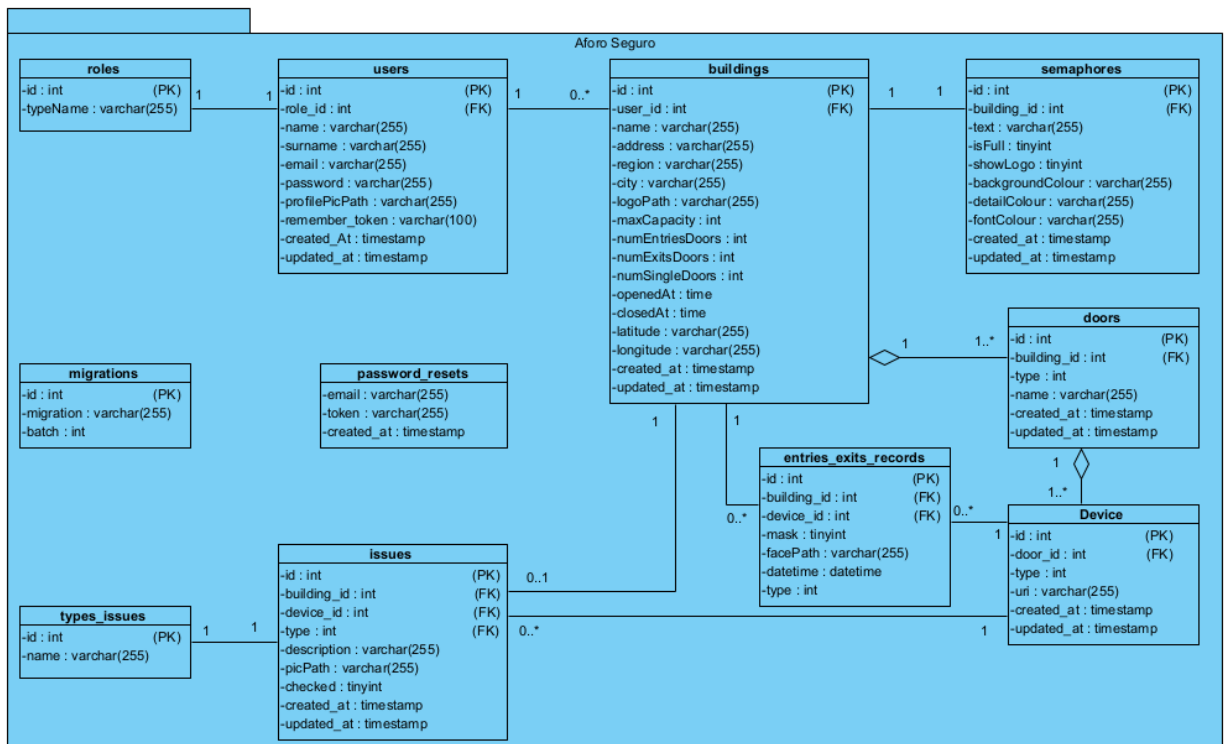


Figura 15: Base de datos

### 6.5.7 Modelo de despliegue

---

En el último apartado de la fase de diseño se ha realizado un diagrama de despliegue, que permite ver como se organizan y como se conectan los nodos del sistema en el proyecto. En la siguiente figura vemos los tres tipos de nodos:

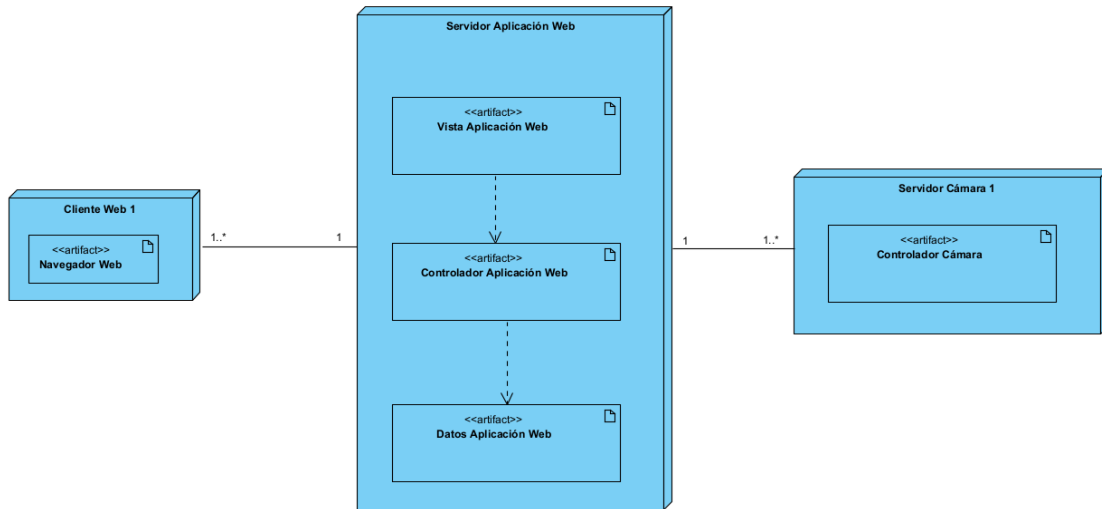


Figura 16: Modelo de despliegue

- **Cliente Web:** es la página web en el artefacto Navegador Web a la cual accederán los usuarios, tanto registrados como no registrados. Cada uno entrará con su dispositivo con su propia instancia del nodo, por lo que habrá múltiples instancias de este tipo de nodo.
- **Servidor Aplicación Web:** es el servidor en el cual está almacenada toda la información los edificios, junto con sus registros de entradas y salidas. Servirá las páginas a los nodos *Cliente Web* y también se encargará de recibir los registros que envíen los nodos *Servidor Cámara*. Solo habrá una instancia de este tipo de nodo.
- **Servidor Cámara:** es el servidor que se instalará en cada puerta junto con una cámara. Esta se encargará de monitorizar a todas las personas que entren (o salgan) del edificio y, cada vez que ocurra, mandará la información de esa entrada (o salida) al *Servidor Aplicación Web*. Habrá tantas instancias de este nodo como puertas en el edificio. En el caso de haber una puerta única para entrar y salir, habrá dos instancias de este nodo.

### 6.6 Implementación

---

Esta fase es la más larga de todas, donde tiene lugar la implementación del sistema. Se codifica toda la funcionalidad utilizando las herramientas y técnicas vistas en el apartado 5 de este documento.

El diseño realizado anteriormente sirve ahora para guiar y facilitar el proceso, consiguiendo como resultado un proyecto ordenado. La implementación se ha realizado en dos partes diferenciadas:

- **Aplicación Web:** servidor que aloja la página de la aplicación web. Esto supone implementar todas las funcionalidades vistas anteriormente, así como las vistas necesarias para que los usuarios puedan hacer uso de las mismas.
- **Servidor cámara:** este servidor lleva instalada una cámara para poder monitorizar todas las entradas o salidas que se producen en una puerta. Se implementa un sistema de detección de personas y una comunicación con el servidor principal.

Para facilitar la comprensión del código a otros programadores, se ha generado una documentación técnica de todo el código del proyecto. Puede encontrarse en el documento *Anexo V – Documentación técnica* que acompaña esta memoria.

### 6.7 Pruebas

---

Es uno de los puntos más importantes, que permite comprobar si la funcionalidad del sistema es correcta y si se cumplen los objetivos marcados.

Durante el desarrollo de cada componente del sistema se ha llevado a cabo pruebas unitarias, comprobando que el comportamiento sea el deseado. Al acabar el desarrollo se realizan pruebas unitarias con los componentes que tenga relación, para confirmar que el módulo funciona correctamente.

Las pruebas se enmarcaron en el supuesto de dos cámaras en dos puertas de un establecimiento ficticio. Esto permite validar el correcto funcionamiento del proyecto, pero lo ideal sería trasladarlo a un entorno de pruebas real, con el correspondiente flujo de clientes.

### 6.8 Funcionalidad del sistema

---

En esta sección se resumen las funcionalidades principales del sistema, clasificadas según los diferentes roles de usuario del sistema. Para ver una información más detallada se puede consultar el *Anexo VI – Manual de Usuario*.

#### 6.8.1 Usuario No Identificado

---

Es aquel usuario que accede a la web para consultar el estado de alguna tienda o establecimiento de su interés.

También permite las acciones relacionadas con la autenticación.

##### 6.8.1.1 Acceder a la página pública

---

Esta es la página principal que se carga al acceder a la web. Desde aquí se puede iniciar sesión, registrar un nuevo usuario y consultar el estado de cualquier edificio añadido al sistema.

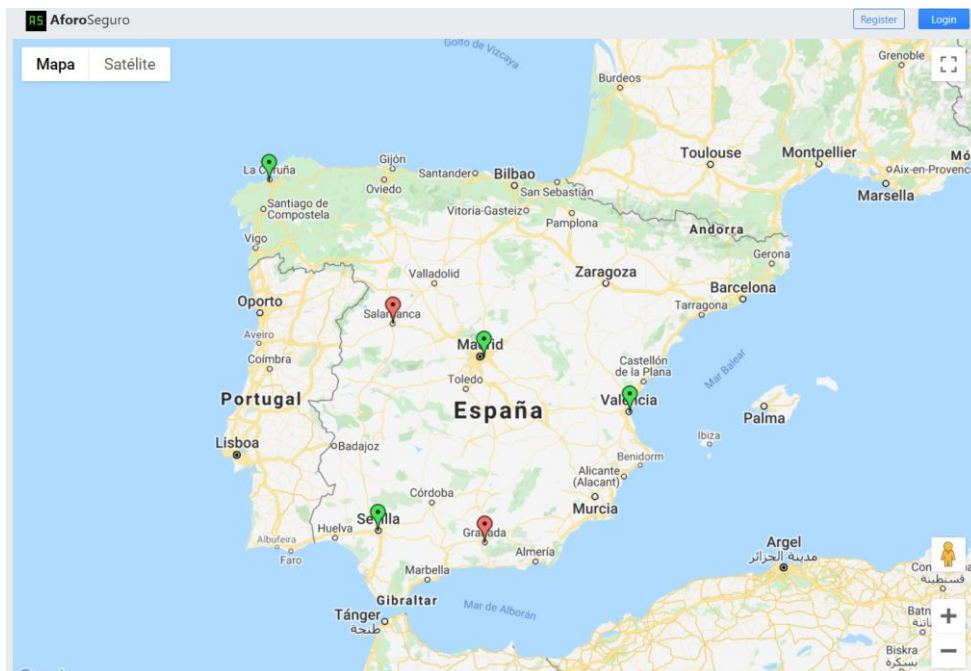


Figura 17: Página pública



## Memoria del proyecto

El color de las etiquetas indica el estado del edificio, verde si está abierto y rojo si está cerrado. Además, si el usuario selecciona alguno de los marcadores se abrirá un cuadro como el de la siguiente figura, con información detallada del edificio en concreto.

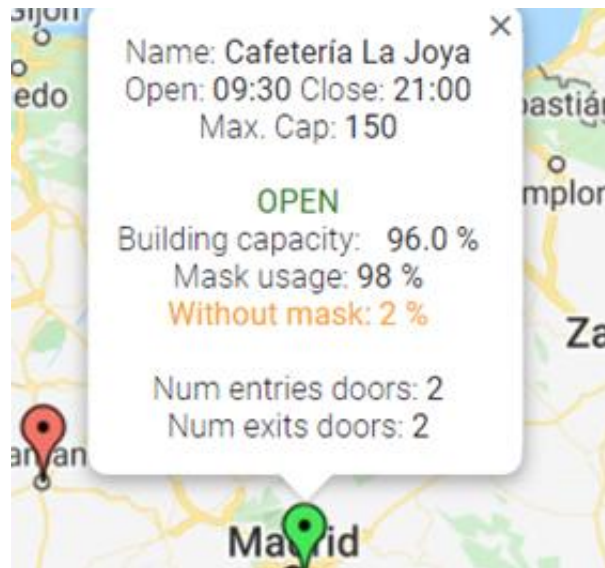


Figura 18: Información detallada

### 6.8.1.2 Iniciar sesión

---

Cuando el usuario quiera iniciar la sesión en su cuenta deberá introducir su correo y contraseña en el siguiente formulario de la página de identificación.

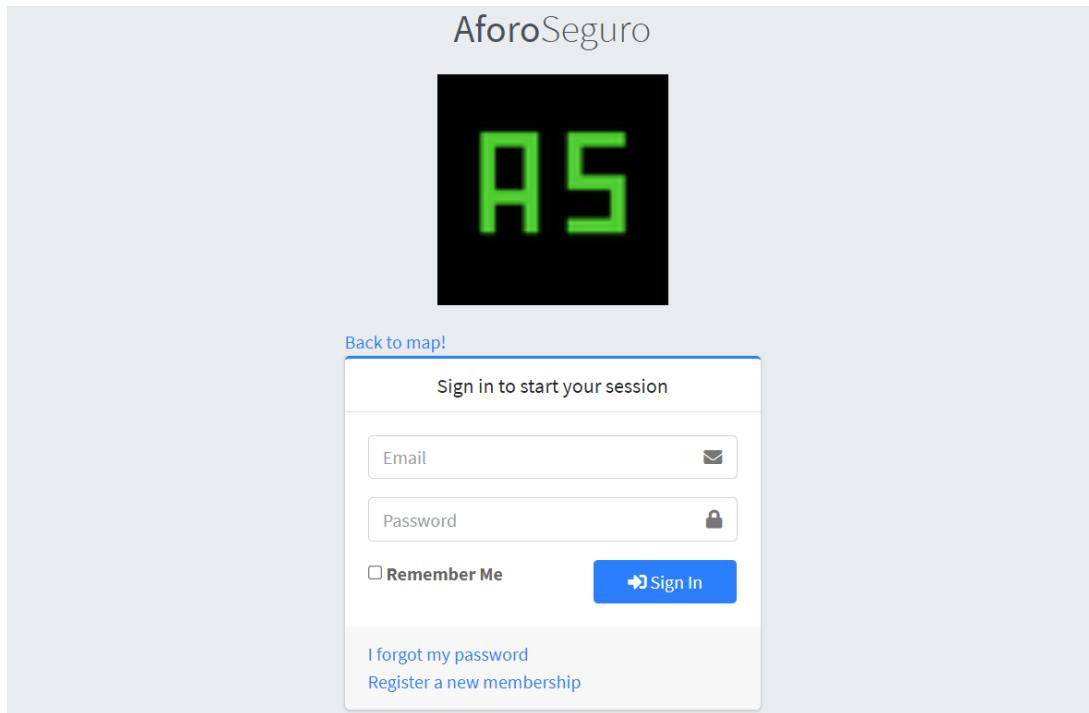


Figura 19: Iniciar sesión

Si alguno de los datos introducidos no coincide con lo almacenado en el sistema devolverá los errores a este mismo formulario. Si los datos con correctos, redirige al usuario a su página privada de bienvenida.

Además, desde esta página se puede solicitar crear un nuevo usuario y recuperar la contraseña con el correo electrónico.

### 6.8.2 Usuario Identificado

---

Este usuario deber ser el propietario de un establecimiento que contrate el servicio de la aplicación. Se realizará una instalación de las cámaras y los servidores en el edificio y se le proporcionará la información necesaria para poder gestionar su establecimiento.

### 6.8.2.1 Página privada de bienvenida

---

Al iniciar sesión cargará la página de la siguiente figura.

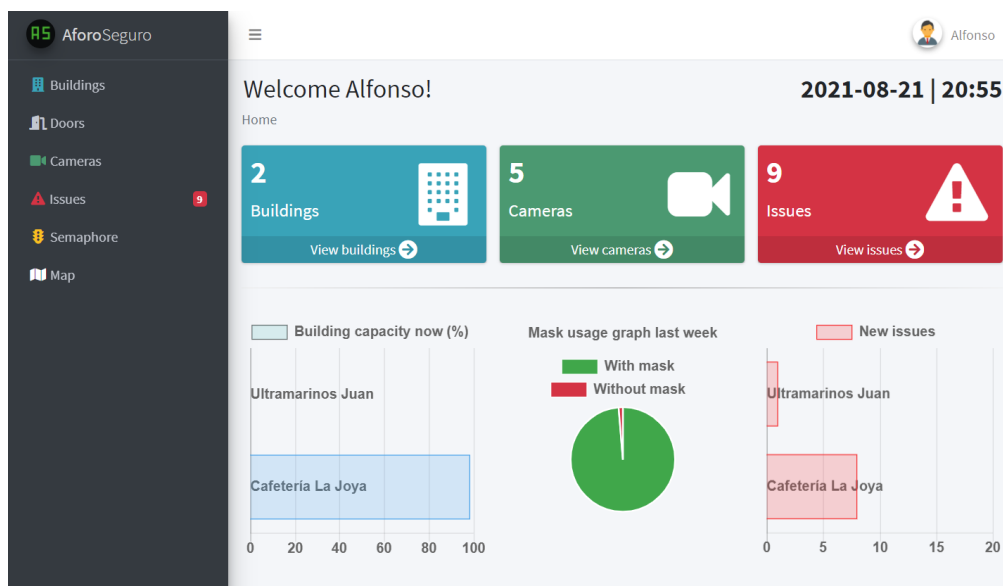


Figura 20: Página privada de bienvenida

Desde aquí podemos ver los gráficos que resumen el estado de cada edificio en tiempo real. También podemos acceder a las diferentes secciones de la aplicación, mediante los enlaces del menú lateral. Dicho menú siempre estará presente mientras naveguemos por la web, pero podrá replegarse ocupar menos espacio y permitir una mejor visualización.

Además, en la esquina superior derecha siempre estará accesible un desplegable que permite cerrar la sesión y ver el perfil.

### 6.8.2.2 Ver perfil

---

Si en la siguiente figura se selecciona la opción de ver su perfil se cargará la página de la siguiente figura.

## Memoria del proyecto

Settings

**Name**  
Alfonso

**Surname**  
Hernandez Noriega

**Email**  
alfonsohnoriega@gmail.com

**Profile pic**  
Choose a file... Browse

Update

**Current password**

**New password**

**Repeat new password**

Change password

I confirm I want to delete my whole account. I understand I'll lose everything related to it.

Delete account

Figura 21: Ver perfil

Desde aquí se podrá modificar los datos personales del usuario: nombre, apellidos, foto de perfil y contraseña. No se permite la modificación del correo electrónico. Si alguno de los campos está vacío o las contraseñas no son correctas, devolverá al formulario los errores.

También permite borrar la cuenta y todos los edificios asociados a ella.

### 6.8.2.3 Listado de edificios

---

Si en el menú lateral se selecciona la opción de ver edificios, cargará la siguiente página con un listado de todos los edificios del usuario.

The screenshot displays a web interface for managing buildings. At the top, there are buttons for 'Compare' and 'Add building'. Below these, a search bar and a 'Show 10 entries' dropdown are visible. The main content is a table with the following data:

Name	Address	City	Capacity	Doors	Open/Close	Compare	Actions
<a href="#">Cafetería La Joya</a>	Calle de la Raya	Madrid	150	4		<input type="checkbox"/>	
<a href="#">Ultramarinos Juan</a>	Avda de la universidad 27	La Coruña	123	1		<input type="checkbox"/>	

At the bottom, it shows 'Showing 1 to 2 of 2 entries' and pagination controls for 'Previous', '1', and 'Next'.

Figura 22: Listado de edificios

Desde aquí podemos añadir un nuevo edificio y comparar información entre más de dos edificios seleccionados.

También se puede ver [ ] el estado de un edificio, editar [ ] la información del edificio, ir a la página de edición [ ] de su semáforo y borrar [ ] un edificio. Para borrar un edificio se pedirá la confirmación al usuario y se mostrará un aviso que toda la información relacionada (puertas, cámaras, incidencias y registros) será eliminada.

### 6.8.2.4 Añadir un edificio

Si se selecciona la opción de añadir un nuevo edificio cargará la siguiente página, en la que se deberán rellenar todos los campos con la información del nuevo edificio y seleccionar su ubicación en el mapa. Únicamente el logo es opcional, si falta algún otro campo y se pulsa en siguiente aparecerá un listado con los errores.

## Memoria del proyecto

The screenshot shows a web form titled "Add building". At the top left, there is a breadcrumb trail: "Home / Buildings / Add building". The form is organized into several sections:

- Name:** A text input field with the placeholder "Name".
- Logo (optional):** A file upload button with a blue icon and the text "Choose a file...", followed by a "Browse" button.
- Address:** A large text input field with the placeholder "Address".
- Region:** A text input field with the placeholder "Region".
- City:** A text input field with the placeholder "City".
- Opened at:** A time selection field showing "00:00" with a clock icon.
- Closed at:** A time selection field showing "00:00" with a clock icon.
- Max capacity:** A text input field with the placeholder "Max people".

At the bottom right of the form is a blue "Next" button. Below the form is a map of the Iberian Peninsula. The map has two tabs: "Mapa" (selected) and "Satélite". The map shows major cities in Spain (Madrid, Valencia, Barcelona) and Portugal (Lisboa, Oporto). Neighboring countries like Andorra and Monaco are also labeled. A person icon is visible in the bottom right corner of the map area.

Figura 23: Añadir edificio

Si la información es válida aparecerá el siguiente paso para añadir el edificio. En la siguiente figura se puede ver la página que se carga, en la que se puede elegir el número de puertas de entrada y salida que tiene el edificio.

Por cada puerta se debe introducir obligatoriamente un nombre y su dirección asociada, proporcionada por los técnicos de la aplicación. Si falta alguno de estos campos volverá a la vista mostrando los errores. Si toda la información es válida se añade el edificio y muestra un mensaje de éxito en la lista de edificios.

## Memoria del proyecto

**Add building**  
Home / Buildings / Add building

[Back](#) [Add Building](#)

**Number of doors**

**Entry**  **Exit**   Single entry and exit door

**Entry door 1**

**Name**

**Associated camera URI**

**Entry door 2**

**Name**

**Associated camera URI**

**Exit door 1**

**Name**

**Associated camera URI**

Figura 24: Añadir edificio, último paso

Si se marca la casilla de una única puerta de entrada y salida se desactivarán los selectores entrados/salidas y se mostrará el siguiente formulario. Todos los campos son obligatorios. La dirección asociada a cada cámara será proporcionada por los técnicos de la aplicación.

**Entry**  **Exit**   Single entry and exit door

**Single door**

**Name**

**Associated entry camera URI**

**Associated exit camera URI**

Figura 25: Añadir edificio, puerta única

### 6.8.2.5 Comparar edificios

Si desde la lista de edificios se seleccionan al menos dos de ellos y se pulsa la opción de comparar, cargará la siguiente página con la información de los edificios. En ella se puede observar, para un día, el aforo por horas de los edificios, así como el porcentaje de uso de mascarillas en cada edificio. Se puede seleccionar otro día diferente en el calendario para ver la comparación de edificios en esa fecha.

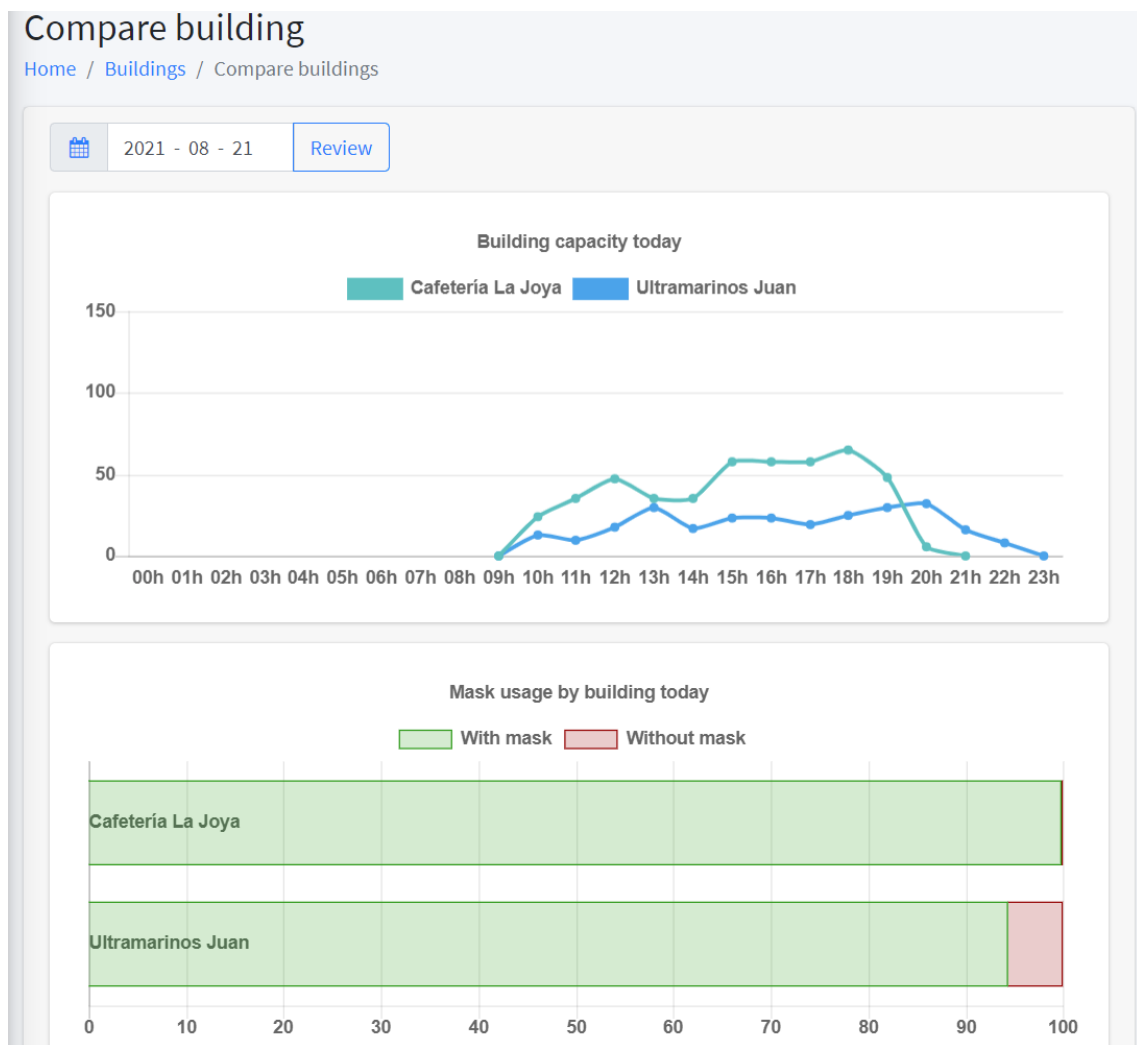


Figura 26: Comparar edificios



### 6.8.2.6 Ver estado de un edificio

Si se selecciona la opción de ver [👁] el estado de un edificio en concreto cargará la siguiente página, con el grafico de aforo y uso de mascarillas. Si se selecciona un día en el calendario se cargará la información de dicha fecha.



Figura 27: Ver edificio

Si se selecciona la opción de ver fotografías cargará la página de la siguiente figura, en la que se muestran las instantáneas de las entradas y salidas de los clientes. Se puede modificar el filtro para ver imágenes de las cámaras de entrada o de salida, o de una cámara en particular. También se puede filtrar por horas y según lleven o no mascarilla.

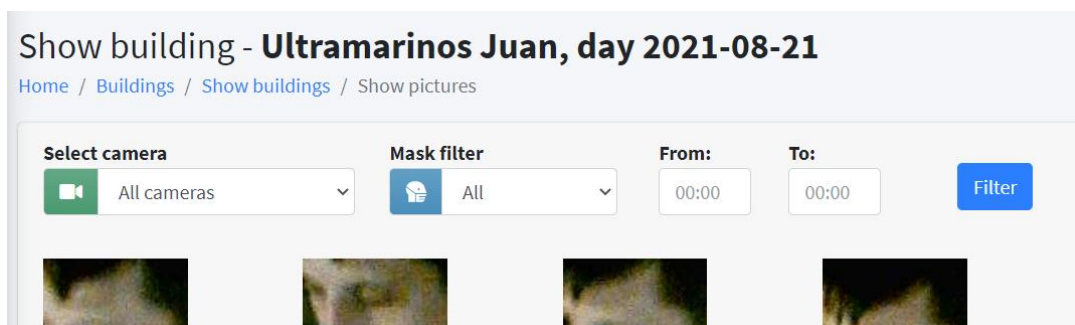


Figura 28: Ver fotografías


Si se pulsa en una foto se abrirá un modal con información más detallada de ese registro.

### 6.8.2.7 Editar edificio

Desde la lista de edificios se puede solicitar editar [✎] un edificio. Cargaré la siguiente página con la información del edificio, la localización en el mapa y su logo. Para editar cualquiera de los campos, se debe introducir la nueva información y solicitar actualizar. Si cumple los requisitos de validación, completará el cambio. Si no mostrará los errores en el formulario.

#### Edit building - Cafetería La Joya, Calle de la Raya

[Home](#) / [Buildings](#) / Edit Building



<b>Name</b>	<input type="text" value="Cafetería La Joya"/>	<b>Logo (optional)</b>	<input type="button" value="Choose a file..."/>	<input type="button" value="Browse"/>		
<b>Address</b>	<input type="text" value="Calle de la Raya"/>					
<b>Region</b>	<input type="text" value="Madrid"/>	<b>City</b>	<input type="text" value="Madrid"/>			
<b>Opened at:</b>	<input type="text" value="09:30"/> <input type="button" value="🕒"/>	<b>Closed at:</b>	<input type="text" value="21:00"/> <input type="button" value="🕒"/>	<b>Max capacity</b>	<input type="text" value="150"/>	<input type="button" value="Update"/>


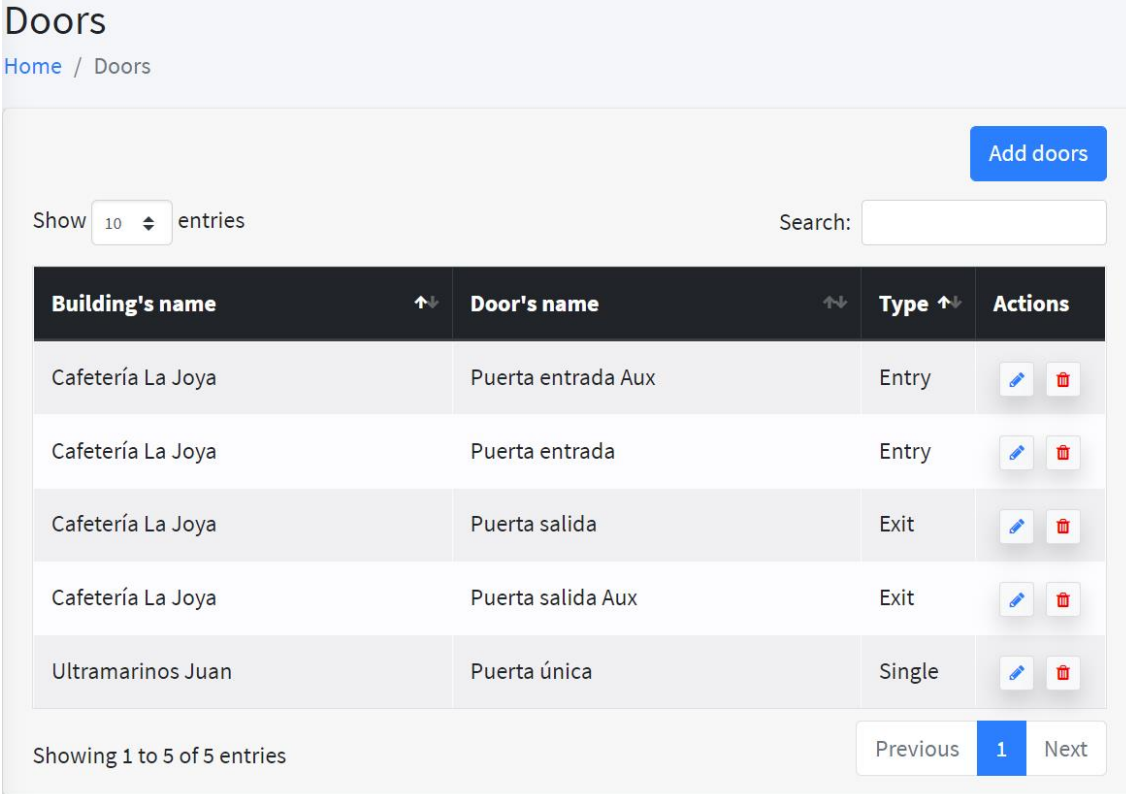


Figura 29: Editar edificio

## 6.8.2.8 Listado de puertas

Para ver el listado de las puertas se debe seleccionar la opción puertas del menú lateral, que devolverá la siguiente vista.



The screenshot shows a web interface titled "Doors" with a breadcrumb "Home / Doors". It features a blue "Add doors" button in the top right. Below the header, there is a "Show 10 entries" dropdown and a "Search:" input field. The main content is a table with the following data:

Building's name	Door's name	Type	Actions
Cafetería La Joya	Puerta entrada Aux	Entry	[edit] [delete]
Cafetería La Joya	Puerta entrada	Entry	[edit] [delete]
Cafetería La Joya	Puerta salida	Exit	[edit] [delete]
Cafetería La Joya	Puerta salida Aux	Exit	[edit] [delete]
Ultramarinos Juan	Puerta única	Single	[edit] [delete]

At the bottom, it says "Showing 1 to 5 of 5 entries" and has pagination controls for "Previous", "1", and "Next".

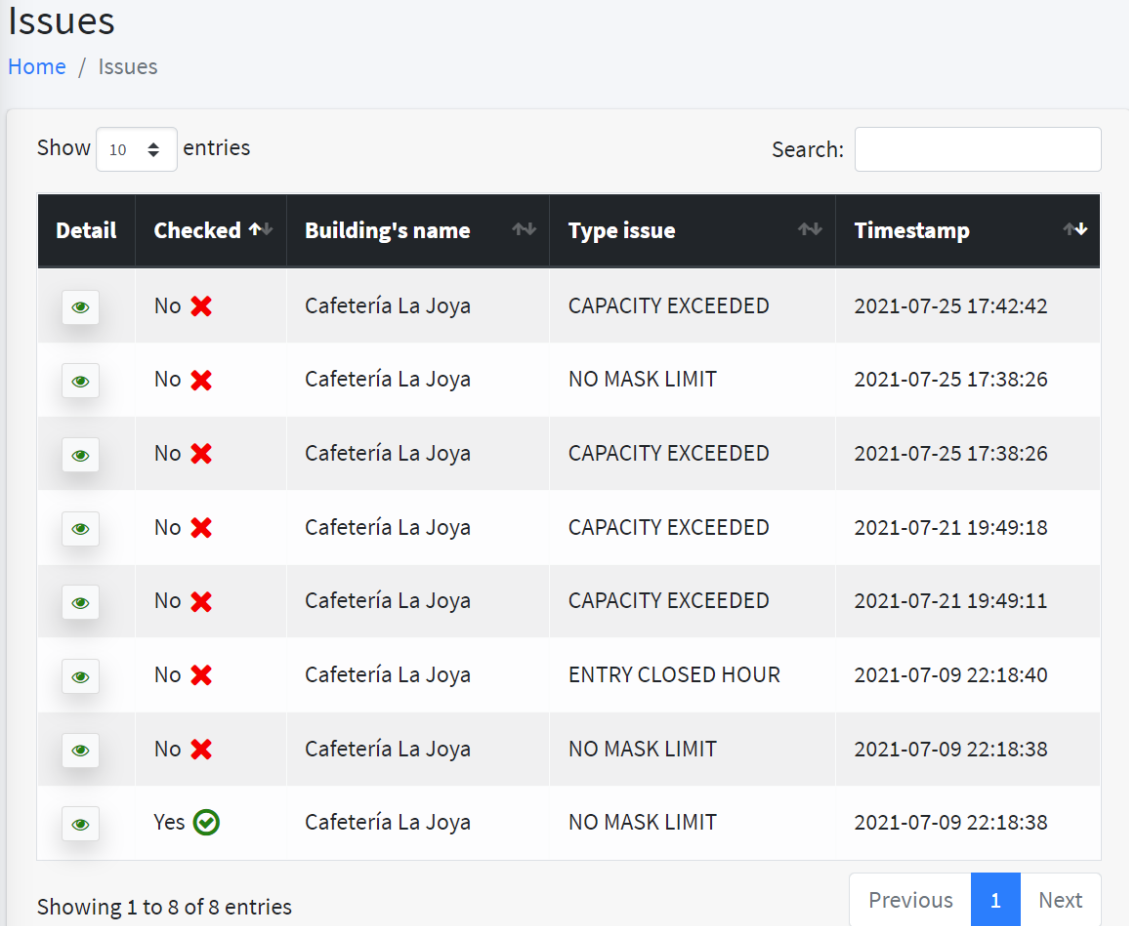
Figura 30: Listado de puertas

Se permite añadir, editar [edit icon] o borrar [delete icon] puertas a un edificio. Para facilitar la visualización de la tabla se puede buscar por nombre, tipo y ordenar las diferentes columnas.

Si se solicita borrar una puerta, se pedirá la confirmación antes de completar el proceso. Se mostrará un aviso que se va a eliminar también la(s) cámara(s) asociada(s), junto con todos sus registros de entradas/salidas y todas sus incidencias.

### 6.8.2.9 Listado de incidencias

Para acceder al listado de las incidencias se debe seleccionar la opción en el menú lateral, lo que cargará la siguiente vista con la información de las incidencias en una tabla.



The screenshot shows a web interface titled "Issues" with a breadcrumb "Home / Issues". It features a table with 5 columns: "Detail", "Checked", "Building's name", "Type issue", and "Timestamp". The table contains 8 rows of incident data. At the top, there is a "Show 10 entries" dropdown and a "Search:" input field. At the bottom, it says "Showing 1 to 8 of 8 entries" and has "Previous", "1", and "Next" navigation buttons.

Detail	Checked	Building's name	Type issue	Timestamp
	No ❌	Cafetería La Joya	CAPACITY EXCEEDED	2021-07-25 17:42:42
	No ❌	Cafetería La Joya	NO MASK LIMIT	2021-07-25 17:38:26
	No ❌	Cafetería La Joya	CAPACITY EXCEEDED	2021-07-25 17:38:26
	No ❌	Cafetería La Joya	CAPACITY EXCEEDED	2021-07-21 19:49:18
	No ❌	Cafetería La Joya	CAPACITY EXCEEDED	2021-07-21 19:49:11
	No ❌	Cafetería La Joya	ENTRY CLOSED HOUR	2021-07-09 22:18:40
	No ❌	Cafetería La Joya	NO MASK LIMIT	2021-07-09 22:18:38
	Yes ✅	Cafetería La Joya	NO MASK LIMIT	2021-07-09 22:18:38

Figura 31: Listado de incidencias

Si se desea ver más detalle de una incidencia se debe pulsar en la opción ver [] incidencia.

## 6.8.2.10 Edición del semáforo

Se puede acceder a esta vista desde el menú lateral o desde el listado de edificios. Permite seleccionar un edificio del desplegable y personalizar el diseño de su semáforo. En la siguiente figura se puede ver dicha vista.

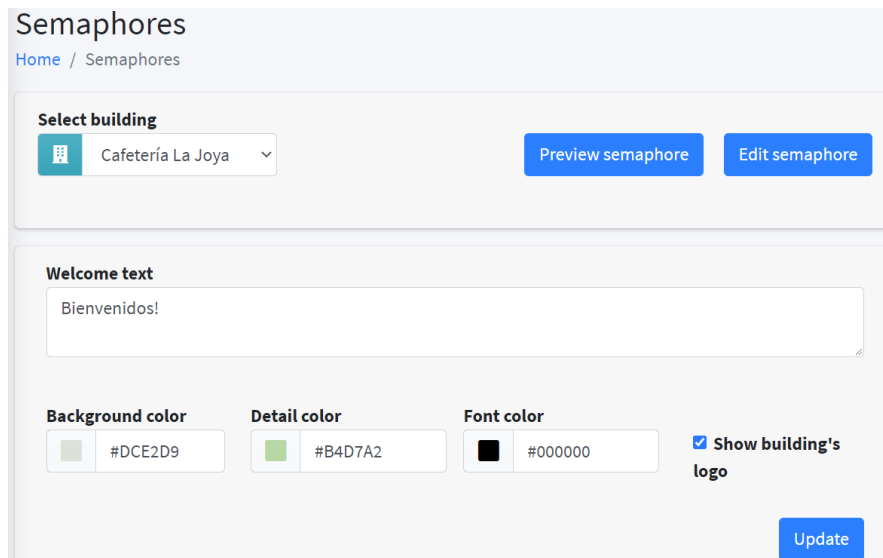


Figura 32: Ver detalles del semáforo

Además, permite previsualizar el semáforo del edificio deseado en una nueva pestaña. Tiene el diseño de la siguiente figura.



Figura 33: Vista previa del semáforo

### 6.8.3 Usuario Administrador

Este usuario es único en el sistema y solo tiene acceso el desarrollador de la aplicación. Tiene la capacidad de ver y editar cualquier información de los edificios añadidos al sistema. Además de todas las funcionalidades vistas hasta ahora, también puede administrar los usuarios registrados en la aplicación.

#### 6.8.3.1 Modo Administrador

Cuando se accede con las credenciales de la cuenta de administrador, cargará la siguiente página privada de bienvenida. En el menú se puede apreciar un aviso de Modo administrador y una nueva opción para ver el listado de usuarios. También aparecerá el resumen de todos los edificios añadidos al sistema. En la siguiente figura se puede ver dicha pantalla.

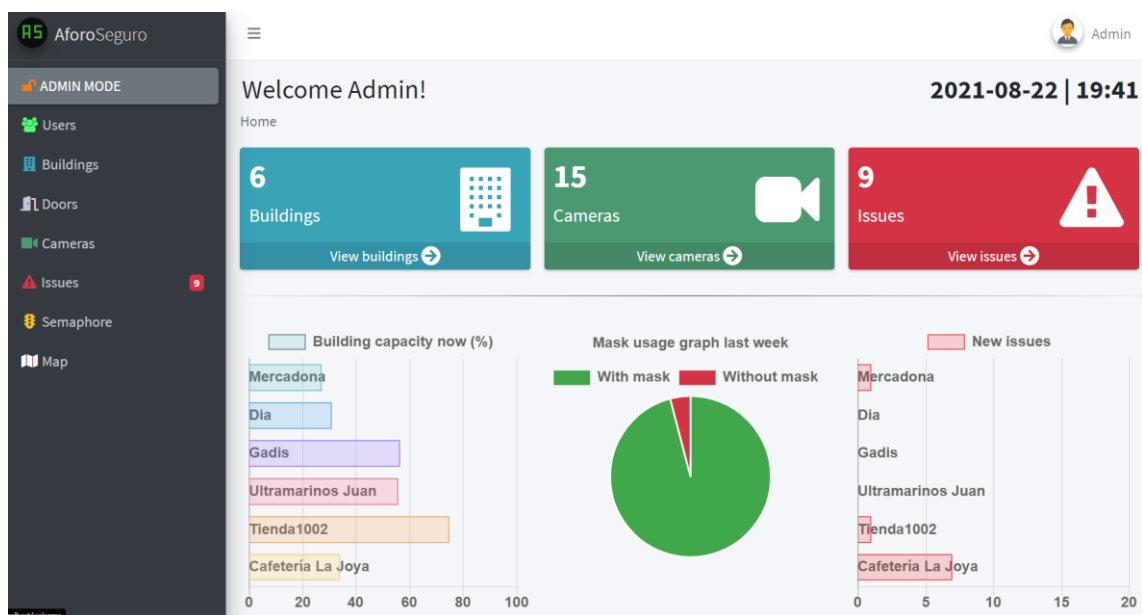


Figura 34: Página privada del administrador

### 6.8.3.2 Listado de usuarios

---

Cuando el administrador solicite el listado de usuarios cargará la página de la siguiente figura. En ella se pueden ver todos los usuarios registrados en el sistema y acceder a su edición [✎]. La tabla proporciona un buscador y se permite ordenar las diferentes columnas.

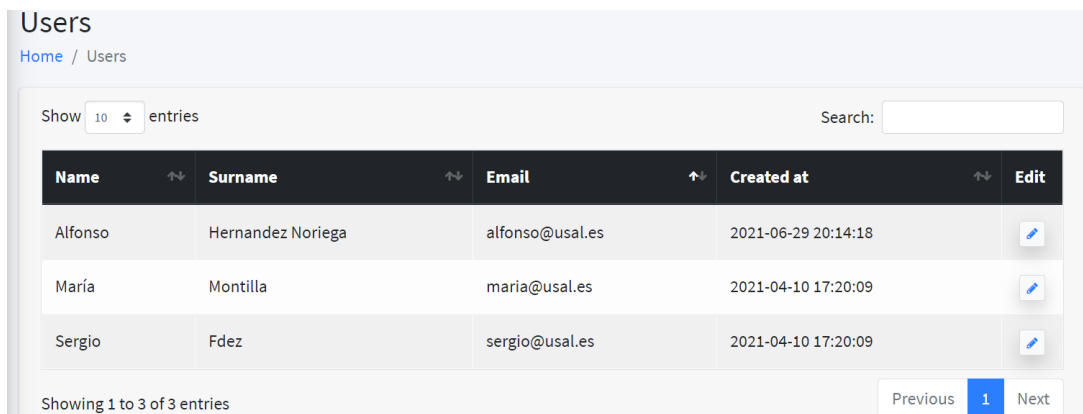


Figura 35: Listado de usuarios

### 6.8.3.3 Editar usuario

---

Cuando el administrador solicite editar [✎] un usuario de la lista, cargará la página de la siguiente figura. Desde aquí podrá editar la información personal y modificar la contraseña sin introducir la anterior. Si alguno de los campos no es válido se mostrará el error en el formulario. También permite eliminar de forma permanente la cuenta del usuario, junto con la información de sus edificios.

The screenshot shows the 'Edit user' page for 'User: Alfonso Hernandez Noriega'. The breadcrumb is 'Home / Users / Edit user'. The form has several sections: 'Name' (Alfonso), 'Surname' (Hernandez Noriega), 'Email' (alfonso@usal.es), and 'Profile pic' (Choose a file... Browse). Below these are 'Created at: 2021-06-29 20:14:18' and 'Last updated: 2021-06-29 21:09:43' with an 'Update' button. At the bottom, there are 'New password' and 'Repeat new password' fields with a 'Change password' button.

Figura 36: Editar usuario

### 7. Limitaciones del prototipo

---

El proyecto realizado no deja de ser un prototipo que tiene una serie de limitaciones. Las más interesantes a mejorar en futuras versiones son las siguientes:

- Es necesario un ordenador por cada una de las cámaras que se instalan en los establecimientos. Estos servidores tienen la función de procesar las imágenes y establecer comunicación con el servidor principal. Esto significa un alto coste de instalación del proyecto, con las desventajas que ello conlleva. La solución pasaría por instalar un único servidor en cada edificio, con la suficiente capacidad de cálculo como para procesar simultáneamente todas las cámaras instaladas en el establecimiento. Por supuesto habría diseñar un nuevo despliegue e implementar la correcta comunicación.
- En el prototipo desarrollado se utilizan cámaras de bajo presupuesto y esto afecta al procesamiento de las imágenes, ya que son muy sensibles a la exposición de la luz y esto dificulta el reconocimiento facial. Esto se traduce en que es necesario calibrar la configuración de la cámara con un nivel de iluminación y mantenerlo constante. La solución viene dada por el uso de cámaras de mayor calidad.
- En relación al conteo de aforo, es posible que en algunos casos no detecte a una persona y por lo tanto no la tenga en cuenta para las estadísticas. Para detectar una persona primero es necesario detectar una cara y, a través de su posición, se determina si ha entrado por la puerta. Después se analiza si llevaba mascarilla puesta o no, según las zonas de interés visibles de la cara. Es decir, si no detecta la boca ni la nariz significa que lleva puesta la mascarilla. Esto puede dar lugar a que no se reconozca la cara en primer lugar, ya que no detectan suficientes zonas de interés como para determinar que sea rostro. Una solución a este problema puede ser la instalación de sensores físicos, que ayuden al conteo de personas, y que las cámaras solo analicen el uso de mascarillas.



## 8. Conclusiones y líneas de trabajo futuras

---

En este apartado se recogen las conclusiones obtenidas una vez finalizadas todas las fases de desarrollo del proyecto.

### 8.1 Conclusiones

---

En primer lugar, cabe destacar que se han cumplido todos los objetivos propuestos, tanto funcionales como no funcionales. Se trata de un sistema cerrado de servidores que comunican las entradas y salidas detectadas a un servidor central, que procesa toda esta información para mantener actualizado el estado del aforo de un establecimiento. Esta información está disponible para su visualización en tiempo real a través de la Aplicación Web, permitiendo a los propietarios gestionar de manera más eficiente sus edificios, puertas y semáforos. También puede ser consultada por los clientes que quieran ver el aforo de un determinado establecimiento, accediendo a la página pública principal.

Por tanto, se puede determinar que se han cumplido los siguientes objetivos establecidos: detectar los clientes y el uso de mascarilla, controlar el aforo, gestionar los usuarios, gestionar los edificios, gestionar las puertas, gestionar los semáforos y controlar las incidencias.

Se ha podido comprobar cuál es el esfuerzo que supone la realización de un proyecto de estas dimensiones, obteniendo una mayor experiencia en las tareas de análisis, diseño e implementación. Cada una de las fases del proyecto ha permitido poner en práctica los conocimientos y habilidades adquiridas en el Grado de Ingeniería Informática.

## 8.2 Líneas de trabajo futuras

---

En este apartado se mencionan las posibles mejoras e ideas sobre el sistema desarrollado. Las principales líneas de trabajo futuras podrían consistir en resolver las limitaciones expuestas en el apartado 7 de este documento.

También sería interesante añadir nuevas funcionalidades al sistema como, por ejemplo:

- Ver en directo la transmisión de las cámaras mediante la Aplicación Web.
- Reconocer la indumentaria de los empleados para no contabilizarlos en el aforo.
- Ofrecer una estadística más detallada, calculando la media por horas, semanas, meses...
- Establecer una base de datos en el servidor de las cámaras para garantizar la consistencia de los datos, si por algún motivo perdiera la conexión.
- Desarrollar un sistema público de comentarios para compartir y valorar las experiencias de cada establecimiento.

## 9. Referencias

---

- AdminLTE. (2021). AdminLTE Bootstrap Admin Dashboard Template. Best open source admin dashboard & control panel theme. Built on top of Bootstrap, AdminLTE provides a range of responsive, reusable, and commonly used components. Recuperado de: <https://adminlte.io/> (accedido en 29/08/2021)
- Aforoseguro.com. (s.f.). Recuperado de: <https://www.aforoseguro.com/> (accedido en 15/05/2021)
- Álvarez Sánchez, R. y Álvarez, M. A. (2018). *Manual de HTML. El manual completo para introducirse en el lenguaje de marcación HTML*. Recuperado de: <https://app.desarrolloweb.com/manuales/manual-html> (accedido en 27/08/2021)
- Cobo, A., Gómez, P., Pérez, D., Rocha, R. (2005). *PHP y MySQL. Tecnologías para el desarrollo de aplicaciones web*. Ediciones Díaz de Santos.
- Consejo interterritorial. (2020). *Actuaciones de respuesta coordinada para el control de la transmisión de COVID-19*. Recuperado de: [https://www.mscbs.gob.es/profesionales/saludPublica/ccayes/alertasActual/nCov/documentos/Actuaciones\\_respuesta\\_COVID\\_22.10.2020.pdf](https://www.mscbs.gob.es/profesionales/saludPublica/ccayes/alertasActual/nCov/documentos/Actuaciones_respuesta_COVID_22.10.2020.pdf) (accedido en 28/08/2021)
- Del Río Díez, L. (2013). Uso sistemático del barbijo quirúrgico. ¿Cuál es la evidencia científica al respecto? *IntraMed Journal*, 5(1), 1-13.
- Eguiluz Pérez, J. (2009). *CSS Avanzado*. Autoedición, recuperado de: <https://openlibra.com/es/book/css-avanzado> (accedido en 29/08/2021)
- Gauchat, J. D. (2012). *El gran libro de HTML5, CSS3 y JavaScript*. Barcelona: Marcombo, S. A.
- GeoSek. (2021). Mapa semáforo de aforos. Recuperado de: <https://geosek.info/mapaforoseguro/> (accedido en 29/08/2021)
- Haro, J., Arcos Chalán, J. S., Chicaiza Inguillay, D. E. (2016). *Análisis comparativo de los Frameworks Laravel y Codeigniter para la implementación del sistema de gestión de concursos de méritos y oposición*

## Memoria del proyecto

en la Universidad Nacional de Chimborazo”. (Tesis de ingeniería en sistemas y computación). Universidad Nacional de Chimborazo.

Juanes, G. G. (2020). Innovadora aplicación de detección de mascarillas. *Cuadernos de seguridad*. Recuperado de: <https://cuadernosdeseguridad.com/2020/07/innovadora-solucion-de-deteccion-de-mascarillas-hanwha/> (accedido en 30/08/2021)

Kinsta. (29 abril 2021). El framework PHP Laravel – Construcción de aplicaciones web para todos. Recuperado de: <https://kinsta.com/es/base-de-conocimiento/que-es-laravel/> (accedido en 30/08/2021)

Marco de desarrollo de la Junta de Andalucía. (s.f.). Doxygen. Recuperado de: <https://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/110> (accedido en 29/08/2021)

Oltra, C. y Sala, R. (2015). La implicación del público en los riesgos derivados de la contaminación atmosférica urbana. *Papers*, 100(4), 477-492.

OpenCV. (2021). OpenCV. Recuperado de: <https://opencv.org/> (accedido en 30/08/2021)

Oracle. (2021). MySQL Database Service. Recuperado de: <https://www.oracle.com/es/mysql/> (accedido en 31/08/2021)

Organización Mundial de la Salud (OMS). (2020a). Vías de transmisión del virus de la COVID-19: repercusiones para las recomendaciones relativas a las precauciones en materia de prevención y control de las infecciones. Recuperado de: <https://www.who.int/es/news-room/commentaries/detail/modes-of-transmission-of-virus-causing-covid-19-implications-for-ipc-precaution-recommendations> (accedido en 31/08/2021)

Organización Mundial de la Salud (OMS). (2020b). Recomendaciones sobre el uso de mascarillas en el contexto de la COVID-19. Recuperado de: [https://apps.who.int/iris/bitstream/handle/10665/332657/WHO-2019-nCov-IPC\\_Masks-2020.4-spa.pdf](https://apps.who.int/iris/bitstream/handle/10665/332657/WHO-2019-nCov-IPC_Masks-2020.4-spa.pdf) (accedido en 31/08/2021)

## Memoria del proyecto

Ortega Alarcón, J. A., Rodríguez López, J. R., Hernández Palma, H. (2017). Importancia de la seguridad de los trabajadores en el cumplimiento de procesos, procedimientos y funciones. *Revista academia y derecho*, 8(14), 155-176.

PHPStorm. (2021). The lightning-smart PHP IDE. Recuperado de: <https://www.jetbrains.com/phpstorm/> (accedido en 31/08/2021)

PyCharm. (2021). Funcionalidades de PyCharm. Recuperado de: <https://www.jetbrains.com/es-es/pycharm/features/> (accedido en 31/08/2021)

Python. (2021). Python. Recuperado de: <https://www.python.org/> (accedido en 31/08/2021)

Rojas Campo, J. (2020). Control de aforo, qué es, cómo implementarlo y cuáles son los beneficios de hacerlo. Recuperado de: <https://www.tecnoseguro.com/analisis/pro/control-de-aforo-que-es-como-implementarlo-beneficios> (accedido en 29/08/2021)

Rubio Martínez, M. I. (2014). Conocimiento sobre el uso correcto de las mascarillas como medida preventiva frente al riesgo biológico por transmisión aérea y por gotas. *Enfermería del trabajo*, 4, 45-54.

Sitio web oficial de Vue. (2021). *The progressive JavaScript Framework. Why Vue.js?* Recuperado de: <https://vuejs.org/> (accedido en 31/08/2021)

Svitla. (2021). Overview of modern computer vision tools. Recuperado de: <https://svitla.com/blog/overview-of-modern-computer-vision-tools> (accedido en 27/08/2021)

Visus, A. (2020). ¿Para qué sirve Python? Razones para utilizar este lenguaje de programación. Recuperado de: <https://www.esic.edu/rethink/tecnologia/para-que-sirve-python> (accedido en 31/08/2021)

Balaji Srinivasan (2020). *Face Mask Detector*. Recuperado de: <https://github.com/balajisrinivas/Face-Mask-Detection> (accedido en 10/01/2021)