



UNIVERSIDAD FRANCISCO DE VITORIA

ESCUELA POLITÉCNICA SUPERIOR

GRADO EN INGENIERÍA INFORMÁTICA

PROYECTO FINAL DE GRADO

MODALIDAD DE INGENIERÍA



Sistema Optimizador de Tiempo de
Aparcamiento (SOTA) para el *Smart
Parking* de un *Smart Campus*

Convocatoria de mayo de 2022

CALIFICACIÓN DEL PROYECTO FINAL DE GRADO

CUALITATIVA:	
NUMÉRICA:	

Conforme Presidente:	Conforme Secretario:
Fdo.:	Fdo.:

Conforme Vocal:	Conforme Vocal:	Conforme Vocal:
Fdo.:	Fdo.:	Fdo.:

Lugar y fecha: Pozuelo de Alarcón, a ____ de _____ de 2022

"El tiempo es la cosa más valiosa que una persona puede gastar" [1]

Theophrastus – Grecia (300-287 a.C.)

Resumen

Este proyecto final de grado estudia, diseña y desarrolla una solución inteligente y segura para optimizar el tiempo de aparcamiento con un sistema de *Smart Parking* dentro de un *Smart Campus*. Ofrece al gestor del aparcamiento la visión en tiempo real sobre un aparcamiento virtual 2D, en un Digital Twin desarrollado en PHP y JS. Permite simular el uso de cámaras IP de reconocimiento de matrículas, utilizando un protocolo JSON seguro, para comunicarse con un servidor socket gestor de cámaras desarrollado en Python, que puede ser controlado desde el Digital Twin. Por otro lado, también ofrece a los usuarios una visión del aparcamiento atractiva y simple en una aplicación web desarrollada utilizando los frameworks de Laravel y Vue.js. El acceso al aparcamiento está limitando a solo los usuarios registrados previamente en la aplicación web. El usuario, al entrar al aparcamiento, recibe un *ticket* con su plaza para aparcar. Este es generado según su tipo de vehículo añadido previamente en la web y los tipos de plazas existentes en el aparcamiento de la universidad.

Gracias a este proyecto se averiguó que existe un gran interés, por parte de la comunidad universitaria UFV, en tener un sistema de asignación de plazas de aparcamiento. El sistema SOTA es un hito importante en el marco del desarrollo e investigación para la transformación del Campus de la Universidad Francisco de Vitoria en un Smart Campus, y si fuera implementado evitaría muchos accidentes, debido a que se asignan las plazas según el tipo y tamaño de vehículo registrado por el usuario.

Palabras clave

IoT, Gemelo Digital, Aparcamiento Inteligente, Criptografía, Campus Inteligente, Ciberseguridad.

Abstract

This final degree project studies, designs and develops an intelligent and secure solution to optimize parking time in a Smart Parking system within a Smart Campus. It offers the parking manager a real-time view over a virtual 2D parking lot map, in a Digital Twin developed in PHP and JS. It allows the simulation of license plate recognition IP cameras, using a secure JSON protocol to communicate with a socket server camera manager developed in Python, which can be controlled from the Digital Twin. On the other hand, it also provides users with an attractive and simple view of the parking lot in a web application developed using Laravel and Vue.js frameworks. Access to the parking lot is limited to users previously registered in the web application. When entering the parking lot, the user receives a ticket with his parking space, this ticket is generated according to his type of vehicle previously added on the WebApp and the types of existing parking spaces in the university parking lot.

Thanks to this project, it was found out that there is a great interest, by the UFV university community, in having a parking space allocation system. The Parking Time Optimization System (PTOS) is an important milestone in the framework of the development and research for the transformation of the Francisco de Vitoria University Campus, into a Smart Campus. If implemented, it would avoid many accidents on the UFV Campus because the parking spaces are assigned according to the type and size of vehicle registered by the user.

Keywords

IoT, Digital Twin, Smart Parking, Cryptography, Smart Campus, Cybersecurity.

Índice de Contenidos

1. Introducción	1
2. Investigación previa	3
2.1. Smart Campus.....	3
2.1.1. Smart Parking	6
2.2. Estudio y Toma de datos de los usuarios del sistema	7
2.2.1. Reunión con Gestor del Aparcamiento UFV.....	7
2.2.2. Estudio del Parking UFV	10
2.2.3. Seminario de Smart Cities UFV.....	16
2.3. Soluciones existentes.....	17
3. Objetivos	18
3.1. Objetivos generales	18
3.1.1. Optimizar el tiempo de búsqueda de plaza.....	18
3.1.2. Presentar el estado actual del aparcamiento.....	18
3.2. Lista de objetivos específicos	18
3.3. Métodos de Verificación y Validación	19
4. Plan de Desarrollo del Proyecto.....	20
4.1. Metodología	20
4.2. Tecnologías	21
4.2.1. Arquitectura de Tecnologías Software.....	21
4.2.2. Lenguajes de Programación	21
4.2.3. Frameworks.....	22
4.2.4. Gestores de Paquetes	23
4.2.5. Bases de Datos.....	23
4.2.6. Almacenamiento de Software y Despliegue	24
4.2.7. Servidores.....	25
4.2.8. Otras tecnologías	25
4.3. Plan de desarrollo del proyecto.....	27
4.3.1. Tareas Planificadas.....	27

4.3.2.	Investigación Previa	27
4.3.3.	Definiciones Iniciales.....	28
4.3.4.	Planificación y Especificación del Sistema	28
4.3.5.	Diseño y Configuración del Entorno	28
4.3.6.	Desarrollo del Software	29
4.3.7.	Integración y Pruebas	30
4.4.	Plan de Trabajo	30
4.5.	Recursos.	32
4.6.	Costes	32
4.7.	Condicionantes y Limitaciones	34
5.	Desarrollo de la Solución Técnica	35
5.1.	Investigación previa	35
5.2.	Definiciones Iniciales.....	35
5.3.	Planificación y Especificación del Sistema.....	36
5.4.	Diseño y Configuración del Entrono	36
5.5.	Desarrollo del Software.....	38
5.5.1.	Base de Datos	38
5.5.2.	Digital Twin.....	39
5.5.3.	WebApp	43
5.5.4.	Seguridad	47
5.6.	Integración y Pruebas.....	50
5.6.1.	Integración	50
5.6.2.	Verificación.....	50
6.	Resultados.....	51
6.1.	Resultados Generales.....	51
6.2.	Resultados Específicos	51
7.	Implicaciones Éticas e Impacto Social.....	52
7.1.	Introducción.....	52
7.2.	Valor del Proyecto.....	52
7.3.	Alcance	52
7.4.	Responsabilidad e Impacto	53
7.4.1.	Riesgos	53

7.4.2.	Leyes	54
7.4.3.	Impacto Social.....	54
7.4.4.	Impacto Económico	54
7.4.5.	Impacto Medioambiental.....	55
7.5.	Conclusiones.....	55
8.	Conclusiones.....	56
8.1.	Conclusiones Generales	56
8.2.	Conclusiones Técnicas.....	57
8.3.	Trabajo Futuro	57
9.	Otros Méritos del Proyecto.....	59
10.	Bibliografía.....	60
Anexo A:	Requisitos de Usuario.....	73
10.1.	Descripción de Atributos	73
10.2.	Requisitos Funcionales	75
10.2.1.	Módulo Digital Twin.....	75
10.2.2.	Módulo Servidor Device Manager	76
10.2.3.	Módulo Protocolo SJMP	76
10.2.4.	Módulo Paquete SJMP	77
10.2.5.	Módulo Flags de Paquete SJMP.....	78
10.2.6.	Módulo Parking.....	78
10.2.7.	Módulo Plaza Parking.....	80
10.2.8.	Módulo WebApp.....	81
10.2.9.	Módulo WebApp Autenticación	81
10.2.10.	Módulo WebApp Área de Usuario.....	82
10.2.11.	Módulo WebApp Menú.....	82
10.2.12.	Módulo WebApp Vehículos	82
10.2.13.	Módulo WebApp Perfil del Usuario	83
10.2.14.	Módulo WebApp Ticket.....	84
10.2.15.	Módulo WebApp Incidencias.....	84
10.2.16.	Módulo Base de Datos	85
10.2.17.	Módulo WebApp Notificaciones.....	86

10.3. Requisitos No Funcionales	86
10.3.1. Módulo Software	86
10.3.2. Módulo Contenedores	88
10.3.3. Módulo Red	88
10.3.4. Módulo Volúmenes	89
10.3.5. Módulo Seguridad	89
10.3.6. Módulo de Accesibilidad	89
10.3.7. Módulo Legal	90
10.3.8. Módulo Control de Accesos	90
10.3.9. Módulo Presentación	90
Anexo B: Diseño del Sistema.....	91
Anexo C: Manuales	95
Manual de Usuario y de Instalación	95
Anexo D: Detalles Técnicos	96
Descripción de Protocolos	96
Protocolo SJMP	99
Anexo E: Detalles Estudio del Parking UFV.....	102
Anexo F: Bocetos y Detalles del Diseño del Sistema.....	121
Investigación Previa.....	121
Parking UFV	121
Tipos de Plaza	122
Número de Plazas y Arquitectura	124
Definición de Alcance	130
Diseño de la Arquitectura	131
Diseño y Integración de Tecnologías.....	139
Diseño del Sistema	141
Planificación	146
Modelo de Datos	150
Anexo G: Detalles Planificación.....	151

Investigación Previa	151
Definiciones Iniciales.....	152
Planificación y Especificación del Sistema.....	152
Diseño y Configuración del Entorno	153
Desarrollo del Software.....	154
Base de Datos	154
Digital Twin	155
WebApp.....	156
Integración y Pruebas.....	159
Diagrama de PERT	160
Anexo H: Registro de Riesgos	161
Tabla de Riesgos.....	161
Matriz de Riesgos	162
Anexo I: Plan de Pruebas.....	163

Índice de Tablas

<i>Tabla 1: Lista de Universidades Españolas con Programa de Smart Campus – Elaboración Propia</i>	4
<i>Tabla 2: Datos del Entrevistado — Elaboración Propia.....</i>	7
<i>Tabla 3: Características del Estudio — Elaboración Propia</i>	11
<i>Tabla 4: Datos del Profesor – Elaboración Propia</i>	16
<i>Tabla 5: Paquete de Trabajo: Investigación Previa – Elaboración Propia.....</i>	27
<i>Tabla 6: Paquete de Trabajo: Definiciones Iniciales– Elaboración Propia.....</i>	28
<i>Tabla 7: Paquete de Trabajo: Planificación y Especificación del Sistema – Elaboración Propia.....</i>	28
<i>Tabla 8: Paquete de Trabajo: Diseño y Configuración del Entorno – Elaboración Propia.....</i>	28
<i>Tabla 9: Paquete de Trabajo: Base de Datos – Elaboración Propia.....</i>	29
<i>Tabla 10: Paquete de Trabajo: Digital Twin – Elaboración Propia</i>	29
<i>Tabla 11: Paquete de Trabajo: WebApp – Elaboración Propia</i>	29
<i>Tabla 12: Paquete de Trabajo: Integración y Pruebas – Elaboración Propia</i>	30
<i>Tabla 13: Recursos del Proyecto – Elaboración Propia</i>	32
<i>Tabla 14: Recursos Materiales – Elaboración Propia.....</i>	33
<i>Tabla 15: Servicios para la Obra Civil – Elaboración Propia</i>	33
<i>Tabla 16: Recursos Humanos para Gestión y Desarrollo – Elaboración Propia.....</i>	34
<i>Tabla 17: Costes totales del proyecto – Elaboración Propia</i>	34
<i>Tabla 18: Registro de Riesgos – Elaboración Propia</i>	53
<i>Tabla 19: Relación Asignatura-Solución – Elaboración Propia – Con Acrónimos de Asignaturas</i>	59
<i>Tabla 20: Prioridades de los Requisitos – Elaboración Propia</i>	73
<i>Tabla 21: Descripción de Campos – Elaboración Propia</i>	73
<i>Tabla 22: Descripción de Módulos – Elaboración Propia</i>	74
<i>Tabla 23: Requisitos del Módulo Digital Twin – Elaboración Propia</i>	75
<i>Tabla 24: Requisitos del Módulo Device Manager – Elaboración Propia.....</i>	76
<i>Tabla 25: Requisitos del Módulo SJMP – Elaboración Propia.....</i>	76
<i>Tabla 26: Requisitos del Módulo Paquete SJMP – Elaboración Propia</i>	77
<i>Tabla 27: Requisitos del Módulo Flags de Paquete SJMP – Elaboración Propia.....</i>	78
<i>Tabla 28: Requisitos del Módulo Parking – Elaboración Propia</i>	78
<i>Tabla 29: Requisitos del Módulo Plaza Parking – Elaboración Propia</i>	80
<i>Tabla 30: Requisitos del Módulo WebApp – Elaboración Propia.....</i>	81
<i>Tabla 31: Requisitos del Módulo WebApp Autenticación – Elaboración Propia</i>	81
<i>Tabla 32: Requisitos del Módulo WebApp Área de Usuario – Elaboración Propia.....</i>	82

<i>Tabla 33: Requisitos del Módulo WebApp Menú – Elaboración Propia.....</i>	82
<i>Tabla 34: Requisitos del Módulo WebApp Vehículos – Elaboración Propia.....</i>	82
<i>Tabla 35: Requisitos del Módulo WebApp Perfil del Usuario – Elaboración Propia</i>	83
<i>Tabla 36: Requisitos del Módulo WebApp Ticket – Elaboración Propia</i>	84
<i>Tabla 37: Requisitos del Módulo WebApp Incidencias – Elaboración Propia.....</i>	84
<i>Tabla 38: Módulo Base de Datos – Elaboración Propia</i>	85
<i>Tabla 39: Requisitos del Módulo WebApp Notificaciones – Elaboración Propia.....</i>	86
<i>Tabla 40: Requisitos del Módulo Software – Elaboración Propia.....</i>	86
<i>Tabla 41: Requisitos del Módulo Contenedores – Elaboración Propia</i>	88
<i>Tabla 42: Requisitos del Módulo Red – Elaboración Propia.....</i>	88
<i>Tabla 43 Requisitos del Módulo Volúmenes – Elaboración Propia.....</i>	89
<i>Tabla 44: Requisitos del Módulo Seguridad – Elaboración Propia</i>	89
<i>Tabla 45: Requisitos del Módulo Accesibilidad – Elaboración Propia.....</i>	89
<i>Tabla 46: Requisitos del Módulo Legal – Elaboración Propia.....</i>	90
<i>Tabla 47: Requisitos del Módulo Control de Accesos – Elaboración Propia</i>	90
<i>Tabla 48: Requisitos del Módulo Presentación – Elaboración Propia.....</i>	90
<i>Tabla 49: Descripción Flags Protocolo SJMP - Elaboración Propia</i>	99
<i>Tabla 50: Tarea PT01-T01: Segunda Reunión con Gestor – Elaboración Propia</i>	151
<i>Tabla 51: Tarea PT01-T02: Investigación del Estado del Arte – Elaboración Propia</i>	151
<i>Tabla 52: Tarea PT01-T03: Estudio del Parking UFV – Elaboración Propia.....</i>	152
<i>Tabla 53: Tarea PT02-T01: Introducción – Elaboración Propia</i>	152
<i>Tabla 54: Tarea PT02-T02: Definición de Objetivos y Alcance – Elaboración Propia.....</i>	152
<i>Tabla 55: Tarea PT03-T01: Requisitos de Usuario – Elaboración Propia</i>	153
<i>Tabla 56: Tarea PT03-T02: Plan de Proyecto – Elaboración Propia.....</i>	153
<i>Tabla 57: Tarea PT03-T03: Modelado 2D del Parking Seleccionado – Elaboración Propia</i>	153
<i>Tabla 58: Tarea PT04-T01: Estudio de Tecnologías – Elaboración Propia</i>	153
<i>Tabla 59: Tarea PT04-T02: Configuración del Entorno – Elaboración Propia</i>	154
<i>Tabla 60: Tarea PT04-T03: Diseño del sistema – Elaboración Propia</i>	154
<i>Tabla 61: Tarea PT05-T01: Configuración de Base de Datos – Elaboración Propia.....</i>	154
<i>Tabla 62: Tarea PT05-T02: Conexión con Servidor – Elaboración Propia</i>	155
<i>Tabla 63: Tarea PT05-T03: Device Manager – Elaboración Propia</i>	155
<i>Tabla 64: Tarea PT05-T04: Cámara IP Python – Elaboración Propia.....</i>	155
<i>Tabla 65: Tarea PT05-T05: Pantalla de Visualización – Elaboración Propia</i>	156
<i>Tabla 66: Tarea PT05-T06: Autenticación – Elaboración Propia</i>	156
<i>Tabla 67: Tarea PT05-T07: Esqueleto Dashboard y Home – Elaboración Propia</i>	157

<i>Tabla 68: Tarea PT05-T08: Perfil de Usuario – Elaboración Propia</i>	157
<i>Tabla 69: Tarea PT05-T09: Notificaciones – Elaboración Propia.....</i>	157
<i>Tabla 70: Tarea PT05-T10: Área de Vehículos – Elaboración Propia</i>	158
<i>Tabla 71: Tarea PT05-T11: Tickets – Elaboración Propia.....</i>	158
<i>Tabla 72: Tarea PT05-T12: Incidencias – Elaboración Propia.....</i>	159
<i>Tabla 73: Tarea PT06-T01: Pruebas de Integración – Elaboración Propia</i>	159
<i>Tabla 74: Tarea PT06-T02: Pruebas de Verificación – Elaboración Propia.....</i>	159
<i>Tabla 75: Registro de Riesgos – Elaboración Propia</i>	161
<i>Tabla 76: Matriz de Riesgos – Elaboración Propia.....</i>	162
<i>Tabla 77: Registro del Plan de Pruebas. – Elaboración Propia</i>	163

Índice de Figuras

<i>Figura 2.1: Componentes abstractos de un Smart Campus, Fuente: [3]</i>	3
<i>Figura 2.2: Objetivos de Desarrollo Sostenible para 2030, Fuente: [10]</i>	5
<i>Figura 2.3: Componentes Visuales de un Smart Campus, Fuente: [12]</i>	6
<i>Figura 2.4: Comparación Campus UFV Actual vs Futuro</i>	8
<i>Figura 2.5: Especificación de la Cámara IP</i>	9
<i>Figura 2.6: Precio de la Cámara IP</i>	9
<i>Figura 2.7: Flujo de Preguntas en Estudio Parking UFV – Elaboración Propia</i>	11
<i>Figura 2.8: Categorías de la Encuesta – Elaboración Propia en Google Forms [18]</i>	12
<i>Figura 2.9: Valoración Final de la Encuesta – Elaboración Propia en Google Forms [18]</i>	12
<i>Figura 2.10: Primera pregunta de la encuesta y resultados. – Elaboración Propia en Google Forms [18]</i>	13
<i>Figura 2.11: Porcentaje de aparcamiento ilegal (izquierda) y respuestas (derecha) – Elaboración Propia en Google Forms [18]</i>	13
<i>Figura 2.12: Impactos positivos – Elaboración Propia en Google Forms [18]</i>	14
<i>Figura 2.13: Necesidad de la solución – Elaboración Propia en Google Forms [18]</i>	15
<i>Figura 4.1: Esquema del modelo en cascada modificado. Fuente: Adaptado desde Sommerville [39] Pg.30, Figura 2.1, Elaboración Propia.</i>	20
<i>Figura 4.2: Arquitetura de Tecnologias Software. Fuente: Elaboración Propia</i>	21
<i>Figura 4.3: Logo Python: [47]</i>	22
<i>Figura 4.4: Estudio de lenguajes más utilizados para Análisis de Datos. Fuentes: [48] [50]</i>	22
<i>Figura 4.5: Logotipos de Laravel (Izquierda) y Vue.js (Derecha). Fuentes: [54] [43]</i>	23
<i>Figura 4.6: Logo Azure DevOps. Fuente: [38]</i>	24
<i>Figura 4.7: Logo Docker (Izquierda) y Docker Compose (Derecha). Fuente: [73] [78]</i>	25
<i>Figura 4.8: WebSocket Frame. Fuente: [82]</i>	26
<i>Figura 4.9: Capas del Protocolo TCP: Fuente: [83]</i>	26
<i>Figura 4.10: Diagrama de GANTT Parte I, Fuente: Elaboración propia en GanttProject [86]</i>	30
<i>Figura 4.11: Diagrama de GANTT Parte II, Fuente: Elaboración propia en GanttProject [86]</i>	31
<i>Figura 4.12: Diagrama de PERT: Relación entre Tareas, Fuente: Elaboración propia en GanttProject [86], Más detalle en: [Anexo G: Detalles Planificación]</i>	31
<i>Cameras IP, lectora de matrículas. [Figura 4.13]</i>	33
<i>Figura 5.1: Zonas de Aparcamiento UFV – Elaboración Propia sobre el plano actual del Campus UFV [16] de Vanesa García</i>	35
<i>Figura 5.2: Foto Panorámica de la Zona Y en Campus UFV – Elaboración Propia</i>	35

<i>Figura 5.3: Modelado 2D (vista aérea) de la Aparcamiento Zona "Y" del Campus UFV. [Ver Figura 5.1] - Zonas identificadas con letras y colores, entradas/salidas y arquitectura – Elaboración Propia</i>	36
<i>Figura 5.4: Esquema Funcional y Arquitectura de Contenedores – Elaboración Propia</i>	36
<i>Figura 5.5: Esquema a Nivel Conceptual – Elaboración Propia</i>	37
<i>Figura 5.6: Captura del Repositorio en Azure DevOps – Elaboración Propia</i>	37
<i>Figura 5.7: Modelo de Datos y relaciones entre Tablas. – Fuente: Elaboración propia utilizando: dbdiagram.io [88]. Descripción: El símbolo * se interpreta como N (muchos).....</i>	38
<i>Figura 5.8: Migraciones establecidas en Laravel – Elaboración Propia.....</i>	39
<i>Figura 5.9: Tablas de la Base de Datos en phpMyAdmin – Elaboración Propia, Captura de Pantalla.</i>	39
<i>Figura 5.10: Intercambio de claves en paquetes SJMP entre Cámara Socket TCP (Izquierda) con Menú de Opciones en CMD y Servidor Socket (Derecha) en Python - Elaboración Propia.</i>	39
<i>Figura 5.11: Esquema del Intercambio de claves asimétricas en SJMP – Elaboración Propia</i>	40
<i>Figura 5.12: Pantalla de Login del Digital Twin – Elaboración Propia</i>	40
<i>Figura 5.13: Pantalla de Visualización del Digital Twin – Elaboración Propia</i>	41
<i>Figura 5.14: Botones Abrir Server / Cerrar y desconectar cámara – Elaboración Propia</i>	41
<i>Figura 5.15: Configuración de apertura de Servidor – Elaboración Propia.....</i>	41
<i>Figura 5.16: Botones de Acción – Elaboración Propia</i>	41
<i>Figura 5.17: Apertura de Servidor y Conexión de Cámara con WebSockets – Elaboración Propia</i>	42
<i>Figura 5.18: Asignación de Plaza (Izquierda) y Simulación de Salida (Derecha) - Elaboración Propia</i>	42
<i>Figura 5.19: Plazas Ocupadas – Elaboración Propia.....</i>	42
<i>Figura 5.20: Simulador de Barrera – Elaboración Propia.....</i>	43
<i>Figura 5.21: Aforo Total del Parking y Botón de Refresh y Auto-Refresh – Elaboración Propia.....</i>	43
<i>Figura 5.22: HTTP POST Request para Registro de Usuario utilizando API (/api/auth/register) creada en Laravel – Fuente: Elaboración Propia utilizando la Herramienta de Postman [90].</i>	43
<i>Figura 5.23: Vistas de Registro y Login en UFV MyParking – Elaboración Propia.....</i>	43
<i>Figura 5.24: Dashboard visualización pública del parking – Elaboración Propia.</i>	44
<i>Figura 5.25: Hub Mis Vehículos – Elaboración Propia</i>	44
<i>Figura 5.26 - Interacción botón añadir e interacción botón borrar – Elaboración Propia</i>	45
<i>Figura 5.27: Tabla de Tickets del Usuario – Elaboración Propia</i>	45
<i>Figura 5.28: Ejemplo de Ticket Válido y No Válido – Elaboración Propia</i>	45
<i>Figura 5.29 – Formulario Incidencia Ticket – Elaboración Propia.....</i>	46
<i>Figura 5.30: Perfil Usuario – Elaboración Propia.....</i>	46
<i>Figura 5.31: Área de Administrador de Usuarios – Elaboración Propia.....</i>	47
<i>Figura 5.32: Actualización de Rol por Administrador – Elaboración Propia.....</i>	47
<i>Figura 5.33: API's con control de autenticación y middleware en PHP Laravel – Elaboración Propia.</i>	48
<i>Figura 5.34: Mensajes de Estado: Carga, Error y Éxito en WebApp – Elaboración Propia</i>	48

<i>Figura 5.35: Cookies de Sesión del Digital Twin en Navegador Web – Elaboración Propia</i>	49
<i>Figura 5.36: Error de Usuario no Autorizado – Elaboración Propia.....</i>	49
<i>Figura 5.37: Mensajes de Estado del Digital Twin - Elaboración Propia</i>	49
<i>Figura 5.38: Plan de Pruebas Detallado – Elaboración Propia</i>	50
<i>Figura 0.1: Esquema a nivel de Comunicaciones – Elaboración Propia</i>	91
<i>Figura 0.2: Esquema a nivel Red/Físico – Elaboración Propia</i>	91
<i>Figura 0.3: Esquema Arquitectura de Contenedores – Elaboración Propia</i>	92
<i>Figura 0.4: Esquema Detallado de Protocolos de Comunicación – Elaboración Propia.....</i>	92
<i>Figura 0.5: Boceto a nivel Conceptual Ampliado. – Elaboración Propia</i>	93
<i>Figura 0.1: README.md En Repositorio de Github – Elaboración Propia</i>	95
<i>Figura 0.1: Formato Cabecera TCP. Fuente: [83]</i>	96
<i>Figura 0.2: Cabecera HTTP General y Métodos Permitidos. Fuente: [85]</i>	96
<i>Figura 0.3: Estructura de un Mensaje HTTP Request: Fuente: [85]</i>	97
<i>Figura 0.4: Estructura de un Mensaje HTTP Response: Fuente: [73]</i>	97
<i>Figura 0.5: Intercambio de claves SJMP – Elaboración propia</i>	101
<i>Figura 0.1: Introducción Encuesta – Elaboración Propia.....</i>	102
<i>Figura 0.2: Pregunta 1 – Elaboración Propia.....</i>	103
<i>Figura 0.3: Pregunta 2 – Elaboración Propia.....</i>	103
<i>Figura 0.4: Pregunta 3 – Elaboración Propia.....</i>	103
<i>Figura 0.5: Pregunta 4 – Elaboración Propia.....</i>	104
<i>Figura 0.6: Pregunta 5 – Elaboración Propia.....</i>	104
<i>Figura 0.7: Pregunta 6 – Elaboración Propia.....</i>	104
<i>Figura 0.8: Pregunta 7 – Elaboración Propia.....</i>	105
<i>Figura 0.9: Pregunta 8 – Elaboración Propia.....</i>	105
<i>Figura 0.10: Pregunta 9 – Elaboración Propia.....</i>	105
<i>Figura 0.11: Pregunta 10 – Elaboración Propia.....</i>	106
<i>Figura 0.12: Pregunta 11 – Elaboración Propia.....</i>	106
<i>Figura 0.13: Pregunta 12 – Elaboración Propia.....</i>	106
<i>Figura 0.14: Pregunta 13 – Elaboración Propia.....</i>	107
<i>Figura 0.15: Pregunta 14 – Elaboración Propia.....</i>	107
<i>Figura 0.16: Pregunta 15 – Elaboración Propia.....</i>	107
<i>Figura 0.17: Pregunta 16 – Elaboración Propia.....</i>	108
<i>Figura 0.18: Pregunta 17 – Elaboración Propia.....</i>	108
<i>Figura 0.19: Pregunta 18 – Elaboración Propia.....</i>	108
<i>Figura 0.20: Pregunta 19 – Elaboración Propia.....</i>	109

<i>Figura 0.21: Pregunta 20 – Elaboración Propia.....</i>	109
<i>Figura 0.22: Pregunta 21 – Elaboración Propia.....</i>	109
<i>Figura 0.23: Pregunta 22 – Elaboración Propia.....</i>	110
<i>Figura 0.24: Pregunta 23 – Elaboración Propia.....</i>	110
<i>Figura 0.25: Pregunta 24 – Elaboración Propia.....</i>	110
<i>Figura 0.26: Pregunta 25 – Elaboración Propia.....</i>	110
<i>Figura 0.27: Pregunta 26 – Elaboración Propia.....</i>	111
<i>Figura 0.28: Pregunta 27 – Elaboración Propia.....</i>	111
<i>Figura 0.29: Pregunta 28 – Elaboración Propia.....</i>	111
<i>Figura 0.30: Pregunta 29 – Elaboración Propia.....</i>	112
<i>Figura 0.31: Pregunta 30 – Elaboración Propia.....</i>	112
<i>Figura 0.32: Pregunta 31 – Elaboración Propia.....</i>	112
<i>Figura 0.33: Pregunta 32 – Elaboración Propia.....</i>	113
<i>Figura 0.34: Pregunta 33 – Elaboración Propia.....</i>	113
<i>Figura 0.35: Pregunta 34 – Elaboración Propia.....</i>	114
<i>Figura 0.36: Pregunta 35 – Elaboración Propia.....</i>	114
<i>Figura 0.37: Pregunta 36 – Elaboración Propia.....</i>	114
<i>Figura 0.38: Pregunta 37 – Elaboración Propia.....</i>	115
<i>Figura 0.39: Pregunta 38 – Elaboración Propia.....</i>	115
<i>Figura 0.40: Pregunta 39 – Elaboración Propia.....</i>	115
<i>Figura 0.41: Pregunta 40 – Elaboración Propia.....</i>	115
<i>Figura 0.42: Pregunta 41 – Elaboración Propia.....</i>	116
<i>Figura 0.43: Pregunta 42 – Elaboración Propia.....</i>	116
<i>Figura 0.44: Pregunta 43 – Elaboración Propia.....</i>	116
<i>Figura 0.45: Pregunta 44 – Elaboración Propia.....</i>	117
<i>Figura 0.46: Pregunta 45 – Elaboración Propia.....</i>	117
<i>Figura 0.47: Pregunta 46 – Elaboración Propia.....</i>	118
<i>Figura 0.48: Pregunta 47 – Elaboración Propia.....</i>	118
<i>Figura 0.49: Pregunta 48 – Elaboración Propia.....</i>	119
<i>Figura 0.50: Pregunta 49 – Elaboración Propia.....</i>	119
<i>Figura 0.51: Pregunta 50 – Elaboración Propia.....</i>	119
<i>Figura 0.52: Pregunta 51 – Elaboración Propia.....</i>	120
<i>Figura 0.1: Boceto para la Elección del Parking - Elaboración Propia sobre Google Maps.....</i>	121
<i>Figura 0.2: Toma de Medidas de las Plazas – Elaboración Propia</i>	122
<i>Figura 0.3: Medidas de los Tipos de Plaza – Elaboración Propia.....</i>	123

<i>Figura 0.4: Foto Nocturna del Parking para el Conteo de Plazas I – Elaboración Propia.....</i>	124
<i>Figura 0.5: Foto Nocturna del Parking para el Conteo de Plazas II – Elaboración Propia.....</i>	124
<i>Figura 0.6: Notas sobre número de plazas.....</i>	125
<i>Figura 0.7: Boceto de la Arquitectura del Aparcamiento– Elaboración Propia.....</i>	126
<i>Figura 0.8: Boceto Versión 1.0 del mapeo de la Zona “Y”. – Elaboración Propia</i>	127
<i>Figura 0.9: Versión 2.0 – Mapa Zona Y– Elaboración Propia</i>	128
<i>Figura 0.10: Foto Interior Vehículo en Entrada E1 del Mapa Zona Y – Elaboración Propia</i>	129
<i>Figura 0.11: Foto Interior Vehículo en Salida S1 del Mapa Zona Y – Elaboración Propia</i>	129
<i>Figura 0.12: Boceto para la elección de alcance– Elaboración Propia.....</i>	130
<i>Figura 0.13: Boceto para la configuración y decisión– Elaboración Propia</i>	131
<i>Figura 0.14: Boceto de arquitectura del sistema– Elaboración Propia.....</i>	132
<i>Figura 0.15: Boceto de Decisiones del Sistema– Elaboración Propia.....</i>	133
<i>Figura 0.16: Boceto de Flujo de Sistema– Elaboración Propia</i>	134
<i>Figura 0.17: Boceto para definición de Componentes del Sistema – Elaboración Propia.....</i>	135
<i>Figura 0.18: Boceto de Arquitectura– Elaboración Propia</i>	136
<i>Figura 0.19: Boceto de Diseño de la Arquitectura– Elaboración Propia</i>	137
<i>Figura 0.20: Boceto para la decisión de la estructura principal– Elaboración Propia</i>	138
<i>Figura 0.21: Definición de tecnologías. – Elaboración Propia</i>	139
<i>Figura 0.22: Boceto de Árbol tecnologías– Elaboración Propia.....</i>	140
<i>Figura 0.23: Diseño de las Vistas para WebApp UFV MyParking I – Elaboración Propia.....</i>	141
<i>Figura 0.24 Diseño de las Vistas para WebApp UFV MyParking II – Elaboración Propia.....</i>	142
<i>Figura 0.25: Diseño de las Vistas para WebApp UFV MyParking III – Elaboración Propia.....</i>	143
<i>Figura 0.26: Diseño de las Vistas para Digital Twin I – Elaboración Propia</i>	144
<i>Figura 0.27: Diseño de las Vistas para Digital Twin II – Elaboración Propia</i>	145
<i>Figura 0.28: Boceto de planificación – Elaboración Propia</i>	146
<i>Figura 0.29: Boceto de planificación II – Elaboración Propia</i>	147
<i>Figura 0.30: Boceto planificación III – Elaboración Propia</i>	148
<i>Figura 0.31: Boceto planificación IV – Elaboración Propia.....</i>	149
<i>Figura 0.32: Modelo Relacional Entidad Relación - Elaboración Propia</i>	150
<i>Figura 0.1: Diagrama de PERT – Parte 1, Fuente Elaboración Propia con GANTT Project [86].....</i>	160
<i>Figura 0.2: Diagrama de PERT – Parte 2, Fuente Elaboración Propia con GANTT Project [86].....</i>	160

Lista de Acrónimos

Acrónimo	Significado
API	Application Programming Interface
ASOP	Administración de Sistemas Operativos
BBDD	Bases de Datos
CI/CD	Continuous Integration and Continuous Delivery
COCO	Complejidad Computacional
DGT	Dirección General de Trafico
DIS	Desarrollo e Integración de Software
HTTP	Hypertext Transfer Protocol
IA	Inteligencia Artificial
IoT	Internet of Things
IP	Internet Protocol
IPO	Interacción Persona Ordenador
IS	Ingeniería del Software
JS	Java Script
JSON	JavaScript Object Notation
JWT	JSON Web Tokens
LAN	Local Area Network
LoRa	Long Range
MPI	Matemáticas para la Ingeniería
N/A	No Aplica
ONU	Organización de las Naciones Unidas
PAS	Personal de Administración y Servicios
PFG	Proyecto Final de Grado
PGPI	Planificación y Gestión de Proyectos Informáticos
PHP	Hypertext Preprocessor
POO	Programación Orientada a Objetos
PTOP	Parking Time Optimization System
QR	Quick Response
RAID	Redundant Array of Independent Disks
REC	Recurso
RSA	Rivest, Shamir y Adleman
SHA	Secure Hash Algorithm
SJMP	Secure JSON Message Protocol
SOTA	Sistema Optimizador de Tiempo de Aparcamiento
SQL	Structured Query Language
TCP	Transmission Control Protocol
TFM	Trabajo Final de Máster
TFT	Thin-film transistor
UFV	Universidad Francisco de Vitoria
UUID	Universally Unique Identifier
WAN	Wide Area Network
XML	eXtensible Markup Language
YAML	Yet Another Markup Language

1. INTRODUCCIÓN

Desde el principio del siglo XXI la sociedad avanza de manera sustancial hacia una nueva era digital, donde el ser humano y la tecnología comparten un espacio contiguo. El dato sumado a las aplicaciones de la técnica permite al ser humano ampliar su capacidad de interacción con la realidad. Nuestra sociedad se agrupa en ciudades inteligentes o *Smart Cities* que se vuelven cada día más complejas y demandan un conocimiento de su realidad para poder manejarse. Ellas utilizan gemelos digitales o *Digital Twins* que proporcionan una visualización de datos en tiempo real y permiten la realización de simulaciones para poder predecir situaciones futuras. Las Universidades cada vez más anhelan conocer la realidad en sus campus, lo que fomenta la creación del concepto de *Smart Campus*, que, entre muchos servicios, dispone de un sistema de *Smart Parking* que permite a los usuarios visualizar el estado en tiempo real del aparcamiento, además de guiar a los conductores a las plazas libres en los distintos aparcamientos disponibles. Estos sistemas utilizan Internet para conectar las cosas (IoT, *Internet of Things*) e intercambiar datos entre sensores de movimiento; cámaras de lectura de matrículas, entre otros dispositivos.

En una universidad puede haber diferentes aparcamientos con configuraciones distintas y separaciones entre ellos de manera no uniforme, lo que produce confusión a la hora de buscar una plaza para aparcar. Si un conductor utiliza de manera regular el aparcamiento, puede llegar a conocer en qué horarios de pico aproximadamente tendrá problemas para aparcar, lo que le permite calcular en qué horario debe llegar al *parking*. Sin embargo, nada le garantiza al conductor que encontrará plaza y si la encuentra será en un tiempo indeterminado.

Debido a esta situación muchos usuarios del *aparcamiento de la Universidad Francisco de Vitoria* (UFV). [2] deciden aparcar en zonas ilegales que no están dimensionadas para albergar sus vehículos, lo que en muchos casos dificulta la circulación interna de coches.

Con la intención de evitar estos incidentes y optimizar el tiempo de búsqueda de plazas libres, se estudiará, diseñará y desarrollará un sistema que permita conectar cámaras IP lectoras de matrículas, con un servidor gestor de cámaras TCP/IP controlado por un gemelo digital o *Digital Twin*. Debido a que el coste de las Cámaras IP con esta tecnología es muy elevado, se permitirá la simulación de la entrada y salida de vehículos, además de visualizar el estado del aparcamiento en tiempo real.

Se desarrollará un sistema modular que soporte múltiples cámaras y acepte datos de múltiples sensores, siempre que posean la estructura adecuada. Para que las comunicaciones

sean seguras se diseñará, desarrollará e implementará un protocolo basado en JSON (*JavaScript Object Notation*) que permite encriptar la comunicación, de extremo a extremo a nivel de dato con independencia de la red utilizada para transmitir la comunicación. Esto mensajes se envían desde un dispositivo (cámara) hasta un servidor que puedan guardar, procesar y generar un *ticket*, asignando una plaza libre al vehículo del usuario.

Para que el usuario pueda acceder a sus *tickets*, se diseñará y desarrollará una aplicación web utilizando Laravel y Vue.js, para poder registrarlo y que el usuario pueda añadir sus vehículos en la aplicación, además de visualizar el aforo de las zonas del aparcamiento en tiempo real.

Esta memoria contiene un estudio previo de las Universidades españolas que implementan modelos de Smart Campus en España, además de investigar sobre el papel del *Smart Parking* dentro de un Smart Campus. Contiene una toma de datos de los usuarios del aparcamiento UFV, de los responsables del *parking* y del responsable del laboratorio de observación de *Smart Cities* en la Universidad Francisco de Vitoria.

A lo largo del documento se definen unos objetivos para el desarrollo de un *Digital Twin* que permita visualizar un aparcamiento en tiempo real; simular la entrada/salida de vehículos y controlar un servidor socket de cámaras TCP/IP (*Transmission Control Protocol/Internet Protocol*) que permita la conexión múltiples cámaras y la simulación de estas utilizando clientes sockets programados en Python. Se desarrollará una aplicación web capaz de autenticar a los usuarios del aparcamiento. Estos usuarios podrán añadir y gestionar sus coches; recibir *tickets* al entrar en el aparcamiento, con la indicación de su plaza para aparcar; ver la ocupación del aparcamiento; añadir incidencias y gestionarlas si son gestores del aparcamiento. Además, si el usuario es administrador podrá borrar y modificar los roles de todos los usuarios de la aplicación.

Se expone también una planificación del proyecto; los paquetes de trabajo; un análisis de costes y un análisis de recursos necesarios. Este será seguido de un análisis de las implicaciones éticas junto al impacto social del proyecto; se enseñarán los resultados obtenidos; se expondrán las conclusiones y al final se enunciarán los méritos de este proyecto, ya que está relacionado con numerosas asignaturas del Grado de Ingeniería Informática y posee una cierta complejidad al integrar diversas tecnologías innovadoras utilizadas actualmente en el mercado.

2. INVESTIGACIÓN PREVIA

Se propone realizar una investigación previa que permita conocer el entorno en el que se encuentra este proyecto, además de identificar a los usuarios finales del sistema, con el fin de comprender mejor el problema identificado y actuar en función de la información obtenida de los usuarios del *parking UFV*.

2.1. SMART CAMPUS

Un Smart Campus está compuesto por diferentes componentes que permiten monitorizar, interactuar y visualizar los datos del campus. Como podemos observar en la siguiente figura:

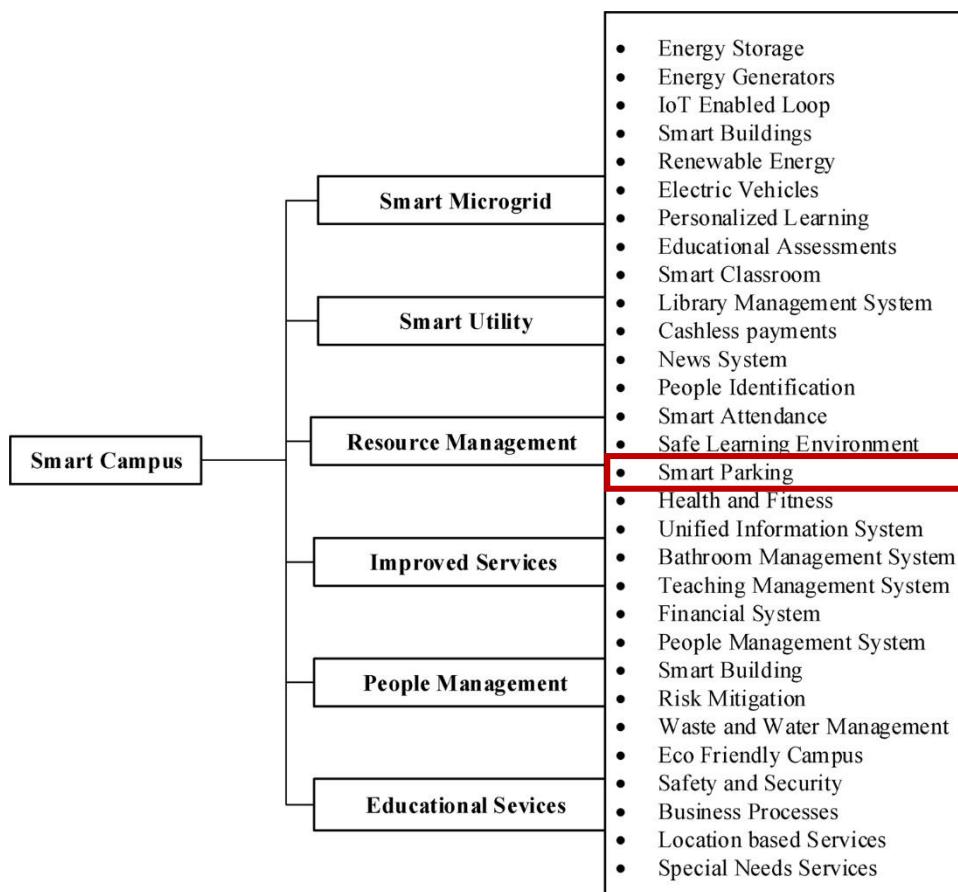


Figura 2.1: Componentes abstractos de un Smart Campus, **Fuente:** [3]

Actualmente en España existen diferentes Universidades que reconocen la necesidad de transformar sus centros académicos en verdaderos Smart Campus. Aquí podemos ver algunas Universidades españolas que empezaron el proceso de transformación a Smart Campus o ya diseñaron su programa de transformación y empiezan a ponerlo en práctica:

Tabla 1: Lista de Universidades Españolas con Programa de Smart Campus – Elaboración Propia

Universidad	Localización	Año	Nombre Programa	Objetivos y Avances
Universidad de Alicante	Alicante, Comunidad Valenciana	2014	UA Smart University [4]	Mejorar Movilidad, Gobierno, Economía etc.
Universidad de Málaga	Málaga, Andalucía	2017	SmartUMA [5]	Cuidar emisiones, cuidar la naturaleza, y movilidad.
Universidad Complutense Madrid	Madrid, Comunidad de Madrid	2021	UCM-Smart Campus [6]	Visualizar datos de Sensores de Humedad, Temperatura y CO2 Online.
LaSalle Universidad Ramon Llull	Barcelona, Catalunya	2016	La Salle-Url Smart Campus [7]	Smart Classroom and Smart Laboratories, Smart Mobility, Smart Facilities
Rey Juan Carlos III	Fuenlabrada, Madrid	2020	Smart Campus de Fuenlabrada /Smart E2 [8]	Detección de CO2, Temperatura, Humedad y TVOC.
Universidad Francisco de Vitoria	Pozuelo de Alarcón, Madrid	2019	Smart Campus UFV [9]	Investigación en soluciones (Dispositivo IoT detector de CO2, Temperatura y Humedad) y Observatorio de Smart Cities: “¿Que hace una ciudad más atractiva al talento?”, control de aforos en instalaciones, instalación de cámaras para conocer el aforo del parking, e-mobility, SharingCarUFV y Smart Flower.

Como podemos observar muchas Universidades tomaron el primer paso; decidiendo investigar e implementar soluciones “Smart” en el entorno de un campus universitario.

Las Universidades al igual que las ciudades están pasando por una transformación, motivadas por metas como el objetivo 11 de la agenda 2030 de la Organización de las Naciones Unidas (ONU), que fomenta la creación de ciudades y comunidades sostenibles. [Ver Figura 2.2]



Figura 2.2: Objetivos de Desarrollo Sostenible para 2030, Fuente: [10]

Dentro del marco de ciudades sostenibles del objetivo 11, nos encontramos a la optimización de la movilidad, donde se fomentan el uso de coches eléctricos y la reducción de la polución. Los campus [9] cada vez más implementan tecnologías que fomentan la movilidad sostenible, con coches eléctricos o coches compartidos. Sin embargo, las Universidades, se olvidan de que el cambio de movilidad a combustión a la eléctrica se realiza da manera gradual y requiere su tiempo [11], por lo que, una solución inteligente que reduzca el tiempo al volante puede ser una solución aceptable para el problema ambiental. A raíz de este problema se generan soluciones que implican la utilización de la tecnología en los aparcamientos, creando a los denominados *Smart Parkings*.

2.1.1. Smart Parking

Dentro de un campus inteligente se encuentran diferentes áreas de aplicación y una de ellas son los llamados Smart Parkings.

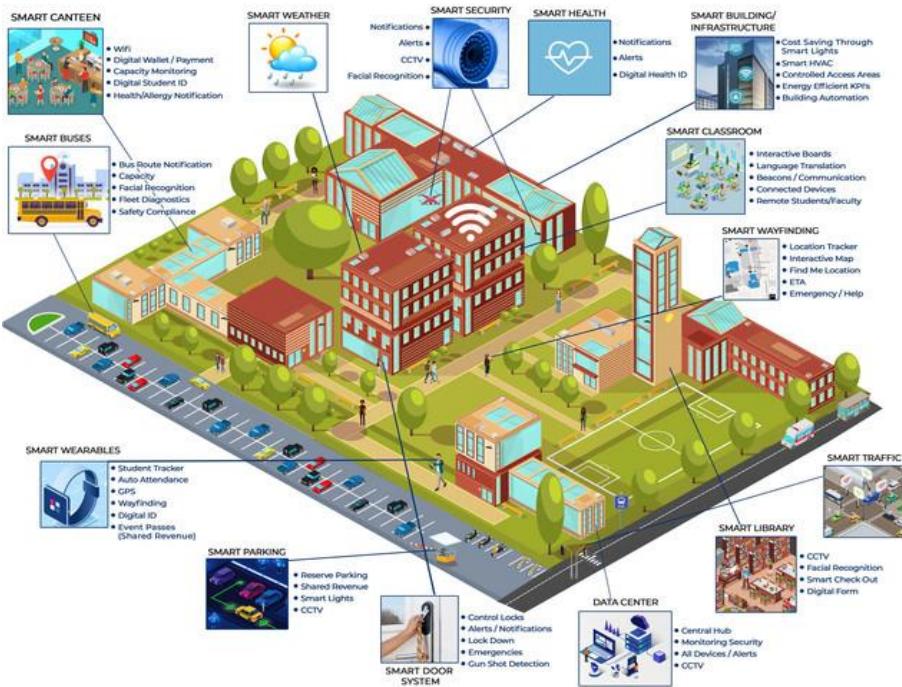


Figura 2.3: Componentes Visuales de un Smart Campus, Fuente: [12]

Los *Smart Parkings* incluyen diferentes tipos de funcionalidades [12]:

- Visualizar en Tiempo Real el estado del aparcamiento.
- Permite el registro de matrículas para identificar a los usuarios y contar los coches.
- Permite la reserva de plazas.
- Permite indicar al usuario donde se encuentra su plaza y guiarlo hasta allí.
- Permite controlar las luces y barreras del aparcamiento.

Todas ellas facilitan a la administración del *parking* efectuar una gestión efectiva del mismo, además de permitir al usuario conocer de manera previa el estado del *parking*, para poder reservar una plaza o decidir ir en transporte público antes de salir de casa en coche, en el caso de que el aparcamiento se encuentre lleno.

2.2. ESTUDIO Y TOMA DE DATOS DE LOS USUARIOS DEL SISTEMA

2.2.1. Reunión con Gestor del Aparcamiento UFV

Debido a que este proyecto final de grado trata de resolver un problema en aparcamientos de Smart Campus, se realizaron dos reuniones con el jefe del departamento [13] [14] de servicios generales de la Universidad Francisco de Vitoria, con el objetivo de conocer más la realidad de un gestor.

Tabla 2: Datos del Entrevistado -- Elaboración Propia

Datos del Entrevistado	
Nombre	Juan Carlos Miranda
Correo	j.miranda@ufv.es
Posición	Jefe del Departamento de Servicios Generales UFV
Fecha primera reunión	30 de septiembre de 2021
Fecha segunda reunión	1 de marzo de 2022

2.2.1.1. Temas Tratados en la Primera Reunión

Permiso y apoyo para realizar proyecto final de grado

Una vez explicada la idea del proyecto, Juan autorizó cualquier colocación de equipamiento como sensores y cámaras en el aparcamiento, siempre y cuando no afecte a la circulación de vehículos. Él aseguró que el alumno tendría el apoyo necesario para facilitarle el desarrollo del Proyecto Final de Grado (PFG) y que el alumno podría consultar más información en el futuro.

Sistema de Vigilancia Actual

Desde hace años en el aparcamiento de la UFV existe una cámara lectora de matrículas antigua, que permite a los guardias conocer si una matrícula introducida accede al campus.

Se detectó la necesidad de conocer el estado actual de aparcamiento, por lo que se contrató una empresa “Lasser” [15] para instalar cámaras y sensores que permitan contar los coches que entran en el aparcamiento.

Futuro del Parking UFV

El aparcamiento está pasando tiempos difíciles, ya que el tráfico en la entrada está congestionado por la mañana y al medio día con el cambio de turnos y la llegada de los autobuses.

El campus de la UFV está siendo actualizado, y el aparcamiento también será actualizado, ya que se incluirá una ronda grande en la entrada que permita entrar y salir a los coches de manera fluida, algunos aparcamientos desaparecerán y otros serán reducidos.



Figura 2.4: Comparación Campus UFV Actual vs Futuro

Fuente: Plano actual y futuro del Campus UFV [16] de Vanesa García

Zonas con más problemas

Según Juan, las zonas que tienen más problemas son la entrada del campus y la zona cerca del edificio M. En estas zonas se producen aparcamientos ilegales que bloquean el tráfico y dificultan la circulación en días de pico, según él los miércoles, debido a que los alumnos del campus del MSI vienen a estudiar en el campus principal.

2.2.1.2. Temas tratados en la Segunda Reunión.

Nuevo Sistema de Vigilancia

Se instalaron una serie de cámaras para poder controlar el aforo total del *parking*. Las 6 instaladas se posicionan de manera estratégica en las 3 entradas del campus UFV [Figura 2.4]. La empresa “Lasser” [15] está probando el sistema en el campus y su software indicará solo la cantidad total de vehículos en el campus. La solución empleada consiste en utilizar cámaras lectoras de matrículas con visión infrarroja, para detectar y contar los coches de manera precisa.



ZOOM TRAFFIC	Distancia máxima 12 m. DISTANCIA MÍNIMA A PLACA DE MATRÍCULA: 2,5 m. DISTANCIA MÁXIMA A PLACA DE MATRÍCULA: 12 m.
ESPECIFICACIONES TÉCNICAS	
BULLET ZOOM TRAFFIC	
Modelo	Cámara IP Día/Noche autorizo DC
Sensor Imagen	CMOS RGB de barrido progresivo de 1/2,8"
Objetivo	4,3-137 mm, F1.4-4.0
Compresión Video	Motion JPEG/MPEG-4/H.264
Resolución	Configurable desde 1920x1080 HDTV 1080p a 160x120 pixels
Velocidad de Imágenes	Con WDR: Hasta 25/50 ips (50/60 Hz). Sin WDR: Hasta 50/60 ips (50/60 Hz)
Transmisión Video	Múltiples secuencias en H.264 y Motion JPEG
Ancho de banda	Velocidad de imagen y ancho de banda controlables VBR/ABR/MBR H.264
Velocidad de Obturación	De 1/66 500 s a 1 s
Conectores	Ethernet 10baseT/100BaseTX, RJ-45
Procesador	ARTPEC, RAM: 256 Mb, Flash: 128 Mb
Alimentación	Alimentación a través de Ethernet (PoE) IEEE 802.3af/802.3at Tipo 1 Clase 3 7,7 típicos, 12,95 W máx
Condiciones de Funcionamiento	De -40°C a 60°C Humedad relativa del 10 - 100 %
Homologaciones	EN 50121-4, EN 55024, EN 61000-3-2, EN 61000-3-3, EN 61000-6-1, EN 61000-6-2, IEC/UL 60950-22, EN/IEC 62471 grupo de riesgo 1 AXIS Q1785-LE: EN 55022 Clase B, C-tick AS/NZS CISPR 22 Clase B, FCC Parte 15 Subparte B Clase B, ICES-003 Clase B, VCCI Clase B, KCC KN22 Clase B, KN24 AXIS Q1765-LE PT Mount: EN 55022 Clase A, C-tick AS/NZS CISPR 22 Clase A, FCC Parte 15 Subparte B Clase A, ICES-003 Clase A, VCCI Clase A, KCC KN22 Clase A, KN24
Dimensiones	Longitud: 386 mm ø 147 mm Peso 2,4 kg

Figura 2.5: Especificación de la Cámara IP.

Fuente: Foto del Presupuesto Grupo Lasser a servicios generales UFV [15]

La cámara tiene el precio de **1.830,93€** como podemos observar en el siguiente presupuesto:

DESCRIPCIÓN	UD.	PVP UNID.	PVP TOTAL
SIRAM BULLET TRAFFIC: Sistema de reconocimiento de matrículas para un vial. Se compone de la unidad de captura BULLET TRAFFIC para exterior y de módulo SIRAM OCR:			
BULLET TRAFFIC. Unidad de captura compuesta por cámara Bullet outdoor de 2 Mpixel con lente de 4,7-84,6mm MOTORIZADA, foco IR de leds. Incluye inyector POE.			
SIRAM OCR. Módulo básico de reconocimiento de matrículas para una cámara de OCR. Utiliza trigger externo que puede ser hardware (consultar CONEX) o software (mediante socket TCP/IP). Realiza la transferencia de información al software de gestión mediante TCP/IP. Incluye programa de configuración, no de gestión.	1	1.830,93€	1.830,93€

Figura 2.6: Precio de la Cámara IP.

Fuente: Foto del Presupuesto Grupo Lasser a servicios generales UFV [15]

Esta cámara utiliza un software que transfiere las matrículas desde un socket TCP/IP, a un servidor gestor de las cámaras, que puede actualizar el número de coches en la Universidad Francisco de Vitoria.

Ofertas de Soluciones Similares Propuestas

Dos consultoras ofrecieron soluciones similares a la solución propuesta en este PFG. Las consultoras realizaron un estudio contando los coches de manera manual con un contador, y concluyeron que la mejor solución sería conocer el estado actual del aparcamiento, para poder tener una visión general del mismo a la hora de aplicar reformas, creando algo muy parecido a un *Digital Twin* que permita visualizar en tiempo real el aparcamiento.

Problemas actuales

Los usuarios del *parking* no tienen tiempo ni voluntad para buscar plaza y deciden aparcar de manera ilegal en el césped o en sitios que bloquean la circulación. Esto implica que los bedeles y los responsables de seguridad tengan que colocar pegatinas en los coches para informar que el local donde aparcaron es ilegal.

Estudios realizados

Los bedeles y los responsables del servicio de seguridad realizaron un estudio coordinado por el jefe del departamento, con el objetivo de determinar si había o no plazas de *parking* disponibles en horarios de pico.

Todos ellos se subieron en un horario de pico problemático al tejado de todos los edificios que tenían visión al *parking* y se coordinaron por radio.

Al final del estudio concluyeron que siempre hay plazas libres para aparcar en horarios de pico, debido a que las personas se van de sus plazas de manera constante y el estado del *parking* cambia de manera continua.

Iniciativas propuestas

Al preguntar sobre soluciones o iniciativas que se propusieron en el departamento para solucionar el problema de la congestión y de la búsqueda exhaustiva de plazas libres, me contestó que se propuso crear un *parking* privado para los usuarios de *SharingcarUFV* [17], donde se introdujera la matrícula del coche y pudieran aparcar durante la semana de manera exclusiva por usar la aplicación.

2.2.1.3. Conclusiones

Con estas reuniones podemos concluir que existe una necesidad real por parte de los responsables del *parking* en conocer el estado en tiempo real de mismo; que se recomendaron soluciones parecidas a la solución propuesta en este proyecto; que uno de los problemas actuales en el *parking* es el aparcamiento ilegal, que se produce según los responsables de parking por pereza de realizar la búsqueda de plaza y por culpa de la necesidad de encontrar plaza de manera rápida para poder llegar a clase con tiempo.

2.2.2. Estudio del Parking UFV

A raíz de las reuniones con el jefe del departamento de servicios generales, se creó un estudio con el fin de confirmar las hipótesis y recoger nuevos datos, con la intención de conocer si de verdad hay o no un problema en el aparcamiento de la UFV. Este estudio pretende ofrecer un segundo punto de vista del problema, debido a una parte del público objetivo son los usuarios del *parking*, que se encuentran con diversas situaciones en el día a día. Los detalles de la encuesta se encuentran en el Anexo E, con el objetivo de apoyar a cualquier estudio o investigación sobre la opinión de los usuarios del *parking* UFV.

2.2.2.1. Características del estudio

Tabla 3: Características del Estudio -- Elaboración Propia

Nombre	Estudio del Parking UFV [2]
Enlace	https://forms.gle/TecpSGXA3GPYXpej6
Duración y Fecha	4 días [04/03/2022 – 08/03/2022]
Medio	Elaboración Propia de forma Virtual utilizando Google Forms [18]
Tipo y Estilo	Anónima de estilo sencillo, rápido y dinámico.
Distribución	Listas de Distribución de Correos Electrónicos de Alumnos, Profesores y PAS de la UFV.
Público Objetivo	Estudiantes UFV, Profesores UFV, PAS UFV, Usuarios del Parking
Categorías	Conductor de Coche, Pasajero de Transporte Público, Conductor de Coche Compartido, Pasajero de Coche Compartido.
Objetivos:	<ol style="list-style-type: none"> 1. Averiguar si se considera en la UFV que existe un problema en el parking. 2. Averiguar el comportamiento de los conductores. 3. Averiguar porque se produce el aparcamiento ilegal y si la solución propuesta evitaría el problema. 4. Averiguar horarios de pico/estrés y distancia de los edificios en que los estudiantes están y el aparcamiento. 5. Averiguar los tiempos medios y máximos de aparcamiento. 6. Averiguar si los conductores de coche pasarían a ser conductores de coche compartido si su aparcamiento estuviera asegurado. 7. Averiguar el impacto que tendría el uso del sistema. 8. Averiguar si el tiempo de aparcamiento existente tiene un impacto en la salud mental y afecta el tiempo de salida de los conductores de casa, con el objetivo de evitar estrés. 9. Averiguar si existe una necesidad de un sistema optimizador de tiempo de aparcamiento.
Nº Total de Preguntas:	51
Nº de Respuestas:	68

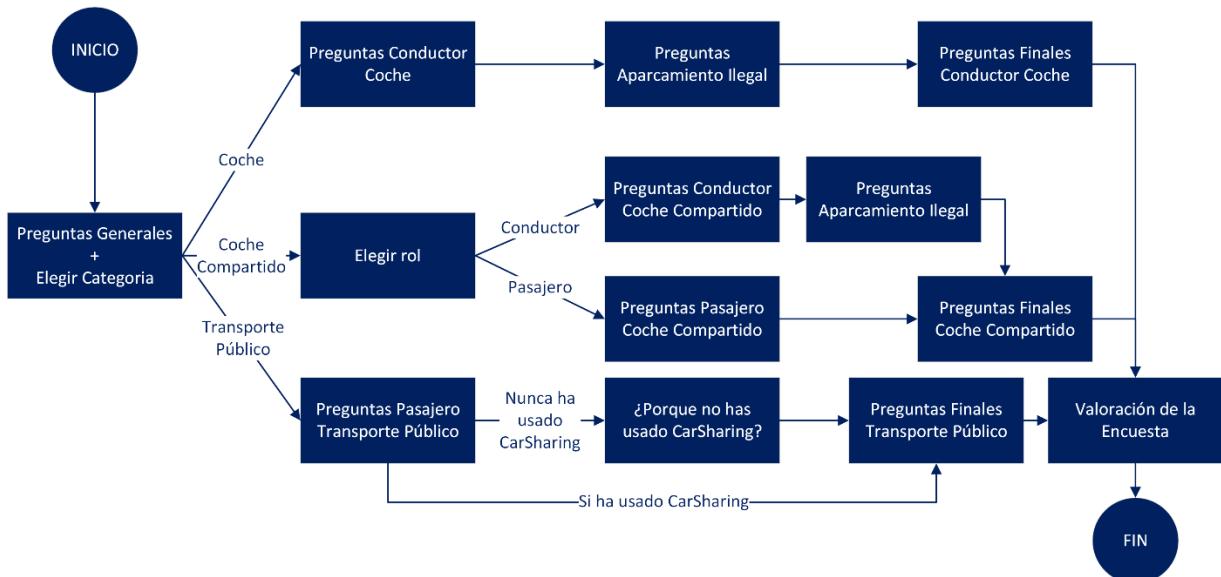


Figura 2.7: Flujo de Preguntas en Estudio Parking UFV – Elaboración Propria

2.2.2.2. Resultados Generales

Porcentaje de Cada Categoría

Un 58,8% de los usuarios entrevistados tienen turno de mañana [Ver Figura 0.3 Anexo E] y las categorías de como vienen los usuarios a la UFV están divididas de la siguiente manera:

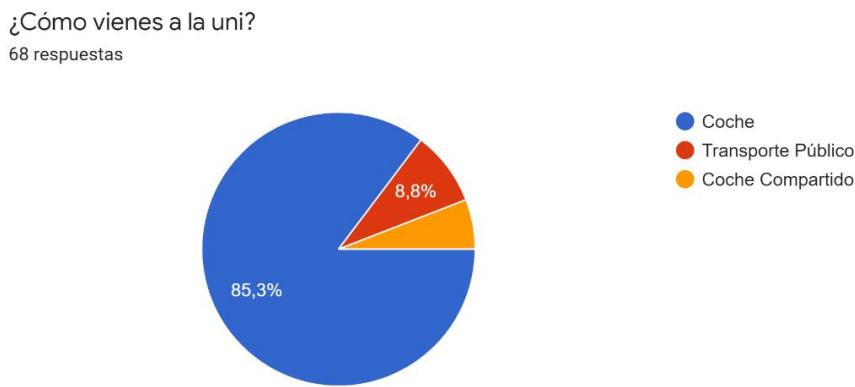


Figura 2.8: Categorías de la Encuesta – Elaboración Propia en Google Forms [18]

Debido a que tenemos más conductores de coche, podemos conocer más la realidad del *parking*. Otro punto interesante de analizar son los usuarios de coche compartido que pueden aportar otro punto de vista a la solución.

Valoración de la encuesta

Al final de todo se pidió a los usuarios que valorasen de 0 (Muy Mala) a 5 (Muy Buena) la encuesta. Como podemos observar en el resultado obtenido el **44,1%** de las personas les parecieron muy buena la encuesta, y el **39,7%** de las personas les pareció buena, sumando el total de **83,8%** de satisfacción.

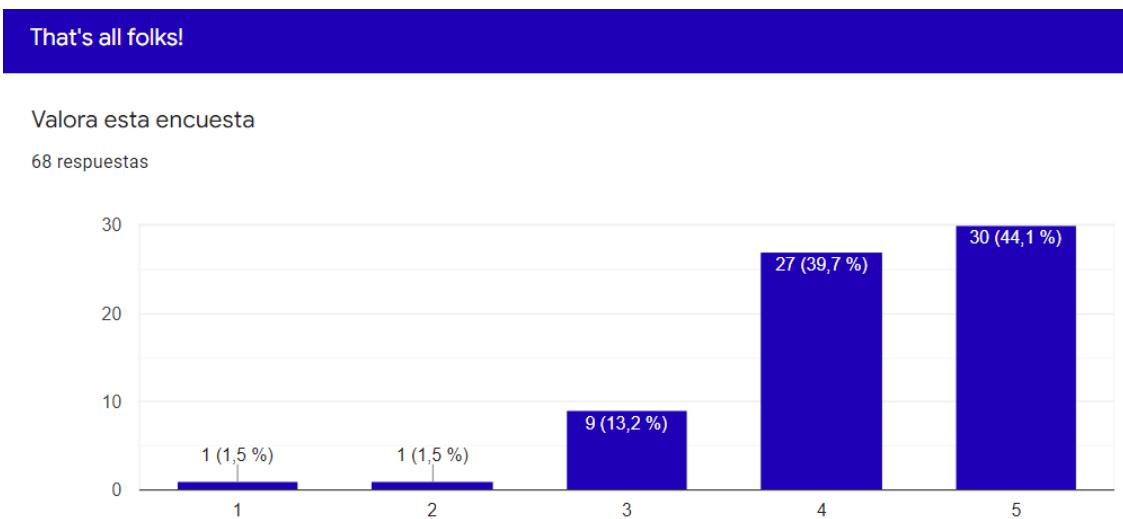


Figura 2.9: Valoración Final de la Encuesta -- Elaboración Propia en Google Forms [18]

2.2.2.3. Objetivo 1: Existencia de un problema

¿Consideras que hay un problema en el Parking UFV?

Para empezar el estudio primero preguntamos a los usuarios del *parking*, si creen que hay un problema (en general) en el aparcamiento UFV. Como podemos observar en la [Figura 2.10] un **89,7%** de los usuarios del *parking* entrevistados verificaron que existe un problema con el parking de la UFV, que necesita ser identificado y solucionado.

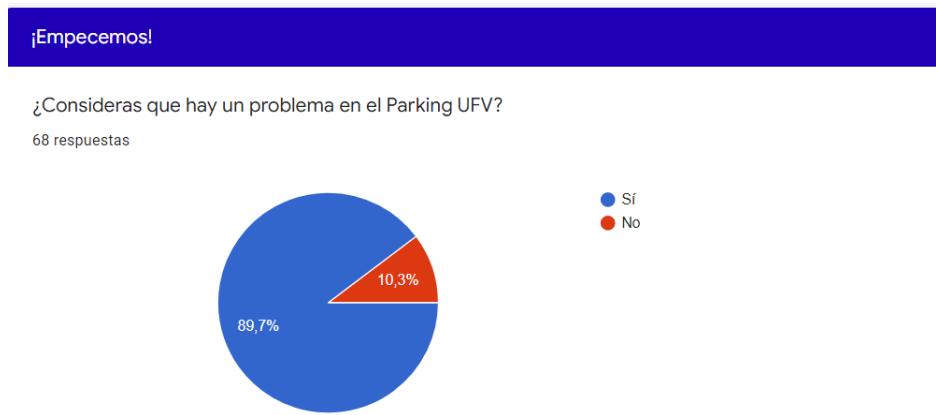


Figura 2.10: Primera pregunta de la encuesta y resultados. – Elaboración Propia en Google Forms [18]

2.2.2.4. Objetivo 2: Comportamiento al entrar al parking

Con el objetivo de conocer el comportamiento de los conductores al entrar en el *parking*, con el objetivo de conocer si hay confusión por parte de los conductores en donde aparcar, se realizó una pregunta [Ver Figura 0.21 Anexo E]. El resultado demuestra que el 94,8% los conductores tienen claro donde aparcar y solo un **5%** no. Por lo tanto, en principio no influye la indecisión de los conductores en el tráfico.

2.2.2.5. Objetivo 3: Aparcamiento ilegal

Con el objetivo de conocer el porcentaje de aparcamientos ilegales y el motivo por el cual se produce el problema se plantearon dos preguntas a los conductores:

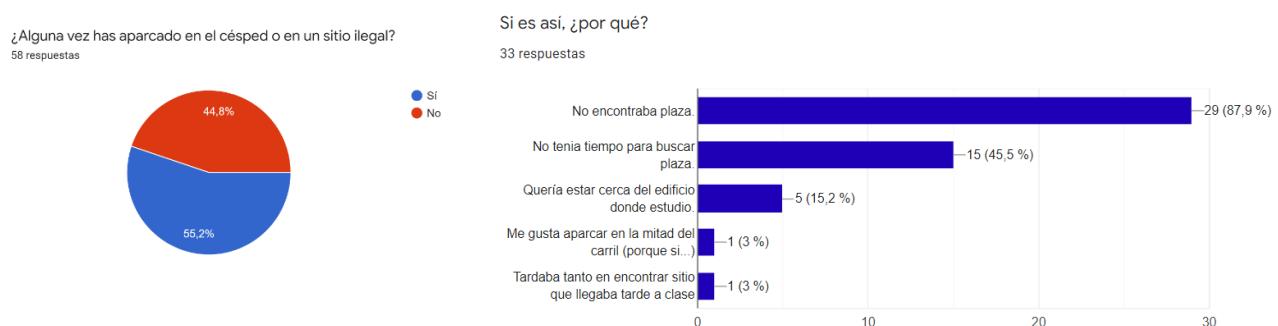


Figura 2.11: Porcentaje de aparcamiento ilegal (izquierda) y respuestas (derecha) – Elaboración Propia en Google Forms [18]

Como podemos observar los resultados obtenidos son sorprendentes debido a que la **mayoría (55,2%)** de los entrevistados aparcaba de manera ilegal. El principal motivo (87,9%) es que no encontraban plaza para aparcar, o simplemente no tenían tiempo de buscar plaza (45,5%). Estos datos refuerzan la necesidad de un sistema que facilite la localización de plazas.

2.2.2.6. Objetivo 4: Horarios de pico y distancia de aparcamiento

Para averiguar los horarios de picos de estrés se realizó una pregunta [Ver Figura 0.9 en Anexo E]. Las horas de pico registradas son de **10 a 11 horas de la mañana** pudiendo a ser también de **14 a 15 horas de la tarde** si el usuario del parking tiene turno de tarde.

Para averiguar la distancia de aparcamiento se realizó una pregunta [Ver Figura 0.6 Anexo E]. El porcentaje se encuentra bastante distribuido, pero en su gran mayoría (51,7%) podemos decir que los usuarios tienen que aparcar algo lejos de sus edificios y un (44.8%) aparcaba cerca de su edificio.

2.2.2.7. Objetivo 5: Tiempos máximos y medios de aparcamiento

Para averiguar los tiempos medios y máximos de aparcamiento se preguntó por los tiempos máximos y medios de aparcamiento [Ver Figuras 0.7 y 0.8 en Anexo E]. Se observó que el tiempo medio de aparcamiento registrado es de entre **5 a 15 minutos**. Un valor alto considerando que es un valor medio. Ya el valor máximo varía entre **15 a 30 minutos**, siendo media hora el valor dominante y pudiendo llegar a ser **45 minutos** de búsqueda de plaza. Lo que supone una pérdida de tiempo desnecesaria para los usuarios.

2.2.2.8. Objetivo 6: Cambio a CarSharing

Para averiguar si los conductores de coche se pasarían a conductores de coche compartido si siempre que llegaran a la universidad tuvieran su plaza reservada, realizamos una pregunta [Ver Figura 0.49 Anexo E]. Los resultados son bastante equitativos, pero en su mayoría los conductores no están interesados en utilizar una aplicación de coche compartido. Sin embargo, el **48,3%** de los conductores sí que pasarían a usar la app de *CarSharing* si tuvieran su plaza asegurada.

2.2.2.9. Objetivo 7: Impactos positivos

Con el objetivo de preguntar la opción de los usuarios sobre el impacto positivo que tendría una aplicación de optimización de tiempo de aparcamiento, se hicieron dos preguntas:

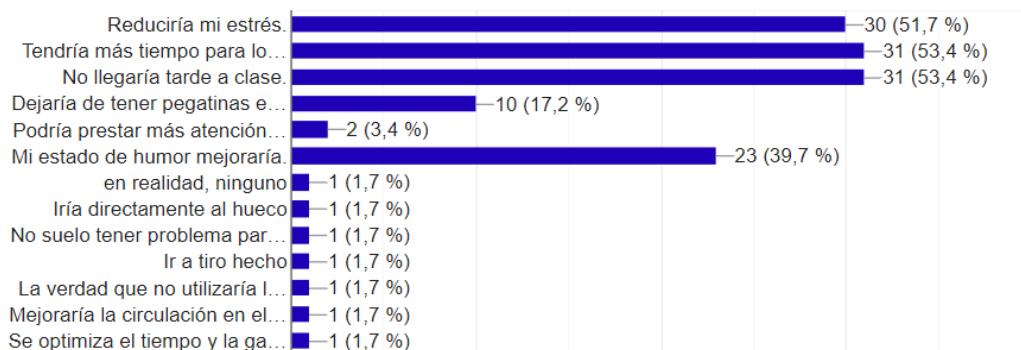


Figura 2.12: Impactos positivos – Elaboración Propia en Google Forms [18]

También preguntamos por el aparcamiento ilegal [Ver Figura 0.48 en Anexo E] y observó que el **85,5%** de los usuarios dejarían de aparcar ilegalmente si existiera un sistema optimizador de aparcamiento, lo que supondría la reducción del caos producido por los aparcamientos ilegales.

En cuanto a impactos positivos se encuentran el tener más tiempo para lo que importa, además de reducir el estrés, mejorar el estado de humor de las personas y permitir que no lleguen tarde a clase por estar buscando plaza.

2.2.2.10. Objetivo 8: Impactos Negativos

Para averiguar si se produce un impacto en la salud mental al buscar plaza, se realizaron dos preguntas [Ver Figura 0.10 Anexo E]. Se observó que el **65,5%** de los conductores creen que si existe un impacto en su salud mental el no encontrar plaza para aparcar. Esto supone que más de la mayoría de los miembros de la universidad no están rendiendo al 100% de sus capacidades mentales, esto produce un efecto negativo en el rendimiento del estudiante al atender a las clases. [19] [Ver Figura 0.11 Anexo E]. Se observó que en un **72,4%** de los casos el estado actual del *parking* afecta la salida de los conductores de sus casas, debido a que tienen en cuenta el posible tiempo de búsqueda de plaza y aparcamiento. Esto nos indica que el sistema SOTA sería utilizado a menudo para consultar los datos actuales del aparcamiento.

2.2.2.11. Objetivo 9: Necesidad de la solución

Para averiguar y verificar que existe la necesidad de una solución como la propuesta en este PFG, realizamos dos preguntas directas que permiten reflejar el interés de los conductores.

¿Te gustaría tener una Web/App que te indicara donde hay una plaza libre para ti nada más entrar al parking?

58 respuestas

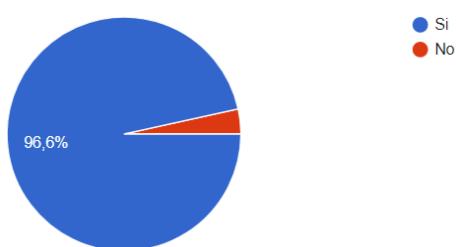


Figura 2.13: Necesidad de la solución – Elaboración Propia en Google Forms [18]

Se observó que un **96,6%** de los entrevistados les gustaría tener la aplicación propuesta en este PFG. Una aplicación web que permita visualizar el estado actual del *parking*.

La segunda pregunta que se realizó [Figura 0.50 en Anexo E] tiene que ver con la visualización en tiempo real, por lo que preguntamos al usuario, si le gustaría aparcar en un *parking* que informara de su estado actual. Los resultados son que el **94,8%** estaría interesados en poder aparcar en un aparcamiento inteligente.

2.2.2.12. Conclusiones

Podemos concluir que existe una necesidad entre los administradores y gestores del aparcamiento UFV y además existe una necesidad entre los alumnos, profesores y PAS de resolver el problema del *parking*.

Si se implementara el sistema según la encuesta:

- Se reduciría el estrés y el estado de ánimo mejoraría.
- Aumentaría el rendimiento académico. [19]
- Las personas no aparcarían siempre de manera ilegal.
- Habría más tiempo para lo que de verdad importa.
- Se podría encontrar plaza de manera efectiva y rápida.
- Se podría fomentar el uso de *CarSharing* creando un *parking* privado.

2.2.3. Seminario de Smart Cities UFV

Tabla 4: Datos del Profesor – Elaboración Propia

Datos del Profesor	
Nombre	Dr. José A. Ondiviela
Correo	josea.ondiviela@ufv.es
Posición	Director de SmartCities para Europa en Microsoft, Codirector Académico del Máster de Transformación de una Ciudad [20]
Fecha reunión	2 de noviembre de 2021

Para un mejor entendimiento de la temática elegida, que se encuentra dentro de un entorno universitario posicionado dentro de una Smart City, se asistió a un seminario [21] impartido con el Director del Observatorio de *Smart Cities* UFV [22], un proyecto/grupo de investigación que se generó en 2020 a raíz de su tesis doctoral [23] y lleva ya dos ediciones publicadas 2020 [24] y 2021 [25] que aportan mucha información relevante y clasifica las diferentes ciudades con programas de Smart City alrededor del mundo.

2.2.3.1. Digital Twin

Según Ondiviela en su seminario [21] las *Smart Cities* ofrecen una cantidad altísima de datos desde dos fuentes mayoritarias, los ciudadanos junto a sus dispositivos y los diferentes servicios públicos ofrecidos por el gobierno, como Vehículos, Edificios, Utilidades, entre otros. Una vez extraídos estos datos se utilizan diferentes técnicas como *data mining* o *inteligencia artificial* (IA), entre otras, con el objetivo de relacionar los diferentes datos y poder sacar conclusiones. Este servicio puede englobarse dentro de un sistema llamado Digital Twin. Según IBM [26]: “Un gemelo digital o Digital Twin es una representación virtual de un objeto o sistema que abarca su ciclo de vida, se actualiza a partir de datos en tiempo real y utiliza la simulación, el aprendizaje automático y el razonamiento para ayudar a la toma de decisiones.” Según lo aprendido en el seminario [21] la potencia de estos sistemas está en la posibilidad de realizar simulaciones como, por ejemplo, en una ciudad real, bloquear la entrada de

vehículos en una calle y visualizar las consecuencias. Esto permite tomar decisiones debido a que en la vida real hay tantos factores para tener en cuenta, por ejemplo, podría pasar que, al cortar una calle en una ciudad, la calidad del aire en una determinada zona empeorara y los ingresos en hospitales aumentasen, debido al incremento de gases en el aire.

2.2.3.2. Futuro de la Movilidad

Según Onvidiela en su seminario [21], la movilidad urbana es un factor muy importante para las *Smart Cities*. Como dijo Arthur D. Little en su informe [27] “La movilidad urbana es uno de los desafíos más complejos que las ciudades se enfrentan hoy, debido a que los sistemas de movilidad existentes están cerca de la decadencia total”. El aumento del número de coches en las ciudades es un problema real [27] y existe una gran oportunidad para integrar a los coches en los sistemas de las ciudades, debido a que la 96% de los nuevos coches producidos podrán conectarse a Internet en 2030, según un estudio realizado en 2021 por Martin Placek [28].

2.3. SOLUCIONES EXISTENTES

Algunas empresas ofrecen soluciones similares como [29] que permite controlar el aforo de un aparcamiento. Otras por ejemplo [30] ofrecen un servicio de control de accesos en oficinas, esto permite a los empleados reservar plaza y poder acceder al *parking* una vez que se detecte su matrícula con la cámara montada.

Lucas Sagarra Fernández en su TFM (*Trabajo Final de Máster*) [31] realiza una aproximación muy interesante con detección y conteo de vehículos utilizando Inteligencia Artificial, que si disponible pudiera ser utilizada en esta solución. Existen también productos como [32] que permiten contar el número de coches que entran y salen del aparcamiento, pero son muy limitados a nivel físico.

Algunas empresas proponen soluciones completas como [33] que tiene un sistema integral de control de luces, vigilancia, aparcamiento inteligente y análisis en *cloud* de los resultados obtenidos, además de permitir visualizar en tiempo real el aparcamiento. También existen numerosos estudios y proyectos para la realización de Smart Parkings dentro de un Smart Campus [34] [35] o en una Smart City [36] [37].

3. OBJETIVOS

3.1. OBJETIVOS GENERALES

Para este proyecto final de grado se definen los siguientes objetivos generales:

3.1.1. Optimizar el tiempo de búsqueda de plaza

Debido a que el problema de buscar plaza en un aparcamiento requiere un tiempo indefinido hasta encontrar plaza, suponiendo que no conocemos donde puede haber plazas libres, se plantea reducir el tiempo de búsqueda indicando al conductor donde se encuentra su plaza libre.

3.1.2. Presentar el estado actual del aparcamiento

Se realizará un Digital Twin que permitirá simular la lectura de una matrícula tanto en la entrada como en la salida, además de visualizar la información del estado actual del aparcamiento.

3.2. LISTA DE OBJETIVOS ESPECÍFICOS

- Desarrollar un simulador de datos de Cámara IP:
 - Un **simulador desarrollado en Python** que podrá **lanzar peticiones TCP** desde un socket utilizando un protocolo adicional de comunicación cifrado de extremo a extremo; simular la lectura de la matrícula de un coche y enviar la información al servidor.
 - Un **simulador desarrollado en JavaScript** que podrá lanzar peticiones **utilizando el protocolo Web Socket** que se encuentra sobre TCP y HTTP (Hypertext Transfer Protocol), y que de manera adicional utilizará el protocolo de comunicación seguro cifrado de extremo a extremo para enviar los datos de las matrículas simuladas.
- Desarrollar un servidor de aplicaciones:
 - El servidor deberá de ser capaz de **recoger y guardar los datos** recibidos desde las cámaras, conectándose de manera segura y almacenando la información en **una base de datos**.
 - Será capaz de **interpretar los protocolos TCP, HTTP, Web Socket** y el protocolo de comunicación segura con cifrado de extremo a extremo (SJMP).

- Diseñar y Desarrollar un protocolo de comunicación segura sobre TCP utilizando JSON:
 - El protocolo debe utilizar una **estructura JSON** para enviar los mensajes desde los diferentes clientes Python y JavaScript.
 - El protocolo debe utilizar el **cifrado asimétrico** de extremo a extremo que se discutirá las claves en un *handshake* inicial.
 - La cámara enviará un mensaje JSON al servidor con su clave pública, luego el servidor encriptará y enviará el mensaje JSON a la cámara, para que ella desencripte y guarde la clave pública del servidor, encriptando de extremo a extremo la comunicación en cada mensaje enviado en el futuro.
- Desarrollar un *Digital Twin* de una zona del *Parking UFV*:
 - El simulador deberá **presentar el estado actual del aparcamiento**, y será capaz de enseñar las plazas libres y se integrará con el simulador JavaScript para poder simular la entrada de coches y plaza recomendada.
- Desarrollar una Web App para conductores:
 - Para que los conductores puedan **registrar sus matriculas** y recibir **un ticket con su plaza al entrar al parking**.
 - Deberá **consultar la tabla en la base de datos** para comprobar si el usuario está o no está registrado.
 - Deberá notificar al usuario si está conectado en que plaza puede aparcar luego al entrar al *parking*.
 - Creará un *Ticket Virtual* con un QRCode (*Quick Response Code*) y token único, para el vehículo del conductor, que caducará a la salida.

3.3. MÉTODOS DE VERIFICACIÓN Y VALIDACIÓN

Los objetivos definidos serán traducidos en una serie de requisitos de usuario [Ver Anexo A] que especificará el sistema a desarrollar en detalle.

A lo largo del proyecto se validará siguiendo estos requisitos definidos, con el objetivo de asegurar que el sistema desarrollado es el correcto. De manera adicional se propone utilizar una plataforma como Azure DevOps [38], con el fin de centralizar el desarrollo y permitir la validación del sistema a lo largo de todo el proyecto.

Al final del desarrollo, se realizará un plan de pruebas [Ver Anexo I] que permita verificar cada requisito funcional, con su respectiva funcionalidad implementada en el sistema.

4. PLAN DE DESARROLLO DEL PROYECTO

4.1. METODOLOGÍA

Para el desarrollo técnico se utilizará el modelo de desarrollo en cascada modificado. Con diferencia del modelo en cascada tradicional, descrito en el manual de Ingeniería del Software, *Ian Sommerville* [39], el modelo en cascada modificado permite volver a la fase anterior para poder validar y consolidar los diferentes elementos [40]. Por ejemplo, se permitiría volver a definir los requisitos tras realizar la fase de “diseño y configuración”. En el modelo en cascada la fase final es “operación y mantenimiento” [39] Pg.31, debido a que está pensado para proyectos reales y escalables, por el contrario, este proyecto, no tiene una fase de mantenimiento, debido a que su ciclo de vida termina con la entrega final, por lo tanto, se adaptará el modelo y solo se tendrán en cuenta las siguientes fases:

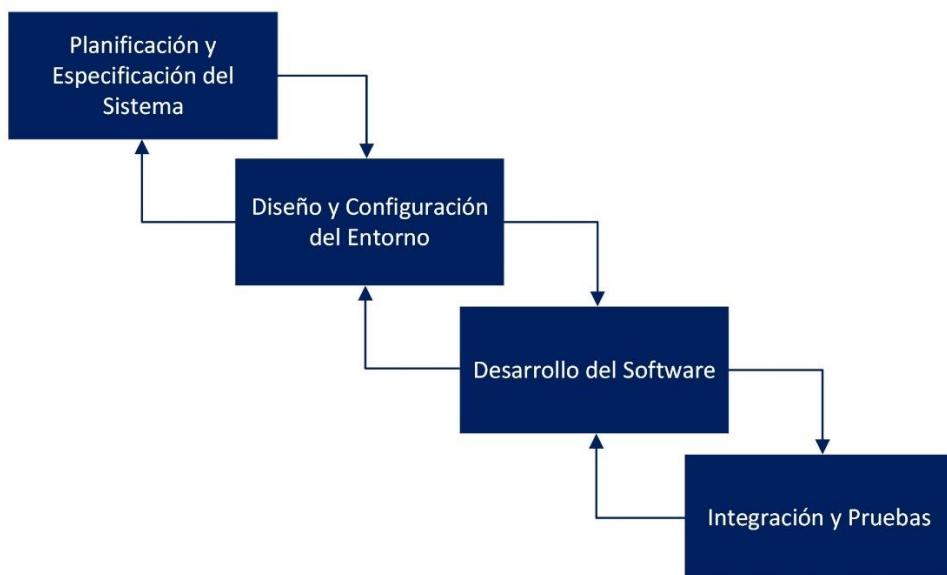


Figura 4.1: Esquema del modelo en cascada modificado. **Fuente:** Adaptado desde Sommerville [39] Pg.30, Figura 2.1, *Elaboración Propia.*

Se seleccionó la siguiente metodología porque este proyecto posee una cierta complejidad, debido a que utiliza diferentes tecnologías para integrar componentes de software específicos. Por consiguiente, se dedicará dentro del proyecto un periodo relativamente largo de tiempo a la planificación, lo que permitirá definir con libertad de adaptación cómo será y que tendrá el sistema antes de empezar con la implementación y validación de este.

4.2. TECNOLOGÍAS

4.2.1. Arquitectura de Tecnologías Software

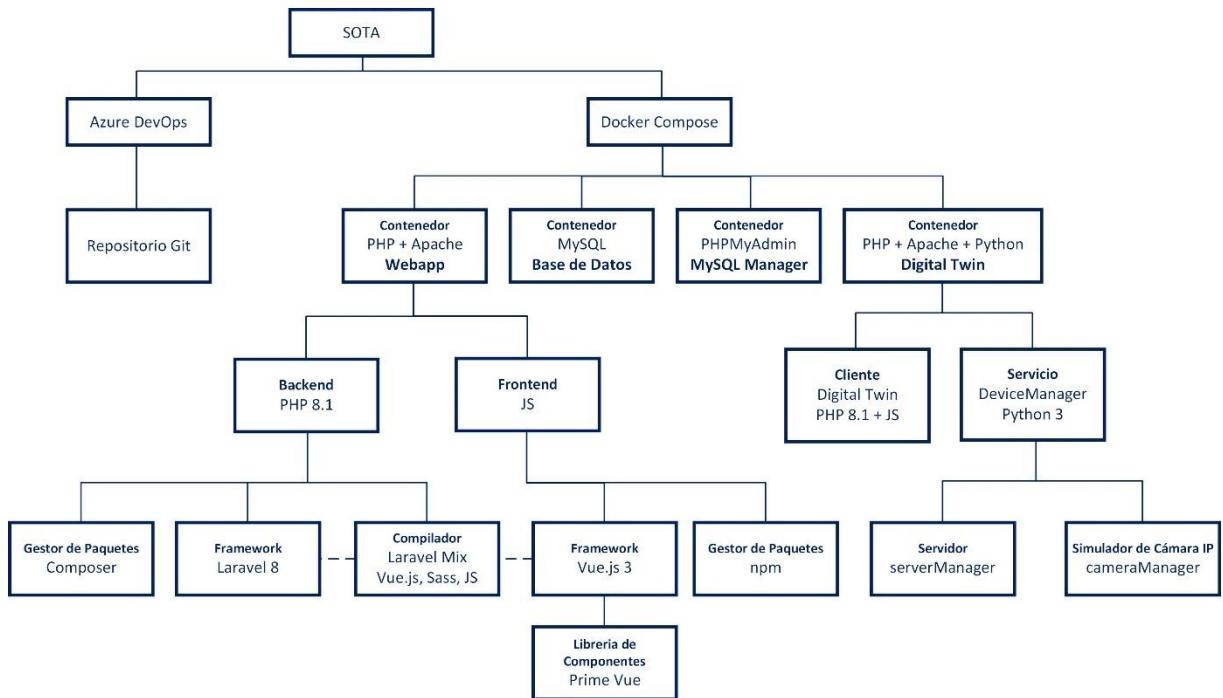


Figura 4.2: Arquitectura de Tecnologías Software. Fuente: Elaboración Propia

4.2.2. Lenguajes de Programación

4.2.2.1. PHP 8.1

Según sus creadores [41]: “PHP (Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.” Lo que facilita mucho el desarrollo y la integración de los servicios de acceso a base de datos (*backend*) con el desarrollo *frontend* (HTML, CSS y JS), permitiendo de manera simple la integración simultánea de los diversos lenguajes.

4.2.2.2. JavaScript

Según la fundación Mozilla [42] principal embajadora de este lenguaje lo define como: “JavaScript (JS) es un lenguaje de programación ligero, interpretado, o compilado justo-a-tiempo (*just-in-time*) con funciones de primera clase.” Poderosos *frameworks* de *frontend* como Vue.js [43] y React.js [44] utilizan JavaScript, lo que permite facilitar el desarrollo y aumentar la estabilidad cuando este lenguaje es compilado en el lado del servidor. Él también nos permite crear clientes como WebSockets [45], que se pueden conectar con servidores utilizando Application Programming Interfaces (API's).

Otra ventaja de utilizar JavaScript es porque está estandarizado [46], lo que permite a todos los navegadores utilizar esta tecnología de manera bien definida, aumentando la calidad.

4.2.2.3. Python 3

Según sus creadores: [47] “Python es un lenguaje de programación potente y fácil de aprender. Cuenta con eficientes estructuras de datos de alto nivel y un enfoque sencillo pero eficaz de la programación orientada a objetos”. Además de esto podemos acrecentar que hoy en día es el lenguaje más escalable y popular según muchos desarrolladores e investigadores [48] [49] [50] [51], además según los datos recogidos utilizando Google Analytics [52] es el lenguaje más buscado actualmente [53].

Como podemos observar la popularidad de Python, permite que haya toda una comunidad de desarrolladores que crean librerías y las dispongan de manera *open source* a la comunidad.



Figura 4.3: Logo Python: [47]

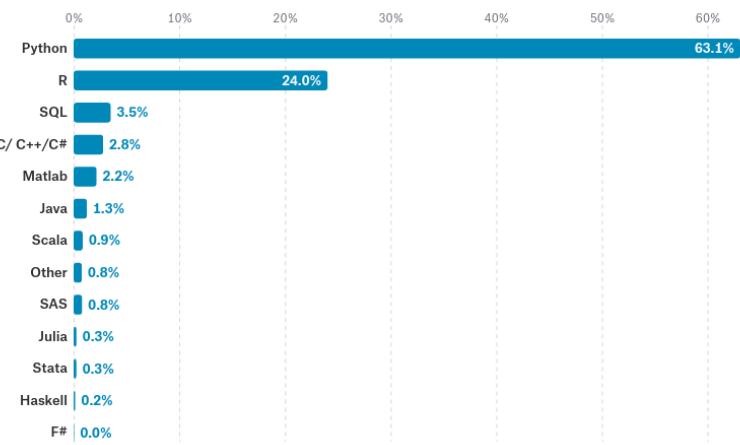


Figura 4.4: Estudio de lenguajes más utilizados para Análisis de Datos. Fuentes: [48] [50]

4.2.3. Frameworks

4.2.3.1. Laravel 8

Laravel [54] es un *framework* de aplicaciones web que utiliza PHP como lenguaje de programación. En comparación a los diversos *frameworks* de desarrollo *backend* para web como, por ejemplo, *Symfony* [55], Laravel se destaca por diversos motivos:

- Obliga desde el principio del desarrollo a seguir buenas prácticas de programación para estructurar las migraciones, los servicios y controladores.
- Permite un desarrollo seguro debido a que todas las comunicaciones están *tokenizadas*, lo que significa que contiene un token de sesión en cada conexión de manera nativa.
- Permite la compilación de ficheros JavaScript y Sass [56], utilizando Laravel.Mix [57], que incluso permite la compilación de *Frameworks* de *frontend* de manera nativa, como es el ejemplo de Vue.js.
- Facilita el acceso a la base de datos y permite la creación de API's seguras con protección contra SQL (*Structured Query Language*) *Injection* de manera nativa.

4.2.3.2. Vue.js 3

Vue.js [43] según sus creadores [58]: “es un *framework* progresivo de JavaScript para la creación de interfaces de usuario. Se basa en el HTML, CSS y JavaScript estándar, y proporciona un modelo de programación declarativo y basado en componentes que le ayuda a desarrollar eficazmente interfaces de usuario, ya sean sencillas o complejas”. Uno de los principales patrocinadores de Vue es el *framework* de Laravel, pues los profesionales que suelen desarrollar usando Laravel en *backend*, normalmente también conocen como utilizar Vue.js para construir las interfaces en *frontend*. Esto supone una oportunidad debido a que los dos *frameworks* están relacionados de manera nativa y permiten una integración compacta, segura y sencilla.



Figura 4.5: Logotipos de Laravel (Izquierda) y Vue.js (Derecha). Fuentes: [54] [43]

4.2.4. Gestores de Paquetes

4.2.4.1. Composer

Composer [59] es un potente gestor de paquetes y dependencias para PHP. Este gestor es necesario para poder descargar la estructura del *framework* de Laravel en PHP y será utilizado para añadir y actualizar las dependencias del entorno.

4.2.4.2. npm

npm [60] es el sistema gestor de paquetes de manera nativa de Node.js [61] y es utilizado por diversos *frameworks* en especial los que requieren una compilación de JavaScript como Vue.js [43], React.js [44]. *npm* permite la descarga de diversos paquetes y librerías como PrimeVue [62] que facilitan la construcción de las diferentes interfaces.

4.2.5. Bases de Datos

4.2.5.1. MySQL

MySQL [63] es un sistema gestor de base de datos relacional desarrollado por Oracle que es muy utilizado para entornos web, debido a su fácil integración con PHP y otros lenguajes de programación de *backend*. Actualmente es la segunda base de datos más utilizada en el mundo, según el ranking RDBMS [64] con una puntuación de 1198.23 [65] en 2021.

4.2.5.2. phpMyAdmin

phpMyAdmin [66] es una herramienta de software libre que permite administrar una base de datos *MySQL* o *MariaDB* [67] de manera sencilla y visual, utilizando una interfaz de usuario web. *phpMyAdmin* es frecuentemente utilizado en servidores tipo host que interpretan *php* [41] como herramienta de acceso a la base de datos por parte de un administrador.

4.2.6. Almacenamiento de Software y Despliegue

4.2.6.1. Azure DevOps

Azure DevOps [68] es una plataforma desarrollada por Microsoft que permite la integración de todas las áreas del desarrollo, desde la gestión de equipos, como la integración continua (CI/CD) y el almacenamiento del código en un repositorio privado. Facilita la gestión de un equipo que utiliza metodologías ágiles como Scrum [69] poniendo a disposición diferentes herramientas de control de backlog o tareas a hacer. Puede ser utilizado también, pero de manera limitada en un proyecto que usa la metodología en cascada, simplemente se deberán añadir los paquetes de trabajo del plan de proyecto al “backlog” y realizar un seguimiento de las tareas del equipo de desarrollo como hace Jira [70], por ejemplo. Esto permite al gestor del proyecto visualizar el avance del desarrollo y generar un espacio controlado y abierto a la colaboración dentro de un equipo, debido a que Azure DevOps permite crear foros de discusión, Wikis, Plan de Pruebas, entre otros servicios.

Se utilizará Azure DevOps para almacenar el código en un repositorio privado Azure Repo que utiliza Git [71]. El modelo de organización del repositorio a seguir será el de *Gitflow* [72] que permite una gestión eficiente y trazable de las ramas a lo largo del desarrollo.



Figura 4.6: Logo Azure DevOps. Fuente: [38]

4.2.6.2. Docker

Docker [73] es una plataforma de código abierto que permite la creación de contenedores (sistemas encapsulados) a partir de imágenes (entornos de sistemas preconfigurados) facilitando la separación de la infraestructura del software desarrollado [74], permitiendo realizar el desarrollo y las pruebas en un entorno controlado que puede replicarse de manera exacta una vez sea necesario desplegar el software. La principal ventaja de esta plataforma es que su arquitectura es muy robusta y compacta, lo que permite crear entornos constantes en infraestructuras cambiantes a partir de ficheros de configuración e imágenes creadas, actualizadas y almacenadas en repositorios *cloud* en *DockerHub* [75].

4.2.6.3. Docker Compose

Docker compose [76] es una herramienta utilizada para crear aplicaciones Docker multi-contenedor donde existen dependencias y un intercambio de información continua entre los contenedores. Para utilizar la herramienta debemos seleccionar las imágenes y nombrar los diferentes servicios a crear dentro de un fichero “*docker-compose.yml*”, que utiliza el lenguaje YAML (*Yet Another Markup Language*) [77], un lenguaje atractivo al ser humano, permitiendo configurar los parámetros de manera sencilla y eficiente.

Se utilizará *Docker Compose* para la creación del entorno de ejecución del sistema, montando los servicios en contenedores independientes, pero añadiendo dependencias e interconexiones que permiten el tráfico de información sea fluido. De esta manera se facilita el despliegue del sistema una vez terminado el desarrollo, debido a que los contenedores son independientes de la infraestructura utilizada y ofrecen una estabilidad a la hora de desarrollar, puesto que se pueden realizar pruebas en un entorno controlado.



Figura 4.7: Logo Docker (Izquierda) y Docker Compose (Derecha). Fuente: [73] [78]

4.2.7. Servidores

4.2.7.1. Apache

Dentro de los contenedores Docker que necesiten servir aplicaciones web, se utilizará el servicio de Apache [79]. Apache nos permite configurar el servidor de la manera que sea necesaria utilizando un lenguaje basado en XML (*eXtensible Markup Language*) [80] [81], siguiendo una estructura para poder configurar los permisos de accesos de usuario en cada directorio, así como gestionar las conexiones TCP/IP, HTTP y HTTPS.

4.2.8. Otras tecnologías

Para la realización de los diferentes productos desarrollados en este proyecto, se utilizarán las siguientes tecnologías. Que no se han mencionado en la arquitectura de software del sistema, pero si están presentes en el diseño del sistema.

4.2.8.1. Web Socket Protocol

Para la comunicación de la cámara JS que presentará los datos del *digital twin*, con el servidor Python dentro de *deviceManager*, se utilizará el protocolo *WebSocket* [82] que permite la comunicación de un socket abierto en JavaScript, en un navegador, con un servidor socket en Python. El protocolo va sobre HTTP, que a su vez va sobre TCP/IP, por lo que debemos tratar los bytes recibidos de una determinada forma, establecido

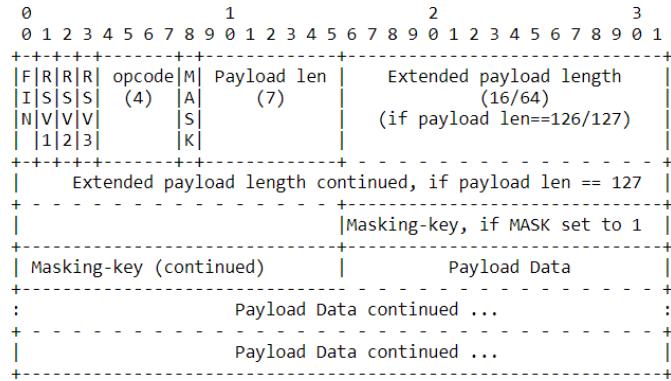


Figura 4.8: WebSocket Frame. Fuente: [82]

4.2.8.2. TCP/IP Protocol

Este protocolo es utilizado ampliamente en casi todas las conexiones de Internet a bajo nivel que implican una transmisión de datos de host a host. Permite enviar paquetes de bytes por medio de túneles establecidos entre diferentes sockets en la red. El protocolo TCP [83] es el responsable de transmitir la información mediante paquetes de datos, mientras que el protocolo IP [84] es el responsable de fragmentar y reorganizar la información enviada a direcciones fijas identificadas de una manera determinada en IPv4 como: [IP:PUERTO].

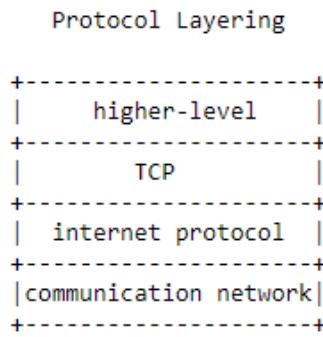


Figura 4.9: Capas del Protocolo TCP: Fuente: [83]

4.2.8.3. Protocolo HTTP

El protocolo HTTP [85] en su versión 1.1 es responsable de gran parte de las comunicaciones entre los clientes web y los servidores de este proyecto.

Para que las peticiones realizadas en HTTP sean validadas, deberán seguir un formato [*Request o Response*] dependiendo del mensaje, además deberán utilizar uno de los métodos permitidos [Ver Anexo D] y utilizar un código de *status* adecuado para cada situación.

4.3. PLAN DE DESARROLLO DEL PROYECTO

El plan de desarrollo del proyecto es clave para la ejecución de cualquier proyecto, establece las diferentes fases y paquetes de trabajo a realizarse a lo largo de un periodo establecido. Debido a que este proyecto intenta dar solución a un problema en Universidades se dedicó un largo periodo de tiempo de planificación y planteamiento del problema debido a que el sistema que se desarrolla en este proyecto consta de diferentes componentes que se integrarán en una estructura relativamente compleja.

La planificación sigue la metodología en cascada, por lo tanto, tendremos una fase de “Planificación y Especificación del Sistema” (Requisitos) seguida de una fase de “Diseño y Configuración” donde se define como se integrarán los diferentes componentes y vistas. Una fase de “Desarrollo” donde se implementa el software y se construye el sistema. En cada fase de desarrollarán primero las API's (*Backend*) para luego poder utilizarlas en las vistas (*Frontend*). Una vez terminado el desarrollo se probará el sistema siguiendo un plan de pruebas que permita verificar y validar a los requisitos de usuario.

4.3.1. Tareas Planificadas

El detalle de las tareas planificadas se encuentra en el Anexo G: Detalles Planificación. Ahí se encuentran desglosadas cada entregable de cada tarea, así como sus entradas y salidas.

4.3.2. Investigación Previa

La fase de investigación previa anhela encontrar y validar la verdadera necesidad del proyecto. En esta fase se deberá recoger información de los diferentes usuarios del sistema, gestores y usuarios del *parking*. Se busca también profundizar más en las tecnologías generadas en las soluciones del proyecto y en que Universidades españolas se aplican.

Tabla 5: Paquete de Trabajo: Investigación Previa – Elaboración Propia

Código	PT01	Nombre	Investigación Previa
Objetivo		Conocer a los usuarios del sistema, gestores, conductores, además del estado del arte de este tipo de tecnologías y el tema.	
Entradas		N/A	
Salidas		Estado del Arte, Problemas Actuales del Parking, Opinión de los Usuarios	
Tareas			<ul style="list-style-type: none">➤ PT01-T01: Reunión con Gestor Parking UFV➤ PT01-T02: Investigación del Estado del Arte de los Smart Campus en España➤ PT01-T03: Estudio del Parking UFV, con profesores, alumnos y PAS.

4.3.3. Definiciones Iniciales

En la fase de definiciones iniciales se planteará el problema, para ello se deberá acotar el alcance del proyecto definiendo objetivos y que tipo de sistema se desarrollará.

Tabla 6: Paquete de Trabajo: Definiciones Iniciales – Elaboración Propia

Código	PT01	Nombre	Definiciones Iniciales
Objetivo	Introducir el tema y poner en contexto el proyecto, definir los objetivos y alcance del proyecto.		
Entradas	Investigación Previa en Anteproyecto.		
Salidas	Estado del Arte, Problemas Actuales del <i>Parking</i> , Opinión de los Usuarios		
Tareas	<ul style="list-style-type: none"> ➤ PT02-T01: Introducción ➤ PT02-T02: Definición de Objetivos y Alcance 		

4.3.4. Planificación y Especificación del Sistema

La primera fase de la metodología en cascada es la definición de los requisitos de usuario para poder definir de manera detallada lo que se pretende hacer. Una vez que tenemos los requisitos podemos empezar a realizar el plan de proyecto y definir las especificaciones.

Tabla 7: Paquete de Trabajo: Planificación y Especificación del Sistema – Elaboración Propia

Código	PT03	Nombre	Planificación y Especificación del Sistema
Objetivo	Planificar y especificar, como, que y cuando se hará cada elemento en el proyecto		
Entradas	Objetivo y Alcance		
Salidas	Requisitos de Usuario, Plan de Proyecto, Mapa <i>Parking</i> 2D		
Tareas	<ul style="list-style-type: none"> ➤ PT03-T01: Requisitos de Usuario ➤ PT03-T02: Plan de Proyecto ➤ PT03-T03: Modelado 2D del <i>Parking</i> Seleccionado 		

4.3.5. Diseño y Configuración del Entorno

Tabla 8: Paquete de Trabajo: Diseño y Configuración del Entorno – Elaboración Propia

Código	PT04	Nombre	Diseño y Configuración del Entorno
Objetivo	Estudiar detalladamente las tecnologías y configurar el entorno.		
Entradas	Requisitos de Usuario y Plan de Proyecto.		
Salidas	Listado de tecnologías a utilizar, entorno configurado para empezar con el desarrollo		
Tareas	<ul style="list-style-type: none"> ➤ PT04-T01: Estudio de Tecnologías ➤ PT04-T02: Configuración del Entorno ➤ PT04-T03: Diseño del sistema 		

4.3.6. Desarrollo del Software

En el desarrollo del software se construirá el sistema planificado y diseñado anteriormente.

4.3.6.1. Base de Datos

Tabla 9: Paquete de Trabajo: Base de Datos – Elaboración Propia

Código	PT05-1	Nombre	Base de Datos
Objetivo	Configurar migraciones y el modelo de datos.		
Entradas	Requisitos de Usuario y Listado de Tecnologías.		
Salidas	Modelo de Datos y Base de Datos Configurada.		
Tareas	➤ PT05-T01: Configuración de Base de Datos		

4.3.6.2. Digital Twin

Tabla 10: Paquete de Trabajo: Digital Twin – Elaboración Propia

Código	PT05-2	Nombre	Digital Twin
Objetivo	Desarrollar el Digital Twin y el DeviceManager.		
Entradas	Requisitos de Usuario y Listado de Tecnologías.		
Salidas	Digital Twin, DeviceManager, Simulador Cámara IP		
Tareas	➤ PT05-T02: Conexión con Servidor ➤ PT05-T03: DeviceManager ➤ PT05-T04: Cámara IP Python ➤ PT05-T05: Pantalla de Visualización		

4.3.6.3. WebApp

Tabla 11: Paquete de Trabajo: WebApp – Elaboración Propia

Código	PT05-3	Nombre	WebApp
Objetivo	Desarrollar la Aplicación Web UFV MyParking		
Entradas	Base de Datos y Digital Twin		
Salidas	WebApp UFV MyParking		
Tareas	➤ PT05-T06: Autenticación ➤ PT05-T07: Esqueleto Dashboard y Home ➤ PT05-T08: Perfil de Usuario ➤ PT05-T09: Notificaciones ➤ PT05-T10: Área de Vehículos ➤ PT05-T11: Tickets ➤ PT05-T12: Incidencias		

4.3.7. Integración y Pruebas

Se realizará un periodo de integración y pruebas con el objetivo de hacer que el sistema sea seguro y estable.

Tabla 12: Paquete de Trabajo: Integración y Pruebas – Elaboración Propia

Código	PT06	Nombre	Integración y Pruebas
Objetivo	Probar el sistema y verificar que cumple los requisitos		
Entradas	<i>Digital Twin y WebApp</i>		
Salidas	Plan de Pruebas y Pruebas de Integración		
Tareas	<ul style="list-style-type: none"> ➤ PT06-T01: Pruebas de Integración ➤ PT06-T02: Pruebas de Verificación 		

4.4. PLAN DE TRABAJO

Para el correcto desarrollo del proyecto se hizo una estimación de la carga de trabajo en un plan de proyecto. Para una mejor visualización de las tareas se realizó un diagrama de Gantt, que permite visualizar el orden de ejecución y dependencias de las diferentes tareas:

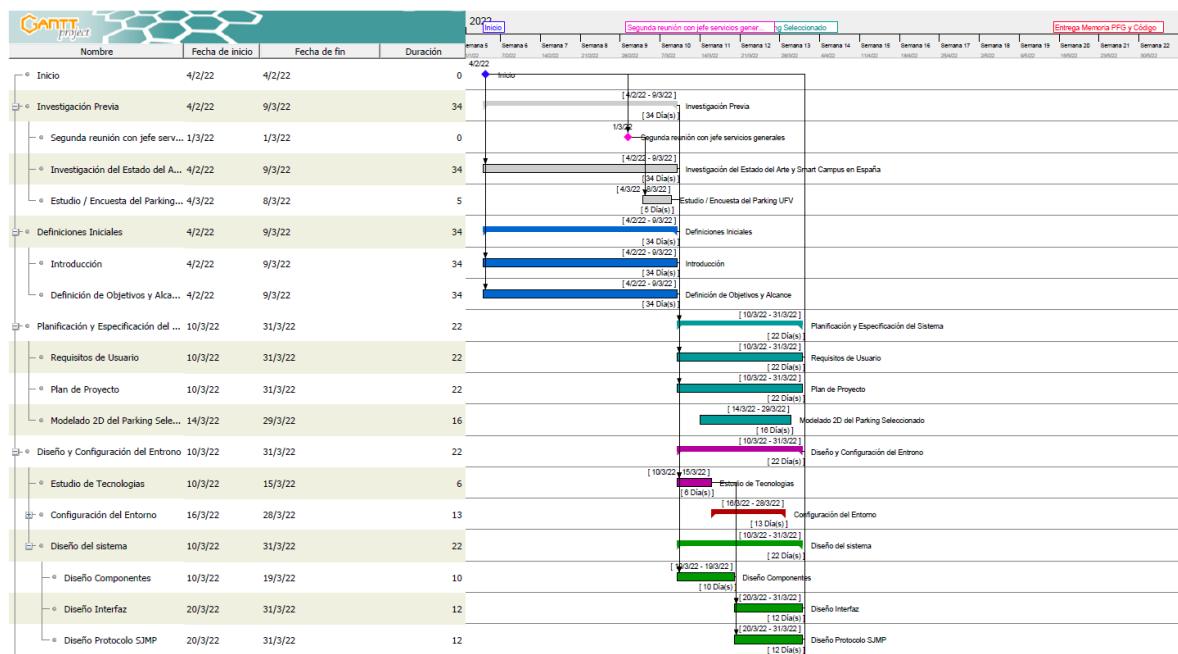


Figura 4.10: Diagrama de GANTT Parte I, Fuente: Elaboración propia en GanttProject [86]

En la primera parte del diagrama [Figura 4.10] se describen las fases iniciales del proyecto donde se realiza la definición, planificación y diseño del sistema estableciendo el camino a seguir.

En la segunda parte del diagrama [Figura 4.11] se describen las diferentes fases del desarrollo del software y sus plazos. Al final se encuentra la fase de pruebas e integración del sistema donde se comprueban los diferentes fallos que pueda haber y se realiza la verificación de los requisitos establecidos.

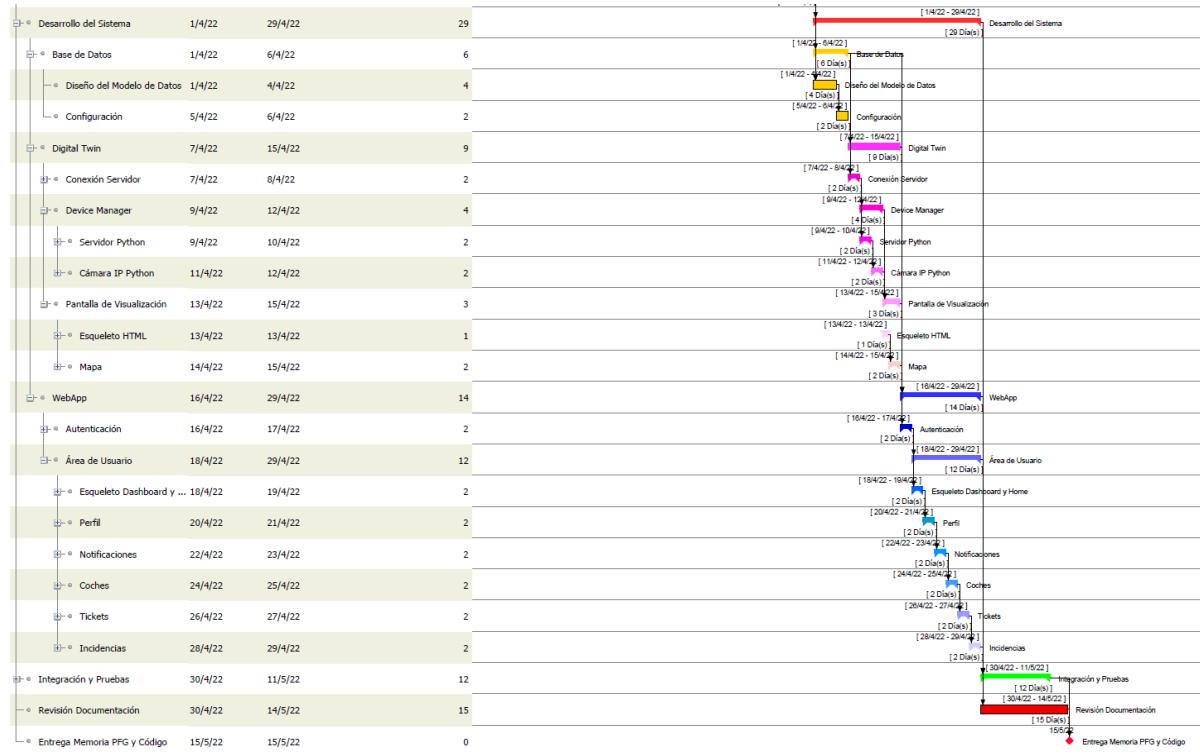


Figura 4.11: Diagrama de GANTT Parte II, Fuente: Elaboración propia en GanttProject [86]

Se generó también un diagrama de PERT que permite visualizar las relaciones entre las diferentes tareas del proyecto. El diagrama se encuentra ampliado en el [Anexo G: Detalles Planificación]

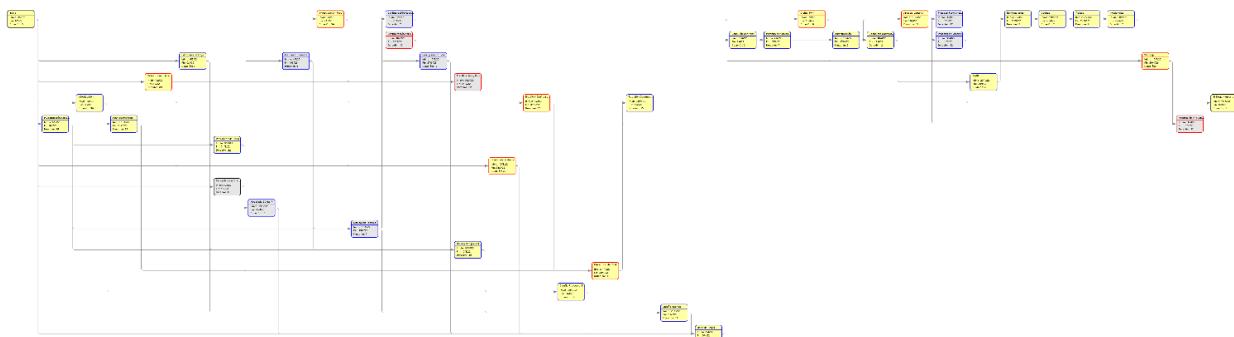


Figura 4.12: Diagrama de PERT: Relación entre Tareas, Fuente: Elaboración propia en GanttProject [86], Más detalle en: [Anexo G: Detalles Planificación]

4.5. RECURSOS.

Respecto a los recursos, que se deben incluir en el proyecto, necesitamos:

Tabla 13: Recursos del Proyecto – Elaboración Propia

Id	Nombre Recurso	Descripción	Precio
REC-01	Jefe de Proyecto	Responsable de coordinar el equipo, gestionar el proyecto, interactuar con el cliente y aprobar los documentos.	18€/hora
REC-02	Arquitecto de la solución.	Responsable de idear a la solución y el modelo de datos. Debe coordinar el desarrollo técnico.	15€/hora
REC-03	Desarrollador <i>Backend</i> I	Responsable del desarrollo <i>backend</i> e integración del software.	12€/hora
REC-04	Desarrollador <i>Backend</i> II	Responsable de la conexión con la base de datos y el tratamiento de los datos en <i>backend</i> .	12€/hora
REC-05	Desarrollador <i>Frontend</i> I	Responsable del <i>frontend</i> y diseño de interfaces gráficas.	10€/hora
REC-06	Desarrollador <i>Frontend</i> II	Responsable de presentar la información recibida de la conexión con el <i>backend</i> .	10€/hora
REC-07	Equipo de Pruebas	Serán los responsables por asegurar la calidad el producto antes de la subida a producción y durante todo el desarrollo.	7€/hora
REC-08	Documentador	Será el responsable de documentar todo el proceso desde el principio hasta el final de la solución.	8€/hora

[87] En este proyecto todos los roles son realizados por la misma persona (proyecto de carácter unipersonal), si bien, el disponer de diversos roles aceleraría el proceso y facilitaría la ejecución de diversas tareas de manera paralela.

Los costes especificados en los recursos son necesarios ya que jugarán un papel vital a la hora de desarrollar el apartado de los costes. En este apartado que explicaremos con más detalle a continuación, incluiremos tanto los costes de personal como los del material requerido para la realización del proyecto

4.6. COSTES

Debemos separar los costes diferenciando claramente entre el material y el personal requerido para el desarrollo del proyecto. Del mismo modo, incluiremos un coste que no conlleva un gasto económico en sí mismo, si bien juega un factor clave, que es el tiempo. Un retraso en un proyecto puede conllevar sobrecoste o incluso la pérdida del proyecto, por lo que es vital cumplir en el plazo establecido con lo acordado.

En cuanto a los materiales necesitamos lo siguiente:

Tabla 14: Recursos Materiales – Elaboración Propia

Nombre del Recurso	Precio	Unidades	Total
Licencia de Obra	1.200,00 €	1	1.200,00 €
Servidor Bases de Datos	60,00 €/mes	6	360,00 €
Servidor Pruebas y Despliegue	250,00 €/mes	6	1.500,00 €
Cameras IP, lectora de matrículas. [Figura 4.13]	1.830,93 €	2	3.661,86 €
Equipos Informáticos	1.500,00 €	2	3.000,00 €
Barreras Automáticas	1.086,58 €	2	2.173,16 €
Conmutadores o Switches	68,97 €	2	137,94 €
Rúter o Enrutador 10 Gigabit Ethernet	72,59 €	2	145,18 €
Cableado de Alimentación	48,13 €	7	336,91 €
Cableado Ethernet	67,22 €	1	67,22 €
Pantalla TFT (<i>Thin-film transistor</i>)	615,86 €	1	615,86 €
TOTAL			13.198,13 €

La estimación para estos costes es de 5 meses, desde febrero hasta junio.

En cuanto al coste humano del proyecto, procedemos a definir la tabla correspondiente detallando más a fondo los costes que incluimos en el apartado de los recursos. Debemos tener en cuenta que calculamos un coste por hora de cada uno de los miembros involucrados en el proyecto y que estos trabajarán 8 horas al día durante los 5 días laborables de la semana.

Teniendo en cuenta que el proyecto se espera desarrollar para el mes de mayo serían 4 meses efectivos de trabajo para los miembros, si bien vamos a incluir los costes hasta el mes de junio al igual que hemos realizado con los recursos materiales. Por lo tanto, el cálculo de horas será de 80 horas al mes, en un total de cinco meses, haciendo un total de 300 horas.

Para la ejecución de la obra civil se necesitan los siguientes servicios:

Tabla 15: Servicios para la Obra Civil – Elaboración Propia

Nombre Recurso	Precio Servicio	Trabajadores	Coste
Equipo de obras	960,00€/persona	2	1.920,00 €
Equipo de instalación de las redes	700,00€/persona	2	1.400,00 €
Equipo de configuración y soporte	700,00€/persona	2	1.400,00 €
Costes de transporte	-	-	160,00 €
TOTAL			4.880,00 €

Para el desarrollo técnico y la gestión del software se necesitan los siguientes recursos:

Tabla 16: Recursos Humanos para Gestión y Desarrollo – Elaboración Propia

Nombre del recurso	Precio	Horas	TOTAL
Jefe de Proyecto	18,00€/hora	400	7.200,00 €
Arquitecto de la solución.	15,00€/hora	250	3.750,00 €
Desarrollador Backend I	12,00€/hora	200	2.400,00 €
Desarrollador Backend II	12,00€/hora	100	1.200,00 €
Desarrollador Frontend I	10,00€/hora	150	1.500,00 €
Desarrollador Frontend II	10,00€/hora	100	1.000,00 €
Equipo de Pruebas	7,00€/hora	80	560,00 €
Documentador	8,00€/hora	100	800,00 €
Total			18.410,00 €

El coste definitivo del proyecto, sumando capital humano necesario para la obra civil y desarrollo, además del capital material hace un total de **36.488,13 euros** para su elaboración. Además, debemos incluir un margen de **8%** sobre el valor total, aproximadamente **2.919,05 euros**, para una posible formación del equipo o algún coste de gestión en cuanto a compra de hardware respecto a los servidores y cámaras. El coste definitivo con este margen quedaría en **39.407,18 euros**.

Tabla 17: Costes totales del proyecto – Elaboración Propia

Resumen General	Total
Materiales	13.198,13 €
Servicios Obra Civil	4.880,00 €
Recursos Gestión y Desarrollo	18.410,00 €
Coste de Gestión (8%)	2.919,05 €
Total	39.407,18 €
IVA (21%)	8.275,51 €
Total (IVA Incluido)	47.682,69 €

4.7. CONDICIONANTES Y LIMITACIONES

Debido al precio elevado de las cámaras IP que permiten la detección de matrículas, el alto coste de las barreras automáticas y que el alumno no tiene permiso ni recursos para la realización de obra civil, se optó por simular el entorno real con cámaras simuladas en Python utilizando Sockets TCP/IP. Además, se representará el aparcamiento en un entorno virtualizado incluido dentro de la solución del Digital Twin, donde se permitirá en un futuro gestionar cámaras IP en un entorno real.

5. DESARROLLO DE LA SOLUCIÓN TÉCNICA

5.1. INVESTIGACIÓN PREVIA

Al inicio de la investigación previa, se realizó una reunión con el jefe del Departamento de Servicios Generales UFV [13] [14] para conocer qué zona aparcamiento se podría utilizar como modelo para el Digital Twin. Sobre el plano proporcionado por él, se generó una delimitación por zonas de aparcamiento con colores y letras.

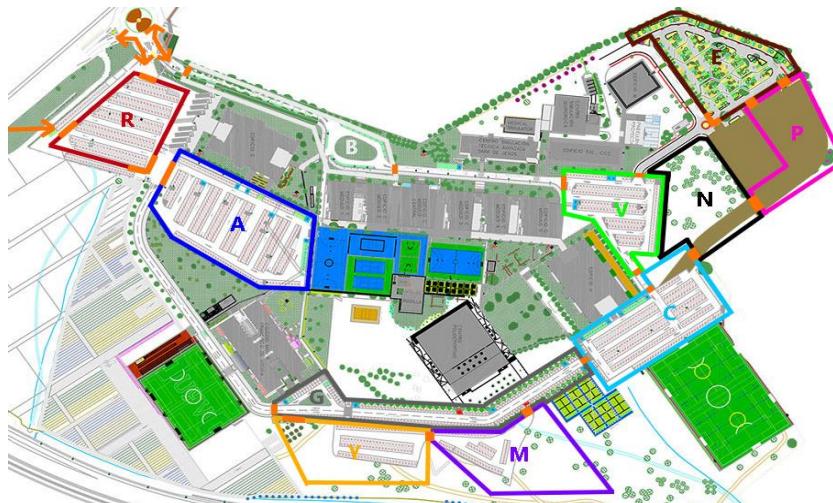


Figura 5.1: Zonas de Aparcamiento UFV – Elaboración Propia sobre el plano actual del Campus UFV [16] de Vanesa García

5.2. DEFINICIONES INICIALES

Según la investigación previa realizada, se tomó la decisión de acotar el alcance a un solo aparcamiento, en este caso el elegido es el de color Amarillo, Zona “Y” [Ver Figura 5.1], ya que puede adaptarse para tener una sola entrada interna entre aparcamientos y una salida a la vía principal. Otro motivo para elegir esta zona es por su arquitectura sencilla con un tipo de plazas estándar, permitiendo asignar zonas internas del parking con un tipo de plaza fijo.



Figura 5.2: Foto Panorámica de la Zona Y en Campus UFV – Elaboración Propia

5.3. PLANIFICACIÓN Y ESPECIFICACIÓN DEL SISTEMA

En esta parte se definieron los requisitos de usuario [Ver Anexo A], se realizó la planificación ya previamente detallada en la memoria y se modeló la Zona “Y” del aparcamiento UFV [Ver Figura 5.1] en 2D (vista aérea), empezando con un boceto en papel [Ver Figura 0.8 Anexo F] hasta modelarlo por completo dividiendo por zonas, colores (para una fácil identificación) y tipo de plaza. Se tomaron medidas de cada tipo de plaza y de las zonas [número de plazas y formato], con el objetivo de incluirlas en los requisitos del usuario [Ver Anexo A-Modulo 006].

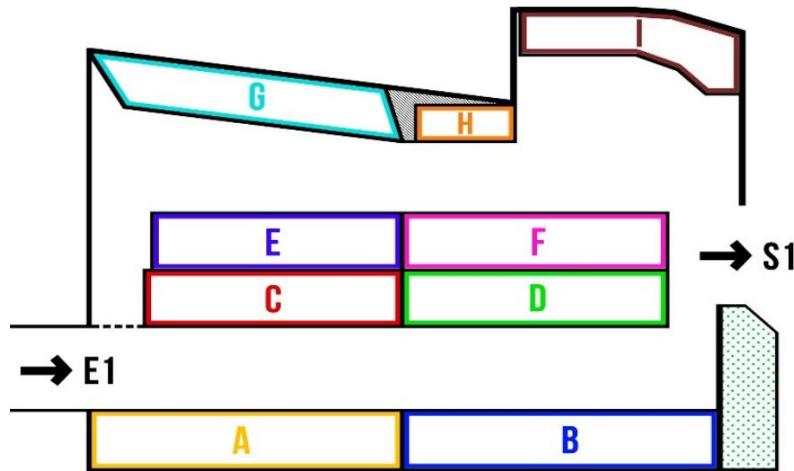


Figura 5.3: Modelado 2D (vista aérea) de la Aparcamiento Zona “Y” del Campus UFV. [Ver Figura 5.1] - Zonas identificadas con letras y colores, entradas/salidas y arquitectura– Elaboración Propia

5.4. DISEÑO Y CONFIGURACIÓN DEL ENTRONO

En la parte de diseño, se diseñó la arquitectura del sistema y se seleccionaron las tecnologías que se podrían utilizar para desarrollarlo. Se crearon diferentes bocetos del sistema [Ver Anexos G] hasta llegar a dos esquemas simplificados:

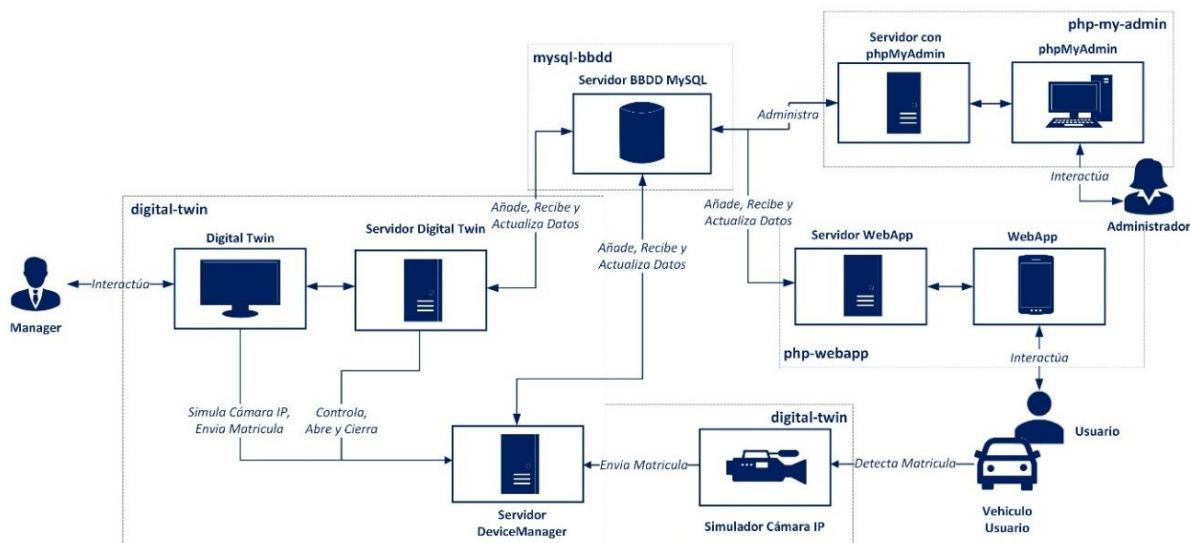


Figura 5.4: Esquema Funcional y Arquitectura de Contenedores – Elaboración Propia

Uno conceptual y uno donde se indica la arquitectura de contenedores Docker, integrados y desplegados utilizando *Docker Compose*. Se configuró comprobando que todos tenían acceso entre sí en una red privada entre contenedores nombrada como *appNet*.

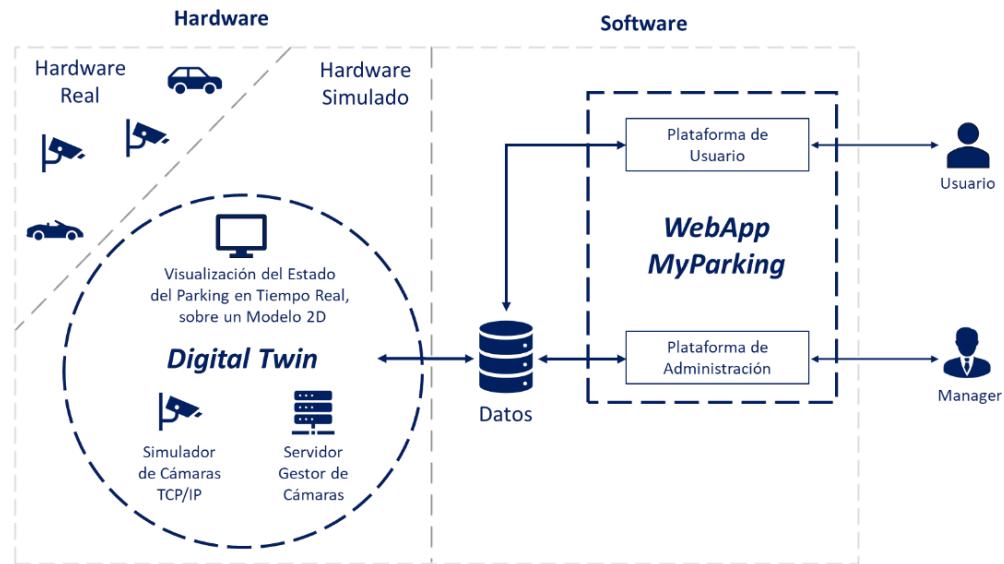


Figura 5.5: Esquema a Nivel Conceptual – Elaboración Propia

Dos esquemas más han sido realizados a nivel de red y protocolos de comunicación. Estos esquemas se encuentran en el Anexo B y contienen más detalles del sistema.

En la parte de configuración del entorno se estudiaron las tecnologías que se podrían utilizar para desarrollar y componer el sistema. Se configuró el entorno de desarrollo en Azure DevOps, creando el repositorio y añadiendo las tareas en la lista de “*Work items*”.

Para almacenar el sistema se eligió la siguiente estructura de directorios:

- **./data** → Contiene el volumen persistente del contenedor de la base de datos.
- **./digital-twin** → Contiene el servidor en: *./deviceManager*, la web en *./src* + configuraciones.
- **./webapp** → Contiene la web en *./src* y configuraciones del servidor apache + Laravel y Vue.js.

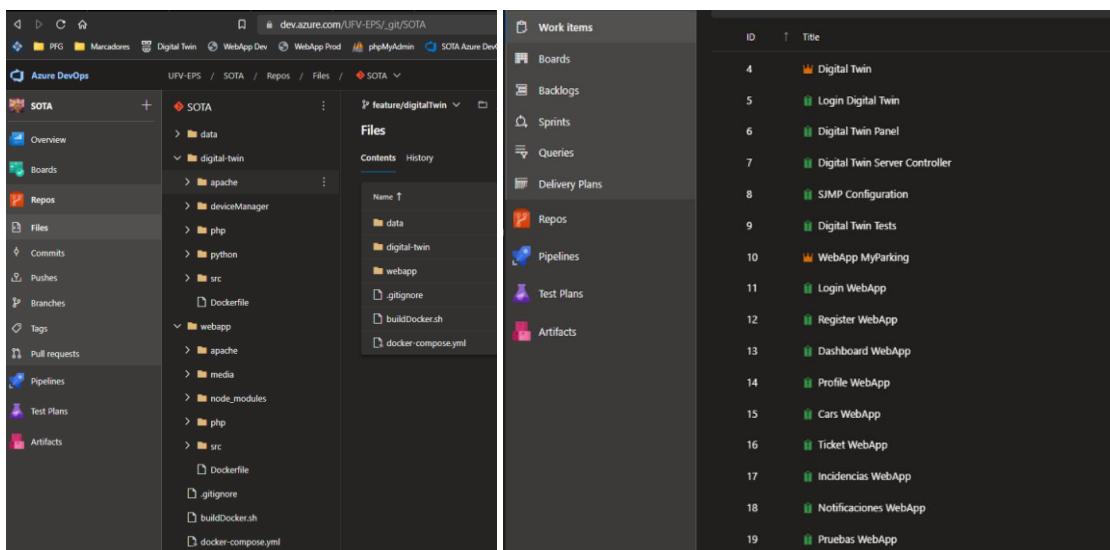


Figura 5.6: Captura del Repositorio en Azure DevOps – Elaboración Propia

5.5. DESARROLLO DEL SOFTWARE

5.5.1. Base de Datos

En la fase de diseño de la base de datos se ideó el modelo de datos [Ver Figura 5.7] relacional. Primero se pensó en las diferentes relaciones de los componentes y usuarios del sistema a papel [modelo entidad-relación] [Ver Figura 0.32 Anexo F], luego utilizando la página dbdiagram.io [88] se diseñó utilizando un lenguaje “pseudo SQL” para poder relacionar las tablas, utilizar claves externas, indicar los tipos de datos y configurar el esquema de distribución de los datos en el sistema, permitiendo convertir en diferentes variaciones de SQL. Se generaron 14 tablas, donde una de ellas [TipoVehiculoTipoPlaza] se genera para eliminar una relación [N:N]. Todas las demás relaciones son de tipo [1:N], donde las claves primarias, “ids”, se añaden como claves externas en otras tablas para indicar la pertenencia.

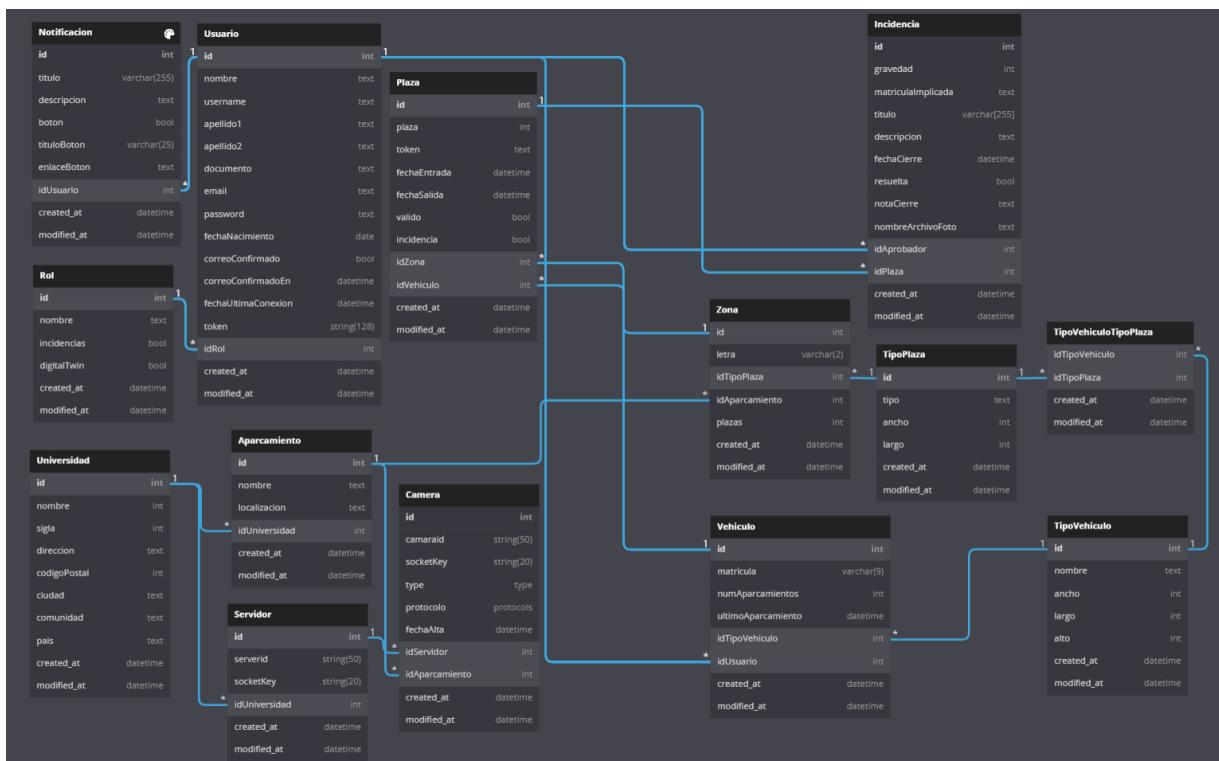


Figura 5.7: Modelo de Datos y relaciones entre Tablas. – Fuente: Elaboración propia utilizando: dbdiagram.io [88]. Descripción: El símbolo * se interpreta como N (muchos).

Después de generar el script en SQL tipo MySQL, se generó una migración de la base de datos utilizando el patrón de diseño “Data Access Object”. Se configuró y probó la base de datos con generadores de datos llamados “Seeders” en Laravel.

Estos generadores permiten construir la estructura inicial de la base de datos, añadiendo datos por defecto necesarios para el funcionamiento de la aplicación como: Roles, Usuarios Administradores, Zonas de Aparcamiento, la Universidad, el Aparcamiento, los tipos de vehículos, los tipos de plaza y las restricciones de aparcamiento de vehículos en plazas específicas.

```

    Vehiculofactory.php
    Zonafactory.php
    migrations
        2022_04_10_151205_create_Universidad_table.php
        2022_04_10_1512106_create_Rol_table.php
        2022_04_10_1645157411_create_Servidor_table.php
        2022_04_10_1649517318_create_Aparcamiento_table.php
        2022_04_10_1649596846_create_Users_table.php
        2022_04_10_1649597116_create_Notificacion_table.php
        2022_04_10_1649597250_create_Vehiculo_table.php
        2022_04_10_1649597284_create_TipoPlaza_table.php
        2022_04_10_1649597297_create_TipoVehiculoTablePlaza...
        2022_04_10_1649597309_create_Zona_table.php
        2022_04_10_1649597360_create_Plaza_table.php
        2022_04_10_1649597384_create_Incidencia_table.php
        2022_04_10_1649597403_create_Camara_table.php
    seeds
        AparcamientoSeeder.php
        ...

```

Figura 5.8: Migraciones establecidas en Laravel – Elaboración Propia

A continuación, se configuró el interfaz de administración de base de datos phpMyAdmin:

Tabla	Acción	Filas	Tipo	Colejamiento	Tamaño	Residuo a depurar
Aparcamiento	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8mb4_unicode_ci	32,0 KB	-
Camara	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_unicode_ci	89,0 KB	-
Incidencia	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8mb4_unicode_ci	48,0 KB	-
migrations	Examinar Estructura Buscar Insertar Vaciar Eliminar	15	InnoDB	utf8mb4_unicode_ci	16,0 KB	-
Notificacion	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8mb4_unicode_ci	32,0 KB	-
personal_access_tokens	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_unicode_ci	16,0 KB	-
Rol	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	utf8mb4_unicode_ci	32,0 KB	-
Servidor	Examinar Estructura Buscar Insertar Vaciar Eliminar	9	InnoDB	utf8mb4_unicode_ci	64,0 KB	-
TipoPlaza	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	utf8mb4_unicode_ci	32,0 KB	-
TipoVehiculo	Examinar Estructura Buscar Insertar Vaciar Eliminar	5	InnoDB	utf8mb4_unicode_ci	48,0 KB	-
TipoVehiculoTipoPlaza	Examinar Estructura Buscar Insertar Vaciar Eliminar	8	InnoDB	utf8mb4_unicode_ci	32,0 KB	-
Universidad	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8mb4_unicode_ci	16,0 KB	-
users	Examinar Estructura Buscar Insertar Vaciar Eliminar	5	InnoDB	utf8mb4_unicode_ci	64,0 KB	-
Vehiculo	Examinar Estructura Buscar Insertar Vaciar Eliminar	7	InnoDB	utf8mb4_unicode_ci	64,0 KB	-
Zona	Examinar Estructura Buscar Insertar Vaciar Eliminar	9	InnoDB	utf8mb4_unicode_ci	64,0 KB	-

Figura 5.9: Tablas de la Base de Datos en phpMyAdmin – Elaboración Propia, Captura de Pantalla.

5.5.2. Digital Twin

5.5.2.1. DeviceManager

Se configuró el servidor de cámaras socket, para poder enviar mensajes utilizando el protocolo de intercambio de paquetes SJMP, que encripta la información a nivel de dato [Ver Anexo D, Detalles Técnicos]. Se creó un menú en el simulador de cámara, para poder simular la conexión de una cámara o desconectarla. Al conectarla se hace el intercambio de claves públicas de la cámara y el servidor:

```

[DEBUG]: [mathias]->[SENT CONNECTION MESSAGE]
['flag': 'SYN', 'clt-time': 1658591103.877672, 'cameraid': 'mathias', 'token': '-----BEGIN PUBLIC KEY-----\nMIIBJANBkgkIg9w0BQAQFAOCQ8AMIIIBCgKCAEAg4uM0CSX+VFYdWrvth/A\nnnDFFBvIecYB1gChyQfsmCrBx030dzWQ8JpP+H+jxhW9D9+cCwzG5/8Mk0+n5HP+PLCJ7mfB0wUe6B/uKu9vZq0830nP2qzgsVv/Xw2R/iy98KXIMWhcAEIA0y\nn+iq+oKCdux/REqdQxDSDpo8xF1l0zqmXCapTc1+FDGNf10@6eyFweee50u0AXHDm0us/n18rNvcN+p\nt:PH6UB0RWiXz0Rwz20GavS19+BH2cd4F3Wtx5up9tnzL0FYBNjEV\n\\nTtFV/5eJBrsauwAxJU\n0/CSZvHPvE0k7BVaaGnyMcLbGIECNZYWdtcnrrect+IZfM\\nQIDAQABn-----END PUBLIC KE
Y-----\n', 'type': 'BOTH'}
[DEBUG]: [mathias]->[SENT SJMP SYN MESSAGE]
[DEBUG]: Camera [mathias], status [CONNECTED] on [127.0.0.1:60290] type [<class 'protocols.TCP SJMPProtocol.TCP SJMPProtocol'>] in [localhost:8888]
[DEBUG]: Camera [mathias] listening in [<class 'camera.socket.TCP SJMPSocketCamera'>->mathias-localhost:8888]

-----n>
1 -> Start new Camera
2 -> Start Default Camera
3 -> Disconnect Camera
4 -> Send Message
5 -> List all cameras
6 -> Exit

Choose option:
[Admin\cameras\cmd] >

```

```

[DEBUG]: [DEFAULT]-[127.0.0.1:8888] BaseRequestHandler.setup()
[DEBUG]: [DEFAULT]-[127.0.0.1:8888] > Received one request from 127.0.0.1:60
286
[DEBUG]: [NEW] RAW Connection Data Received: []
[DEBUG]: Incoming [camera.TCP Camera.TCP Camera] Protocol Connection...
[DEBUG]: Camera [127.0.0.1:60288] logged in with sessionid = [8f2e90e13fe297a1bedd2fb5ad5cd87] type [<class 'prot
ocols.TCPProtocol.TCPProtocol'>]
[DEBUG]: [DEFAULT]-[127.0.0.1:8888] BaseRequestHandler.setup()
[DEBUG]: [DEFAULT]-[127.0.0.1:8888] > Received one request from 127.0.0.1:60
288
[DEBUG]: [NEW] RAW Connection Data Received: [{"flag": "SYN", "clt-time": 1658591103.877672, "cameraid": "mathias", "token": "-----BEGIN PUBLIC KEY-----\nMIIBJANBkgkIg9w0BQAQFAOCQ8AMIIIBCgKCAEAg4uM0CSX+VFYdWrvth/A\nnnDFFBvIecYB1gChyQfsmCrBx030dzWQ8JpP+H+jxhW9D9+cCwzG5/8Mk0+n5HP+PLCJ7mfB0wUe6B/uKu9vZq0830nP2qzgsVv/Xw2R/iy98KXIMWhcAEIA0y\nn+iq+oKCdux/REqdQxDSDpo8xF1l0zqmXCapTc1+FDGNf10@6eyFweee50u0AXHDm0us/n18rNvcN+p\nt:PH6UB0RWiXz0Rwz20GavS19+BH2cd4F3Wtx5up9tnzL0FYBNjEV\n\\nTtFV/5eJBrsauwAxJU\n0/CSZvHPvE0k7BVaaGnyMcLbGIECNZYWdtcnrrect+IZfM\\nQIDAQABn-----END PUBLIC KEY-----\n", "type": "BOTH"}]
[DEBUG]: Incoming [camera.TCP Camera.TCP Camera] Protocol Connection...
[DEBUG]: Camera [mathias], status [ONLINE] on [127.0.0.1:60290] logged in wi
th sessionid = [f7c62510f7f10f00314d055fda1f5497] type [<class 'protocols.TC
PSJMPProtocol.TCP SJMPProtocol'>]
[DEBUG]: [DEFAULT]-[127.0.0.1:8888] BaseRequestHandler.setup()
[DEBUG]: [DEFAULT]-[127.0.0.1:8888] > Received one request from mathias
[STATS]: [18] tables found in [0:00:00.912843] DBManager.refreshDatabase()

```

Figura 5.10: Intercambio de claves en paquetes SJMP entre Cámara Socket TCP (Izquierda) con Menú de Opciones en CMD y Servidor Socket (Derecha) en Python - Elaboración Propia.

Cuando se conecta una cámara posicionada en aparcamiento se genera un intercambio de claves de cifrado asimétrico RSA, lo que permite la encriptación a nivel de dato, haciendo que los datos estén seguros independientemente de la red utilizada para el envío.

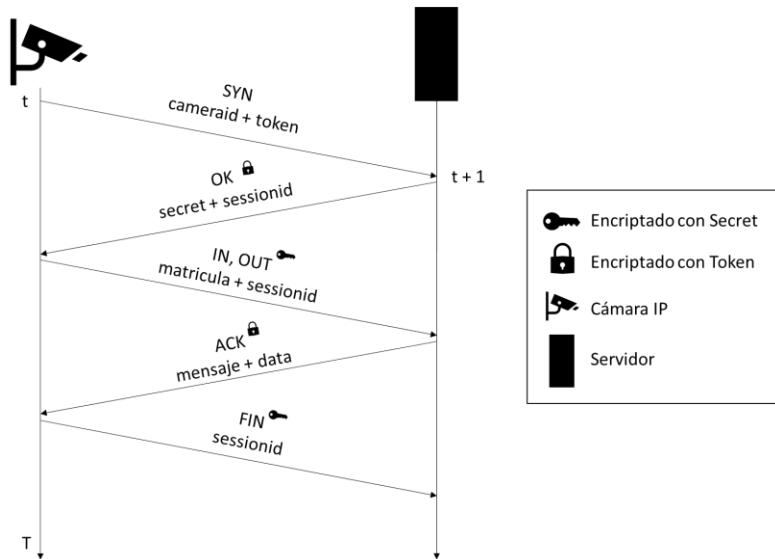


Figura 5.11: Esquema del Intercambio de claves asimétricas en SJMP – Elaboración Propia

Una vez diseñado y configurado el protocolo, se desarrolló un controlador (*serverController*) programado en “*bash script*” y *php*, que permite abrir los servidores, cerrarlos, visualizarlos y buscarlos. El controlador permite abstraer los comandos del sistema operativo, para poder controlar los servidores desde el Digital Twin en una aplicación web.

5.5.2.2. Pantalla de Visualización

Al entrar en la pantalla de visualización del Digital Twin se solicita al usuario que se autentifique utilizando sus credenciales de administrador registradas en la otra aplicación de “*WebApp MyParking*”. Una vez que el usuario introduce su correo y la contraseña, el sistema se utiliza un token de sesión único (un hash sha256) que permite identificar al usuario a lo largo de todo el sistema, independiente de la aplicación. Se genera a partir de la concatenación del correo y la contraseña *hasheada* con *sha512*, haciendo el *login* seguro.

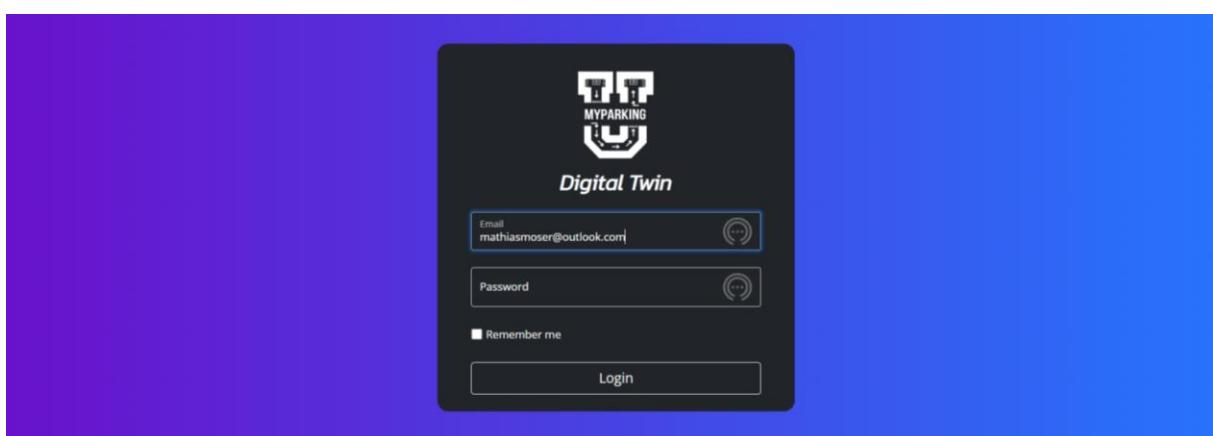


Figura 5.12: Pantalla de Login del Digital Twin – Elaboración Propia

La aplicación cuenta con la posibilidad de cambio a modo oscuro, permite al gestor simular la cámara de entrada, la cámara de salida, ver los detalles de las plazas ocupadas y simular la apertura y cierre de las barreras. También se visualizan los datos configurados en la Base de Datos (BBDD). Para poder interactuar con las cámaras primero se deberá de abrir el servidor, presionando el botón “Open Server”. Esto abrirá un modal que permite introducir los datos de conexión para desplegar el servidor sobre un socket (IP:PUERTO) y conectar el simulador de la Cámara TCP/IP utilizando el protocolo SJMP, enviado sobre el protocolo de *WebSocket*.

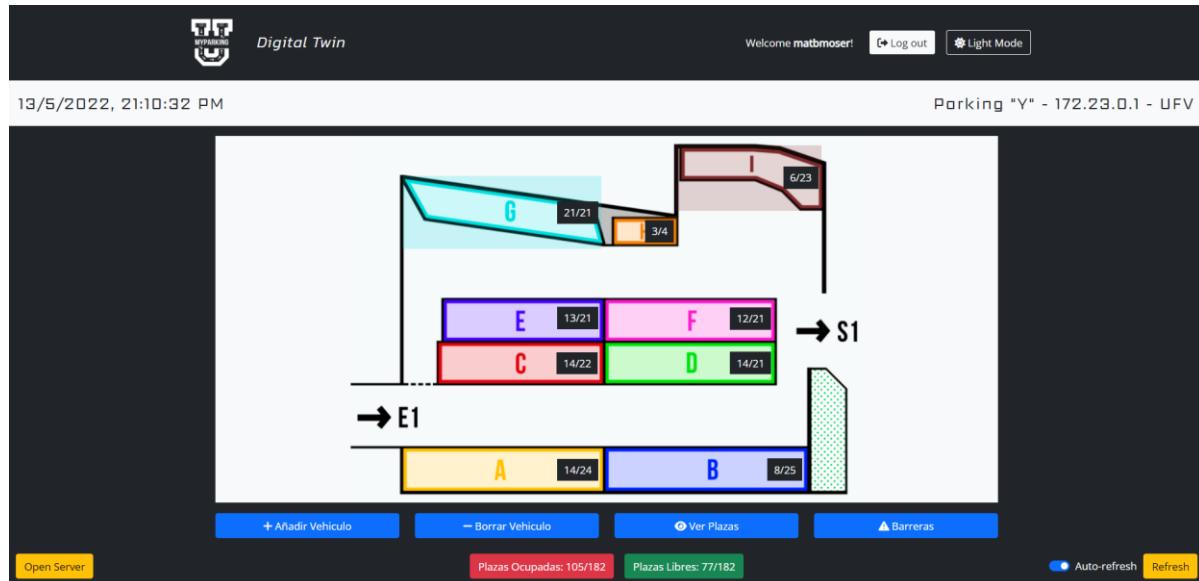


Figura 5.13: Pantalla de Visualización del Digital Twin – Elaboración Propia

Tanto el servidor como la cámara podrán abrirse y cerrarse. También podemos recargar la página y se volverá a reconectar la cámara siempre que el servidor siga abierto.

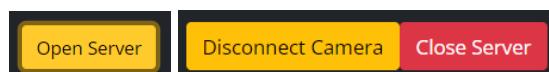


Figura 5.14: Botones Abrir Server / Cerrar y desconectar cámara – Elaboración Propia

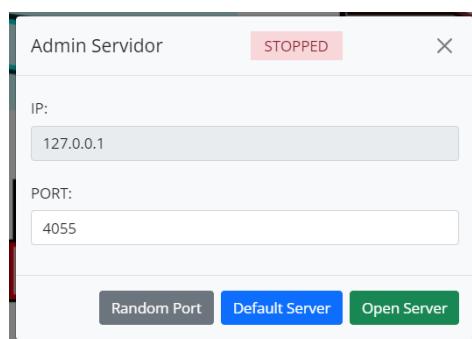


Figura 5.15: Configuración de apertura de Servidor – Elaboración Propia

El *digital twin* cuenta con botones de acción para poder realizar simulaciones en el *parking* y ver su estado actual, haciendo clic sobre el botón de Ver Plazas.



Figura 5.16: Botones de Acción – Elaboración Propria

Se configuró la conexión con el servidor de cámaras, y el simulador de la cámara que utiliza el protocolo *WebSocket* para comunicarse. La cámara es capaz de conectarse al servidor y recibir su clave de sesión para poder enviar matrículas, siempre que esté conectada,

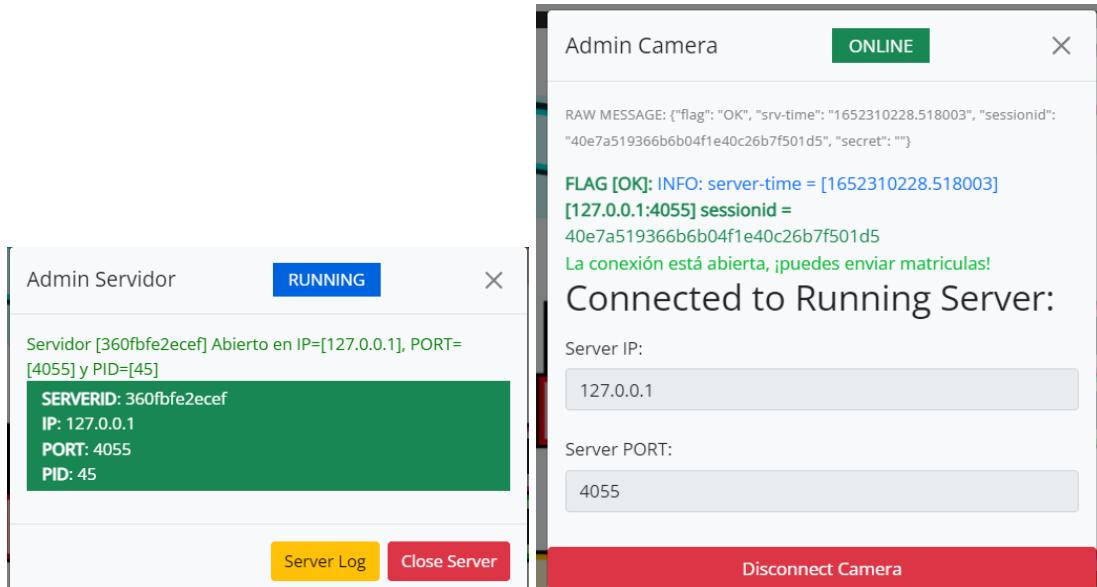


Figura 5.17: Apertura de Servidor y Conexión de Cámara con WebSockets – Elaboración Propia

Se implementó la simulación de envío de matrículas. Si el vehículo esta registrado en la base de datos, el servidor asignará la plaza según el tipo del coche con la matrícula indicada.



Figura 5.18: Asignación de Plaza (Izquierda) y Simulación de Salida (Derecha) - Elaboración Propia

Si el gestor/manager hace clic sobre ver plazas podrá visualizar las plazas ocupadas:

Plazas Ocupadas				
idPlaza	Matrícula	Segmento	Tamaño	Fecha Entrada
E1	3201-PGL	E	grande	2022-05-09 21:29:51
E2	1234-ZXC	D	medio grande	2022-05-09 21:29:51
F3	4561-QWR	A	pequeño	2022-05-09 21:29:51
G1	1605-LDJ	C	medio	2022-05-11 23:15:00
H1	4321-ZXC	E	grande	2022-05-09 21:29:51
H2	1247-LDG	A	pequeño	2022-05-09 22:15:27

Figura 5.19: Plazas Ocupadas – Elaboración Propia

Utilizando el Simulador de la barrera el gestor tiene el control para abrir y cerrar el *parking* manualmente si es necesario. En este caso se esperan 10 segundos para cerrarla de manera automática, pero podrá ser cerrada por el administrador haciendo clic sobre cerrar barrera.



Figura 5.20: Simulador de Barrera – Elaboración Propia



Figura 5.21: Aforo Total del Parking y Botón de Refresh y Auto-Refresh – Elaboración Propia

5.5.3. WebApp

5.5.3.1. Autentificación

Para la autentificación (registro y *login*) se configuró dos *APIs* que generan tokens JWT (JSON Web Tokens) [89] que identifican la sesión del usuario y permiten al *WebApp* identificar si el usuario está registrado durante todo el uso de la aplicación.



Figura 5.22: HTTP POST Request para Registro de Usuario utilizando API (/api/auth/register) creada en Laravel – Fuente: Elaboración Propia utilizando la Herramienta de Postman [90].

A continuación, se crearon dos formularios en Vue.js para el *login* y el registro, que hacen uso de las *APIs* desarrolladas y descritas previamente. Para el registro el usuario deberá aceptar una política de privacidad definida en el sistema. Las contraseñas se encriptan con *bcrypt* [91], para asegurar la seguridad de las contraseñas de los usuarios.

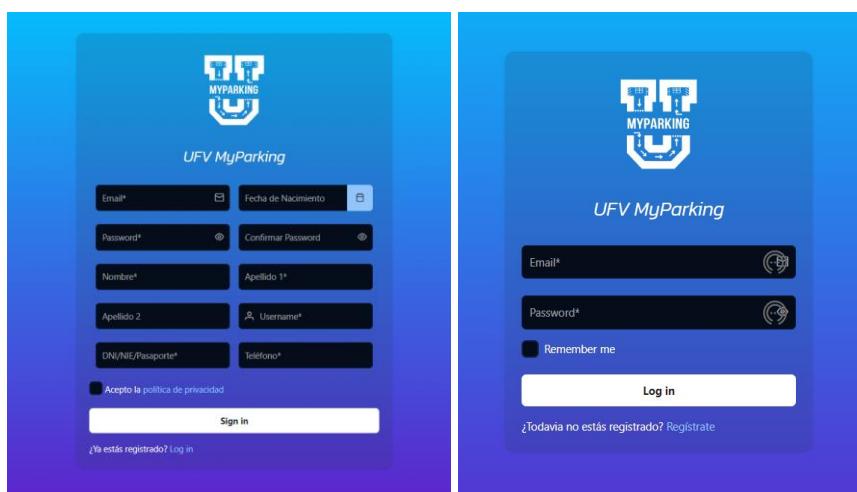


Figura 5.23: Vistas de Registro y Login en UFV MyParking – Elaboración Propia

5.5.3.2. Área de Usuario

Para el área del usuario se definió la estructura de menú y los “routes” para cada página del WebApp. El menú se despliega a la izquierda y contiene botones dinámicos que se ocultan/aparecen según el rol del usuario. Si el usuario tiene permisos de administrador tendrá en su menú más opciones para el área de administración del sistema, para borrar usuarios, modificar roles y resolver incidencias.

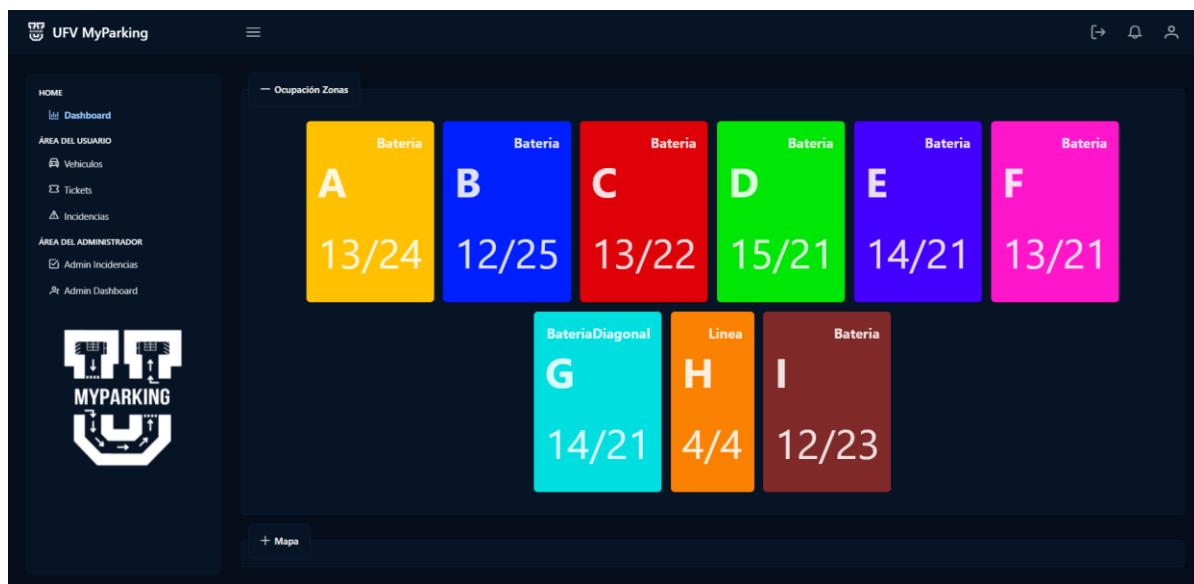


Figura 5.24: Dashboard visualización pública del parking – Elaboración Propia.

El usuario puede visualizar sus vehículos en una lista. En el caso que no tenga ningún vehículo, podrá añadirlos en el botón de “+” en la parte inferior derecha. Si el considera necesario podrá borrar sus vehículos dando al botón de borrar.

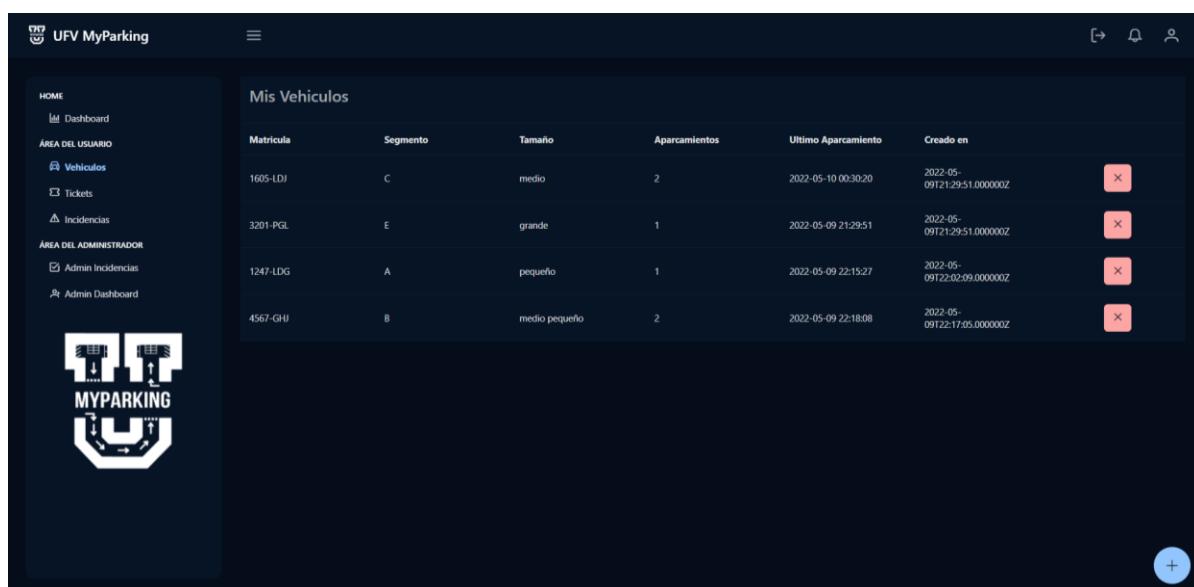


Figura 5.25: Hub Mis Vehículos – Elaboración Propia

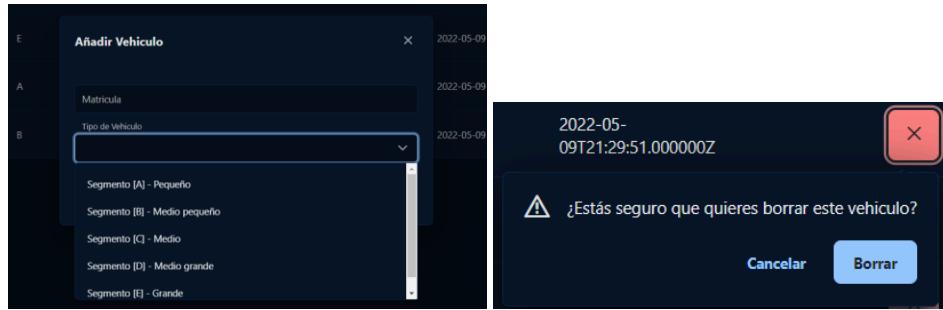


Figura 5.26 - Interacción botón añadir e interacción botón borrar – Elaboración Propia

5.5.3.3. Área Mis Tickets

Para poder indicar a los usuarios en que plazas deben aparcar, se desarrolló una lista con los *tickets* pertenecientes al usuario. El usuario puede visualizar la plaza y tiene la opción de visualizar su *ticket*.

Mis Tickets				
Número de Plaza	Matrícula	Entrada Parking	Salida Parking	
1E	3201-PGL	2022-05-09T22:29:51.000000Z		<button>Ver Ticket</button>
2H	1247-LDG	2022-05-09T22:15:27.000000Z		<button>Ver Ticket</button>
9A	4567-GHI	2022-05-09T22:18:08.000000Z	2022-05-09 22:18:35	<button>Ver Ticket</button>
10F	1605-LDJ	2022-05-10T00:30:20.000000Z		<button>Ver Ticket</button>

Figura 5.27: Tabla de Tickets del Usuario – Elaboración Propia

Una vez abierto el *ticket* se podrá comprobar si existe, ya que cada ticket posee un token único, ligado la plaza, el vehículo y el usuario. Escaneando el QRCode podemos visualizar el *ticket* que se valida al entrar en la vista. El usuario podrá utilizarlo en el móvil, ya que el WebApp es responsive.



Figura 5.28: Ejemplo de Ticket Válido y No Válido – Elaboración Propia

El usuario puede abrir una incidencia, dando al botón abajo del *ticket*, así podrá subir una foto de la incidencia relacionada con su ticket.

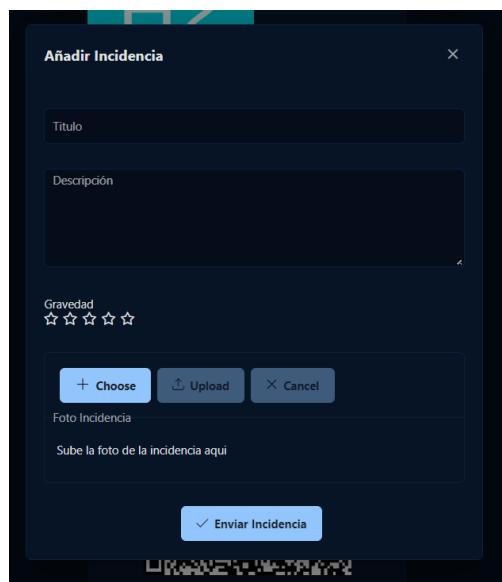


Figura 5.29 – Formulario Incidencia Ticket – Elaboración Propia

El usuario puede visualizar su perfil, donde encontrará sus datos principales. El usuario puede cambiar su contraseña introduciéndola dos veces. Cada rol tiene sus permisos asignados en la base de datos, el usuario puede visualizar sus permisos en su perfil, donde se indica si puede acceder al Digital Twin o no.

The image displays two side-by-side screenshots of a user profile editing interface. Both screenshots feature a large circular profile picture with 'MB' initials. The left screenshot shows the following fields:

- Username: matbmoser
- Nombre: Mathias Brunkow Moser
- Email: mathiasmoser@outlook.com
- Rol: Administrador

Below these are 'Permisos:' (Permissions) settings for three roles, each with a checked checkbox:

- Digital Twin
- Portal de Incidencias
- Portal de Administración

A blue 'Editar Perfil' (Edit Profile) button is at the bottom. The right screenshot shows the same profile information but includes additional fields for password management:

- Contraseña: (password field)
- Confirmar Contraseña: (confirm password field)

At the bottom are 'Back' and 'Actualizar Perfil' (Update Profile) buttons.

Figura 5.30: Perfil Usuario – Elaboración Propia

El administrador posee dos pestañas extras en su menú, donde tiene el área del administrador. Ahí si tiene permiso de acceso, podrá visualizar las incidencias y resolverlas, además de editar el rol del usuario o borrarlos.

Nombre	Apellido	Username	Email	Rol	Creado en
Mathias	Brunkow	matbmoser	mathiasmoser@outlook.com	Administrador	2022-05-09T21:29:51.000000Z
Conductor	MyParking	conductor	conductor@email.com	Usuario	2022-05-09T21:29:51.000000Z
Admin	MyParking	admin	admin@myparking.com	Administrador	2022-05-09T21:29:51.000000Z
Manager	MyParking	manager	manager@myparking.com	Manager	2022-05-09T21:29:51.000000Z
Paco	Gil	pacogil	paco@gmail.com	Usuario	2022-05-09T22:22:49.000000Z

Figura 5.31: Área de Administrador de Usuarios – Elaboración Propia

El administrador tiene el permiso de modificar los roles del usuario, como definido en los requisitos de usuario [Ver Anexo A]:

- El gestor o *manager* podrá acceder al Digital Twin, pero no podrá acceder al área de administración de usuarios, solo al área de resolución de incidencias.
- El administrador puede acceder a cualquier parte de la aplicación.
- El usuario no tiene acceso al Digital Twin, por lo tanto, un administrador deberá darle de alta cuando este sea un gestor.



Figura 5.32: Actualización de Rol por Administrador – Elaboración Propia

5.5.4. Seguridad

Para que los sistemas fueran seguros, se aseguró que cada operación realizada, tanto en el Digital Twin como en el WebApp UFV MyParking, se pudiera realizar solo por el usuario que tiene permisos para realizarla u acceder a la información solicitada. En el caso de envío de datos solo se puede enviar los datos que debe de enviar, por quien debe de enviarlos. En el caso que el usuario no pueda acceder a la información, o no pueda realizar una operación, se enseñará un mensaje de estado para indicar que la operación ha fallado, o se ha ejecutado de manera correcta.

5.5.4.1. Seguridad en WebApp

En el *WebApp* utilizamos tokens JWT (JSON Web Tokens) [89] para autenticar las sesiones del usuario y comprobar en el “middleware” [92] si el usuario es administrador o solicita información de sí mismo siempre que este autenticado y pueda acceder a los datos.

```
Route::prefix('auth')->group(function () {
    Route::post('register', 'App\Http\Controllers\AuthController@register');
    Route::post('login', 'App\Http\Controllers\AuthController@login');
    Route::get('refresh', 'App\Http\Controllers\AuthController@refresh');
    Route::get('check', 'App\Http\Controllers\AuthController@check');
    Route::group(['middleware' => 'auth:api'], function(){
        Route::get('user', 'App\Http\Controllers\AuthController@user');
        Route::post('logout', 'App\Http\Controllers\AuthController@logout');
    });
});

Route::group(['middleware' => 'auth:api'], function(){
    // Users
    Route::get('users', 'App\Http\Controllers\UserController@index')->middleware('isAdmin');
    Route::delete('user', 'App\Http\Controllers\UserController@delete')->middleware('isAdmin');
    Route::get('user/{id}', 'App\Http\Controllers\UserController@show')->middleware('isAdminOrSelf');
    Route::patch('user/{id}', 'App\Http\Controllers\UserController@updateData')->middleware('isAdminOrSelf');

    // Roles
    Route::get('roles', 'App\Http\Controllers\RoleController@index')->middleware('isAdmin');
    Route::patch('role/user', 'App\Http\Controllers\UserController@updateRole')->middleware('isAdmin');
    Route::get('role/{id}', 'App\Http\Controllers\RoleController@show')->middleware('isAdminOrSelfRole');

    // Zonas
    Route::get('zonas', 'App\Http\Controllers\ZonaController@index');
    Route::get('zona/{id}', 'App\Http\Controllers\ZonaController@show');

    // Tipo Vehiculo
    Route::get('vehiculos/tipos', 'App\Http\Controllers\TipoVehiculoController@index');
    Route::get('vehiculo/tipo', 'App\Http\Controllers\TipoVehiculoController@show');

    // Incidencias
    Route::get('incidencias', 'App\Http\Controllers\IncidentController@index')->middleware('isAdmin');
    Route::get('incidencias/user/{id}', 'App\Http\Controllers\IncidentController@getIncidencias')->middleware('isAdminOrSelf');

    // Vehiculos
    Route::get('vehiculos', 'App\Http\Controllers\VehiculoController@index')->middleware('isAdmin');
    Route::get('vehiculo/{id}', 'App\Http\Controllers\VehiculoController@show')->middleware('isAdmin');
    Route::get('vehiculos/user/{id}', 'App\Http\Controllers\VehiculoController@getVehiculosUser')->middleware('isAdminOrSelf');
    Route::put('vehiculo/user/{id}', 'App\Http\Controllers\VehiculoController@store')->middleware('isAdminOrSelf');
    Route::delete('vehiculo/user', 'App\Http\Controllers\VehiculoController@delete');

    // Plazas
    Route::get('plazas', 'App\Http\Controllers\PlazaController@index');
    Route::get('plaza/{id}', 'App\Http\Controllers\PlazaController@show');
    Route::get('plazas/validas', 'App\Http\Controllers\PlazaController@validas');
    Route::get('plazas/validas/zonas', 'App\Http\Controllers\PlazaController@validasZona');
    Route::get('tickets/user/{id}', 'App\Http\Controllers\PlazaController@getTickets')->middleware('isAdminOrSelf');
});
Route::get('ticket/{token}', 'App\Http\Controllers\PlazaController@getTicket');
```

Figura 5.33: API's con control de autenticación y middleware en PHP Laravel – Elaboración Propia

Cuando el usuario no puede acceder a un recurso, está cargando el contenido o ha podido realizar una operación, imprimimos mensajes de estado para informarle del resultado.

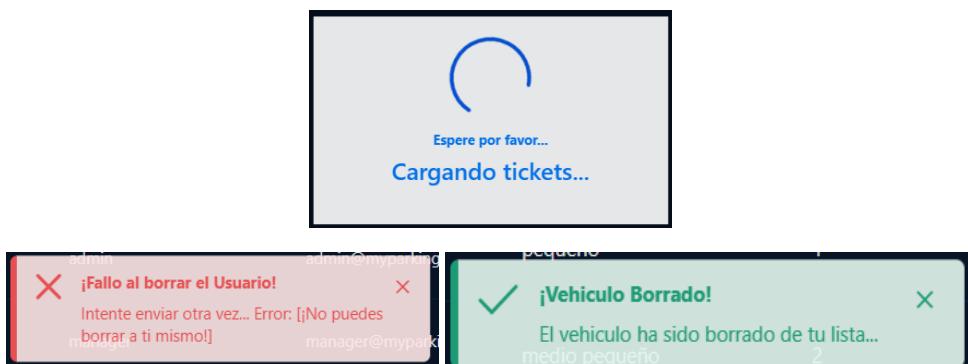


Figura 5.34: Mensajes de Estado: Carga, Error y Éxito en WebApp – Elaboración Propia

Otro punto para destacar son los *tickets*, que se identifican con un token hash 512 único, generado al asignar una plaza al usuario en el servidor, con toda la información del ticket.

5.5.4.2. Seguridad en Digital Twin

Aparte de la utilización del protocolo SJMP diseñado en este proyecto, que encripta las comunicaciones entre las cámaras y el servidor *DeviceManager*, utilizando criptografía asimétrica (RSA) tras un intercambio de claves inicial, también se implementó más sistemas y comprobaciones de seguridad en el Digital Twin.

Para cada usuario en su creación (Registro en *WebApp*), se genera un token único resultado de la siguiente operación utilizando funciones hash, más el correo único y la contraseña:

```
token = sha256(correo + sha512(contraseña))
```

Ecuación 1: Configuración de Generación de Tokens para Digital Twin – Elaboración Propia

El token se almacena en la sesión de PHP y no es visible por el usuario. Adicionalmente se utiliza un UUID (*Universally Unique Identifier*), un token único hash sha512, que autoriza el usuario a realizar peticiones al servidor. Otro punto es que, si se selecciona la opción de “remember me”, en el *login* el correo se encriptará con el identificador único de sesión PHPSESSID utilizando AES-256-CBC [93] y se almacena en la cookie “_chgn”, para luego desencriptar e imprimir el correo al volver con el *logout* tras invalidarse la sesión.

Nombre	Valor	Domair	Path	Expire..
_chgn	U2FsdGVkX19Woex4Sl9lnT6QY4ROO/...	localh...	/	2022...
LOGIN	TRUE	localh...	/	2022...
UUID	e3b0c44298fc1c149afb4c8996fb92427...	localh...	/	2022...
PHPSESSID	de2768651b8f400532400be5a5233d51	localh...	/	Sesión

Figura 5.35: Cookies de Sesión del Digital Twin en Navegador Web – Elaboración Propia

Al Digital Twin solo podrán acceder los usuarios que posean un rol con permisos de acceso al Digital Twin. En el caso que un usuario sin permisos intente acceder, saldrá este error indicando el estado de no Autorizado:

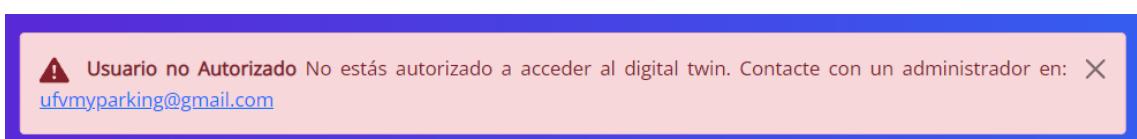


Figura 5.36: Error de Usuario no Autorizado – Elaboración Propia

Otros errores y mensajes de estado se desplegarán, por ejemplo, cuando se expulsa a un usuario de la sesión por fallos de seguridad (cambio de token o invalidez de este). Para cada componente del Digital Twin que interactúa con el servidor y la cámara, existe un monitor dentro de cada modal, que imprime los mensajes de estado de cada operación que se intente realizar. Por ejemplo, para enviar una matrícula debemos de tener una cámara conectada, y si no está saltará un error, informando al usuario que deberá conectarla.

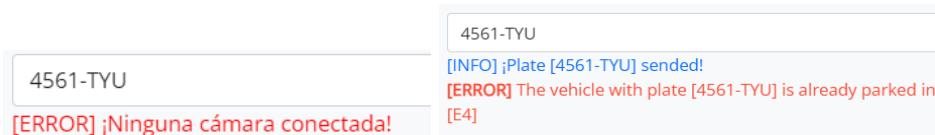


Figura 5.37: Mensajes de Estado del Digital Twin - Elaboración Propia

5.6. INTEGRACIÓN Y PRUEBAS

5.6.1. Integración

- ✓ Prueba de simulación el envío de las matrículas al servidor utilizando el Digital Twin y el Simulador Python de Cámaras IP, para ver que se actualiza en el Digital Twin.
- ✓ Nuevo registro en la aplicación *UFV MyParking* y prueba de *Login* en Digital Twin.
- ✓ Intento de romper el sistema borrando coches que no están en el *parking*.
- ✓ Intento de romper el sistema añadiendo coches que están ya dentro del *parking*.
- ✓ Intento de romper el sistema intentando entrar con usuario sin permiso en el Digital Twin. Además, se intentó acceder a recursos no permitidos sin entrar.
- ✓ Se ha probado las funcionalidades del sistema (borrar matricula, añadir matricula, abrir servidor, cerrar servidor, conectar cámara a servidor, desconectar cámara del servidor, añadir vehículo, borrar vehículo, modificar perfil, invalidar *ticket*, escanear el *QR Code*, Ver *Dashboard* e Información)
- ✓ Pruebas de flujo del usuario con registro, cambio de rol, acceso en el *digital twin*, manejo de vehículos y cámaras.

5.6.2. Verificación

Se realizó un plan de pruebas [Ver Anexo I] para comprobar que los requisitos se cumplen en el producto final. Se comprobó que **113 de 126 requisitos se cumplieron**. Por otro lado, **13 requisitos no se pudieron cumplir**, debido a que sus funcionalidades no estaban incluidas en el Producto Mínimo Viable (MVP) [94], pues su nivel de prioridad en los requisitos establecidos al principio de la planificación es igual o inferior a P3, significando que el tiempo de desarrollo sería inviable debido a la corta duración de este proyecto.

A Requirement No.	B Modulo	C Prioridad	D Test Case Description	E Input	F Prerequisite / Impediments	G Expected Result	H Result	I Status (OK / NOK)
1	Digital Twin	P1	Se debe poder visualizar el mapa del parking, (señalizado por zonas)	-	-	Visualizar Plazas	Visualizar Plazas	OK
2	Digital Twin	P2	Visualizar sobre el mapa el número de plazas en cada zona.	-	-	Se visualizan las plazas	Se visualizan las plazas	OK
3	Digital Twin	P1	Podrá abrir el servidor de Device Manager	Click sobre OpenServer	Login	OpenServer Opens Server	OpenServer Opens Server	OK
4	Digital Twin	P1	Podrá cerrar el servidor de Device Manager	Click sobre CloseServer	OpenServer	CloseServer Close Server	CloseServer Close Server	OK
5	Digital Twin	P1	Habrá un formulario para configurar los datos de conexión de la socket (IP/PUERTO) y el usuario al abrir el digital twin.	-	-	Implementado	Implementado	OK
6	Digital Twin	P1	Visualizar el total de plazas libres	Botón Plazas Libres		Implementado	Implementado	OK
7	Digital Twin	P1	Visualizar el total de plazas ocupadas	Botón Plazas Ocupadas		Implementado	Implementado	OK
8	Digital Twin	P1	Visualizar el total de plazas libres	Botón Plazas Ocupadas y Libres		Implementado	Implementado	OK
9	Digital Twin	P1	Se podrá añadir un vehículo al parking indicando su matrícula.	Añadir Matrícula	Existe en la BBDD	Añade	Añade	OK
10	Digital Twin	P1	Se podrá borrar un vehículo al parking indicando su matrícula.	Añadir Matrícula	Existe en la BBDD del Parking	Borra	Borra	OK
11	Digital Twin	P1	Se podrá ver los vehículos dentro del parking en una lista con campos (Matrícula, Plaza, Fecha Entrada, Tipo).	Dar a Botón Ver Plazas	Existe en la BBDD del Parking	Visualizar Plazas	Visualizar Plazas	OK
12	Digital Twin	P3	Al añadir un vehículo se podrá generar la matrícula de manera aleatoria.			NO IMPLEMENTADO		NOK
13	Digital Twin	P1	Se actualizará la información a cada 10 segundos.	AutoRefresh		Implementado	Implementado	OK
14	Digital Twin	P1	Se indicará la hora actual en el panel.	Reloj		Implementado	Implementado	OK

Figura 5.38: Plan de Pruebas Detallado – Elaboración Propia

6. RESULTADOS

6.1. RESULTADOS GENERALES

- ✓ **Optimizar Tiempo de Aparcamiento:** El sistema desarrollado, como podemos comprobar en el capítulo anterior, tiene todo para poder optimizar el tiempo de aparcamiento de un conductor, debido a que al entrar en el *parking* y estando registrado en la aplicación, podrá recibir su ticket generado para poder aparcar en la plaza asignada. Otra ventaja es que puede visualizar el *parking*, por lo tanto, podrá conocer donde existen plazas libres para aparcar reduciendo su tiempo de búsqueda de manera significativa.
- ✓ **Presentar Estado Actual del Aparcamiento:** Presentamos el estado actual del aparcamiento tanto en el Digital Twin como en la aplicación web para los conductores. Eso permite a todos los usuarios conocer su aforo actual y visualizar las plazas que pueden aparcar. Cuando llegan al aparcamiento y tras la lectura de la matrícula de sus vehículos registrados, el servidor les asigna la mejor plaza disponible para el tipo de vehículo.

6.2. RESULTADOS ESPECÍFICOS

Como podemos comprobar en el capítulo anterior, el desarrollo de la solución técnica es un éxito, ya que se cumplieron todos los objetivos establecidos al principio del proyecto:

- ✓ **Desarrollar un simulador de datos de Cámara IP:** Podemos simular el envío de matrículas utilizando el simulador de Cámaras IP desarrollado en Python.
- ✓ **Desarrollar un servidor de aplicaciones:** El servidor de aplicaciones en Python, puede conectarse con cualquier dispositivo (Cámara) si se envía el paquete correcto.
- ✓ **Diseñar y Desarrollar un protocolo de comunicación segura sobre TCP utilizando JSON:** Se diseñó y desarrolló el protocolo SJMP que permite la encriptación entre la cámara y el servidor de aplicaciones (*DeviceManager*) utilizando criptografía asimétrica (algoritmo RSA). El protocolo puede ir sobre TCP o cualquier red y encripta a nivel de dato cada paquete.
- ✓ **Desarrollar un Digital Twin de una zona del Parking UFV:** Se desarrolló un *Digital Twin*, que permite visualizar el parking UFV, realizar simulaciones de entrada y salida de vehículos, controlar barreras, abrir servidores y simular cámaras.
- ✓ **Desarrollar una Web App para conductores:** Se desarrolló el *WebApp UFV MyParking* utilizando Laravel y Vue.js, y permite el registro de los conductores, así como la visualización de sus *tickets* con las plazas de aparcamiento asignadas, siempre que sus vehículos estén registrados en la aplicación.

7. IMPLICACIONES ÉTICAS E IMPACTO SOCIAL

7.1. INTRODUCCIÓN

Como ingenieros a menudo estamos confrontados con la moralidad de nuestras soluciones [95]. Este proyecto final de grado, como cualquier otro proyecto, posee una cierta carga moral que no debe ser ignorada. El Sistema Optimizador de Tiempo de Aparcamiento (SOTA) consiste en conectar al usuario, persona dueña de un vehículo, con un gestor, responsable de un aparcamiento privado. El sistema entrega al gestor un Digital Twin que ofrece en tiempo real una visualización del *parking*. Este le permite realizar simulaciones de entrada y salida de coches en la Universidad. Estas simulaciones le permiten tomar decisiones a futuro, para buscar el bien de los usuarios en el aparcamiento, minimizando accidentes y otras incidencias que se pueden producir en el campus.

El usuario, por otro lado, conoce el estado del *parking* en una WebApp llamada “*MyParking*”, lo que le permite conocer antes de salir de casa, si existen plazas libres en este momento. Conociendo esto puede ir en transporte público a la Universidad, en lugar de ir en su vehículo privado, minimizando la probabilidad de accidentes que se podrían producir y minimizando la producción de gases contaminantes en la atmósfera.

7.2. VALOR DEL PROYECTO

El proyecto aporta a la sociedad una manera eficiente de encontrar plaza de aparcamiento, reduciendo el estrés producido al buscarla, minimizando los diferentes impactos producidos por los vehículos y sus conductores. Como ingenieros, nuestra misión es mejorar la sociedad mediante el uso de la técnica. A lo largo de este proyecto se realizaron estudios en el *parking* de la Universidad Francisco de Vitoria [2] con el objetivo de aumentar el conocimiento actual del problema existente con el aparcamiento, donde se producen atascos y accidentes semanales. El Sistema Optimizador de Tiempo de Aparcamiento (SOTA) facilita mediante elementos software y hardware tanto la gestión, como la utilización del aparcamiento en una Universidad, aportando valor a los estudiantes y a la imagen de la Universidad Francisco de Vitoria.

7.3. ALCANCE

Este proyecto trata de ofrecer una visión general a los usuarios del estado en tiempo real del aparcamiento del campus de la Universidad Francisco de Vitoria, como parte del proyecto de *Smart Campus* que está en investigación actualmente [21].

Los resultados que se esperan buscan el bien con la utilización del sistema, donde se aspira a conseguir:

- Reducir los atascos producidos en el aparcamiento.
- Reducir el tiempo al volante para encontrar plaza.
- Minimizar los accidentes e incidencias que ocurren en el aparcamiento.
- Mejorar la calidad del aprendizaje y el estado de ánimo.
- Facilitar la integración de un *Smart Parking* en el *Smart Campus*.

7.4. RESPONSABILIDAD E IMPACTO

7.4.1. Riesgos

La prevención de riesgos en un proyecto es una de las piezas clave para su éxito, tanto si este proyecto es informático [96] o no [97]. Es esencial que se realice un buen análisis de riesgos, con el objetivo de minimizarlos según su probabilidad de ocurrencia e impacto. Pudiendo ser este en la sociedad, en el medioambiente o de forma económica. Por ello, con la intención de identificar y prevenir los diferentes riesgos en este proyecto final de grado, se realizó un análisis de riesgos, donde se calcula la probabilidad y el impacto de cada riesgo. Además, se asigna una acción preventiva para minimizar el impacto si el riesgo llega a ocurrir. Los riesgos están identificados en un registro y situados en una matriz para una mejor visualización. [Ver Anexo H para más detalle]

7.4.1.1. Registro de Riesgos

Se identificaron 18 riesgos [Ver Anexo H para más detalles] donde los más relevantes son:

Tabla 18: Registro de Riesgos – Elaboración Propia

Riesgo	Descripción	Prob.	Acción	Imp.
<i>R2</i>	Aparcamiento ilegal o en plaza equivocada.	Alta	Permitir el reporte de incidencias e infracciones.	Medio
<i>R5</i>	Ataques tipo “ <i>Man in the Middle</i> ”, de visualización de comunicaciones en cualquier red utilizada (IP, LoRa (<i>LongRange</i>), etc).	Alta	Crear un protocolo seguro con encriptación a nivel de dato, utilizando encriptación asimétrica RSA.	Alto
<i>R7</i>	Atascos en la entrada del parking.	Alta	Posicionamiento de la entrada al lado de un aparcamiento.	Alto
<i>R12</i>	Incidencias, como atropellamiento o choques con otros coches.	Media	Añadido al reporte de incidencias una gravedad de la situación donde un administrador puede borrar al usuario y contactar con las autoridades.	Alto

7.4.2. Leyes

Si se llega a implementar el sistema de manera real, siendo este no el caso actual donde simulamos la entrada de los vehículos, una de las leyes que debemos seguir es la nueva ley de tráfico de la Dirección General de Tráfico (DGT) [98]. Esta nueva ley del 2022 prohíbe la utilización de teléfonos móviles durante la circulación, lo que supone un riesgo para nuestro proyecto si consideramos que las personas mirarán a sus móviles para ver la plaza asignada. Para seguir esta ley añadimos barreras para que el coche esté fuera de circulación, en estado de semi aparcamiento y añadimos la posibilidad de colocar una pantalla TFT en la entrada, con el objetivo de que el usuario vea su plaza asignada sin abrir el móvil.

7.4.3. Impacto Social

El impacto social producido por este proyecto consiste en la disminución del tiempo de aparcamiento, además de reducir el estrés de las personas. Según este estudio [99] sobre los efectos de un estado de ánimo positivo y negativo en el aprendizaje y rendimiento académico, el estudiante posee peor rendimiento académico y menor aprendizaje si su estado de ánimo es negativo. Por otro lado, el tráfico y los atascos según este estudio [100], afectan negativamente el estado de salud mental, lo que produce un efecto en el estado de ánimo. Lo que nos permite concluir que el estrés generado al no encontrar plaza de aparcamiento afecta a la calidad de aprendizaje del alumno y dificulta la labor docente.

A raíz de las decisiones tomadas en este proyecto, pueden producirse impactos negativos en la sociedad, ya que reducimos el alcance a solo coches y furgonetas con el objetivo de evitar incidencias. Esto puede provocar una insatisfacción por parte de los conductores de motocicletas o ciclo motores, ya que no se les permite la entrada. Como trabajo a futuro se debería incluir a los conductores de motocicletas, añadiendo otro tipo de detección de matrículas o incluyendo nuevas plazas de aparcamiento. Para que no se genere el sentimiento de exclusión social por parte de la comunidad de motociclistas, buscando la paz y armonía entre los conductores de la Universidad Francisco de Vitoria.

7.4.4. Impacto Económico

El precio de la gasolina y de los demás carburantes está muy elevado actualmente [101]. Según un estudio de la asociación ANFAC [11], la compra de coches eléctricos en España crece a un ritmo mucho más lento comparado con los demás países europeos. Esto significa que la cantidad de coches eléctricos utilizados ahora en España todavía es pequeña, por lo tanto, la reducción del tiempo al volante supondría una reducción de costes en carburantes y el uso de vehículos contaminantes. Otro punto donde impacta el proyecto es en la gestión del aparcamiento. Según las entrevistas realizadas en este PFG [13] [14], un gestor de un aparcamiento privado está constantemente intentando mejorar la experiencia de los usuarios. El sistema ahorraría costes a estos gestores, ya que permitiría la monitorización del aparcamiento sin ayudantes de tráfico. Permite también conocer las incidencias producidas en el mismo, donde conociendo el funcionamiento del *parking*, el gestor puede tomar decisiones con el objetivo de mejorarlo.

7.4.5. Impacto Medioambiental

El campus de la Universidad Francisco de Vitoria está sufriendo una actualización hacia un Smart Campus, buscando reducir la huella medio ambiental producida por los vehículos de combustión [9] en sus aparcamientos. Los Smart Campus están dentro de los objetivos de la ONU para 2030 [10], donde se promueve el desarrollo sostenible. El sistema desarrollado en este PFG tiene como objetivo reducir el tiempo al volante del conductor al buscar plaza. Esto minimizaría a su vez la emisión de gases contaminantes en la atmósfera, mejorando a su vez la calidad del aire en el campus.

7.5. CONCLUSIONES

Este proyecto busca el bien en todos sus aspectos, tanto a nivel conceptual como a nivel técnico, haciendo el bien y evitando el mal, como dice el precepto primero y formal de la ley natural de Santo Tomás, [102] - “*Fac bonum et vita malum*” (Haz el bien y evita el mal). El proyecto busca reducir de manera significativa muchos aspectos negativos producidos en el *parking* de la Universidad Francisco de Vitoria actualmente, además de proponer soluciones moralmente aceptables para minimizar el impacto negativo y maximizar el impacto positivo generado por el sistema en la sociedad.

Por todo lo expuesto, debido al valor presentado por el proyecto y que los riesgos identificados son asumibles y están controlados, se concluye que es viable desde el punto de vista ético y es muy recomendable llevar a cabo iniciativas como estas.

8. CONCLUSIONES

8.1. CONCLUSIONES GENERALES

- Existe **gran un interés** por parte de la comunidad Universitaria UVF, en el desarrollo de un sistema/aplicación para solucionar el problema del aparcamiento.
 - Según las encuestas y entrevistas realizadas dentro del marco de este PFG [2] [13] [14], se concluye que existe una necesidad para el desarrollo e implementación de la solución realizada en este proyecto final de grado.
 - Se concluye que un sistema como el desarrollado en este PFG sería muy útil, ahorraría mucho tiempo, mejorando la salud mental de los conductores que utilizan el aparcamiento del campus de la Universidad Francisco de Vitoria.
- La utilización del sistema propuesto en este PFG evitaría muchos accidentes en el campus de la Universidad Francisco de Vitoria, debido a que se asignan las plazas según el tipo y tamaño de vehículo registrado.
- Si el sistema desarrollado en este PFG se implementa, permitirá a los gestores identificar a todos los conductores y vehículos del aparcamiento. Restringiendo la utilización a solo los usuarios registrados en la aplicación.
- El Sistema Optimizador de Tiempo de Aparcamiento (SOTA), es un **hito importante** en el marco del desarrollo e investigación para la transformación del Campus UVF en un verdadero Smart Campus.
 - El sistema contiene un Digital Twin (sistema implementado en *Smart Cities*) que permite visualizar un aparcamiento en tiempo real y realizar simulaciones, pudiendo ser escalable a todo el aparcamiento en un futuro.
 - El sistema permite además asignar plazas a los conductores que entran al *parking*, simulando la misma tecnología y funcionamiento de las cámaras que ya se encuentran instaladas en las entradas y salidas del Campus de la Universidad Francisco de Vitoria.
 - Por lo tanto, el Sistema Optimizador de Tiempo de Aparcamiento (SOTA) es un sistema innovador y avanzado en su área, posicionando a la Universidad Francisco de Vitoria a la altura de las demás Universidades Españolas citadas en este proyecto [4] [5] [6] [7] [8] que tienen programas de Smart Campus.
- En un proyecto de Ingeniería Informática debemos de tener en cuenta factores importantes como la seguridad, la usabilidad y la escalabilidad de un sistema. Si el alcance está bien definido; si se realiza una buena planificación; si se establecen los requisitos adecuados y se diseña el sistema, se facilitará el desarrollo del software y las pruebas a realizar. Si se siguen estos factores no siempre se garantiza el éxito del proyecto, sin embargo, siguiéndolos nos acercaremos al éxito significativamente.

8.2. CONCLUSIONES TÉCNICAS

- La curva de aprendizaje de la estructura de Laravel es relativamente lenta ya que el *framework* tiene el patrón de diseño Modelo-Vista-Controlador [103] y requiere la creación de migraciones para la base de datos, implicando el desarrollo de los modelos de datos en el *backend* previamente. Por lo tanto, no se puede seguir una planificación, enfocada al desarrollo alternado de *API's* primero y luego Vistas. En resumen, se concluye que Laravel es muy potente para aplicaciones que necesitan un alto nivel de escalabilidad y robustez, debido a que obliga a seguir un desarrollo seguro, restringiendo el acceso a los datos por parte del *frontend* (Vue.js) desde el principio del desarrollo. Esto supone también un esfuerzo extra a la hora de desarrollar la aplicación, pudiendo complicar las cosas para el desarrollador, si este no está familiarizado con el entorno del *framework*.
- La diferencia entre el desarrollo utilizando JS nativo y Vue.js en el *frontend* es notable. Vue.js cuenta con un ciclo de vida bastante complejo [104], por lo tanto, a lo largo del desarrollo, si el desarrollador no está familiarizado con él, surgen problemas para guardar la información y acceder a los datos locales, así como realizar la comunicación con las *API's* en Laravel.
- Los contenedores *Docker* y la herramienta de *Docker Compose*, son muy útiles a la hora de encapsular un sistema para su despliegue futuro en cualquier máquina o servidor, manteniendo la estructura establecida en el desarrollo y haciendo que el sistema sea portable.
- El protocolo de comunicación SJMP diseñado e implementado en este proyecto, ofrece un modelo de encriptación de comunicaciones a nivel de dato seguro. Utilizando criptografía asimétrica permite mantener la confidencialidad e integridad de los datos enviados desde las cámaras al servidor. Además, independiente de la red utilizada (TCP, LoraWan, etc), permite la configuración personalizada para cada lógica de negocio en cada sistema, haciéndole personalizable y escalable.

8.3. TRABAJO FUTURO

- Implementar el sistema con cámaras IP reales:
 - Debido a que el sistema no se ha implementado todavía, por falta de tiempo, recursos y autorizaciones, el siguiente paso a futuro sería implementar el sistema en el aparcamiento con cámaras IP reales conectadas al servidor.
- Analizar los datos utilizando Inteligencia Artificial y Técnicas de *Data Mining*:
 - Debido a que el sistema recogerá datos reales, se puede utilizar IA y técnicas de *data mining* con el fin de hallar patrones y predecir el comportamiento del *parking*. A partir de un modelo permitiremos al usuario indicar un horario y fecha para conocer el estado precedido del *parking* en el futuro.

- Integrar con la nueva aplicación de la Universidad Francisco de Vitoria para móvil:
 - La Universidad Francisco de Vitoria posee una aplicación móvil y tiene la intención de actualizarla, por lo tanto, una línea de trabajo futuro seria, añadir el *WebApp UFV MyParking* a la aplicación, para permitir a los usuarios del aparcamiento UFV encontrar plaza de manera óptima.
- Control de aforo para miembros de la UFV añadiendo el *Single Sign-On* [105] de Microsoft [106] utilizado para distribuir licencias de Office 365 [107] en la UFV.
 - Si se implementa el *Single Sign-On* cada estudiante utilizando su correo educativo con dominio “@ufv.es”, tendrían acceso a la información de ciertas zonas del aparcamiento, donde se permite la entrada a solo los miembros de la Universidad registrados.
 - Esto permitirá relacionar a cada conductor con su vehículo, restringiendo por un tiempo el acceso a conductores que provoquen incidentes graves como accidentes o aparcamientos ilegales.
- Integración del *WebApp MyParking* con la aplicación *SharingCarUFV* [17] y creación de *parking* privado para usuarios conductores en la aplicación:
 - Como comprobamos en el estudio del aparcamiento realizado en el marco de este PFG [2], el número de usuarios de la aplicación de *SharingCarUFV* es muy bajo.
 - Como comprobamos en las entrevistas con el gestor del aparcamiento [13] [14], existe un interés en promover y atraer usuarios a utilizar la aplicación de coches compartidos, con el objetivo de reducir el número de coches en el *parking*.
 - Por lo tanto, si se reserva un aparcamiento solo para los conductores de coches compartidos y se añaden sus matrículas al sistema, promoveremos el uso de la aplicación de manera significativa.
- Conexión entre coche y aparcamiento:
 - Para evitar el uso de móviles al conducir una solución sería crear una aplicación que pueda acceder al sistema desde Android Auto [108]. Según este estudio [109] realizado por el Instituto de Telemáticas de la Universidad de Karlsruhe, en Alemania, es posible utilizar tecnologías Wireless para transmitir las comunicaciones desde un vehículo a un servidor o dispositivo.
 - Según el estudio citado en la investigación previa en este proyecto [28] un 96% de los nuevos coches producidos podrán conectarse a Internet en 2030.
 - Por lo tanto, siguiendo el concepto de Internet de las Cosas (IoT), si conectamos el aparcamiento con el vehículo del usuario, este podría recibir y visualizar su *ticket* en la pantalla táctil interna e incluso si el coche es autónomo, se podría comunicar una plaza al vehículo y que el aparcara solo donde el aparcamiento le indique.

9. OTROS MÉRITOS DEL PROYECTO

En este proyecto se realizaron diversas tomas de datos de usuarios reales en el Campus UFV, como encuestas, entrevistas, asistencias a seminarios, estudios y mediciones de los aparcamientos de la UFV. Contiene diferentes componentes específicos dentro de las soluciones generales. Posee una variedad de tecnologías innovadoras utilizadas actualmente en el mercado, como Laravel, Vue.js, Python y Docker. La integración de los sistemas desarrollados es un mérito destacable. A lo largo del proyecto surgieron obstáculos en el desarrollo, debido al alto grado de complejidad. Estos obstáculos se superaron con éxito y se integraron los sistemas de manera estable y segura, pensando siempre en la seguridad desde el principio del proyecto. Además, las soluciones desarrolladas relacionan diversas asignaturas del Grado en Ingeniería Informática:

Tabla 19: Relación Asignatura-Solución – Elaboración Propia – Con Acrónimos de Asignaturas

Soluciones Asignatura \	Digital Twin	WebApp	SJMP	Device Manager	Contenido, Solución y Memoria	Gestión y Desarrollo
Redes	X	X	X	X		
IS I y II	X	X	X		X	X
Desarrollo Web	X	X				
BBDD Avanzadas	X	X	X	X		
IPO	X	X				
COCO			X	X		
POO	X	X	X	X		
DIS		X				
Proyectos I y II	X	X			X	X
PGPI					X	X
Prácticas en Empresa			X	X	X	X
Seguridad	X	X	X			
Criptografía	X		X	X	X	
Seguridad en IoT	X		X	X		
Seguridad en Web y Móviles	X	X				
Hacking ético y Pentesting I y II	X	X	X			
IA I y II		X				
MPI	X		X			X
Ética	X	X	X		X	
<u>ASOP I, II, y III</u>	X		X	X		
Seminario de IoT			X	X	X	
Seminario de Smart Cities	X				X	

10. BIBLIOGRAFÍA

- [1] T. 3.-2. A.C., «Frases Celebres sobre el Tiempo,» [En línea]. Available: <http://frasescelebres.org/tiempo/>. [Último acceso: 01 03 2022].
- [2] M. B. Moser, «Encuesta sobre el Parking UFV,» Elaboración propia en Google Forms, 58 Respuestas, 04 Marzo 2022. [En línea]. Available: https://docs.google.com/forms/d/e/1FAIpQLSdu4UxVdgFJfa-LOosW8AQKBn02Fzv6maH3X7QSiS-p74EyfA/viewform?usp=sf_link. [Último acceso: 05 03 2022].
- [3] N. Min-Allah y S. Alrashed, «Smart campus—A sketch,» 2020. [En línea]. Available: <https://doi.org/10.1016/j.scs.2020.102231>. [Último acceso: 05 03 2022].
- [4] Universidad de Alicante, «UA Smart University,» [En línea]. Available: <https://web.ua.es/es/smart/el-proyecto.html>. [Último acceso: 10 03 2022].
- [5] «SmartUMA, Smart Campus Universidad Málaga,» [En línea]. Available: <https://www.uma.es/smart-campus>. [Último acceso: 10 03 2022].
- [6] «Smart Campus UCM,» [En línea]. Available: <https://www.ucm.es/smartcampus/>. [Último acceso: 10 03 2022].
- [7] «Smart Campus, La Salle,» [En línea]. Available: <https://www.salleurl.edu/es/la-salle/medidas-e-iniciativas-covid-19/la-salle-url-smart-campus>. [Último acceso: 10 03 2022].
- [8] «Rey Juan Carlos III, Smart Campus Fuenlabrada,» [En línea]. Available: <https://www.urjc.es/todas-las-noticias-de-actualidad/5849-se-da-el-primer-paso-en-el-proyecto-smart-campus-fuenlabrada>. [Último acceso: 10 03 2022].

- [9] «Universidad Francisco de Vitoria - Postes de Recarga Eléctrica,» 13 10 2021. [En línea]. Available: <https://www.ufv.es/la-ufv-abre-los-postes-de-recarga-de-coches-de-manera-gratuita-a-la-comunidad-universitaria/>. [Último acceso: 05 03 2022].
- [10] ONU, «17 Objetivos Agenda 2030 para el Desarrollo Sostenible,» 25 09 2015. [En línea]. Available: <https://www.un.org/sustainabledevelopment/es/2015/09/la-asamblea-general-adopta-la-agenda-2030-para-el-desarrollo-sostenible/>. [Último acceso: 10 03 2022].
- [11] «Las ventas de turismos electrificados, híbridos y de gas suponen el 45% del mercado en enero,» ANFAC, 1 Febrero 2022. [En línea]. Available: <https://anfac.com/actualidad/notas-de-matriculacion/las-ventas-de-turismos-electrificados-hibridos-y-de-gas-suponen-el-45-del-mercado-en-enero/>. [Último acceso: 18 04 2022].
- [12] «Clovity, Smart Campus Proposition,» [En línea]. Available: <https://www.clovity.com/smart-campus-proposition>. [Último acceso: 05 03 2022].
- [13] J. C. M. Jefe del Departamento de Servicios Generales UFV, Interviewee, *Primera Reunión con Gestor del Aparcamiento de la Universidad Francisco de Vitoria*. [Entrevista]. 30 Septiembre 2021.
- [14] J. C. M. Jefe del Departamento de Servicios Generales UFV, Interviewee, *Segunda Reunión con Gestor del Aparcamiento de la Universidad Francisco de Vitoria*. [Entrevista]. 1 Marzo 2022.
- [15] «Grupo Lasser,» [En línea]. Available: <https://grupolasser.com/>. [Último acceso: 9 03 2022].
- [16] «Portal Gerencia a tu lado, Universidad Francisco de Vitoria,» [En línea]. Available: <https://www.ufv.es/gerencia-a-tu-lado/direccion-de-espacios-infraestructuras/en-nuestro-campus/>. [Último acceso: 12 11 2021].
- [17] U. F. d. Vitoria, «SharingCar UFV,» [En línea]. Available: <https://sharingcarufv.ufv.es/>. [Último acceso: 20 03 2022].
- [18] Google, «Google Forms,» [En línea]. Available: <https://www.google.es/intl/es/forms/about/>. [Último acceso: 08 03 2022].

- [19] «The Effects of Positive and Negative Mood on University Students Learning and Academic Performance: Evidence from Indonesia,» 2 April 2011. [En línea]. Available: <http://fs.libarts.psu.ac.th/research/conference/proceedings-3/2pdf/007.pdf>. [Último acceso: 12 Nov 2021].
- [20] J. A. Ondiviela, «Postgrado Universidad Francisco de Vitoria,» [En línea]. Available: <https://postgrado.ufv.es/curso-transformando-la-ciudad/>. [Último acceso: 30 03 2022].
- [21] J. A. Ondiviela Garcia, *Seminario de Smart Cities*, Pozuelo de Alarcón, 2021-2022.
- [22] J. A. Ondiviela Garcia, «WorldWide Observatory for Attractive Cities,» [En línea]. Available: <https://attractivecities.com/>. [Último acceso: 30 03 2022].
- [23] J. A. Ondiviela García, «Beyond SmartCities: Creating the most Attractive Cities for Talented Citizens. The necessary imagination: possible worlds theory foundation and applications.,» 2020. [En línea]. Available: <http://hdl.handle.net/10641/2725>. [Último acceso: 30 03 2022].
- [24] J. A. Ondiviela Garcia, «WORLDWIDE OBSERVATORY FOR ATTRACTIVE CITIES - ANNUAL WHITEPAPER for SmartCity Expo,» 2020. [En línea]. Available: <http://hdl.handle.net/10641/2292>. [Último acceso: 30 03 2022].
- [25] J. A. Ondiviela Garcia, «WORLDWIDE OBSERVATORY FOR ATTRACTIVE CITIES - ANNUAL WHITEPAPER for SmartCity Expo,» 2021. [En línea]. Available: <http://hdl.handle.net/10641/2619>. [Último acceso: 30 03 2022].
- [26] IBM, «How a Digital Twin Work?,» [En línea]. Available: <https://www.ibm.com/topics/what-is-a-digital-twin>. [Último acceso: 18 04 2022].
- [27] A. D. Little, «Future of Urban Mobility 2.0,» [En línea]. Available: https://www.adlittle.com/sites/default/files/viewpoints/2014_ADL UITP_Future_of_Urban_Mobility_2_0_Full_study.pdf. [Último acceso: 30 03 2022].
- [28] M. Placek, «Statista,» 8 Diciembre 2021. [En línea]. Available: <https://www.statista.com/topics/1918/connected-cars/#dossierKeyfigures>. [Último acceso: 18 04 2022].

- [29] «Solución de guiado de parking para zonas comerciales,» urbiotica, [En línea]. Available: <https://www.urbiotica.com/soluciones-inteligentes-3-es/guiado-parking-zonas-comerciales/>. [Último acceso: 05 03 2022].
- [30] I+D3, «4Park Business,» [En línea]. Available: <https://imasdetres.com/control-accesos-parking-oficinas-empresas/>. [Último acceso: 05 03 2022].
- [31] L. Segarra Fernández, «Gestión inteligente de plazas de aparcamiento mediante procesamiento de imágenes en IoT: Redes neuronales convolucionales y predicción,» TFM Universidad Complutense, 2019. [En línea]. Available: <https://eprints.ucm.es/id/eprint/57092/>. [Último acceso: 05 03 2022].
- [32] Quadrex, «QX-T100 Contador de Vehículos para aparcamientos,» [En línea]. Available: <https://www.quadrex.es/productos/qx-t100-contador-de-vehiculos-para-aparcamientos/>. [Último acceso: 10 03 2022].
- [33] S. Parking, «Smart Parking for Universities,» [En línea]. Available: <https://www.smartparking.com/solutions/universities>. [Último acceso: 10 03 2022].
- [34] O. C. Yee, N. Yaakob y F. A. M Elshaikh, «A cloud-based automated parking system for smart campus,» Article of School of Computer and Communication Engineering, Universiti Malaysia Perlis, 2020. [En línea]. Available: <https://iopscience.iop.org/article/10.1088/1757-899X/767/1/012049/pdf>. [Último acceso: 05 03 2022].
- [35] P. N. Antúnez, «Smart Parking App,» TFM Universidad Politécnica Madrid, Julio 2019. [En línea]. Available: https://oa.upm.es/65622/1/TESIS_MASTER_PABLO_NAVARRO_ANTUNEZ.pdf. [Último acceso: 05 03 2022].
- [36] libelium, Powering the IoT Revolution , «Smart Parking project in Montpellier to relieve traffic congestion and reduce car parking search,» 24 Octubre 2017. [En línea]. Available: <https://www.libelium.com/libeliumworld/success-stories/smart-parking-project-in-montpellier-to-relieve-traffic-congestion-and-reduce-car-parking-search/>. [Último acceso: 05 03 2022].

- [37] I. Aydin, M. Karakose y E. Karakose, «A Navigation and Reservation Based Smart Parking Platform Using Genetic Optimization for Smart Cities,» Article for 5th International Istanbul Smart Grid and Cities Congress and Fair (ICSG), 19 Abril 2017. [En línea]. Available: <http://www.kresttechnology.com/krest-academic-projects/krest-mtech-projects/IOT/Mech%20IOT-2017-18/IOT%20Basepaper%202017-18/37.%20A%20Navigation%20and%20Reservation%20Based.pdf>. [Último acceso: 05 03 2022].
- [38] «KingPNG Logo Azure DevOps,» [En línea]. Available: https://www.kindpng.com/imgv/TbobwTx_azure-devops-server-logo-hd-png-download/. [Último acceso: 18 04 2022].
- [39] I. Sommerville, Manual de Ingeniería del Software, 9th ed ed., Pearson Educación, 2011.
- [40] E. Conrad, «WaterFall Model Science Direct Eleventh Hour CISSP Fragment,» 2011. [En línea]. Available: <https://www.sciencedirect.com/topics/computer-science/waterfall-model>. [Último acceso: 18 04 2022].
- [41] T. P. Group, «PHP Manual,» [En línea]. Available: <https://www.php.net/manual/es/intro-whatis.php>. [Último acceso: 30 03 2022].
- [42] F. Mozilla, «Web Developer Mozilla JavaScript,» [En línea]. Available: <https://developer.mozilla.org/es/docs/Web/JavaScript>. [Último acceso: 30 03 2022].
- [43] «Vue.js Official Page,» [En línea]. Available: <https://vuejs.org/>. [Último acceso: 30 03 2022].
- [44] M. Platforms, «React.js Framework,» [En línea]. Available: <https://es.reactjs.org/>. [Último acceso: 30 03 2022].
- [45] M. Foundation, «Developer Mozilla WebSocket,» [En línea]. Available: https://developer.mozilla.org/es/docs/Web/API/WebSockets_API. [Último acceso: 30 03 2022].

- [46] E. International, «Estándar JavaScript ECMA-262,» Junio 2021. [En línea]. Available: <https://www.ecma-international.org/publications-and-standards/standards/ecma-262/>. [Último acceso: 30 03 2022].
- [47] T. P. S. Foundation, «Docs Python Introduction,» [En línea]. Available: <https://docs.python.org/3/tutorial/index.html>. [Último acceso: 30 03 2022].
- [48] M. Zaforas, «Paradigma Digital - ¿Es Python el lenguaje del futuro?,» [En línea]. Available: <https://www.paradigmadigital.com/dev/es-python-el-lenguaje-del-futuro/>. [Último acceso: 30 03 2022].
- [49] D. .. L. a. S. Prabhakaran, «A STUDY ON PYTHON PROGRAMMING LANGUAGE,» 12 Julio 2020. [En línea]. Available: http://www.drsrjournal.com/no_12_july_20/15.pdf?i=1. [Último acceso: 30 03 2022].
- [50] Kaggle, «Kaggle Datasets, Programming Language Survey,» 2018. [En línea]. Available: <https://www.kaggle.com/datasets/kaggle/kaggle-survey-2018>. [Último acceso: 30 03 2022].
- [51] K. R. Srinath, «Python – The Fastest Growing Programming Language,» Diciembre 2017. [En línea]. Available: <https://www.irjet.net/archives/V4/i12/IRJET-V4I1266.pdf>. [Último acceso: 30 03 2022].
- [52] Google, «Google Analytics,» [En línea]. Available: <https://analytics.google.com/analytics/>. [Último acceso: 30 03 2022].
- [53] P. Carbonnelle, «PYPL PopularitY of Programming Language Data,» Marzo 2022 . [En línea]. Available: <https://pypl.github.io/PYPL.html>. [Último acceso: 30 03 2022].
- [54] Laravel, «Laravel Framework,» [En línea]. Available: <https://laravel.com/>. [Último acceso: 30 03 2022].
- [55] Symfony, «Symfony Framework,» [En línea]. Available: <https://symfony.com/>. [Último acceso: 30 03 2022].
- [56] S. Team, «Sass Lang,» [En línea]. Available: <https://sass-lang.com/>.

- [57] L. -. T. Otwell, «Laravel Mix Docs,» [En línea]. Available: <https://laravel.com/docs/9.x/mix>. [Último acceso: 30 03 2022].
- [58] V. Team, «Vue Guidelines,» [En línea]. Available: <https://vuejs.org/guide/introduction.html>. [Último acceso: 30 03 2022].
- [59] J. B. Nils Adermann, «Composer,» [En línea]. Available: <https://getcomposer.org/>. [Último acceso: 30 03 2022].
- [60] I. i. a. s. o. G. (. s. o. M. npm, «NPM JS,» [En línea]. Available: <https://www.npmjs.com/>. [Último acceso: 30 03 2022].
- [61] O. Foundation, «Node.js Offical Page,» [En línea]. Available: <https://nodejs.org/es/>. [Último acceso: 30 03 2022].
- [62] P. Faces, «Prime Vue Library,» [En línea]. Available: <https://www.primefaces.org/primevue/#/>. [Último acceso: 30 03 2022].
- [63] Oracle, «MySQL Database,» [En línea]. Available: <https://www.mysql.com/>. [Último acceso: 30 03 2022].
- [64] s. IT, «Ranking RDBMS,» [En línea]. Available: <https://db-engines.com/en/ranking>. [Último acceso: 31 03 2022].
- [65] s. IT, «Ranking RDBMS MySQL,» [En línea]. Available: <https://db-engines.com/en/system/MySQL>. [Último acceso: 30 03 2022].
- [66] p. contributors, «phpMyAdmin Tool,» [En línea]. Available: <https://www.phpmyadmin.net/>. [Último acceso: 30 03 2022].
- [67] M. Foundation, «Maria DB,» [En línea]. Available: <https://mariadb.org/>. [Último acceso: 30 03 2022].
- [68] Microsoft, «Azure DevOps,» [En línea]. Available: <https://azure.microsoft.com/es-es/services/devops/#overview>. [Último acceso: 30 03 2022].

- [69] Atlassian, «What is Scrum?» [En línea]. Available: <https://www.atlassian.com/es/agile/scrum>. [Último acceso: 30 03 2022].
- [70] Atlassian, «Jira Software,» [En línea]. Available: <https://www.atlassian.com/es/software/jira>. [Último acceso: 30 02 2022].
- [71] Git. [En línea]. Available: <https://git-scm.com/>. [Último acceso: 30 03 2022].
- [72] Atlassian, «Gitflow Tutorial,» [En línea]. Available: <https://www.atlassian.com/es/git/tutorials/comparing-workflows/gitflow-workflow>. [Último acceso: 30 03 2022].
- [73] D. Inc., «Docker,» [En línea]. Available: <https://www.docker.com/>. [Último acceso: 30 03 2022].
- [74] D. Inc., «Docker Overview,» [En línea]. Available: <https://docs.docker.com/get-started/overview/>. [Último acceso: 30 03 2022].
- [75] D. Inc., «Docker Hub,» [En línea]. Available: <https://hub.docker.com/>. [Último acceso: 30 03 2022].
- [76] D. Inc., «Docker Compose Docs,» [En línea]. Available: <https://docs.docker.com/compose/>. [Último acceso: 30 02 2022].
- [77] C. Evans, «YAML human-friendly data serialization,» [En línea]. Available: <https://yaml.org/>. [Último acceso: 30 03 2022].
- [78] S. -. V. Coisne, «What is Docker Compose?,» 3 Diciembre 2020. [En línea]. Available: <https://strapi.io/blog/what-is-docker-compose-all-you-need-to-know>. [Último acceso: 30 03 2022].
- [79] T. A. S. Foundation., «Apache HTTPD,» [En línea]. Available: <https://httpd.apache.org/>. [Último acceso: 30 03 2022].

- [80] W. -. E. Prud'hommeaux, «XML-based HTTP Server Configuration Language,» 1998. [En línea]. Available: <https://www.w3.org/1999/07/9-http-server-conf.html>. [Último acceso: 30 03 2022].
- [81] M. Foundation, «Introducción a XML,» [En línea]. Available: https://developer.mozilla.org/es/docs/Web/XML/XML_introduction. [Último acceso: 30 03 2022].
- [82] I. Google, «Estándar WebSocket rfc6455,» Diciembre 2011. [En línea]. Available: <https://datatracker.ietf.org/doc/html/rfc6455>. [Último acceso: 30 03 2022].
- [83] D. I. PROGRAM, «TRANSMISSION CONTROL PROTOCOL - rfc793,» Septiembre 1981. [En línea]. Available: <https://datatracker.ietf.org/doc/html/rfc793>. [Último acceso: 30 03 2022].
- [84] D. I. PROGRAM, «INTERNET PROTOCOL - rfc791,» Septiembre 1981. [En línea]. Available: <https://datatracker.ietf.org/doc/html/rfc791>. [Último acceso: 30 03 2022].
- [85] «Hypertext Transfer Protocol -- HTTP/1.1,» Junio 1999. [En línea]. Available: <https://datatracker.ietf.org/doc/html/rfc2616>. [Último acceso: 30 03 2022].
- [86] A. Thomas y M. Bezemer, «Gantt Project,» [En línea]. Available: <https://www.ganttpoint.biz/>. [Último acceso: 18 04 2022].
- [87] R. R. Galán, «Modelo para realización de estimación de costes del apartado 4 del PFG.,» 2022.
- [88] «dbdiagram.io,» [En línea]. Available: <https://dbdiagram.io/>. [Último acceso: 08 04 2022].
- [89] «JSON Web Token,» [En línea]. Available: <https://jwt.io/>. [Último acceso: 27 04 2022].
- [90] Google, «Postman API Platform,» [En línea]. Available: <https://www.postman.com/>. [Último acceso: 27 04 2022].

- [91] N. P. a. D. Mazières, «A Future Adaptable Password Scheme (Bycrypt) - The Open BSD Proyect,» [En línea]. Available: <https://www.openbsd.org/papers/bcrypt-paper.pdf>. [Último acceso: 18 04 2022].
- [92] R. Hat, «What is Middleware,» [En línea]. Available: <https://www.redhat.com/es/topics/middleware/what-is-middleware>. [Último acceso: 13 05 2022].
- [93] I. N. d. E. y. T. (NIST), «ADVANCED ENCRYPTION STANDARD (AES),» 26 Noviembre 2001. [En línea]. Available: <https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.197.pdf>. [Último acceso: 13 05 2022].
- [94] P. Plan, «Producto Minimo Viable / Minimum Viable Product,» [En línea]. Available: <https://www.productplan.com/glossary/minimum-viable-product/>. [Último acceso: 10 05 2022].
- [95] «El mayor estudio sobre la ética de los coches autónomos trae malas noticias para los ancianos (e incertidumbre sobre el futuro),» Xataka, 23 Abril 2018. [En línea]. Available: <https://www.xataka.com/automovil/el-mayor-estudio-sobre-la-etica-de-los-coches-autonomos-trae-malas-noticias-para-los-ancianos-e-incertidumbre-para-la-industria>. [Último acceso: 18 04 2022].
- [96] A. Council, «ACM Code of Ethics and Professional Conduct,» 22 Junio 2018. [En línea]. Available: <https://www.acm.org/code-of-ethics>. [Último acceso: 18 04 2022].
- [97] «RTVE - Error informatico en Ariane 5,» [En línea]. Available: <https://www.rtve.es/noticias/20140604/error-software-convirtio-lanzamiento-espacial-carismos-fuegos-artificiales/948262.shtml>. [Último acceso: 18 04 2022].
- [98] «DGT,» [En línea]. Available: <https://www.dgt.es/muevete-con-seguridad/conoce-las-normas-de-trafico/nuevas-leyes/>. [Último acceso: 18 04 2022].
- [99] .. A. W. Ika Febrilia, «The Effects of Positive and Negative Mood on University,» Faculty of Liberal Arts, Prince of Sogkla University, 2 Abril 2011. [En línea]. Available:

<http://fs.libarts.psu.ac.th/research/conference/proceedings-3/2pdf/007.pdf>. [Último acceso: 18 04 2022].

- [100] G. C. Gee y D. T. Takeuchi, «Traffic stress, vehicular burden and well-being: A multilevel analysis,» *Social Science & Medicine*, 21 Enero 2004. [En línea]. Available: <https://doi.org/10.1016/j.socscimed.2003.10.027>. [Último acceso: 18 04 2022].
- [101] J. L. ARANDA, «El precio de la gasolina en España bate su récord histórico por sexta semana consecutiva,» *El País Economía*, 10 03 2022. [En línea]. Available: <https://elpais.com/economia/2022-03-10/el-precio-de-la-gasolina-en-espana-bate-su-record-historico-por-sexta-semana-consecutiva.html>. [Último acceso: 18 04 2022].
- [102] U. D. CORDOBA, «LA LEY NATURAL EN SANTO TOMÁS,» 2010. [En línea]. Available: <http://www.uco.es/dptos/ciencias-juridicas/filosofia-derecho/diego/Nuevo/TeoriaII/lecciones/materiales/LeternaST.htm>. [Último acceso: 18 04 2022].
- [103] N. Ighodaro, «Pusher Blog Laravel MVC Use,» 24 Maio 2018. [En línea]. Available: <https://blog.pusher.com/laravel-mvc-use/>. [Último acceso: 02 05 2022].
- [104] «Vue.js Guidelines Life Cicle,» [En línea]. Available: <https://es.vuejs.org/v2/guide/instance.html#Diagrama-del-Ciclo-de-vida>. [Último acceso: 02 05 2022].
- [105] T. Teravainen, «Single Sign-On TechTarget,» [En línea]. Available: <https://www.techtarget.com/searchsecurity/definition/single-sign-on>. [Último acceso: 13 05 2022].
- [106] Microsoft, «Microsoft Dynamics Single Sign-on,» [En línea]. Available: <https://www.microsoft.com/es-es/security/business/identity-access-management/single-sign-on>. [Último acceso: 03 05 2022].
- [107] Microsoft, «Microsoft Office 365,» [En línea]. Available: <https://www.microsoft.com/es-es/microsoft-365>. [Último acceso: 13 05 2022].

- [108] G. Android, «Android Auto,» [En línea]. Available: <https://play.google.com/store/apps/details?id=com.google.android.projection.gearhead&hl=es&gl=US>. [Último acceso: 13 05 2022].
- [109] J. S. L. W. Marc Bechler, «IN-CAR COMMUNICATION USING WIRELESS TECHNOLOGY,» Institute of Telematics, Universität Karlsruhe (TH), Germany, [En línea]. Available: https://www.ibr.cs.tu-bs.de/users/bechler/myPublications/marc_BeSW01.pdf. [Último acceso: 13 05 2022].
- [110] M. B. Moser, «Repositorio Público SOTA en Github,» 15 05 2022. [En línea]. Available: <https://github.com/matbmoser/SOTA>. [Último acceso: 15 05 2022].

ANEXO A: REQUISITOS DE USUARIO

10.1. DESCRIPCIÓN DE ATRIBUTOS

10.1.1.1. Descripción de Prioridades

Tabla 20: Prioridades de los Requisitos – Elaboración Propia

Prioridad	Nombre	Descripción
P1	Debe Tener	La funcionalidad es obligatoria y debe estar en el entregable final.
P2	Debería Tener	La funcionalidad no es obligatoria, pero se recomienda que este en el entregable.
P3	Puede Tener	La funcionalidad no es obligatoria, pero se puede incluir de manera opcional.
P4	No debería tener	La funcionalidad no es obligatoria y es opcional, pero encima no se recomienda o exige un coste demasiado alto.
P5	Desestimado	La funcionalidad es inalcanzable, y no se puede incluir el pliego de requisitos.

10.1.1.2. Descripción de Campos

Tabla 21: Descripción de Campos – Elaboración Propia

Nombre	Descripción
Id	Identificador único de requisito, está compuesto por: <i>TIPO_IDMod_IDReq</i>
Pedido por	Entidad que solicitó el requisito, suele ser el cliente o el equipo.
Descripción	Descripción del requisito en detalle.
Prioridad	Indica el nivel de importancia/prioridad del requisito
E	Si el requisito es estable, en otras palabras, no cambia en el desarrollo.
C	Si el requisito es claro, para ello tanto cliente y equipo deben entenderlo.
V	Si el requisito es verificable, si lo es será posible ver si se cumple o no.

10.1.1.3. Descripción de Módulos

Tabla 22: Descripción de Módulos – Elaboración Propia

Id	Tipo	Nombre	Descripción
001	RF	Digital Twin	Se agrupan los requisitos que especifican que deberá tener el <i>Digital Twin</i> .
002	RF	Servidor Device Manager	Se agrupan los requisitos que especifican que deberá tener el Servidor <i>DeviceManager</i> .
003	RF	Protocolo SJMP	Se describe el funcionamiento del protocolo SJMP y su lógica de negocio.
004	RF	Paquete SJMP	Se describe los componentes de un paquete.
005	RF	Flags Paquete SJMP	Se describe los componentes de las <i>Flags SJMP</i> .
006	RF	Parking	Se describe las especificaciones del <i>parking</i> .
007	RF	Plazas Parking	Se describen las especificaciones de las diferentes plazas dentro del <i>parking</i> seleccionado.
008	RF	WebApp	Se especifica que contiene la aplicación web <i>MyParking</i> .
009	RF	WebApp Autenticación	Especifica como será la autenticación.
010	RF	WebApp Área de Usuario	Especifica los componentes del área de usuario.
011	RF	WebApp Menú	Especifica los componentes del menú.
012	RF	WebApp Notificaciones	Especifica la estructura de las notificaciones.
013	RF	WebApp Vehículos	Especifica los componentes de la lógica y vista de vehículos.
014	RF	WebApp Perfil del Usuario	Especifica los componentes del Perfil del Usuario.
015	RF	WebApp Ticket	Especifica la estructura de un <i>Ticket</i> y su lógica.
016	RF	WebApp Incidencias	Especifica el portal de incidencias y su lógica.
017	RF	Base de Datos	Especifica los requisitos de la base de datos.
018	RNF	Software	Especifica las restricciones de software.
019	RNF	Contenedores	Especifica la estructura de configuración de los contenedores Docker.
020	RNF	Red	Especifica la estructura de red y los puertos que serán abiertos en los contenedores.
021	RNF	Volúmenes	Especifica los volúmenes que se deberán crear para guardar los ficheros de los contenedores de manera persistente.
022	RNF	Seguridad	Se especifica las pautas de seguridad generales y del sistema.
023	RNF	Accesibilidad	Se especifica las pautas de accesibilidad de los diferentes sistemas.
024	RNF	Legal	Se especifica los requisitos legales relacionados con el sistema.
025	RNF	Control de Accesos	Se especifica los elementos de control de accesos necesarios para poder implementar el sistema de manera física en el <i>parking</i> .
026	RNF	Presentación	Se especifica como se realizará la presentación de manera física en el <i>parking</i> .

10.2. REQUISITOS FUNCIONALES

10.2.1. Módulo Digital Twin

Tabla 23: Requisitos del Módulo Digital Twin – Elaboración Propia

Id	Pedido por	Descripción	Prioridad	E	C	V
RF_001_01	<i>Cliente</i>	Se debe poder visualizar el mapa del <i>parking</i> , (señalizado por zonas)	P1	X	X	X
RF_001_02	<i>Cliente</i>	Visualizar sobre el mapa el número de plazas en cada zona.	P2	X	X	X
RF_001_03	<i>Cliente</i>	Podrá abrir el servidor de <i>DeviceManager</i>	P1	X	X	X
RF_001_04	<i>Cliente</i>	Podrá cerrar el servidor de <i>DeviceManager</i>	P1	X	X	X
RF_001_05	<i>Cliente</i>	Habrá un formulario para configurar los datos de conexión de la socket (IP:PUERTO) y el usuario al abrir el Digital Twin.	P1	X	X	X
RF_001_06	<i>Cliente</i>	Visualizar el total de plazas libres	P1	X	X	X
RF_001_07	<i>Cliente</i>	Visualizar el total de plazas ocupadas	P1	X	X	X
RF_001_08	<i>Cliente</i>	Visualizar el total de plazas	P1	X	X	X
RF_001_09	<i>Cliente</i>	Se podrá añadir un vehículo al <i>parking</i> indicando su matrícula.	P1	X	X	X
RF_001_10	<i>Cliente</i>	Se podrá borrar un vehículo al <i>parking</i> indicando su matrícula.	P1	X	X	X
RF_001_11	<i>Cliente</i>	Se podrá ver los vehículos dentro del <i>parking</i> en una lista con campos (Matricula, Plaza, Fecha Entrada, Tipo)	P1	X	X	X
RF_001_12	<i>Cliente</i>	Al añadir un vehículo se podrá generar la matrícula de manera aleatoria.	P5	X	X	X
RF_001_13	<i>Cliente</i>	Se actualizará la información a cada 10 segundos.	P1	X	X	X
RF_001_14	<i>Cliente</i>	Se indicará la hora actual en el panel.	P1	X	X	X
RF_001_15	<i>Cliente</i>	Se indicará la fecha actual en el panel.	P1	X	X	X
RF_001_16	<i>Equipo</i>	Deberá incluir el username del usuario.	P1	X	X	X
RF_001_17	<i>Cliente</i>	Se indicará la ip del cliente en el panel.	P2	X	X	X
RF_001_18	<i>Cliente</i>	Tendrá una pantalla de <i>login</i> .	P1	X	X	X

Id	Pedido por	Descripción	Prioridad	E	C	V
RF_001_19	Cliente	El <i>login</i> será seguro	P1	X	X	X
RF_001_20	Cliente	Solo se permitirá la entrada de roles autorizados.	P1	X	X	X
RF_001_21	Cliente	Podrá levantar un servidor en un socket en su servidor local.	P1	X	X	X
RF_001_22	Equipo	Habrá un botón de <i>refresh</i> , para volver a solicitar la información.	P1	X	X	X
RF_001_23	Equipo	Habrá un botón de <i>auto-refresh</i> , para volver a solicitar la información automáticamente.	P2	X	X	X

10.2.2. Módulo Servidor Device Manager

Tabla 24: Requisitos del Módulo Device Manager – Elaboración Propia

Id	Pedido por	Descripción	Prioridad	E	C	V
RF_002_01	Cliente	Contará con un servidor <i>hosteado</i> en un socket	P1	X	X	X
RF_002_02	Cliente	El servidor interpretará TCP	P1	X	X	X
RF_002_03	Cliente	El servidor interpretará <i>WebSocket</i>	P1	X	X	X
RF_002_04	Cliente	Se deberá rechazar cualquier conexión que no utilice SJMP.	P1	X	X	X
RF_002_05	Cliente	Tendrá conexión con la base de datos	P1	X	X	X
RF_002_06	Cliente	El servidor interpretará SJMP.	P1	X	X	X

10.2.3. Módulo Protocolo SJMP

Tabla 25: Requisitos del Módulo SJMP – Elaboración Propia

Id	Pedido por	Descripción	Prioridad	E	C	V
RF_003_01	Cliente	Deberá ir sobre TCP.	P1	X	X	X
RF_003_02	Cliente	Deberá ir sobre HTTP y <i>WebSocket</i>	P1	X	X	X
RF_003_03	Cliente	Su lógica de negocio será transmitida en paquetes SJMP.	P1	X	X	X

10.2.4. Módulo Paquete SJMP

Tabla 26: Requisitos del Módulo Paquete SJMP – Elaboración Propia

Id	Pedido por	Descripción	Prioridad	E	C	V
RF_004_01	<i>Cliente</i>	Deberá incluir campo "srv-time" con el timestamp de la hora de envío si sale del servidor.	P1	X	X	X
RF_004_02	<i>Cliente</i>	Deberá incluir campo "clt-time" con el timestamp de la hora de envío si sale de cualquier cámara.	P1	X	X	X
RF_004_03	<i>Cliente</i>	Deberá contener una flag.	P1	X	X	X
RF_004_04	<i>Cliente</i>	Cada cliente tendrá un "token" único que servirá de clave publica, que deberá estar relacionada matemáticamente con la clave privada. Solo te intercambiará en el mensaje de conexión.	P1	X	X	X
RF_004_05	<i>Cliente</i>	Cada servidor tendrá una clave publica llamada "secret" que permita encriptar las comunicaciones desde la cámara.	P1	X	X	X
RF_004_06	<i>Cliente</i>	Cada sesión iniciada por el cliente deberá tener un sessionid único.	P1	X	X	X
RF_004_07	<i>Cliente</i>	En la primera conexión se indicará un cameraid que permita identificar a la cámara y se devolverá un OK.	P1	X	X	X
RF_004_09	<i>Cliente</i>	Una vez encriptadas las sesiones el cliente encriptará los paquetes con la clave pública del servidor "secret" y el servidor encriptará los paquetes con la clave publica del cliente "token".	P1	X	X	X

10.2.5. Módulo Flags de Paquete SJMP

Tabla 27: Requisitos del Módulo Flags de Paquete SJMP – Elaboración Propia

Id	Pedido por	Descripción	Prioridad	E	C	V
RF_005_01	<i>Cliente</i>	El <i>flag</i> para añadir coches será "IN" y contendrá los siguientes campos: <i>clt-time, flag, plate, sessionid</i> .	P1	X	X	X
RF_005_02	<i>Cliente</i>	El flag para quitar coches será "OUT" y contendrá los siguientes campos: <i>clt-time, flag, plate, sessionid</i> .	P1	X	X	X
RF_005_03	<i>Cliente</i>	Se devolverá el flag "ACK" para confirmar que la conexión ha llegado al destino.: <i>srv-time, flag, response</i> .	P1	X	X	X
RF_005_04	<i>Cliente</i>	Se devolverá el flag "ERR" para indicar que hubo un error en el servidor a raíz de la petición del cliente. <i>srv-time, flag, response</i> .	P1	X	X	X
RF_005_05	<i>Cliente</i>	Se utilizará el flag "SYN" para realizar una petición de apertura de conexión por parte del cliente. Contendrá la siguiente estructura: <i>clt-time, flag, cameraid, token, type ["IN", "OUT", "BOTH"]</i>	P1	X	X	X
RF_005_06	<i>Cliente</i>	Se utilizará el flag "OK" para indicar al cliente su <i>sessionid</i> y la clave pública <i>secret</i> del servidor. Este paquete irá encriptado con la clave publica del cliente desde el servidor, contiene: <i>srv-time, flag, sessionid, secret</i> .	P1	X	X	X

10.2.6. Módulo Parking

Tabla 28: Requisitos del Módulo Parking – Elaboración Propia

Id	Pedido por	Descripción	Prioridad	E	C	V
RF_006_01	<i>Cliente</i>	El <i>parking</i> tendrá un aforo total de 182 plazas.	P1	X	X	X
RF_006_02	<i>Cliente</i>	El <i>parking</i> tendrá 3 tipos de plazas (Línea, Batería Diagonal, Batería)	P1	X	X	X
RF_006_03	<i>Cliente</i>	El <i>parking</i> estará dividido en 9 zonas nombradas del (A al I)	P1	X	X	X
RF_006_04	<i>Cliente</i>	Solo se permitirán entrar a los <i>usuarios</i> registrados en la Aplicación Web (<i>MyParking</i>)	P1	X	X	X

Id	Pedido por	Descripción	Prioridad	E	C	V
RF_006_05	<i>Cliente</i>	Habrá una única entrada y salida con barreras. La zona estará cerrada alrededor.	P1	X	X	X
RF_006_06	<i>Cliente</i>	Las barreras se abrirán si la matrícula leída cuadra con un usuario registrado y hay plazas disponibles.	P1	X	X	X
RF_006_07	<i>Cliente</i>	Solo se permitirán coches y furgonetas.	P1	X	X	X
RF_006_08	<i>Cliente</i>	Se podrán abrir y cerrar las barreras de entrada y salida desde el Digital Twin.	P1	X	X	X

10.2.7. Módulo Plaza Parking

Tabla 29: Requisitos del Módulo Plaza Parking – Elaboración Propia

Id	Pedido por	Descripción	Prioridad	E	C	V
RF_007_01	<i>Cliente</i>	La zona A tiene 24 plazas de tipo batería	P1	X	X	X
RF_007_02	<i>Cliente</i>	La zona B tiene 25 plazas de tipo batería.	P1	X	X	X
RF_007_03	<i>Cliente</i>	La zona C tiene 22 plazas de tipo batería	P1	X	X	X
RF_007_04	<i>Cliente</i>	La zona D tiene 21 plazas de tipo batería	P1	X	X	X
RF_007_05	<i>Cliente</i>	La zona E tiene 21 plazas de tipo batería.	P1	X	X	X
RF_007_06	<i>Cliente</i>	La zona F tiene 21 plazas de tipo batería.	P1	X	X	X
RF_007_07	<i>Cliente</i>	La zona G tiene 14 plazas de tipo batería diagonal.	P1	X	X	X
RF_007_08	<i>Cliente</i>	La zona H tiene 4 plazas de tipo línea.	P1	X	X	X
RF_007_09	<i>Cliente</i>	La zona I tiene 23 plazas de tipo batería.	P1	X	X	X
RF_007_10	<i>Cliente</i>	Una plaza de tipo batería tiene: Largo: 515 cm, Ancho: 250 cm. Puede albergar todos los tipos menos Pequeños	P1	X	X	X
RF_007_11	<i>Cliente</i>	Una plaza de tipo batería diagonal tiene: Largo: 555cm, Ancho: 300 cm, y puede todos los tipos.	P1	X	X	X
RF_007_12	<i>Cliente</i>	Una plaza de tipo línea tiene: Largo:500 cm, Ancho: 200cm y puede albergar vehículos tipos Pequeño y Medianos.	P1	X	X	X
RF_007_13	<i>Cliente</i>	Se deberá priorizar a los tipos de vehículos que permita la plaza.	P1	X	X	X
RF_007_14	<i>Cliente</i>	La plaza será asignada según el tipo de vehículo.	P1	X	X	X
RF_007_15	<i>Cliente</i>	Se utilizarán para los				

10.2.8. Módulo WebApp

Tabla 30: Requisitos del Módulo WebApp – Elaboración Propia

Id	Pedido por	Descripción	Prioridad	E	C	V
RF_008_01	<i>Cliente</i>	El nombre de la aplicación web será: "UFV MyParking".	P1	X	X	X
RF_008_02	<i>Cliente</i>	Tendrá un menú que permita navegar entre vistas.	P1	X	X	X
RF_008_03	<i>Cliente</i>	Tendrá una vista de área de usuario	P1	X	X	X
RF_008_04	<i>Cliente</i>	Tendrá una vista de perfil	P1	X	X	X
RF_008_05	<i>Cliente</i>	Tendrá una vista de notificaciones	P1	X	X	X
RF_008_06	<i>Cliente</i>	Tendrá una vista para gestionar los coches del usuario.	P1	X	X	X
RF_008_07	<i>Cliente</i>	Tendrá una vista para visualizar los tickets.	P1	X	X	X
RF_008_08	<i>Cliente</i>	Tendrá una vista para visualizar y resolver las incidencias.	P3	X	X	X
RF_008_09	<i>Cliente</i>	Tendrá una vista para visualizar todos los tickets y un solo ticket.	P3	X	X	X
RF_008_10	<i>Cliente</i>	Tendrá una vista para administrar todos los usuarios, si es administrador.	P2	X	X	X
RF_008_11	<i>Cliente</i>	El administrador podrá modificar el rol de los usuarios.	P2	X	X	X

10.2.9. Módulo WebApp Autenticación

Tabla 31: Requisitos del Módulo WebApp Autenticación – Elaboración Propia

Id	Pedido por	Descripción	Prioridad	E	C	V
RF_009_01	<i>Cliente</i>	Para acceder a la WebApp se deberá insertar el correo al hacer login.	P1	X	X	X
RF_009_02	<i>Cliente</i>	Para acceder a la WebApp se deberá insertar una contraseña al hacer login.	P1	X	X	X
RF_009_03	<i>Cliente</i>	Si el usuario no está registrado habrá un formulario de registro.	P1	X	X	X
RF_009_04	<i>Cliente</i>	Se enviará un correo al usuario al registrarse para confirmar el correo.	P3	X	X	X

10.2.10. Módulo WebApp Área de Usuario

Tabla 32: Requisitos del Módulo WebApp Área de Usuario – Elaboración Propia

Id	Pedido por	Descripción	Prioridad	E	C	V
RF_010_01	Cliente	Tendrá un <i>dashboard</i> que permitirá visualizar el estado del <i>parking</i> .	P1	X	X	X
RF_010_02	Cliente	El <i>dashboard</i> tendrá el número de plazas libres por zona.	P3	X	X	X
RF_010_03	Cliente	Se enseñarán las diferentes zonas en cajas.	P1	X	X	X
RF_010_04	Cliente	Se mostrará el aforo total.	P1	X	X	X

10.2.11. Módulo WebApp Menú

Tabla 33: Requisitos del Módulo WebApp Menú – Elaboración Propia

Id	Pedido por	Descripción	Prioridad	E	C	V
RF_011_01	Cliente	El menú será accesible desde cualquier página.	P1	X	X	X
RF_011_02	Cliente	El menú permitirá navegar a todas las vistas.	P1	X	X	X
RF_011_03	Cliente	Tendrá un ícono de campana como botón para ver las notificaciones	P1	X	X	X
RF_011_04	Cliente	Tendrá un ícono de usuario como botón para ver el perfil.	P1	X	X	X
RF_011_05	Cliente	Será variable según el rol, permitiendo más o menos funcionalidades.	P1	X	X	X

10.2.12. Módulo WebApp Vehículos

Tabla 34: Requisitos del Módulo WebApp Vehículos – Elaboración Propia

Id	Pedido por	Descripción	Prioridad	E	C	V
RF_013_01	Cliente	Se podrán añadir vehículos insertando la matrícula y el tipo de vehículos.	P1	X	X	X
RF_013_02	Cliente	Un coche debe pertenecer a un solo usuario.	P1	X	X	X
RF_013_03	Cliente	Se podrá borrar un vehículo.	P1	X	X	X
RF_013_04	Cliente	Se visualizarán los vehículos en una tabla	P1	X	X	X

10.2.13. Módulo WebApp Perfil del Usuario

Tabla 35: Requisitos del Módulo WebApp Perfil del Usuario – Elaboración Propia

Id	Pedido por	Descripción	Prioridad	E	C	V
RF_014_01	<i>Cliente</i>	Se visualizarán los datos introducidos por el usuario del usuario	P1	X	X	X
RF_014_02	<i>Equipo</i>	El usuario podrá modificar su contraseña	P3	X	X	X
RF_014_03	<i>Cliente</i>	Se indicará el número de incidencias	P3	X	X	X
RF_014_04	<i>Cliente</i>	Se indicará el número de <i>tickets</i>	P3	X	X	X
RF_014_05	<i>Cliente</i>	Se indicará el número de vehículos	P3	X	X	X

10.2.14. Módulo WebApp Ticket

Tabla 36: Requisitos del Módulo WebApp Ticket – Elaboración Propia

Id	Pedido por	Descripción	Prioridad	E	C	V
RF_015_01	<i>Cliente</i>	Habrá una opción para abrir un <i>ticket</i>	P1	X	X	X
RF_015_02	<i>Cliente</i>	Tendrá un QR que podrá ser validado.	P2	X	X	X
RF_015_03	<i>Cliente</i>	Se indicará si el <i>ticket</i> es válido o no.	P1	X	X	X
RF_015_04	<i>Cliente</i>	Se podrán visualizar todos los <i>tickets</i> en una tabla.	P1	X	X	X
RF_015_05	<i>Cliente</i>	Un <i>ticket</i> quedará invalidado cuando el vehículo haya salido del parking.	P1	X	X	X
RF_015_06	<i>Cliente</i>	Se asignará una plaza al usuario cuando entre en el parking.	P2	X	X	X
RF_015_07	<i>Cliente</i>	Habrá un botón para crear una incidencia.	P3	X	X	X
RF_015_08	<i>Cliente</i>	Se visualizarán los datos en un <i>template</i> predefinido.	P2	X	X	X
RF_015_09	<i>Cliente</i>	Habrá una opción para abrir un <i>ticket</i>	P2	X	X	X

10.2.15. Módulo WebApp Incidencias

Tabla 37: Requisitos del Módulo WebApp Incidencias – Elaboración Propia

Id	Pedido por	Descripción	Prioridad	E	C	V
RF_016_01	<i>Cliente</i>	Solo un usuario gerente o administrador podrá resolver las incidencias.	P2	X	X	X
RF_016_01	<i>Cliente</i>	Se podrán visualizar las incidencias	P2	X	X	X
RF_016_01	<i>Cliente</i>	Una vez rellenado el formulario de incidencia, se asignará una nueva plaza.	P2	X	X	X
RF_016_01	<i>Cliente</i>	Se podrá subir una foto, enseñando la matrícula.	P3	X	X	X
RF_016_01	<i>Cliente</i>	Se podrá resolver una incidencia indicando una nota de cierre.	P2	X	X	X
RF_016_01	<i>Cliente</i>	Todos los usuarios podrán visualizar sus propias incidencias	P3	X	X	X

10.2.16. Módulo Base de Datos

Tabla 38: Módulo Base de Datos – Elaboración Propia

Id	Pedido por	Descripción	Prioridad	E	C	V
RF_017_01	<i>Cliente</i>	La base de datos deberá ser segura	P1	X	X	X
RF_017_02	<i>Cliente</i>	La base de datos tendrá un usuario de administración	P1	X	X	X
RF_017_03	<i>Cliente</i>	La base de datos deberá poder ser administrada utilizando un administrador de base de datos como PhpMyAdmin	P1	X	X	X
RF_017_04	<i>Equipo</i>	La base de datos deberá ser construida utilizando una sola migración en Laravel.	P1	X	X	X
RF_017_05	<i>Equipo</i>	La base de datos deberá contener un usuario de administración por defecto	P1	X	X	X
RF_017_06	<i>Equipo</i>	Se deberá cargar las configuraciones iniciales de la base de datos con datos prefijados al iniciar una migración.	P1	X	X	X

10.2.17. Módulo WebApp Notificaciones

Tabla 39: Requisitos del Módulo WebApp Notificaciones – Elaboración Propia

Id	Pedido por	Descripción	Prioridad	E	C	V
RF_012_01	<i>Cliente</i>	Permitirá visualizar las notificaciones en una tabla	P3	X	X	X
RF_012_02	<i>Cliente</i>	Permitirá borrar las notificaciones	P3	X	X	X
RF_012_03	<i>Cliente</i>	Una notificación tendrá un título	P2	X	X	X
RF_012_04	<i>Cliente</i>	Una notificación tendrá una descripción.	P2	X	X	X
RF_012_05	<i>Cliente</i>	Tendrá un botón y un enlace de manera opcional.	P2	X	X	X
RF_012_06	<i>Cliente</i>	Tendrá un botón para borrar notificación	P2	X	X	X
RF_012_07	<i>Cliente</i>	El usuario recibirá una notificación cuando se inicialice el aparcamiento incluyendo el <i>ticket</i> .	P3	X	X	X
RF_012_08	<i>Cliente</i>	El usuario recibirá una notificación cuando se termine el aparcamiento incluyendo el <i>ticket</i> .	P3	X	X	X
RF_012_09	<i>Cliente</i>	El usuario recibirá una notificación cuando se envíe la incidencia.	P3	X	X	X
RF_012_10	<i>Cliente</i>	El usuario recibirá una notificación cuando se resuelva la incidencia, incluyendo la nota de cierre.	P3	X	X	X

10.3. REQUISITOS NO FUNCIONALES

10.3.1. Módulo Software

Tabla 40: Requisitos del Módulo Software – Elaboración Propia

Id	Pedido por	Descripción	Prioridad	E	C	V
RNF_018_01	<i>Cliente</i>	El <i>Digital Twin</i> será desarrollado utilizando PHP 8.1 y JS Nativo	P1	X	X	X
RNF_018_02	<i>Cliente</i>	El protocolo SJMP y el <i>DeviceManager</i> deberán ser implementados en Python 3.9 o en una versión mayor	P1	X	X	X

Id	Pedido por	Descripción	Prioridad	E	C	V
RNF_018_03	<i>Cliente</i>	La aplicación web App será desarrollado utilizando PHP 8.1 con el Framework Laravel 8 en el <i>backend</i> y Vue.js 3 en el <i>frontend</i> .	P1	X	X	X
RNF_018_04	<i>Cliente</i>	Se desplegará la aplicación en contenedores <i>docker</i> .	P1	X	X	X
RNF_018_05	<i>Cliente</i>	Se desplegarán y se configurarán los contenedores Docker utilizando un fichero de <i>docker-composer</i> .	P1	X	X	X
RNF_018_06	<i>Cliente</i>	Se utilizarán el gestor de paquetes npm 8.7.0 para gestionar las dependencias de la aplicación WebApp y los frameworks JS.	P1	X	X	X
RNF_018_07	<i>Cliente</i>	Se utilizará WebApp Mix 6.0.43 para compilar el framework de Vue.js 3 y sus dependencias.	P1	X	X	X
RNF_018_08	<i>Cliente</i>	Se utilizará el gestor de paquetes composer v2.3.5 para gestionar las dependencias la aplicación WebApp y los frameworks PHP.	P1	X	X	X
RNF_018_09	<i>Cliente</i>	Se utilizará la librería de componentes PrimeVue para dar soporte al desarrollo frontend con Vue.js 3	P1	X	X	X
RNF_018_10	<i>Equipo</i>	El <i>Digital Twin</i> y el WebApp se desarrollará con HTML5	P1	X	X	X
RNF_018_11	<i>Equipo</i>	El <i>Digital Twin</i> se desarrollará utilizando Bootstrap 5.1.3	P1	X	X	X
RNF_018_12	<i>Equipo</i>	El <i>Digital Twin</i> y el WebApp utilizarán CSS3	P1	X	X	X
RNF_018_13	<i>Equipo</i>	En el <i>Digital Twin</i> usará jQuery v3.5.	P1	X	X	X
RNF_018_14	<i>Cliente</i>	La base de datos utilizada será MySQL 8.0.28	P1	X	X	X
RNF_018_15	<i>Cliente</i>	Se utilizará el administrador de base de datos phpMyAdmin 5.1.3 para poder administrar la base de datos MySQL.	P1	X	X	X
RNF_018_16	<i>Cliente</i>	El <i>Digital Twin</i> y WebApp utilizarán servidores Apache 2	P1	X	X	X
RNF_018_17	<i>Cliente</i>	Se utilizará Azure DevOps 2021 como repositorio y entorno de seguimiento del proyecto y su fase de desarrollo.	P1	X	X	X

10.3.2. Módulo Contenedores

Tabla 41: Requisitos del Módulo Contenedores – Elaboración Propia

Id	Pedido por	Descripción	Prioridad	E	C	V
RNF_019_01	Equipo	El grupo de contenedores se llamará SOTA.	P1	X	X	X
RNF_019_02	Equipo	El contenedor del Digital Twin se llamará "digital-twin" y contendrá el servidor de <i>DiviceManager</i> y el Digital Twin.	P1	X	X	X
RNF_019_03	Equipo	El contenedor de la aplicación <i>MyParking</i> se llamará "php-webapp" y contendrá la aplicación Web.	P1	X	X	X
RNF_019_04	Equipo	El contenedor de la base de datos se llamará "mysql-bbdd" y será una dependencia de todos los contenedores.	P1	X	X	X
RNF_019_05	Equipo	El contenedor del administrador de la bbdd se llamará "php-my-admin" y contendrá el php-my-admin.	P1	X	X	X

10.3.3. Módulo Red

Tabla 42: Requisitos del Módulo Red – Elaboración Propia

Id	Pedido por	Descripción	Prioridad	E	C	V
RNF_020_01	Equipo	Los contenedores estarán interconectados en una red interna llamada <i>appNet</i>	P1	X	X	X
RNF_020_02	Equipo	Los servicios estarán disponibles en puertos	P1	X	X	X
RNF_020_03	Equipo	El contenedor "digital-twin" deberá tener los puertos 3333 y 8888 abiertos.	P1	X	X	X
RNF_020_04	Equipo	El contenedor "php-webapp" deberá tener el puerto 8080 para servidor de producción y los puertos de 3000 a 3002 abiertos para servidores de desarrollo.	P1	X	X	X
RNF_020_05	Equipo	El contenedor "mysql-bbdd" deberá tener el puerto 3306 abierto.	P1	X	X	X
RNF_020_06	Equipo	El contenedor "php-my-admin" deberá tener el puerto 7777 abierto	P1	X	X	X

10.3.4. Módulo Volúmenes

Tabla 43 Requisitos del Módulo Volúmenes – Elaboración Propia

Id	Pedido por	Descripción	Prioridad	E	C	V
RNF_021_01	Equipo	La base de datos deberá guardar sus datos en el volumen local: "./data"	P1	X	X	X
RNF_021_02	Equipo	El <i>Digital Twin</i> guardará sus ficheros en el volumen: "./digital-twin/src"	P1	X	X	X
RNF_021_03	Equipo	La aplicación <i>WebApp</i> guardará sus ficheros en el volumen: "./webapp/src"	P1	X	X	X

10.3.5. Módulo Seguridad

Tabla 44: Requisitos del Módulo Seguridad – Elaboración Propia

Id	Pedido por	Descripción	Prioridad	E	C	V
RNF_022_01	Equipo	Las contraseñas serán <i>hasheadas</i> con SHA-512	P1	X	X	X
RNF_022_02	Equipo	Las conexiones estarán <i>tokenizadas</i> con <i>tokens hasheados</i> en SHA-256	P1	X	X	X
RNF_022_03	Cliente	Las conexiones deberán ser seguras.	P1	X	X	

10.3.6. Módulo de Accesibilidad

Tabla 45: Requisitos del Módulo Accesibilidad – Elaboración Propia

Id	Pedido por	Descripción	Prioridad	E	C	V
RNF_023_01	Cliente	La aplicación <i>WebApp</i> deberá adaptarse a móviles (<i>responsive</i>).	P1	X	X	X
RNF_023_02	Cliente	El <i>Digital Twin</i> deberá podrá desplegarse en una pantalla TFT 75"	P1	X	X	X
RNF_023_03	Cliente	Todas las aplicaciones deberán ser atractivas e intuitivas de utilizar.	P1	X	X	
RNF_023_04	Equipo	Será accesible desde un navegador web Firefox versión 97.0 en adelante	P1	X	X	X
RNF_023_05	Equipo	Será accesible desde un navegador web Chrome versión 100.0 en adelante	P1	X	X	X

10.3.7. Módulo Legal

Tabla 46: Requisitos del Módulo Legal – Elaboración Propia

Id	Pedido por	Descripción	Prioridad	E	C	V
RNF_024_01	Cliente	La aplicación WebApp deberá tener un correo de soporte para poder contactar con los administradores	P1	X	X	X
RNF_024_02	Cliente	La aplicación WebApp de haber una política de privacidad.	P3	X	X	X
RNF_024_03	Cliente	La aplicación WebApp de haber una política de cookies.	P3	X	X	X

10.3.8. Módulo Control de Accesos

Tabla 47: Requisitos del Módulo Control de Accesos – Elaboración Propia

Id	Pedido por	Descripción	Prioridad	E	C	V
RNF_025_01	Cliente	El sistema debe disponer de dos barreras automática <i>Came Card G3750</i> conectadas para mástiles de hasta 4M colocadas en la entrada y en la salida del aparcamiento.	P3	X	X	X
RNF_025_02	Equipo	Se dispondrá de dos cámaras de visión infrarroja para la lectura de matrículas en la entrada y en la salida del aparcamiento.	P3	X	X	X
RNF_025_03	Cliente	Habrá una sola entrada y salida en el aparcamiento.	P1	X	X	X

10.3.9. Módulo Presentación

Tabla 48: Requisitos del Módulo Presentación – Elaboración Propia

Id	Pedido por	Descripción	Prioridad	E	C	V
RNF_026_01	Cliente	Habrá una pantalla TFT 75" en la entrada para indicar la plaza al conductor en el Digital Twin.	P4	X	X	X

ANEXO B: DISEÑO DEL SISTEMA

Para el diseño del sistema se realizaron una serie de esquemas:

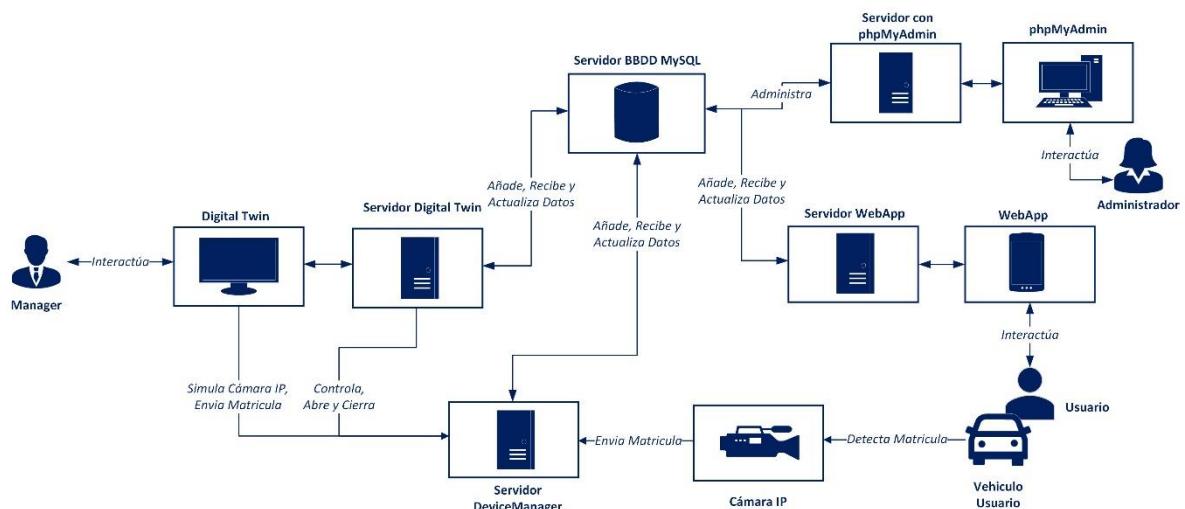


Figura 0.1: Esquema a nivel de Comunicaciones – Elaboración Propia

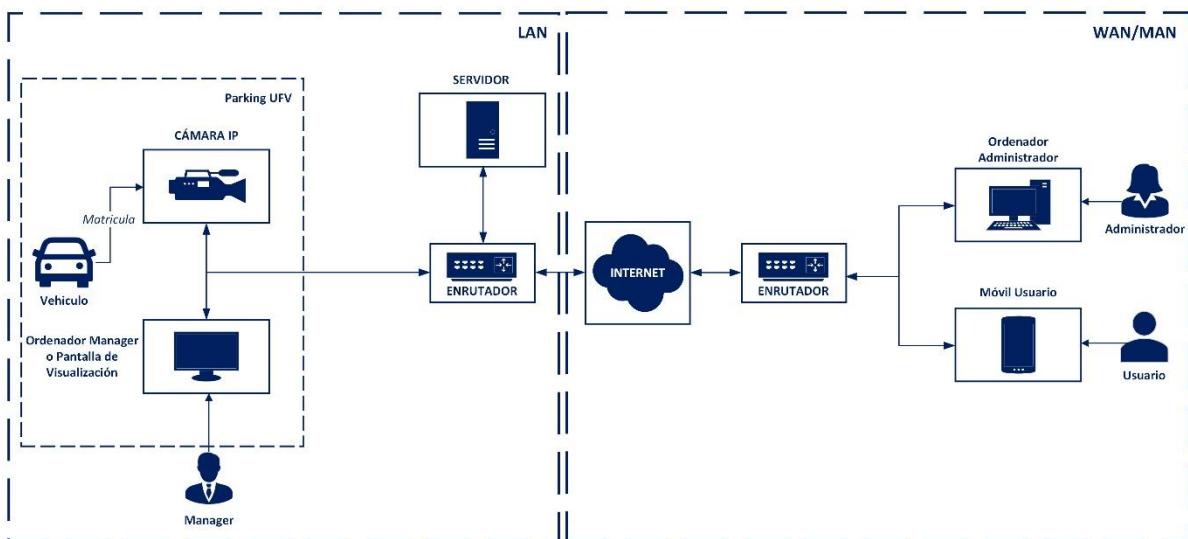


Figura 0.2: Esquema a nivel Red/Físico – Elaboración Propria

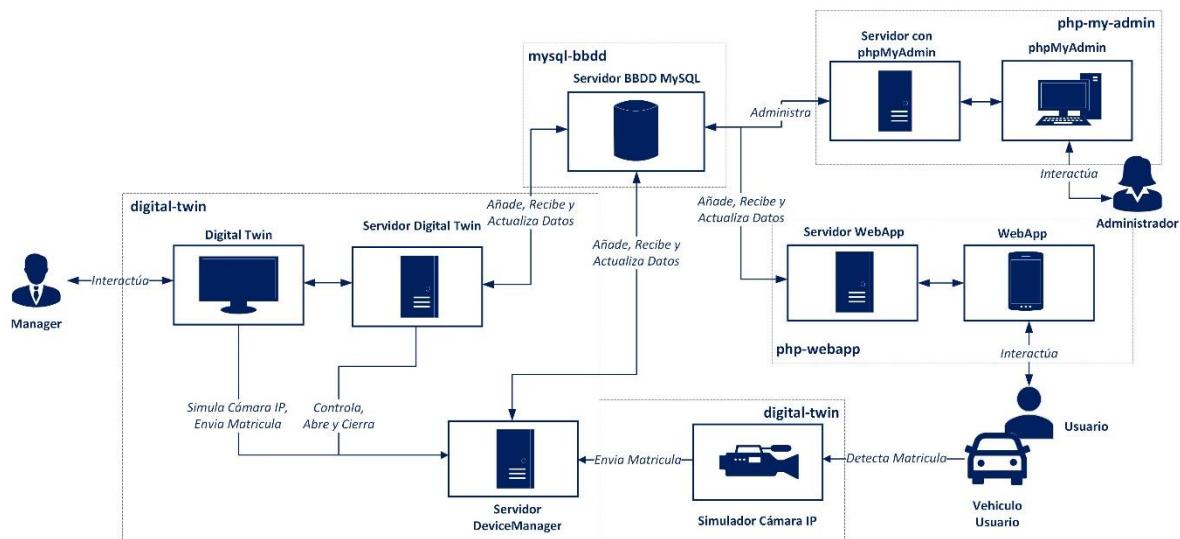


Figura 0.3: Esquema Arquitectura de Contenedores – Elaboración Propia

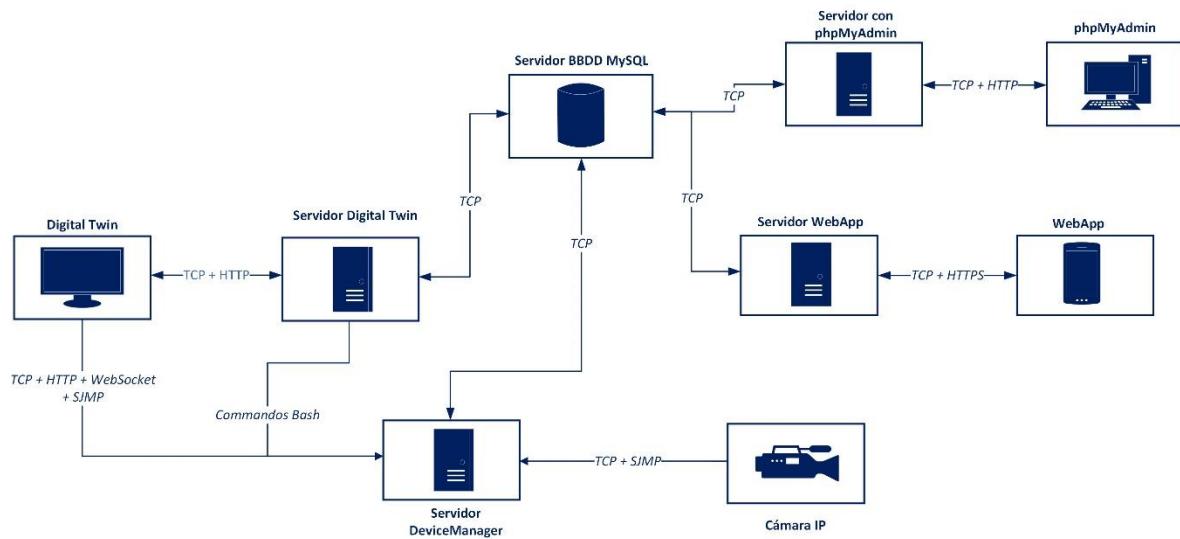


Figura 0.4: Esquema Detallado de Protocolos de Comunicación – Elaboración Propia

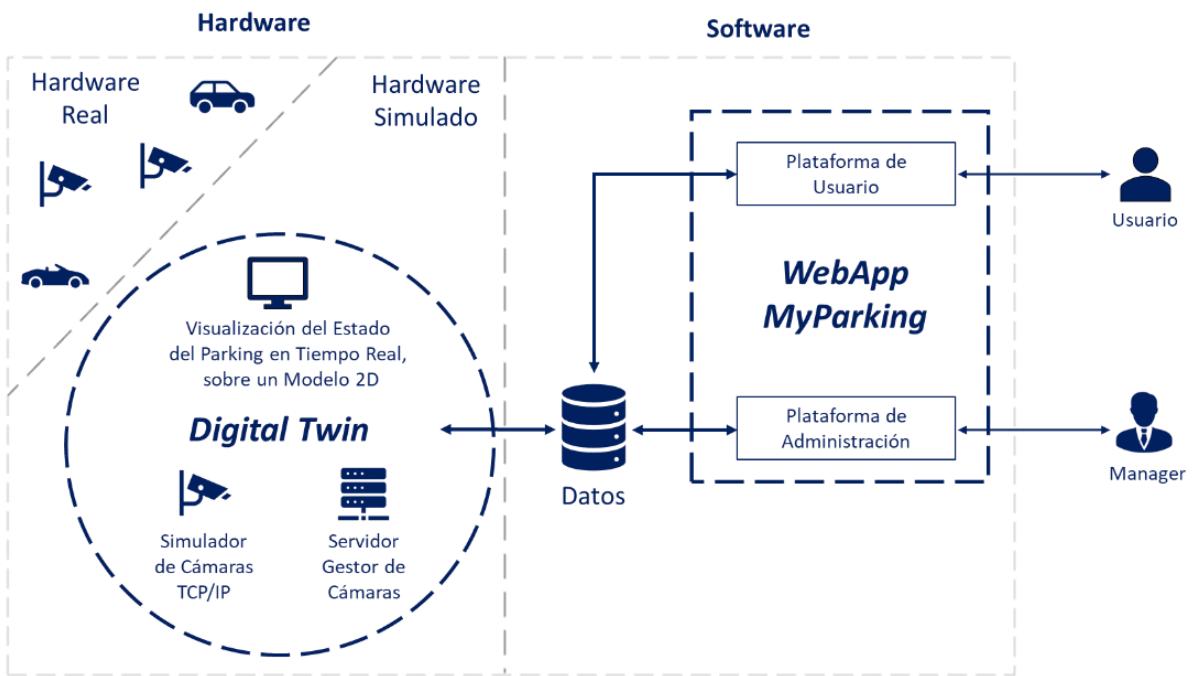


Figura 0.5: Boceto a nivel Conceptual Ampliado. – Elaboración Propia

ANEXO C: MANUALES

MANUAL DE USUARIO Y DE INSTALACIÓN

Debido a que la mayoría de los programas de tipo “*open source*” se almacenan en un repositorio público y que este puede ser accedido por individuos de cualquier país. Se redactó el manual de usuario y de instalación en inglés dentro de un archivo tipo “*README.md*”, codificado en Markdown. El archivo está almacenado junto al código fuente e indica al usuario como deberá proceder para instalar y hacer uso del software.

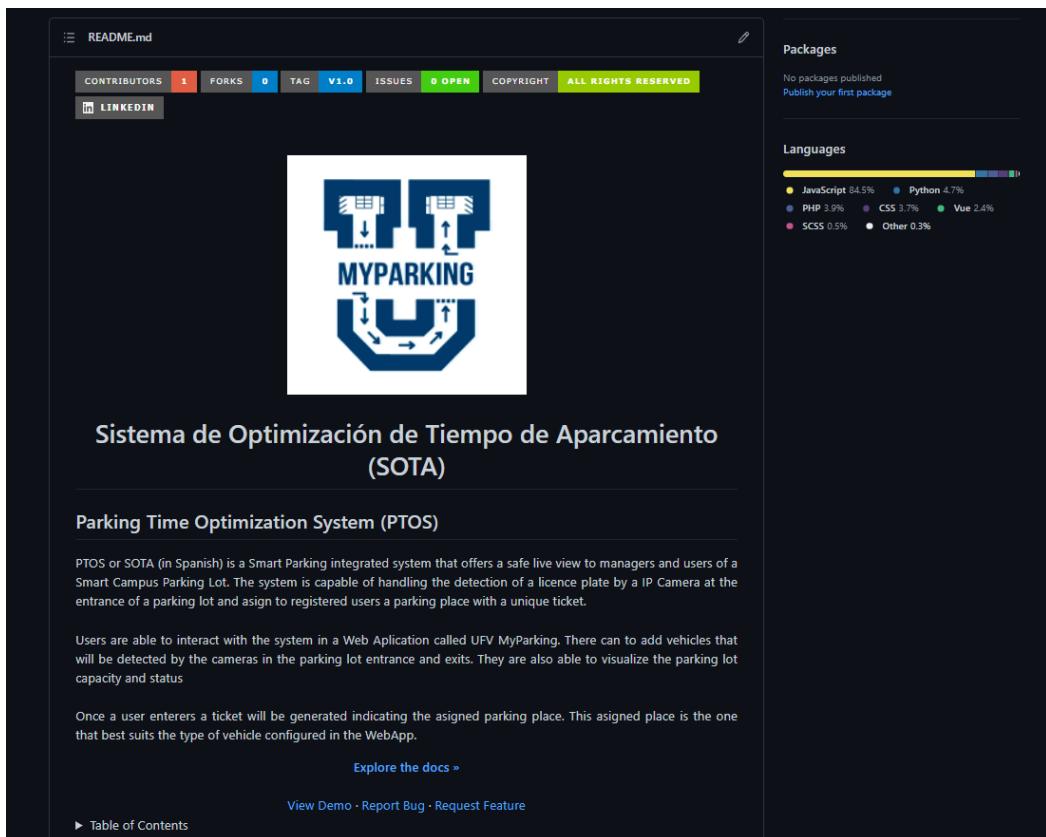


Figura 0.1: README.md en Repositorio de Github [110] – Elaboración Propia

Enlace al Repositorio con el archivo README: <https://github.com/matbmoser/SOTA>

Enlace al Manual de Instalación: <https://github.com/matbmoser/SOTA#getting-started>

Enlace al Manual de Usuario: <https://github.com/matbmoser/SOTA#user-manual>

ANEXO D: DETALLES TÉCNICOS

DESCRIPCIÓN DE PROTOCOLOS

Aquí se incluirán los detalles técnicos del protocolo SJMP y las cabeceras de los protocolos que utiliza:

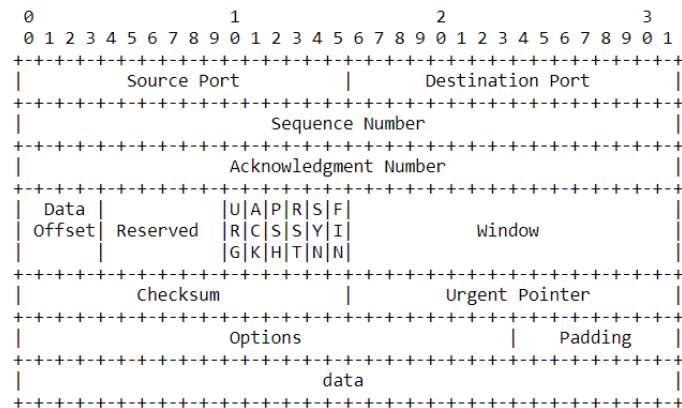


Figura 0.1: Formato Cabecera TCP. **Fuente:** [83]

general-header	= Cache-Control Connection Date Pragma Trailer Transfer-Encoding Upgrade Via Warning	Method	= "OPTIONS" "GET" "HEAD" "POST" "PUT" "DELETE" "TRACE" "CONNECT" extension-method extension-method = token
----------------	--	--------	---

Figura 0.2: Cabecera HTTP General y Métodos Permitidos. **Fuente:** [85]

```
Request      = Request-Line
              *(( general-header
                  | request-header
                  | entity-header ) CRLF)
CRLF
[ message-body ]
```

```

request-header = Accept
               | Accept-Charset
               | Accept-Encoding
               | Accept-Language
               | Authorization
               | Expect
               | From
               | Host
               | If-Match
               | If-Modified-Since
               | If-None-Match
               | If-Range
               | If-Unmodified-Since
               | Max-Forwards
               | Proxy-Authorization
               | Range
               | Referer
               | TE
               | User-Agent

```

Figura 0.3: Estructura de un Mensaje HTTP Request: **Fuente:** [85]

```

Response      = Status-Line
               *(( general-header
                   | response-header
                   | entity-header ) CRLF)
               CRLF
               [ message-body ]

```



```

response-header = Accept-Ranges
                  | Age
                  | ETag
                  | Location
                  | Proxy-Authenticate
                  | Retry-After
                  | Server
                  | Vary
                  | WWW-Authenticate

```

Figura 0.4: Estructura de un Mensaje HTTP Response: **Fuente:** [73]

PROTOCOLO SJMP

El protocolo SJMP (**Secure JSON Message Protocol**) es utilizado en este proyecto para intercambiar datos entre el servidor y un cliente (cámara). SJMP se desarrolló en este PFG y permite el intercambio seguro de información dentro de paquetes que se encriptan a nivel de dato, utilizando criptografía asimétrica para encriptar y desencriptar los paquetes, lo que significa que, utiliza las claves públicas de los clientes/servidor para encriptar y las claves privadas de cada cliente/servidor para desencriptar.

Al principio, al establecer la comunicación TCP, podemos enviar al servidor un paquete con FLAG SYN, que permite indicar que queremos establecer una conexión. En este paquete enviamos la clave pública del cliente (TOKEN) al servidor, para que este pueda encriptar el paquete de vuelta con la clave pública y enviar la suya (SECRET) de vuelta al cliente. Esto permite que los demás paquetes enviados (IN, OUT, ACK y FIN) puedan encriptarse por el cliente y solo podrán ser abiertos por el servidor y viceversa.

Al realizar el intercambio de claves iniciales, existe el riesgo de que el paquete de tipo SYN sea interceptado, por lo tanto, puede ocurrir un ataque de tipo *Man-in-the-middle*. Este protocolo tiene como objetivo proteger el dato, con independencia de la red utilizada (TCP, LoraWan, etc), por lo tanto, se recomienda utilizar, dentro del rango de opciones disponibles, otros protocolos seguros a nivel de transporte, como TLS, con el fin de poder encriptar el envío de paquetes y manejar la encriptación a nivel de transporte.

Todos los paquetes son enriquecidos con el *timestamp* de salida del cliente (clt-time) o servidor (srv-time). Esto permite determinar el tiempo de latencia entre el envío y llegada de los paquetes y por tener una estructura JSON, el protocolo permite la estructuración y rápido acceso a los datos.

Tabla 49: Descripción Flags Protocolo SJMP - Elaboración Propia

[SJMP] Secure JSON Message Protocol Descripción					
Tipo	Flag	Clave Encriptación	Descripción	Server to Client	Client to Server
ACEPTACIÓN HANDSHAKE	OK	token	Indica al cliente que el servidor ha recibido la	{"flag": "OK", "srv-time": "1639143605.659215", "sessionid": "39f66c27b382aa22468588513227}	N/A

			conexión y añadida la nueva cámara e indica la sesión.	6dfe', 'secret':'<clave pública SEVIDOR RSA>'}	
EMPEZAR HANDSHAKE	SYN	N/A	Indica al servidor el nombre de la cámara. Puede ser de tipo IN, OUT o BOTH.	N/A	{"flag": 'SYN', 'clt-time': '1639143605.659215', 'token': '<clave pública camera RSA>', "cameraid": "fd512f541a5", "tipo": "IN"}
AÑADIR COCHE	IN	secret	Cuando la cámara lee una matrícula, y está posicionada en la entrada, envía una flag para añadir un vehículo del parking.	N/A	{"flag": "IN", "matricula": "5213-LDJ?", "clt-time": "1639147366.94", "sessionid": "DAFAUL1234"}
BORRAR COCHE	OUT	secret	Cuando la cámara lee una matrícula, y está posicionada en la salida, envía una flag para añadir un vehículo del parking.	N/A	{"flag": "out", "plate": "5213-LDJ?", "clt-time": "1639147366.94", "sessionid": "e71aeccc9f91f05d46783cd2ad5cf53cc470fb8047f5de7abd387c60f18d4aa0"}
CIERRE DE CONEXIÓN	FIN	secret o token	Cuando la cámara o el servidor quieren cerrar la conexión.	'srv-time': '1639143605.659215', 'flag': 'FIN', response: 'Cierre de Conexión.'}	{"device-time": '1639147205648', 'sessionid': '39f66c27b382aa224685885132276dfe', 'flag': 'FIN', 'message': 'Client disconnected!'}
ERROR	ERR	token	Indica que hay un error.	{'srv-time': '1639143605.659215', 'flag': 'ERR', response: 'Sessionid no valido!'}	N/A
ACKNOWLEDGEMENT	ACK	token	Devuelve la respuesta de la información pedida al servidor.	{'srv-time': '1639143605.659215', 'flag': 'ACK', response: 'Coche Añadido/Borrado'}	N/A

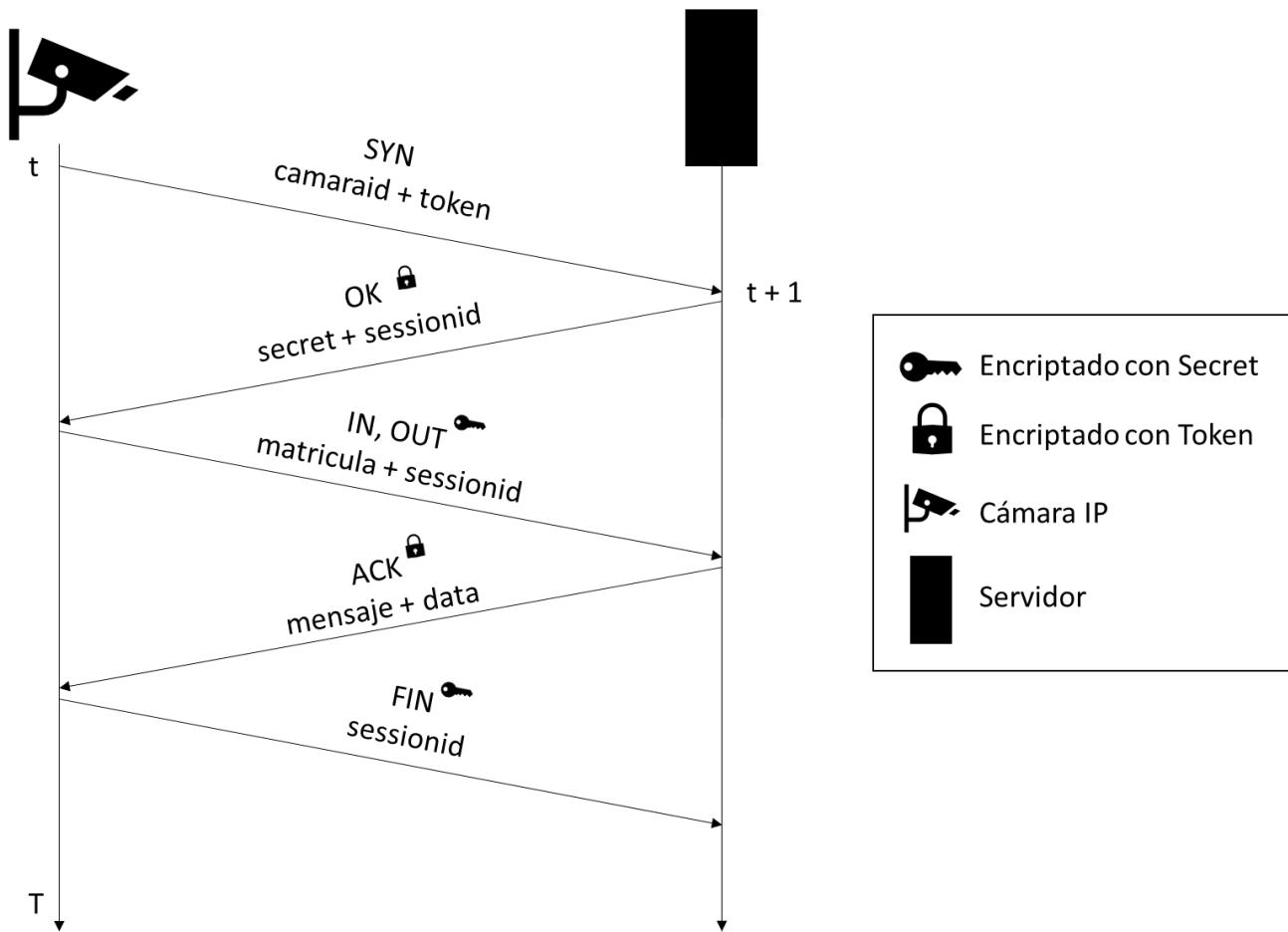


Figura 0.5: Intercambio de claves SJMP – Elaboración propia

ANEXO E: DETALLES ESTUDIO DEL PARKING UFV

En el caso de que alguien en el futuro necesite conocer estos datos de un estudio del Parking UFV, constan aquí las preguntas realizadas, número de respuesta por pregunta y los resultados.



Estudio sobre el Parking UFV

¡Hola! Soy un alumno de 4º de Ingeniería Informática. Actualmente, estoy haciendo mi PFG (Proyecto Final de Grado), e intento proponer una solución informática inteligente para el parking de la UFV.

Te voy a pedir que contestes unas preguntas básicas y rápidas, que me ayudarán a saber si de verdad tenemos un problema en nuestro aparcamiento UFV.

Da igual si tienes coche o no, no tendrás ningún problema en contestar estas preguntas, te lo aseguro...

¡Tardarás 1 minuto y medio como mucho, pero seguro que esto no es nada comparado con lo que tardas en buscar plaza para aparcar!

Por cierto, esta encuesta es totalmente anónima...



matbmoser@gmail.com (no compartidos) [Cambiar de cuenta](#)



Figura 0.1: Introducción Encuesta – Elaboración Propia

¡Empecemos!

¿Consideras que hay un problema en el Parking UFV?

68 respuestas

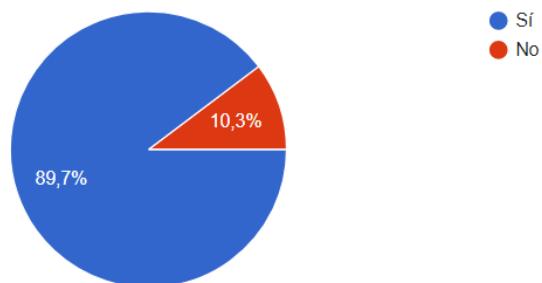


Figura 0.2: Pregunta 1 – Elaboración Propia

¿En que turno del día vienes a la uni?

68 respuestas

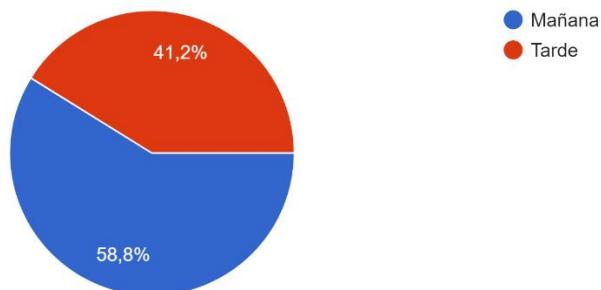


Figura 0.3: Pregunta 2 – Elaboración Propia

¿Cómo vienes a la uni?

68 respuestas

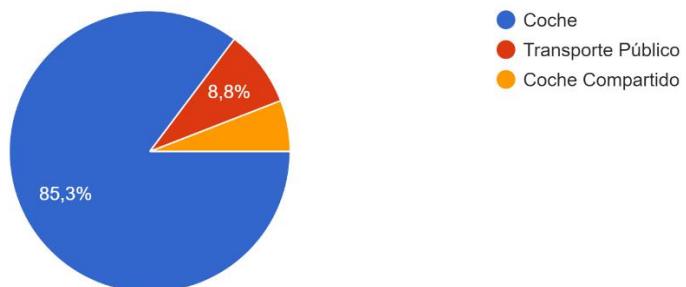


Figura 0.4: Pregunta 3 – Elaboración Propia

Vengo en coche...

Al entrar en el parking, ¿sabes donde quieres aparcar?

58 respuestas

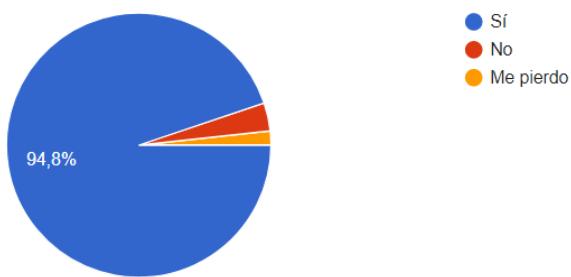


Figura 0.5: Pregunta 4 – Elaboración Propia

¿Cuán lejos de tu edificio tienes que aparcar?

58 respuestas

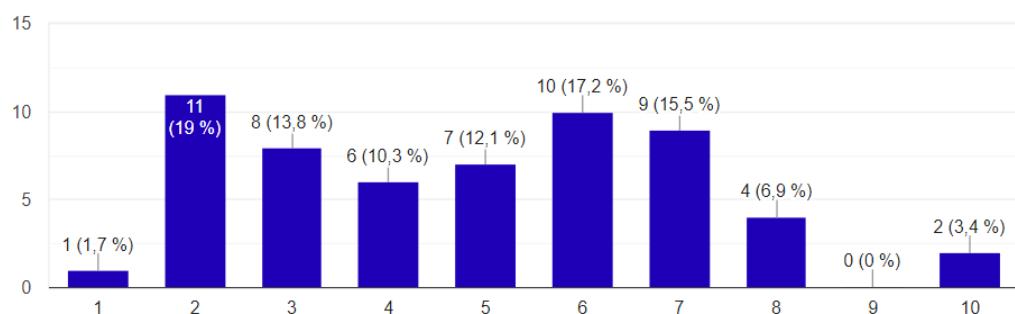


Figura 0.6: Pregunta 5 – Elaboración Propia

¿Cuánto tiempo tardas de media en encontrar plaza?

58 respuestas

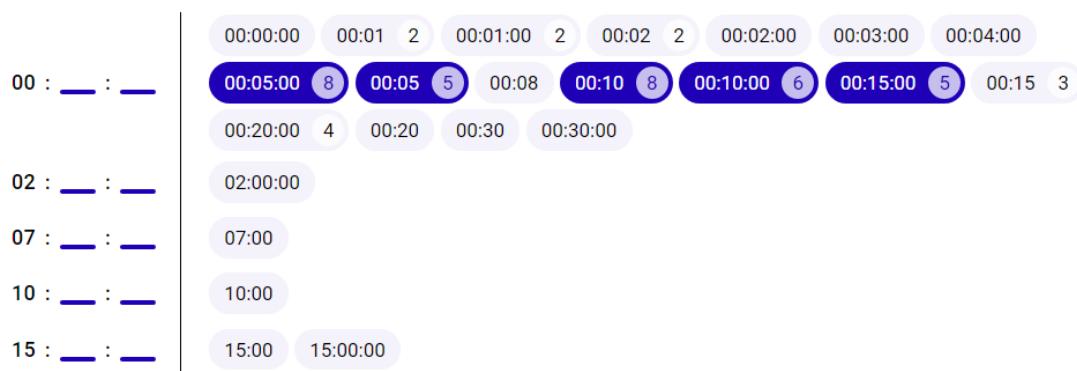


Figura 0.7: Pregunta 6 – Elaboración Propia

¿Cuál es el máximo tiempo que estuviste buscando plaza?

58 respuestas

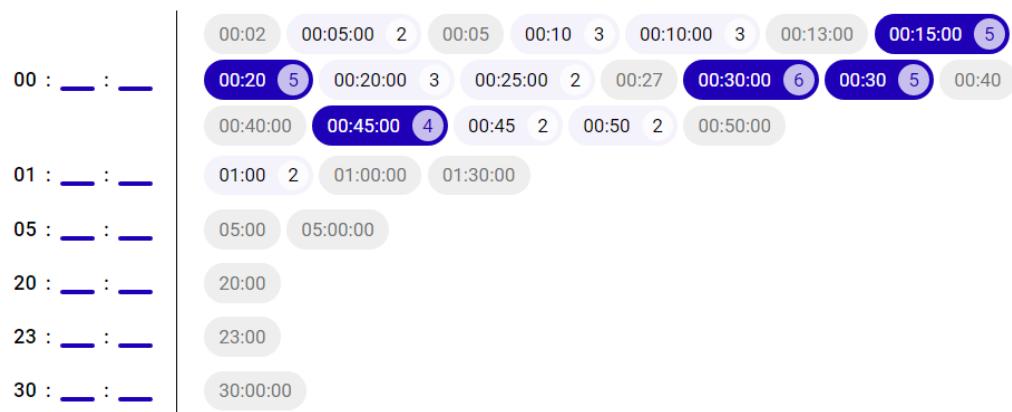


Figura 0.8: Pregunta 7 – Elaboración Propia

¿A que hora sueles tener problemas para encontrar plaza?

58 respuestas

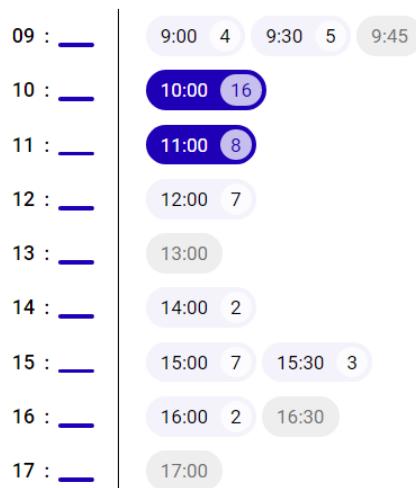


Figura 0.9: Pregunta 8 – Elaboración Propia

¿Crees que el tiempo que tardas en buscar plaza tiene un impacto negativo en tu salud mental?

58 respuestas

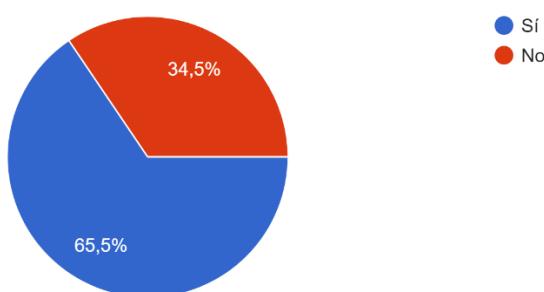


Figura 0.10: Pregunta 9 – Elaboración Propia

¿Influye el estado del parking en tu hora de salida de casa?

58 respuestas

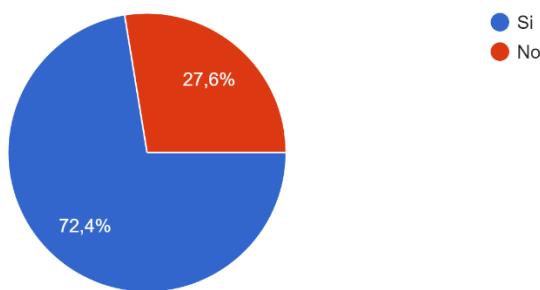


Figura 0.11: Pregunta 10 – Elaboración Propia

¿Alguna vez has ido en transporte público pues creías que no podías encontrar sitio para aparcar?

58 respuestas

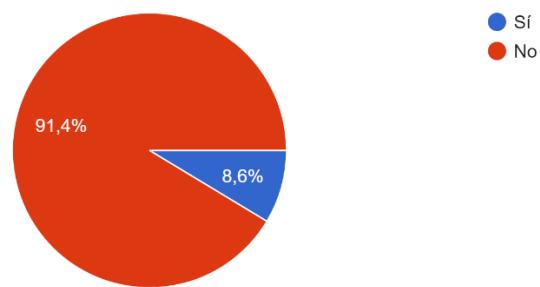


Figura 0.12: Pregunta 11 – Elaboración Propia

Vengo en coche compartido...

¿Qué rol tienes?

4 respuestas

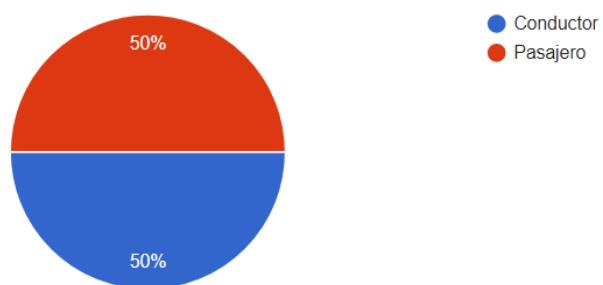


Figura 0.13: Pregunta 12 – Elaboración Propia

Soy Pasajero de Coche Compartido...

¿Utilizas SharingcarUFV?

2 respuestas

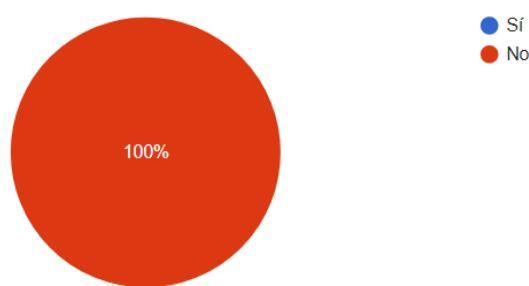


Figura 0.14: Pregunta 13 – Elaboración Propia

¿Cuán lejos de tu edificio tienes que aparcar?

2 respuestas

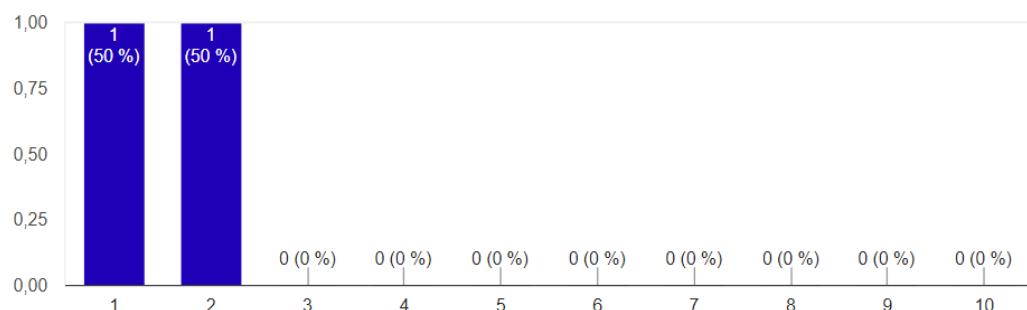


Figura 0.15: Pregunta 14 – Elaboración Propia

¿Tienes problemas para encontrar plaza con el conductor?

2 respuestas

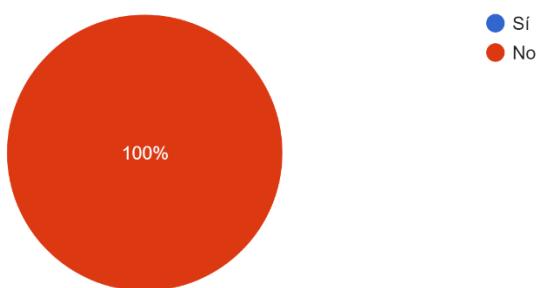


Figura 0.16: Pregunta 15 – Elaboración Propia

¿A que hora sueles tener problemas para encontrar plaza?

2 respuestas



Figura 0.17: Pregunta 16 – Elaboración Propia

¿Te molesta que tengas que gastar más dinero pues el conductor tiene que buscar una plaza y para ello da más vueltas en el parking?

2 respuestas

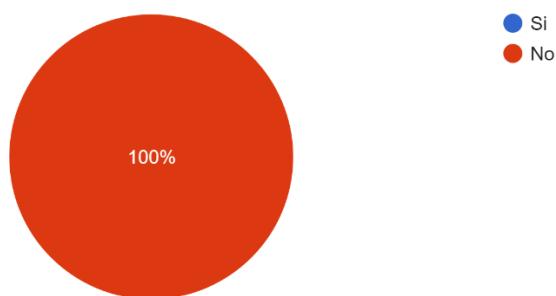


Figura 0.18: Pregunta 17 – Elaboración Propia

¿El buscar plaza junto al conductor justo al llegar a la uni, afecta a tu salud mental?

2 respuestas

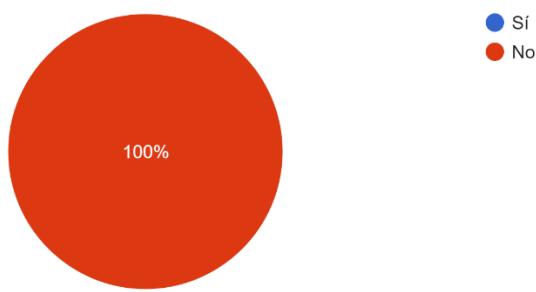


Figura 0.19: Pregunta 18 – Elaboración Propia

Soy Conductor de Coche Compartido...

¿Utilizas SharingcarUFV?

2 respuestas

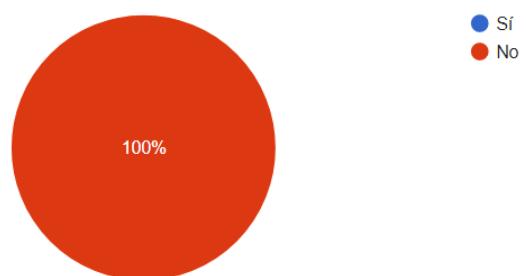


Figura 0.20: Pregunta 19 – Elaboración Propia

Al entrar en el parking, ¿sabes donde quieres aparcar?

2 respuestas

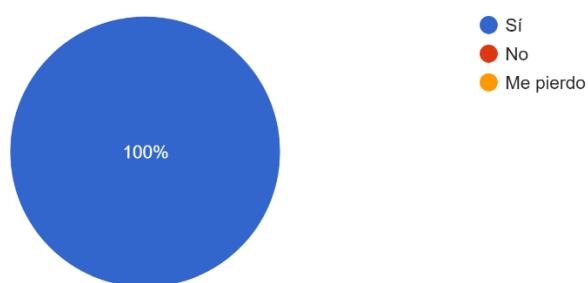


Figura 0.21: Pregunta 20 – Elaboración Propia

¿Cuán lejos de tu edificio tienes que aparcar?

2 respuestas

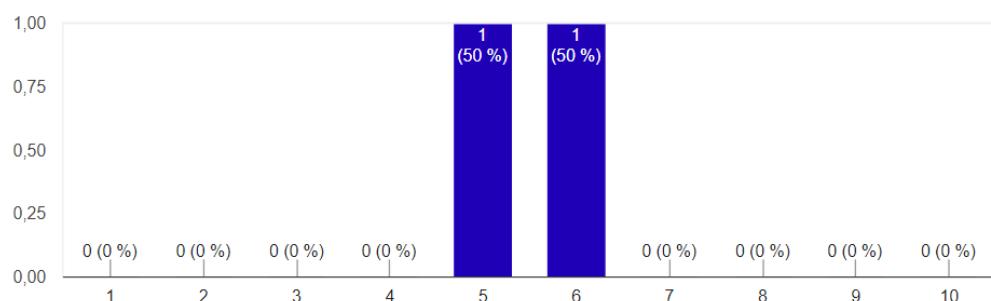


Figura 0.22: Pregunta 21 – Elaboración Propia

¿Cuánto tiempo tardas de media en encontrar plaza?

2 respuestas



Figura 0.23: Pregunta 22 – Elaboración Propia

¿Cuál es el máximo tiempo que estuviste buscando plaza?

2 respuestas



Figura 0.24: Pregunta 23 – Elaboración Propia

¿A qué hora sueles tener problemas para encontrar plaza?

2 respuestas



Figura 0.25: Pregunta 24 – Elaboración Propia

¿Crees que el tiempo que tardas en buscar plaza tiene un impacto negativo en tu salud mental?

2 respuestas

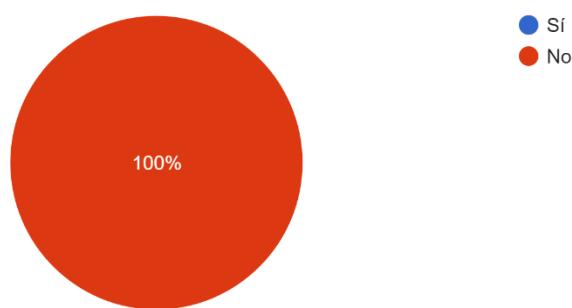


Figura 0.26: Pregunta 25 – Elaboración Propia

¿Influye el estado del parking en tu hora de salida de casa?

2 respuestas

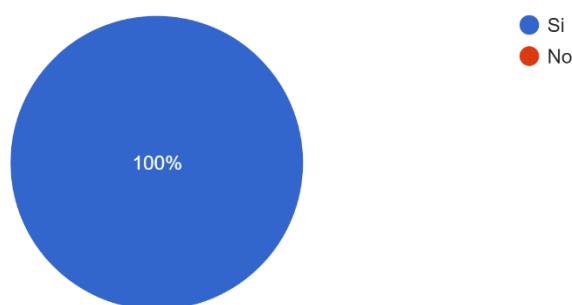


Figura 0.27: Pregunta 26 – Elaboración Propia

¿Alguna vez has ido en transporte público pues creías que no podías encontrar sitio para aparcar?

2 respuestas

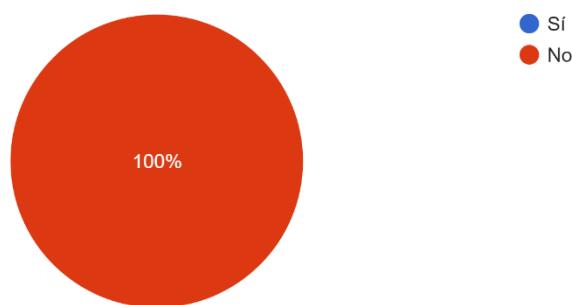


Figura 0.28: Pregunta 27 – Elaboración Propia

Vengo en Transporte Público...

¿Por qué vienes en Transporte Público?

6 respuestas

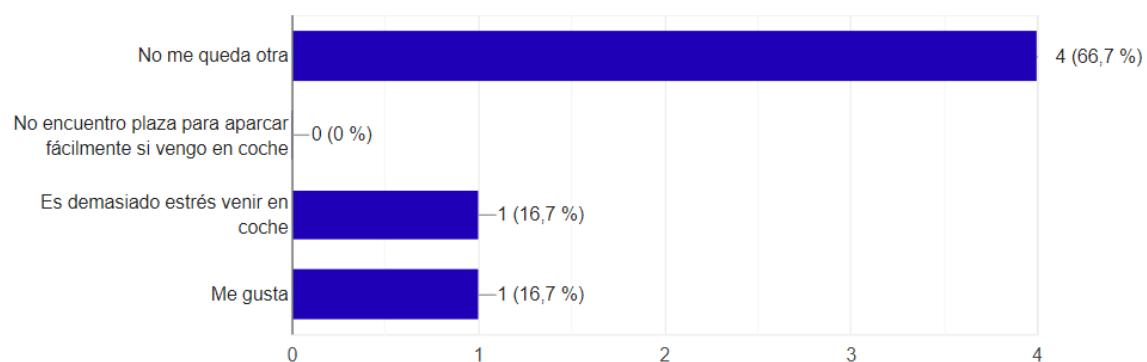


Figura 0.29: Pregunta 28 – Elaboración Propia

¿Has usado alguna vez una aplicación de coche compartido?

6 respuestas

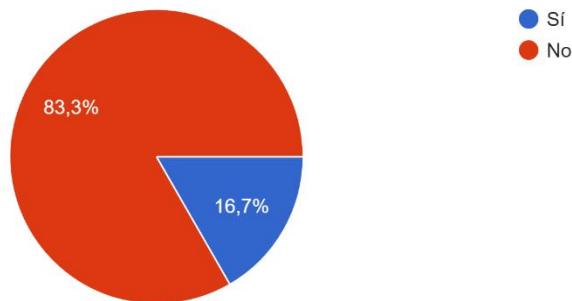


Figura 0.30: Pregunta 29 – Elaboración Propia

¡Ya casi estamos!



¿Sabías que siempre hay plazas disponibles en todo momento en el parking UFV? Según un estudio realizado por los bedeles y los responsables de servicios generales, siempre hay plazas disponibles en todo momento, por muy lleno que esté el parking...

¡Ya casi estamos!

Tranquilo que no te vamos a denunciar ni nada...

¿Alguna vez has aparcado en el césped o en un sitio ilegal?

2 respuestas

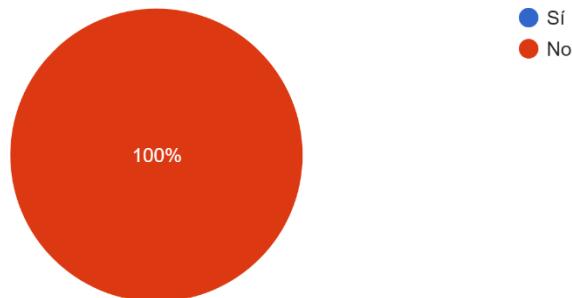


Figura 0.31: Pregunta 30 – Elaboración Propia

Si es así, ¿por qué?

0 respuestas

Aún no hay respuestas para esta pregunta.

Figura 0.32: Pregunta 31 – Elaboración Propia

Estamos llegando al final...

¡Ya casi estamos!

Tranquilo que no te vamos a denunciar ni nada...

¿Alguna vez has aparcado en el césped o en un sitio ilegal?

58 respuestas

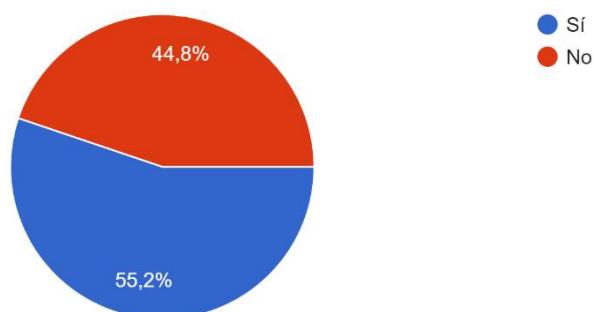


Figura 0.33: Pregunta 32 – Elaboración Propia

Si es así, ¿por qué?

33 respuestas

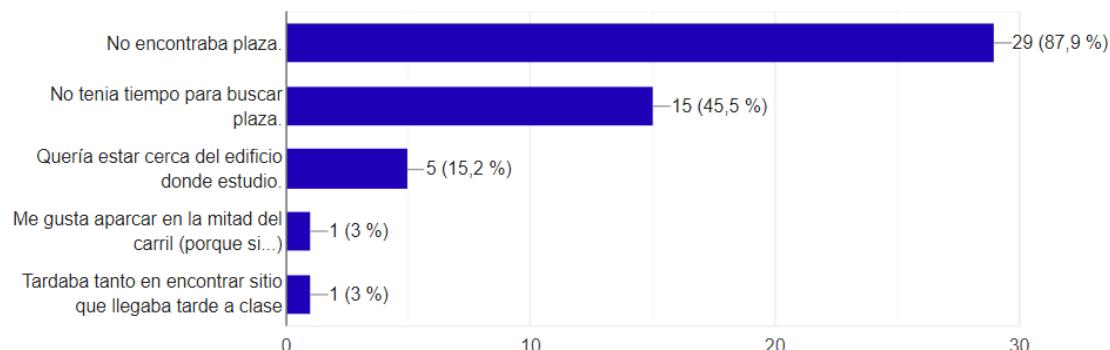


Figura 0.34: Pregunta 33 – Elaboración Propia

Estamos llegando al final...

SharingcarUFV

¿Sabías que la UVF tiene una aplicación para compartir coche? Puedes encontrar gente que te lleve a la universidad por un precio justo y con comodidad sin tener que coger transporte público.

¿Por qué nunca utilizaste la app?

5 respuestas

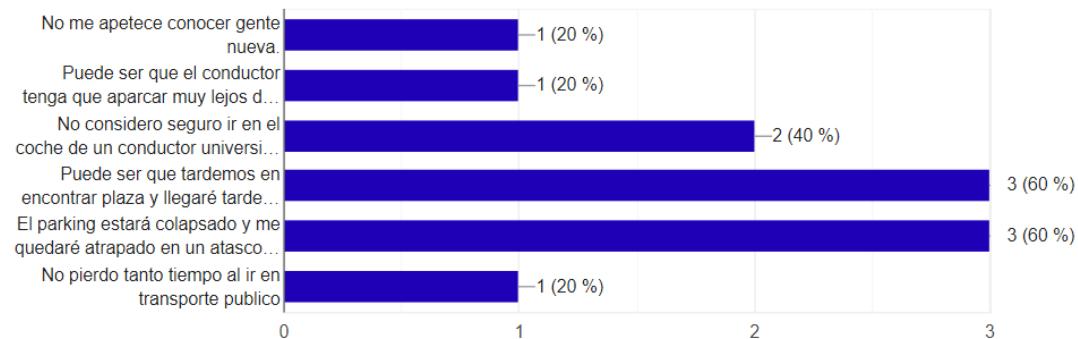


Figura 0.35: Pregunta 34 – Elaboración Propia

El gran finale...

¿Utilizarías la aplicación de SharingcarUFV si pudieras aparcar al lado de tu edificio en un parking reservado solo para CarSharing?

6 respuestas

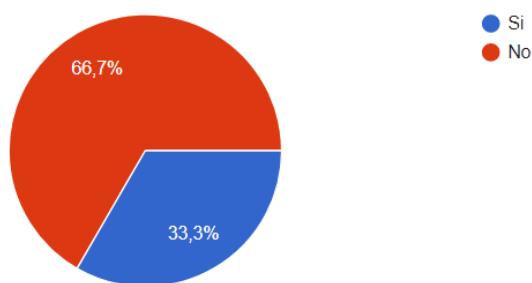


Figura 0.36: Pregunta 35 – Elaboración Propia

¿Qué impactos positivos crees que tendría en tu vida?

6 respuestas

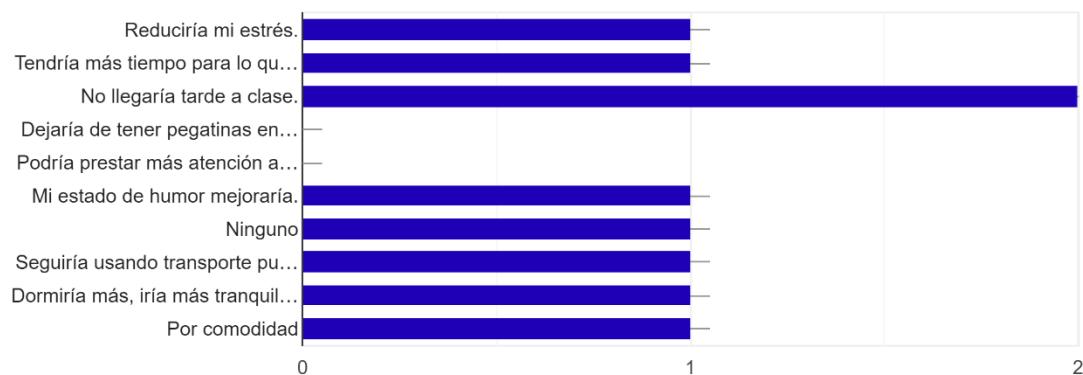


Figura 0.37: Pregunta 36 – Elaboración Propia

Si crees que tendría algún impacto negativo puedes escribirlo aquí...

0 respuestas

Aún no hay respuestas para esta pregunta.

Figura 0.38: Pregunta 37 – Elaboración Propia

El gran finale...

¿Te gustaría tener una Web/App que te indicara donde hay una plaza libre para ti nada más entrar al parking?

4 respuestas

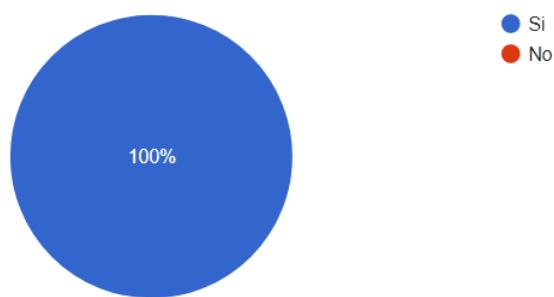


Figura 0.39: Pregunta 38 – Elaboración Propia

¿Qué impactos positivos crees que tendría en tu vida?

4 respuestas

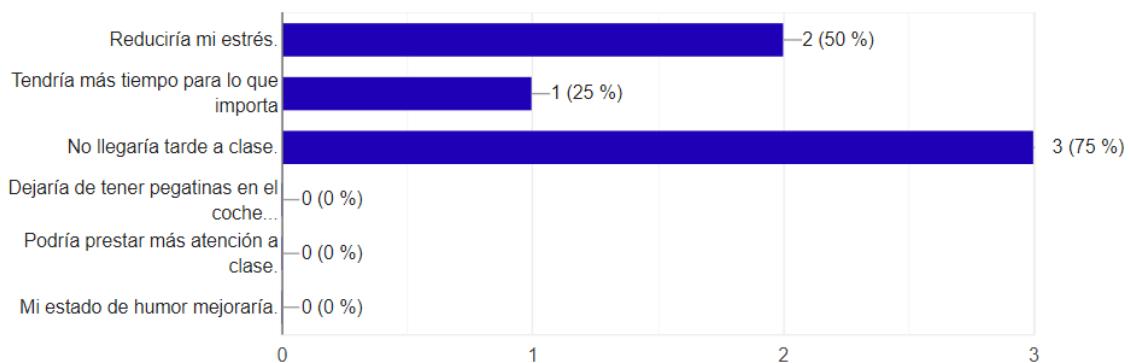


Figura 0.40: Pregunta 39 – Elaboración Propia

Si crees que tendría algún impacto negativo puedes escribirlo aquí...

0 respuestas

Aún no hay respuestas para esta pregunta.

Figura 0.41: Pregunta 40 – Elaboración Propia

Si es tu caso ¿Dejarías de aparcar ilegalmente entonces?

1 respuesta

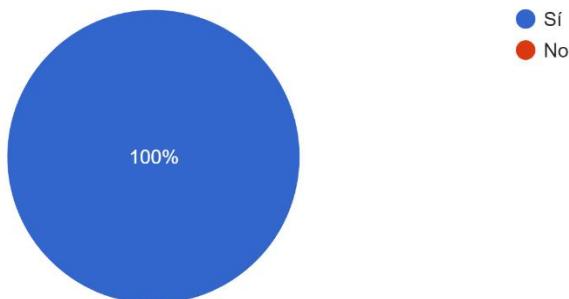


Figura 0.42: Pregunta 41 – Elaboración Propia

¿Te gustaría poder aparcar en un parking que te permitiera conocer su estado actual en tiempo real antes de salir de casa?

4 respuestas

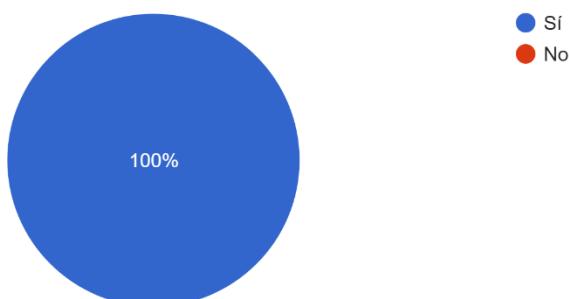


Figura 0.43: Pregunta 42 – Elaboración Propia

Si no, ¿por qué?

0 respuestas

Aún no hay respuestas para esta pregunta.

Figura 0.44: Pregunta 43 – Elaboración Propia

El gran final...



El tiempo no para y no puede ir hacia atrás, así que te planteo una solución:

¿Te gustaría tener una Web/App que te indicara donde hay una plaza libre para ti nada más entrar al parking?

58 respuestas

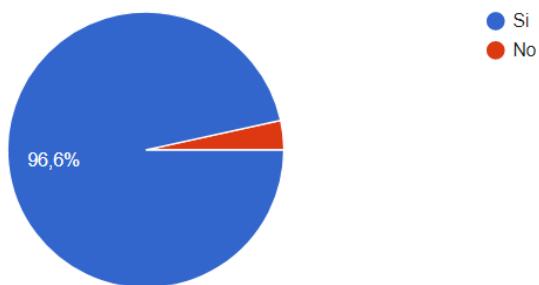


Figura 0.45: Pregunta 44 – Elaboración Propia

¿Qué impactos positivos crees que tendría en tu vida?

58 respuestas

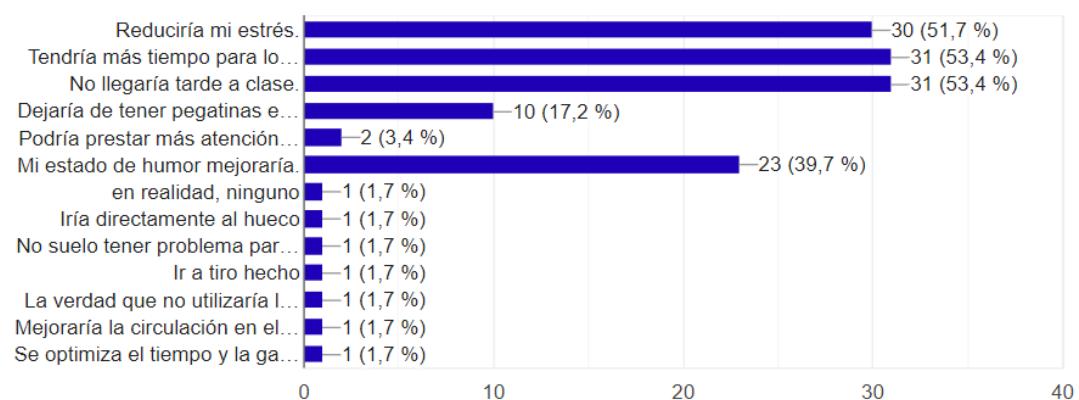


Figura 0.46: Pregunta 45 – Elaboración Propia

Si crees que tendría algún impacto negativo puedes escribirlo aquí...

7 respuestas

Quizá sea más difícil encontrar plaza porque puede motivar más uso del coche por otros.

Incremento de choques y accidentes por distracciones para consultarla

Conduciendo no deberías usar el móvil y en el trayecto la plaza puede variar. Lo suyo sería reservar la plaza antes de salir de casa pero... como la bloqueas para que nadie aparque?

Dado que considero que no es cierto que siempre haya plazas libres, que 50 vehículos busquen plaza al mismo tiempo cuando no hay plazas o cuando hay pocas, provocaría que no se utilizara la aplicación por considerarla inútil. Este sistema ya está inventado y se utiliza en muchos parkings, donde se te informa por secciones zonas el número de plazas libres en tiempo real.

no

No estoy muy seguro de que la app sea una solución operativa si vas conduciendo tú solo. Si no se plantea de manera que te oriente con una locución, prestar atención a la pantalla de tu móvil, si no se dispone de un soporte adecuado, puede aumentar el riesgo de accidentes en el campus

Mirar el móvil mientras se conduce para buscar sitio libre

Figura 0.47: Pregunta 46 – Elaboración Propia

Si es tu caso ¿Dejarías de aparcar ilegalmente entonces?

37 respuestas

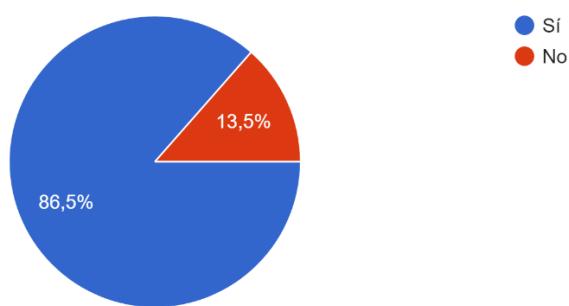


Figura 0.48: Pregunta 47 – Elaboración Propia

¿Utilizarías una aplicación de coche compartido como SharingcarUFV, si supieras que podrás aparcar sin problemas si te haces conductor?

58 respuestas

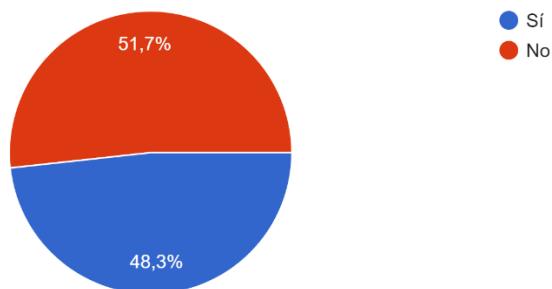


Figura 0.49: Pregunta 48 – Elaboración Propia

¿Te gustaría poder aparcar en un parking que te permitiera conocer su estado actual en tiempo real antes de salir de casa?

58 respuestas

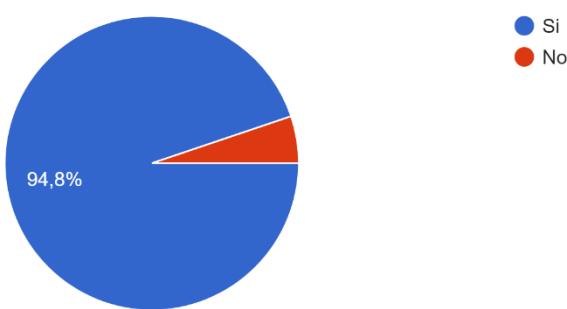


Figura 0.50: Pregunta 49 – Elaboración Propia

Si no, ¿por qué?

5 respuestas

Comodidad

No me es necesario

Tardo 20 minutos, en ese tiempo el estado de plazas puede haber cambiado mucho.

Mi trayecto es de 45 minutos, por lo que no me aportaría nada

Para ganar

Figura 0.51: Pregunta 50 – Elaboración Propia

That's all folks!

Valora esta encuesta

68 respuestas

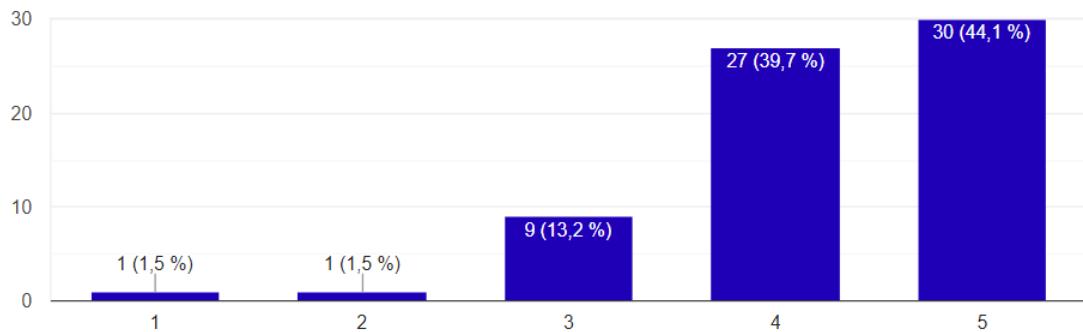


Figura 0.52: Pregunta 51 – Elaboración Propia

ANEXO F: BOCETOS Y DETALLES DEL DISEÑO DEL SISTEMA

INVESTIGACIÓN PREVIA

Parking UFV

Aquí se detallan los bocetos realizados por el alumno con el objetivo de planificar el sistema. Ellos se incluyen en la memoria con el objetivo de exponer el trabajo realizado por el alumno en el diseño del sistema y planificación del proyecto. Los bocetos son informales, pero exponen el trabajo y las notas realizadas por el alumno.

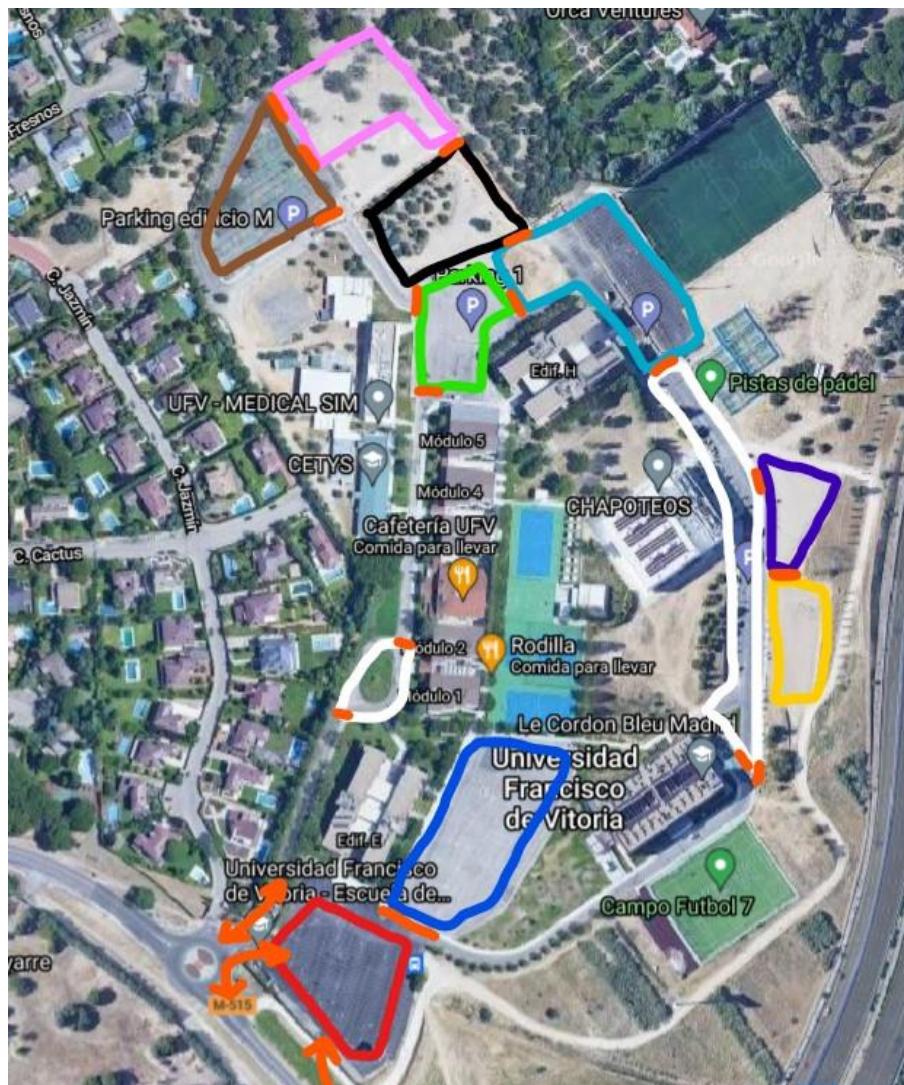


Figura 0.1: Boceto para la Elección del Parking - Elaboración Propia sobre Google Maps

Tipos de Plaza

Se tomaron medidas de las plazas del parking elegido.

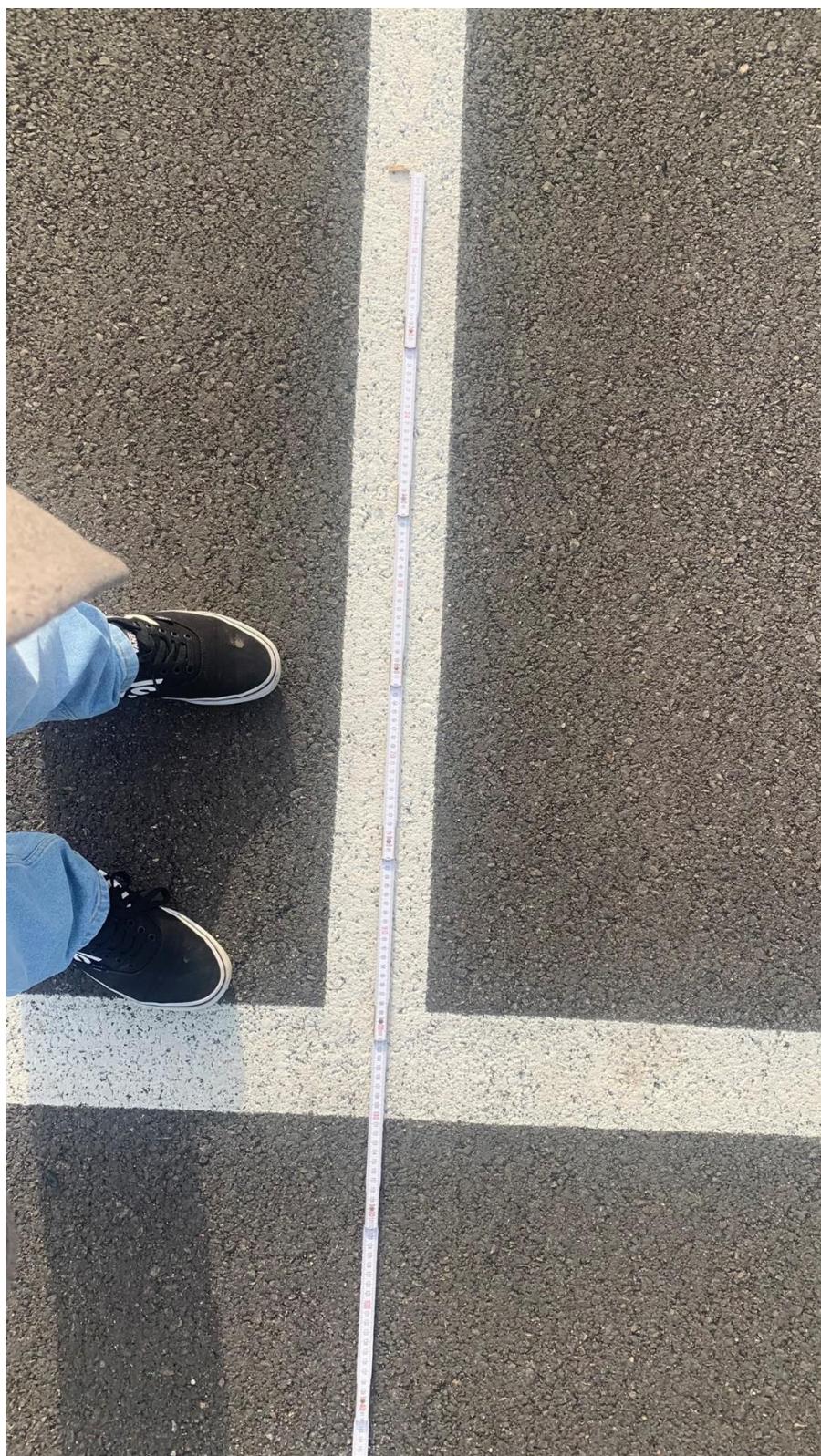


Figura 0.2: Toma de Medidas de las Plazas – Elaboración Propia

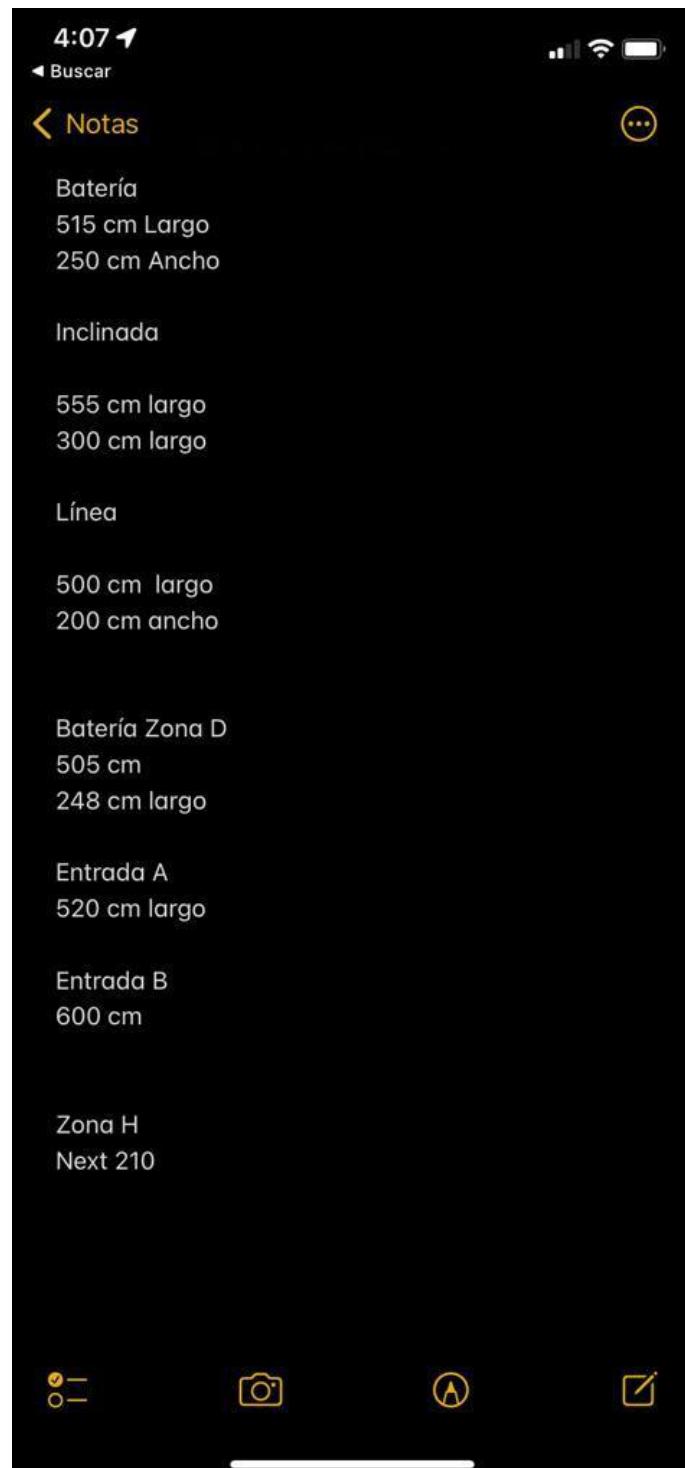


Figura 0.3: Medidas de los Tipos de Plaza – Elaboración Propia

Número de Plazas y Arquitectura

Se contó el número de plazas de aparcamiento de un aparcamiento de la Universidad Francisco de Vitoria, por la noche.



Figura 0.4: Foto Nocturna del Parking para el Conteo de Plazas I – Elaboración Propia

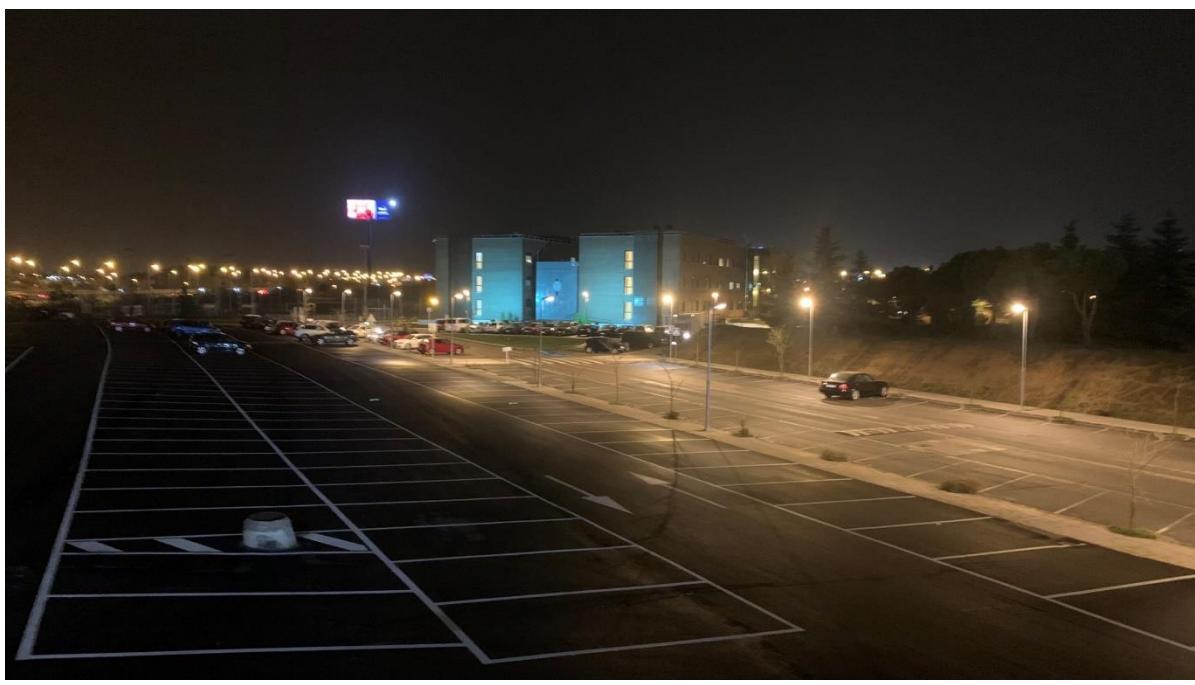


Figura 0.5: Foto Nocturna del Parking para el Conteo de Plazas II – Elaboración Propia

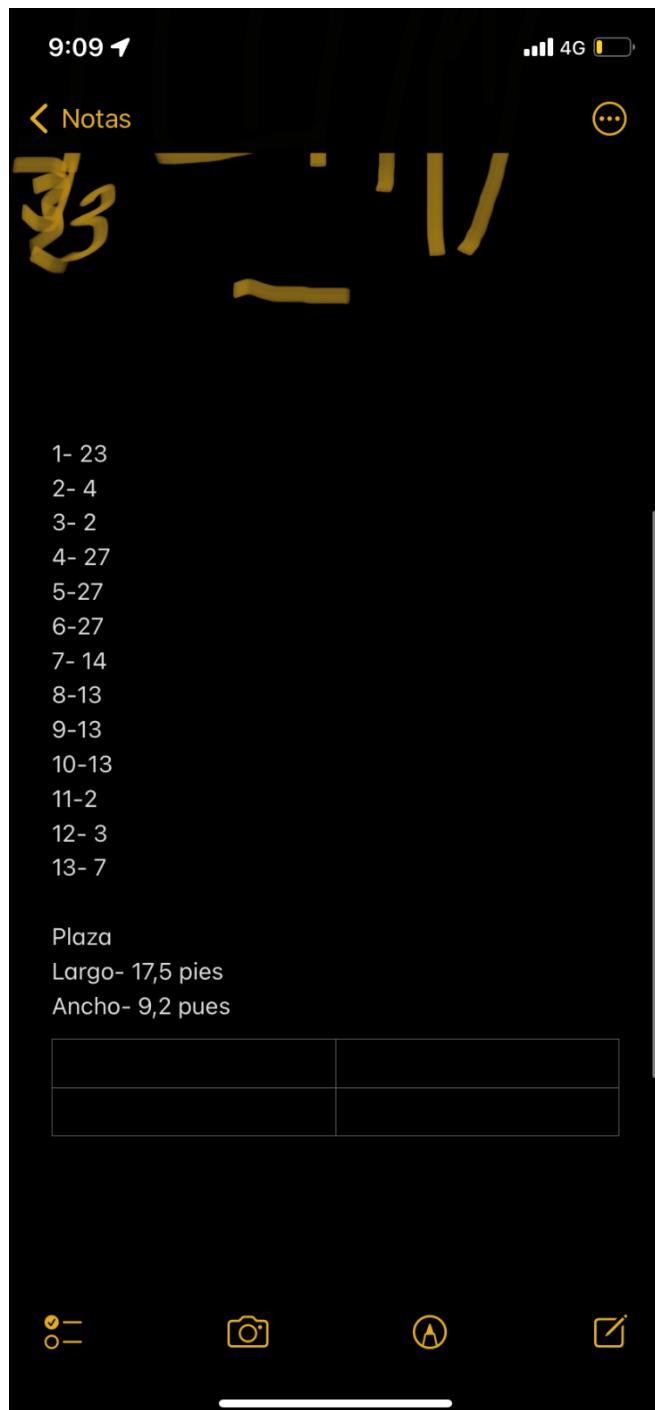


Figura 0.6: Notas sobre número de plazas

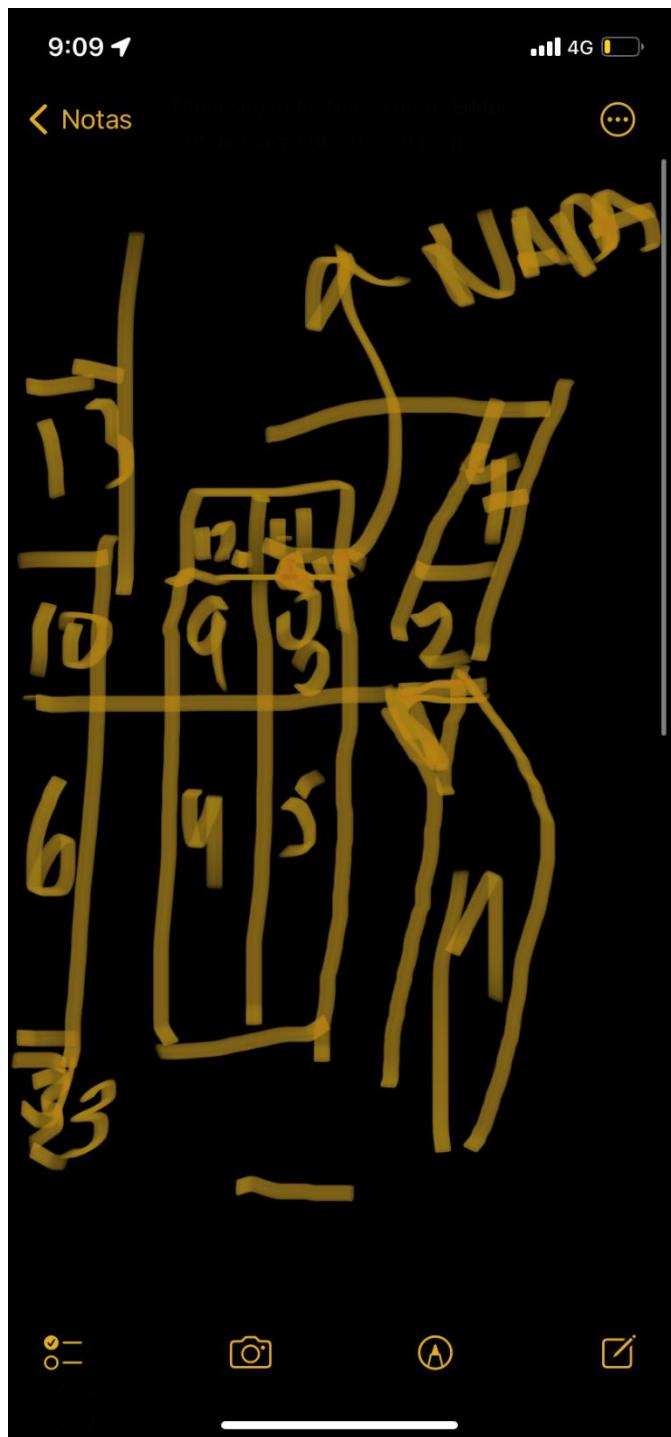


Figura 0.7: Boceto de la Arquitectura del Aparcamiento– Elaboración Propia

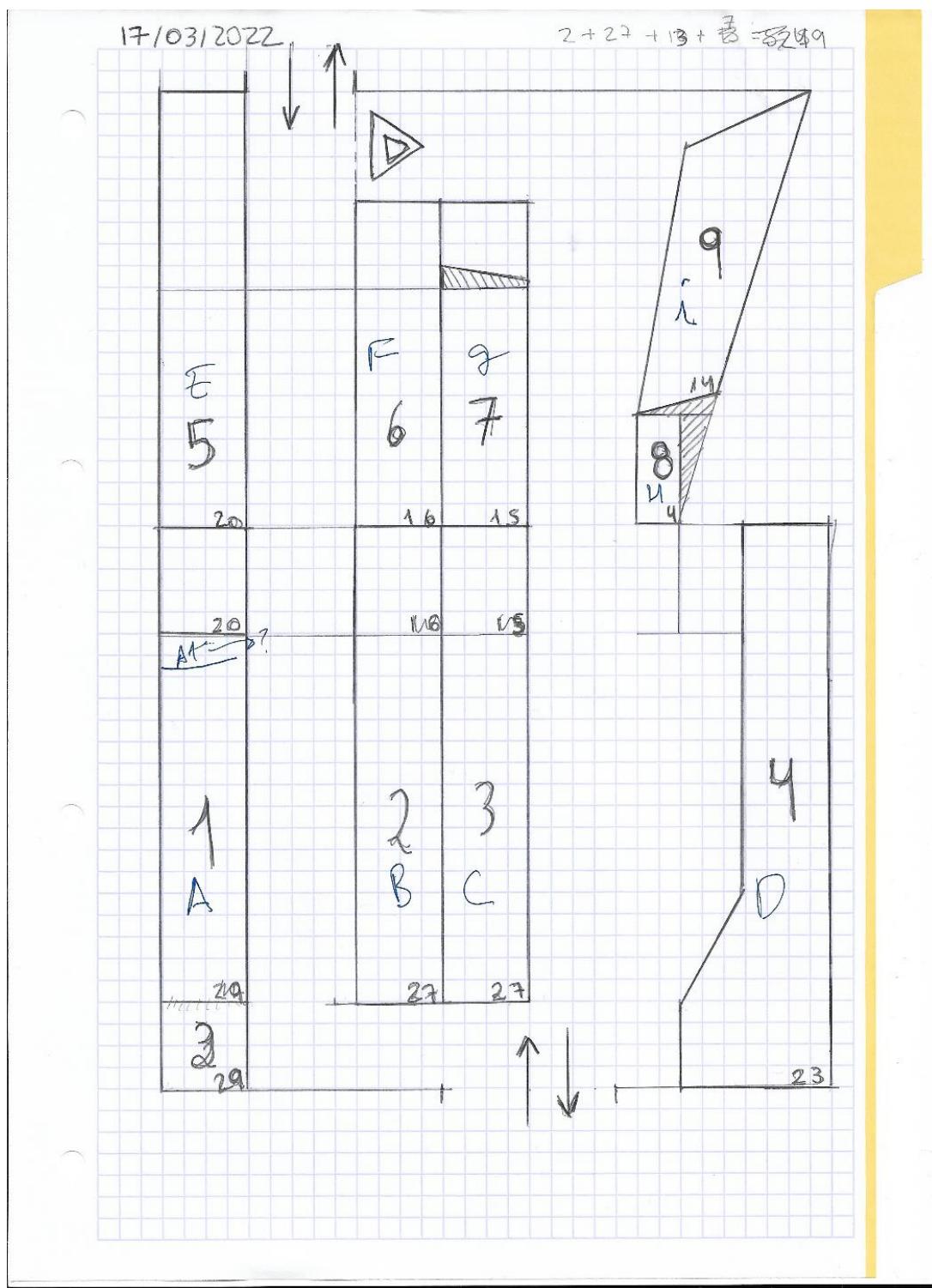


Figura 0.8: Boceto Versión 1.0 del mapeo de la Zona "Y". – Elaboración Propia

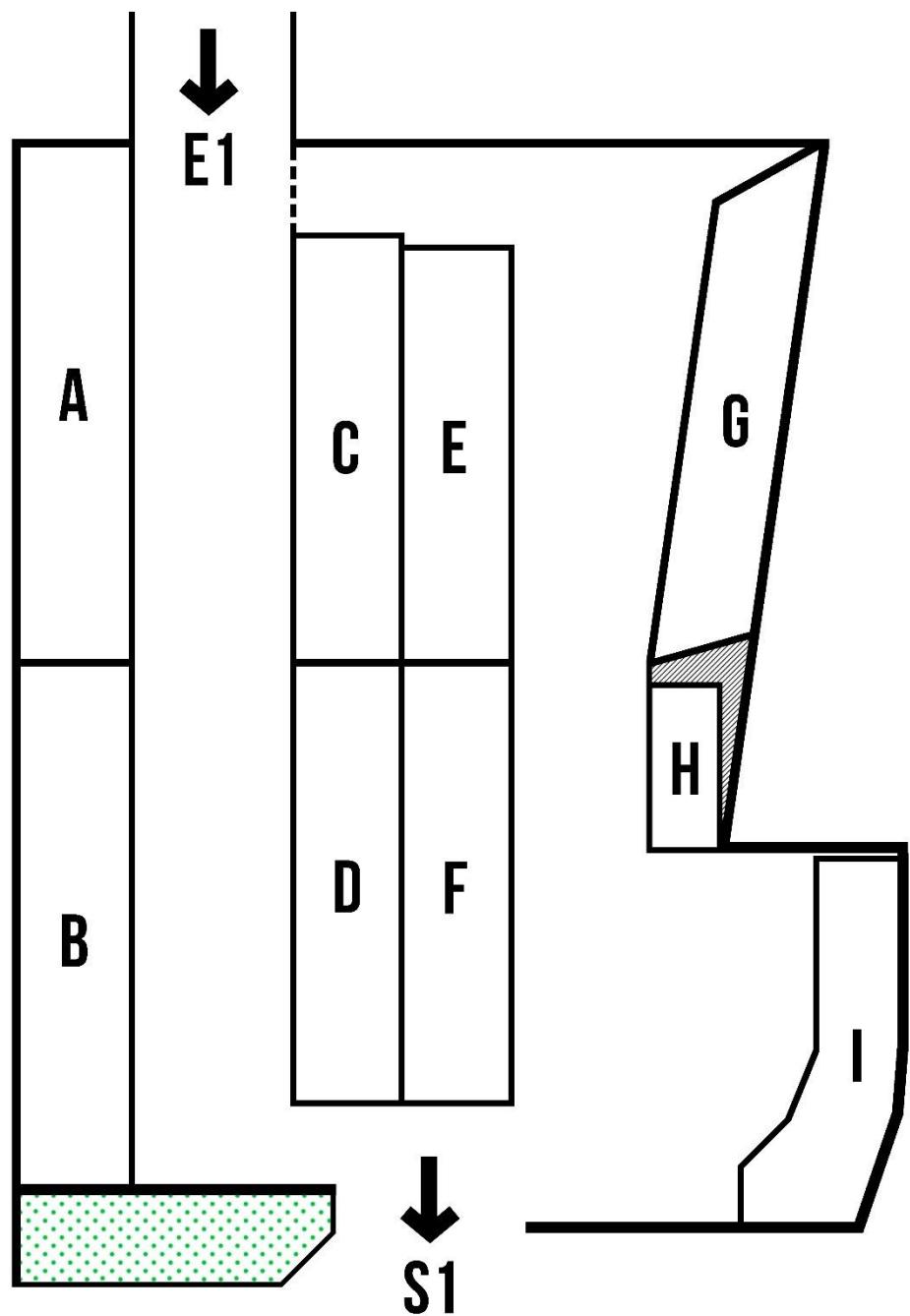


Figura 0.9: Versión 2.0 – Mapa Zona Y– Elaboración Propia

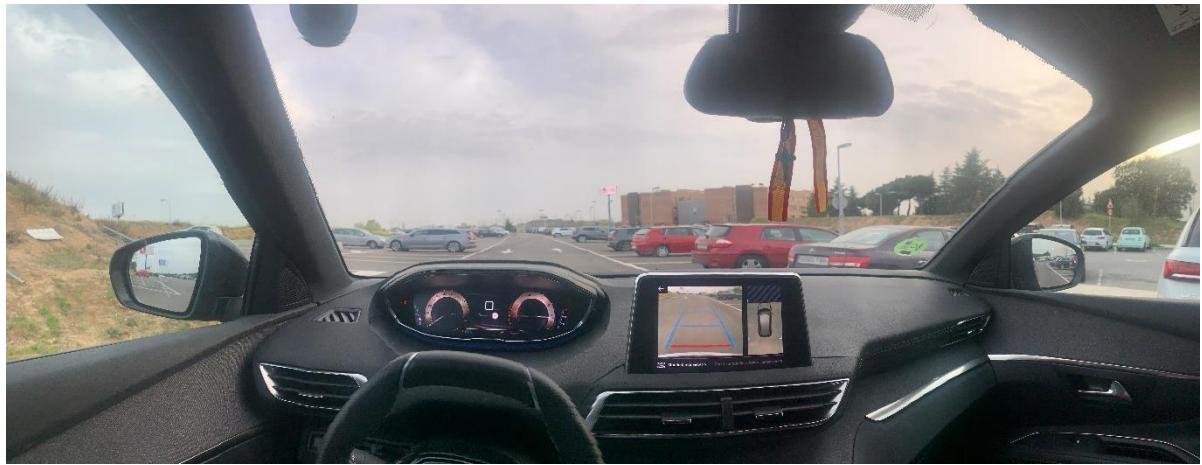


Figura 0.10: Foto Interior Vehículo en Entrada E1 del Mapa Zona Y – Elaboración Propia



Figura 0.11: Foto Interior Vehículo en Salida S1 del Mapa Zona Y – Elaboración Propia

DEFINICIÓN DE ALCANCE

Creado: 10/03/2022
Dúdar (Refran)

S N

1 - ¿Tienen que estar registrados para acceder al portal? <input checked="" type="checkbox"/>	<input type="checkbox"/>
2 - ¿Las barreras estén siempre cerradas? <input checked="" type="checkbox"/>	<input type="checkbox"/>
3 - Estimado el tiempo en los trámites? <input type="checkbox"/>	<input checked="" type="checkbox"/>
4 - Envío carta hacia plaza <input type="checkbox"/>	<input checked="" type="checkbox"/>
5 - Formulario Novena <input type="checkbox"/>	<input checked="" type="checkbox"/>
6 - Notif. al cliente <input checked="" type="checkbox"/>	<input type="checkbox"/>
7 - Envío notificación de alta <input checked="" type="checkbox"/>	<input type="checkbox"/>

Notificación por correo

Figura 0.12: Boceto para la elección de alcance– Elaboración Propia

DISEÑO DE LA ARQUITECTURA

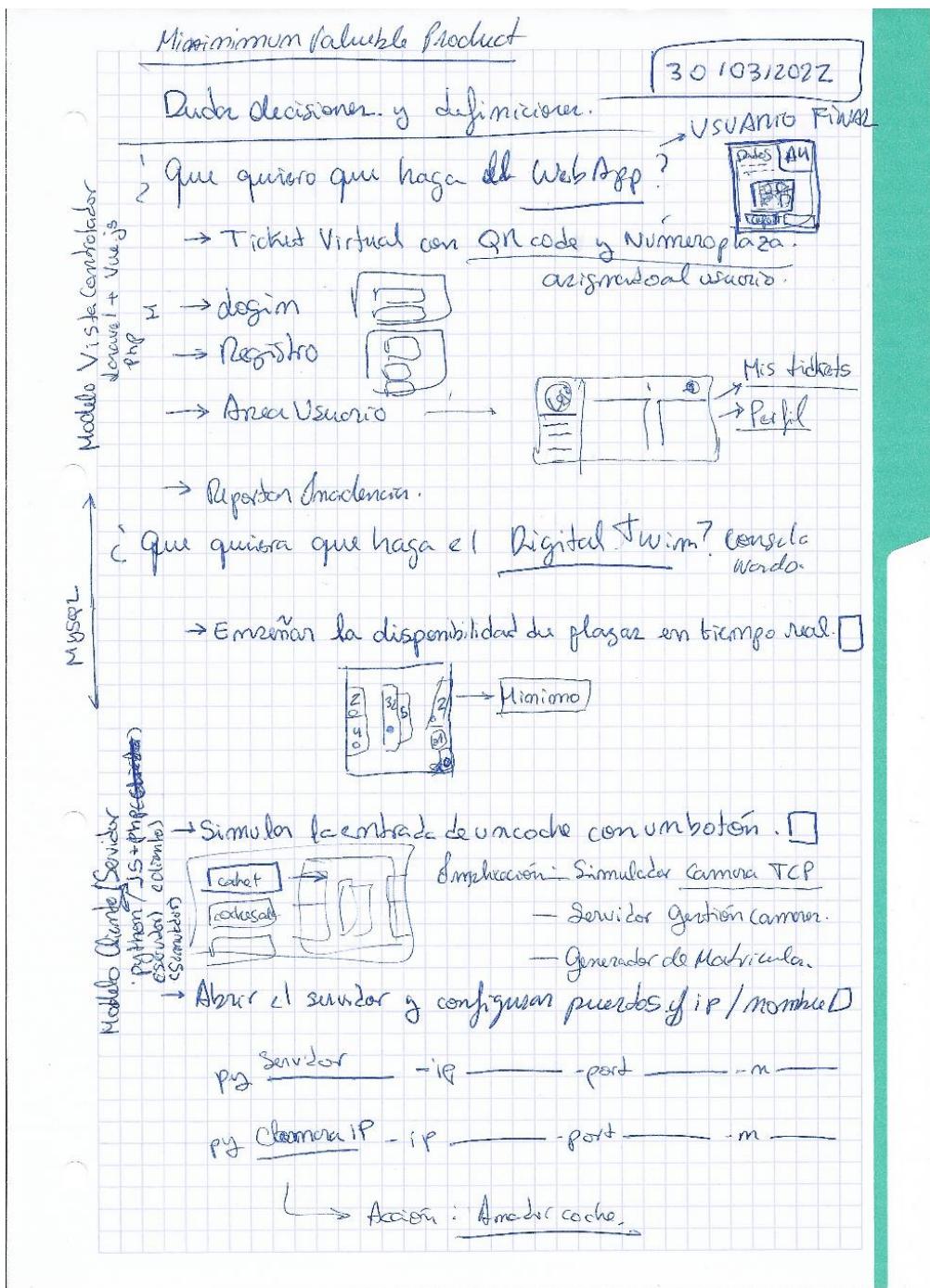


Figura 0.13: Boceto para la configuración y decisión– Elaboración Propia

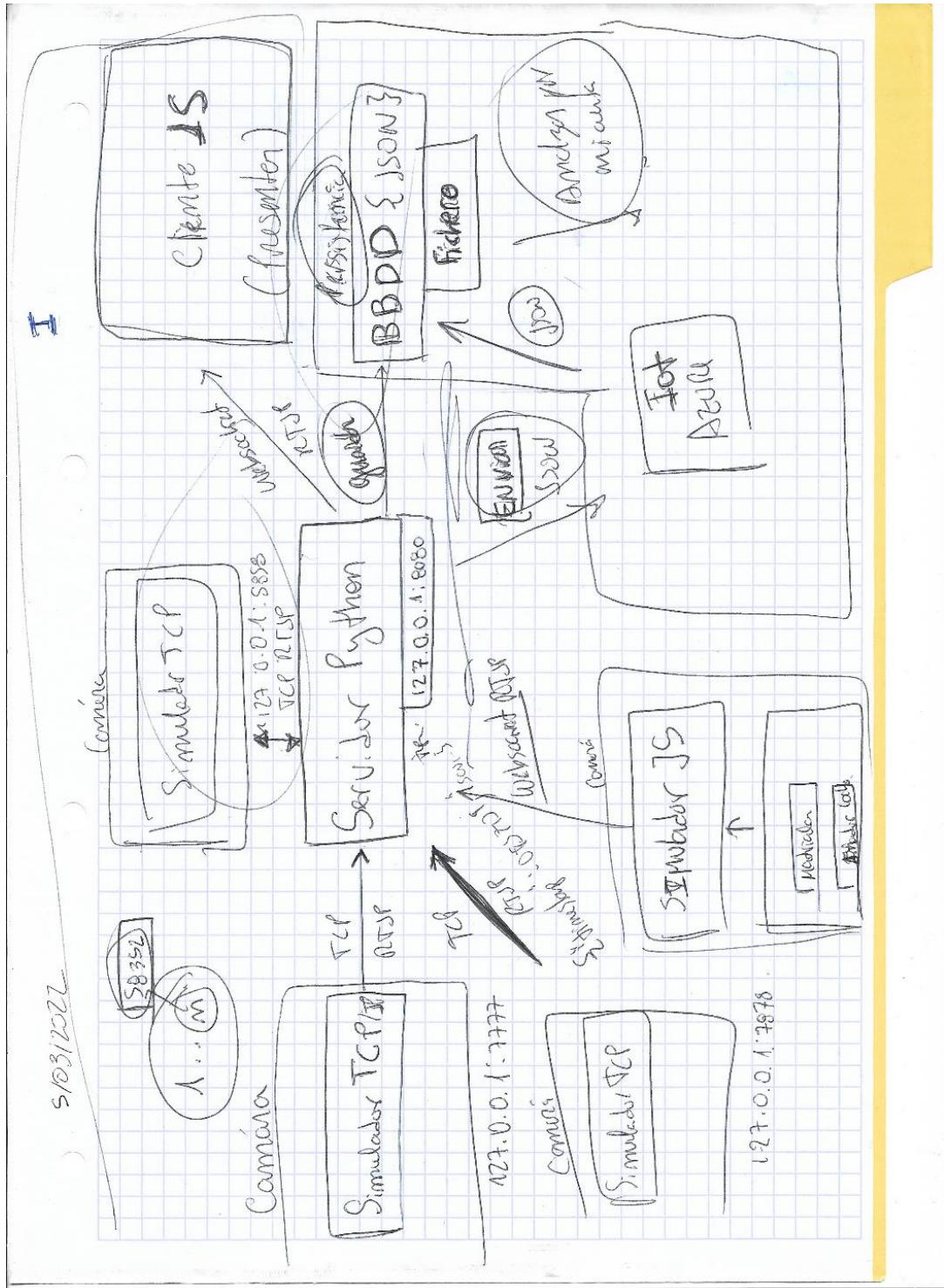


Figura 0.14: Boceto de arquitectura del sistema– Elaboración Propia

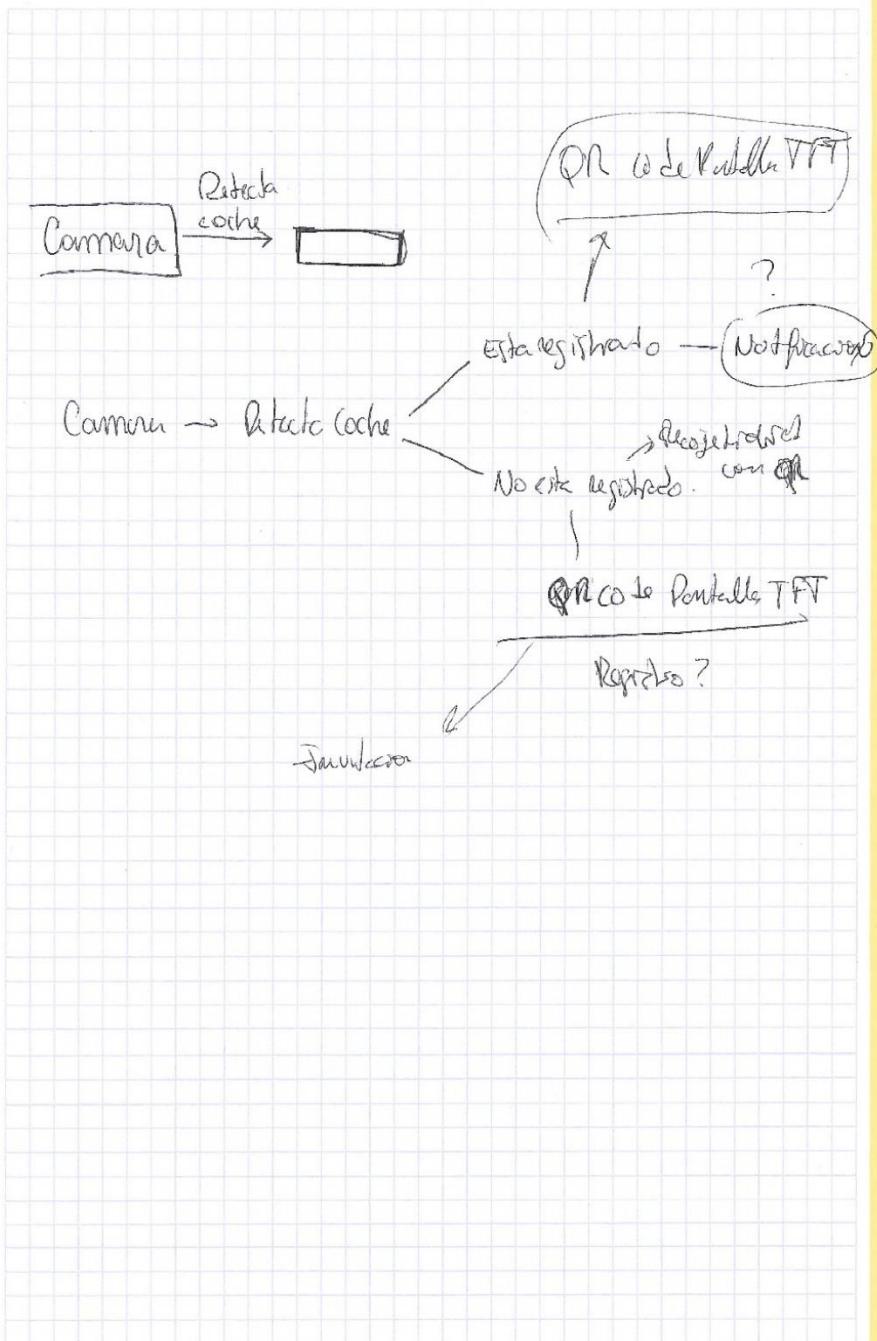


Figura 0.15: Boceto de Decisiones del Sistema– Elaboración Propia

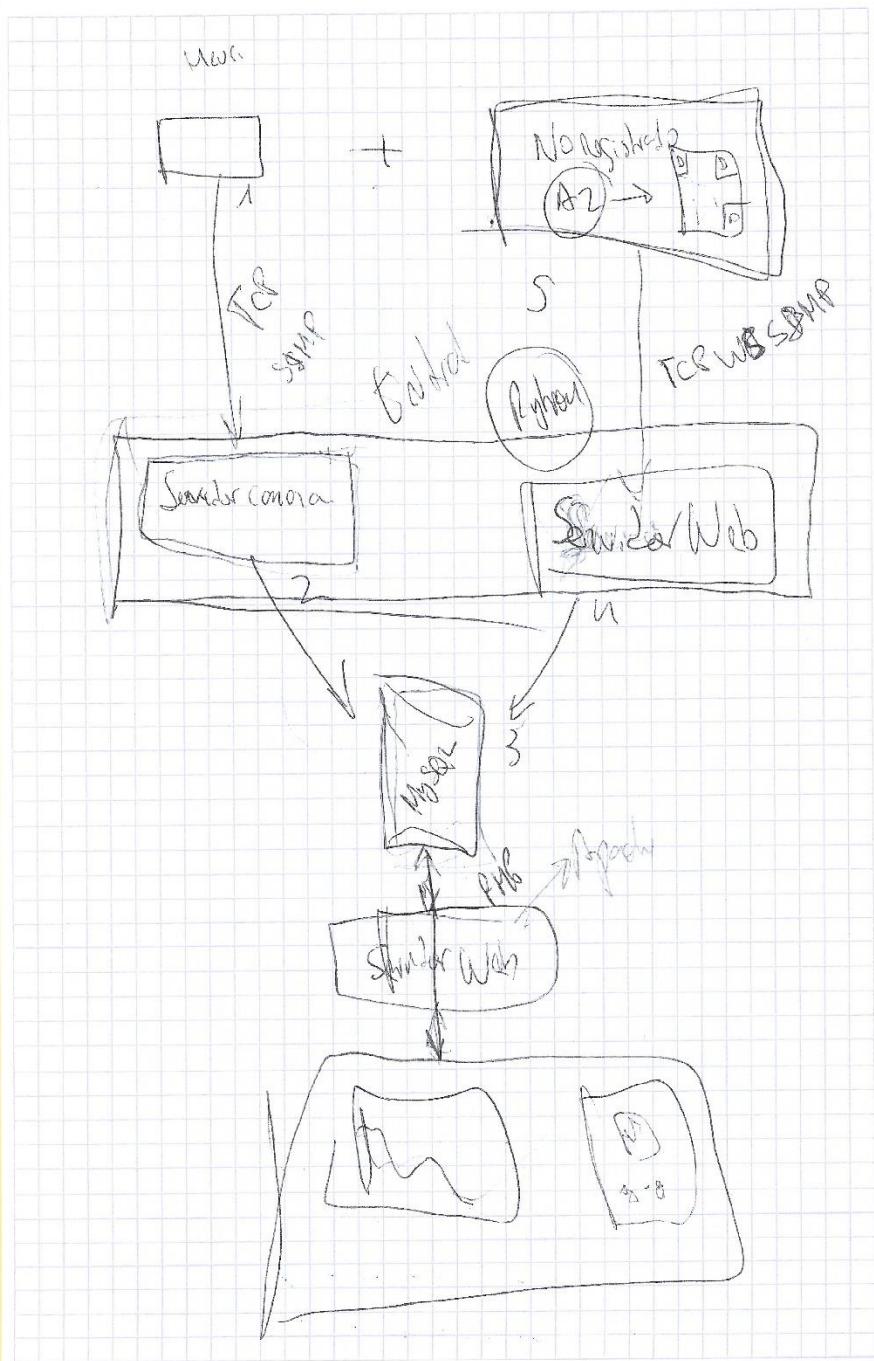


Figura 0.16: Boceto de Flujo de Sistema– Elaboración Propia

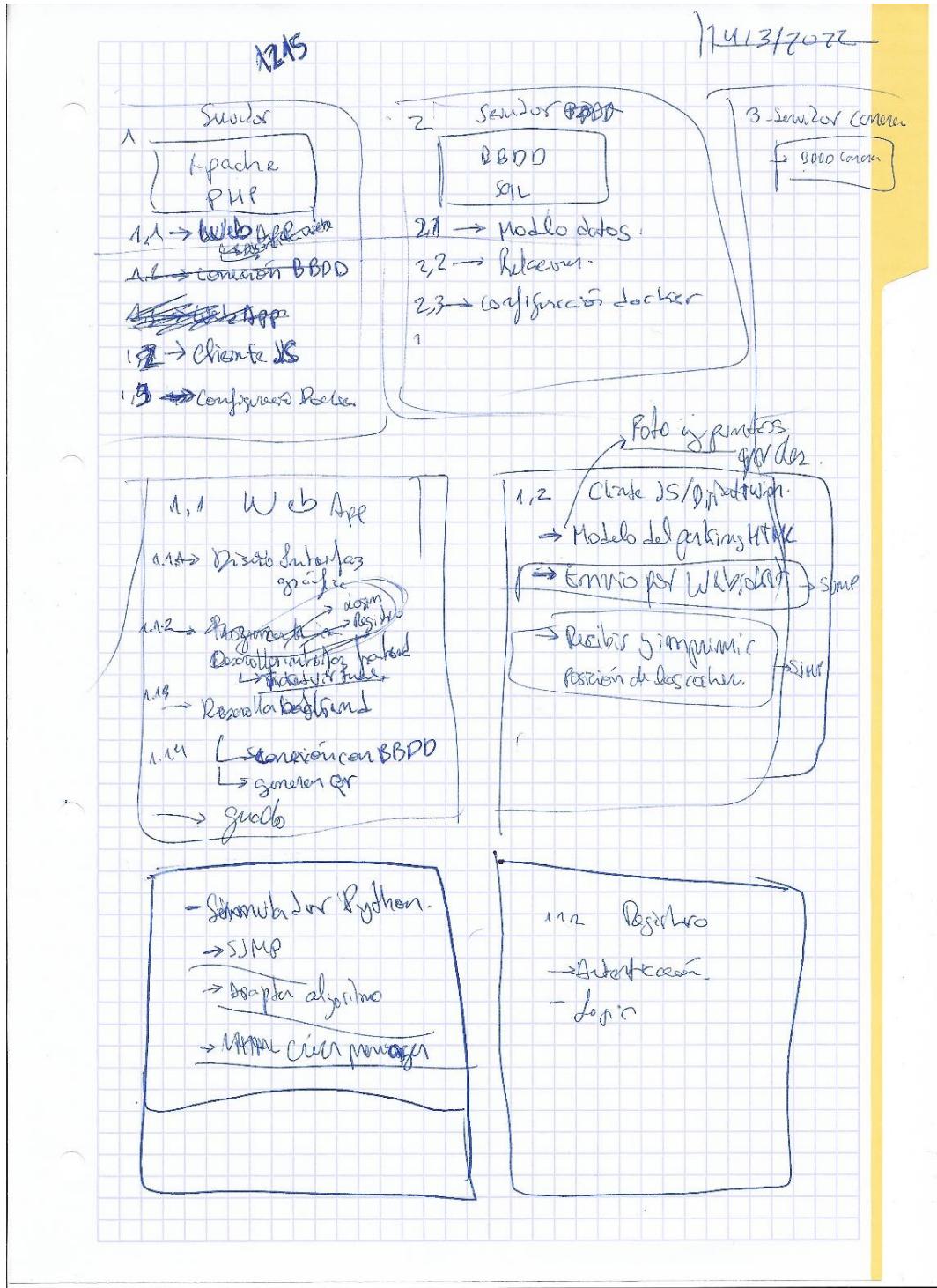


Figura 0.17: Boceto para definición de Componentes del Sistema – Elaboración Propia

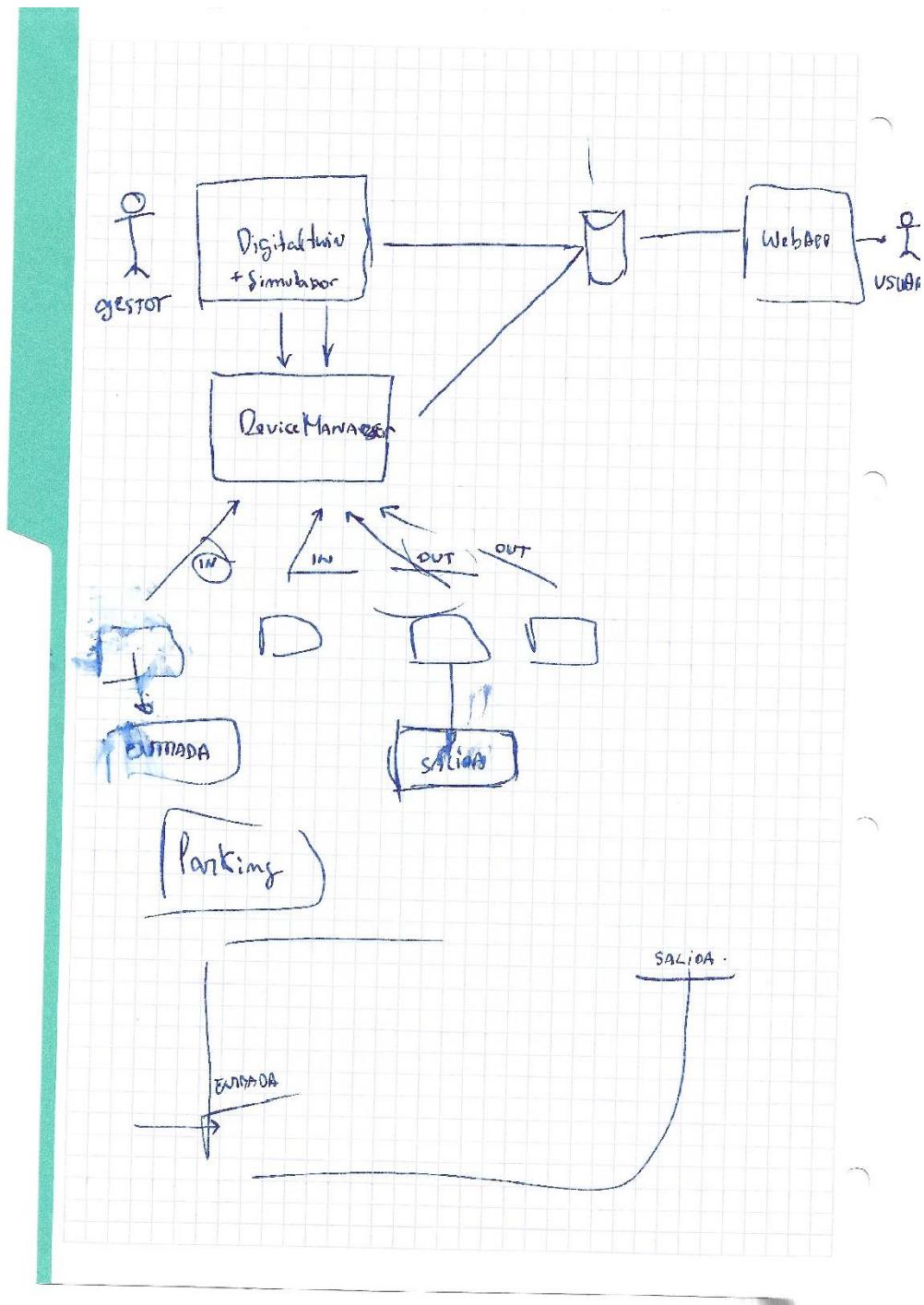


Figura 0.18: Boceto de Arquitectura– Elaboración Propia

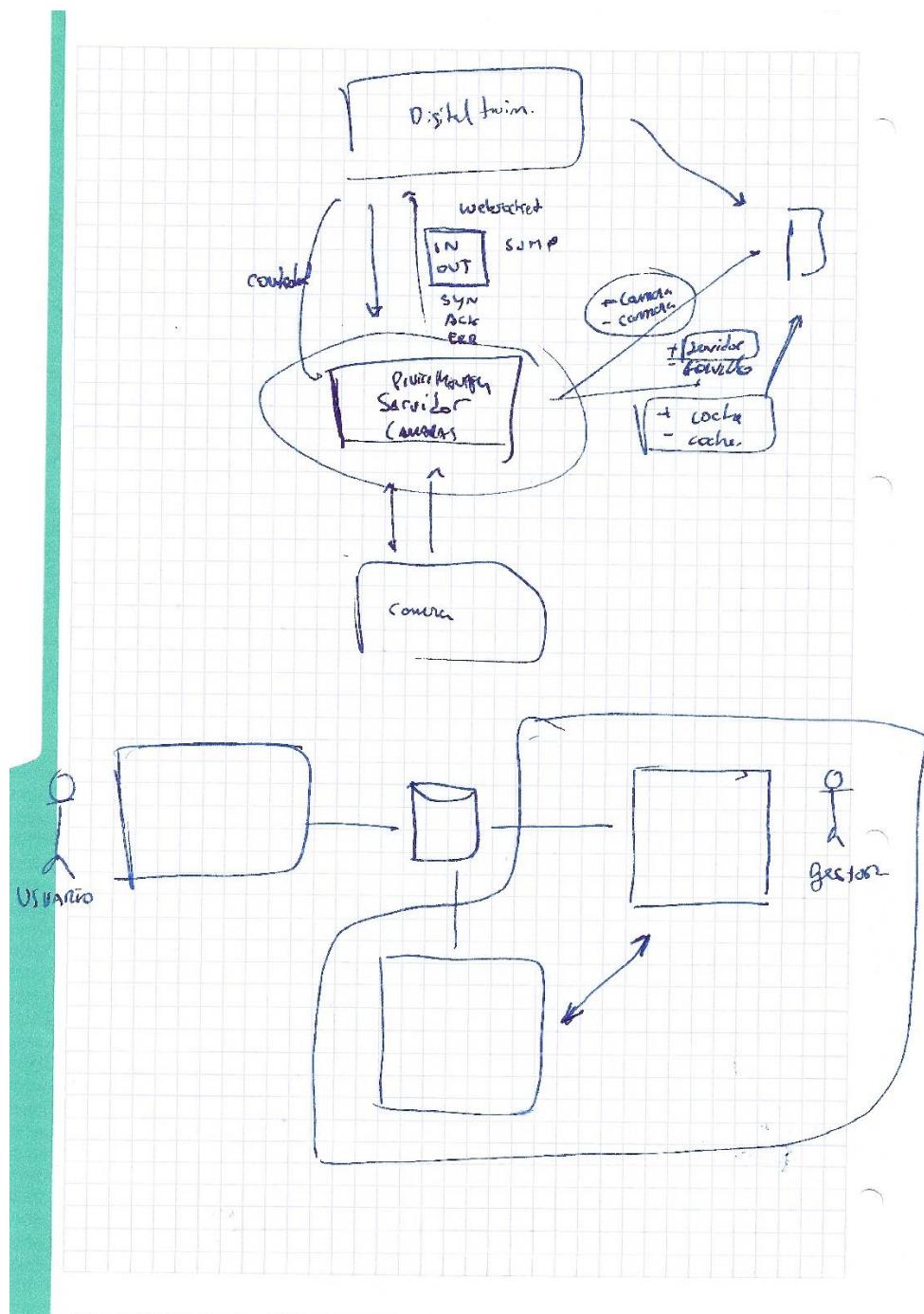


Figura 0.19: Boceto de Diseño de la Arquitectura– Elaboración Propia

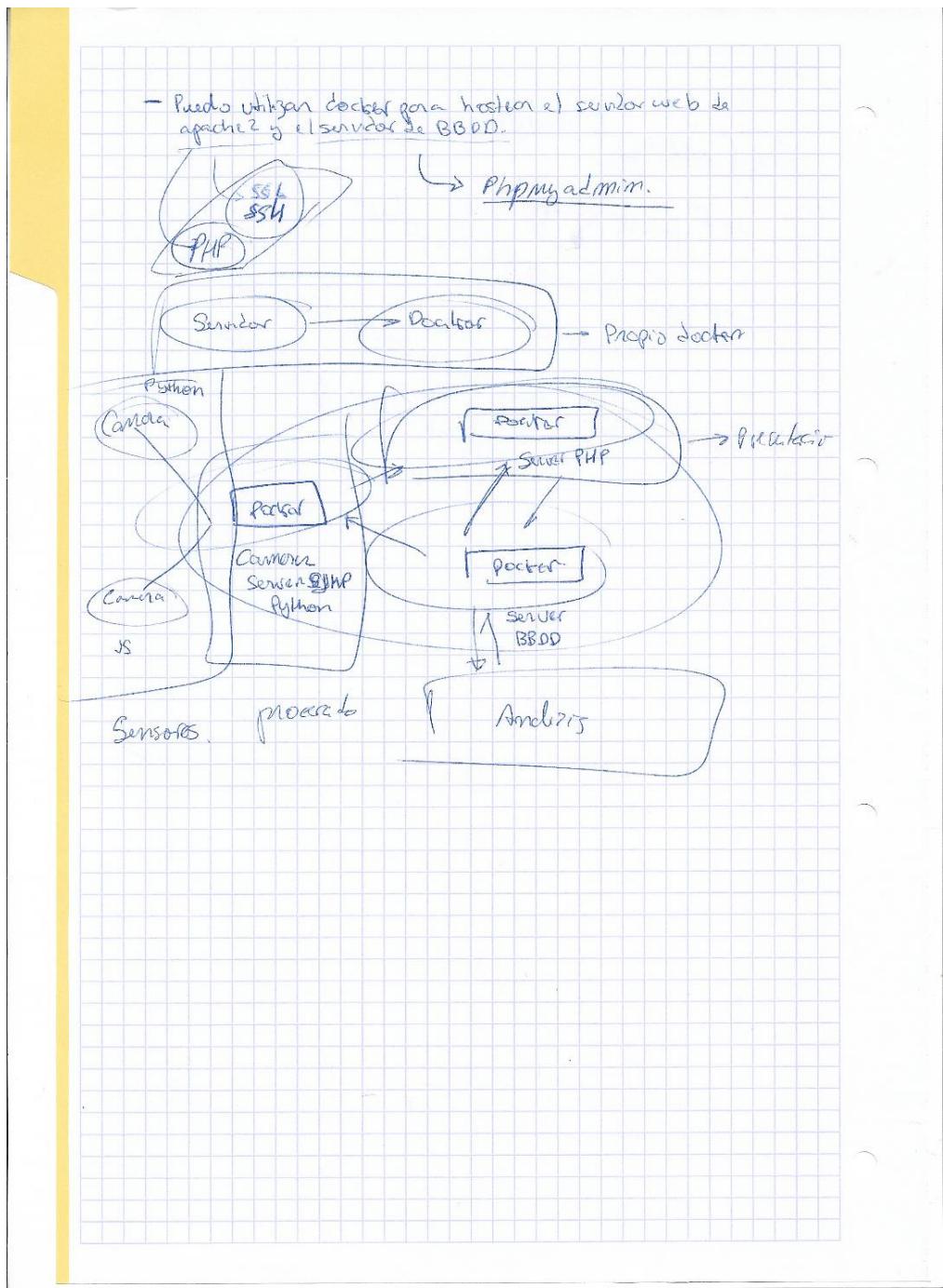


Figura 0.20: Boceto para la decisión de la estructura principal– Elaboración Propia

Diseño y Integración de Tecnologías

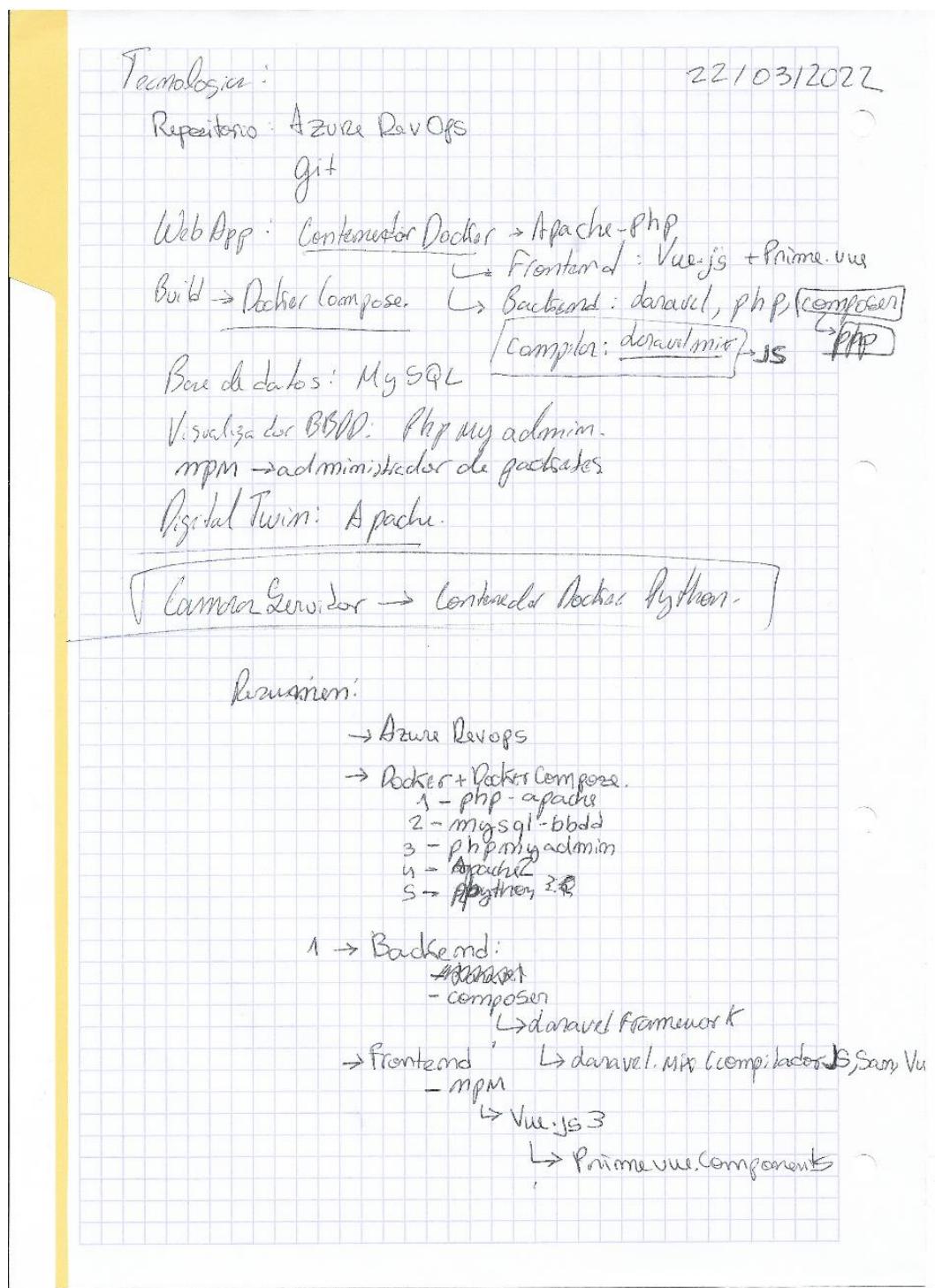


Figura 0.21: Definición de tecnologías. – Elaboración Propia

22/03/2022

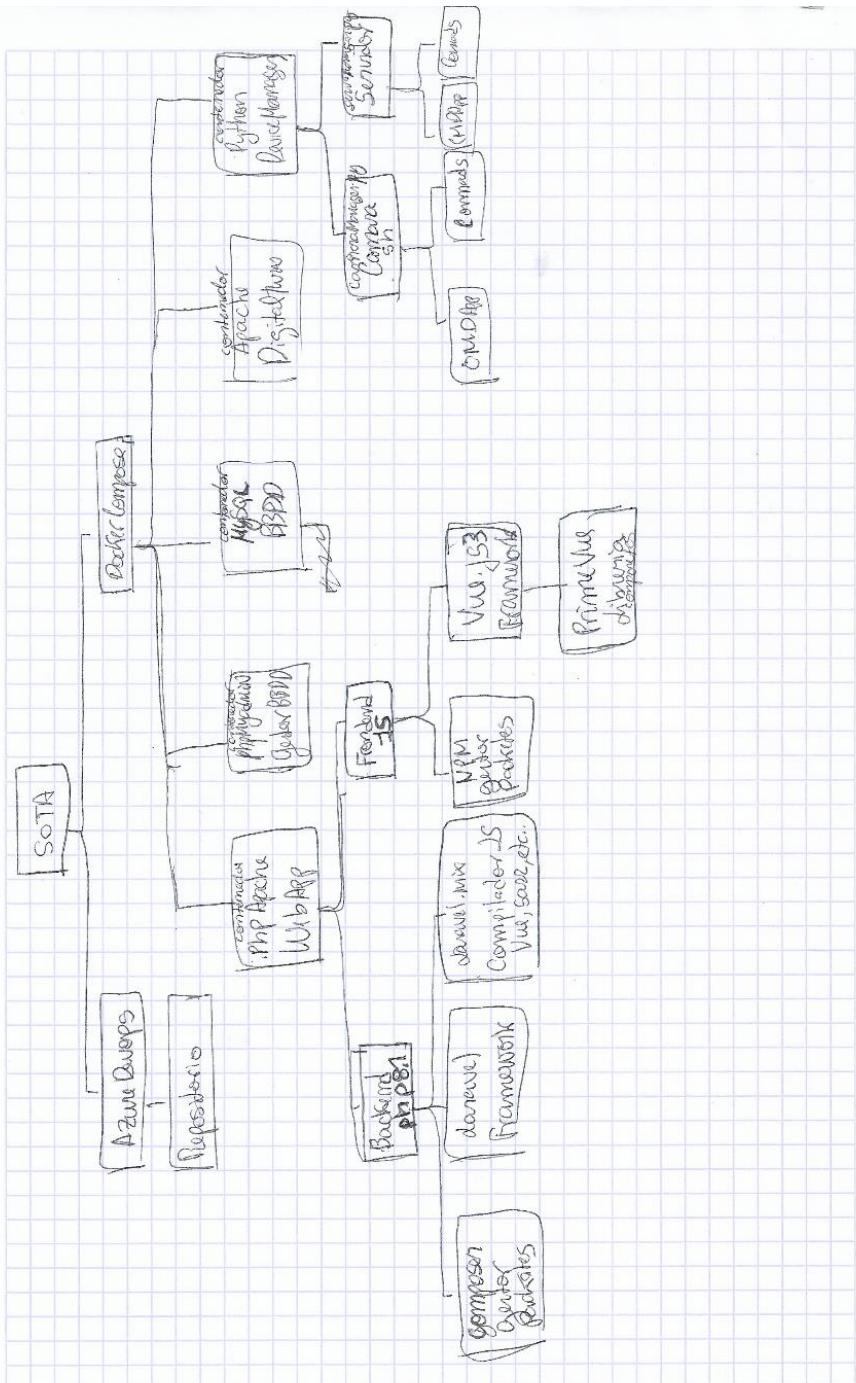


Figura 0.22: Boceto de Árbol tecnologías– Elaboración Propia

DISEÑO DEL SISTEMA

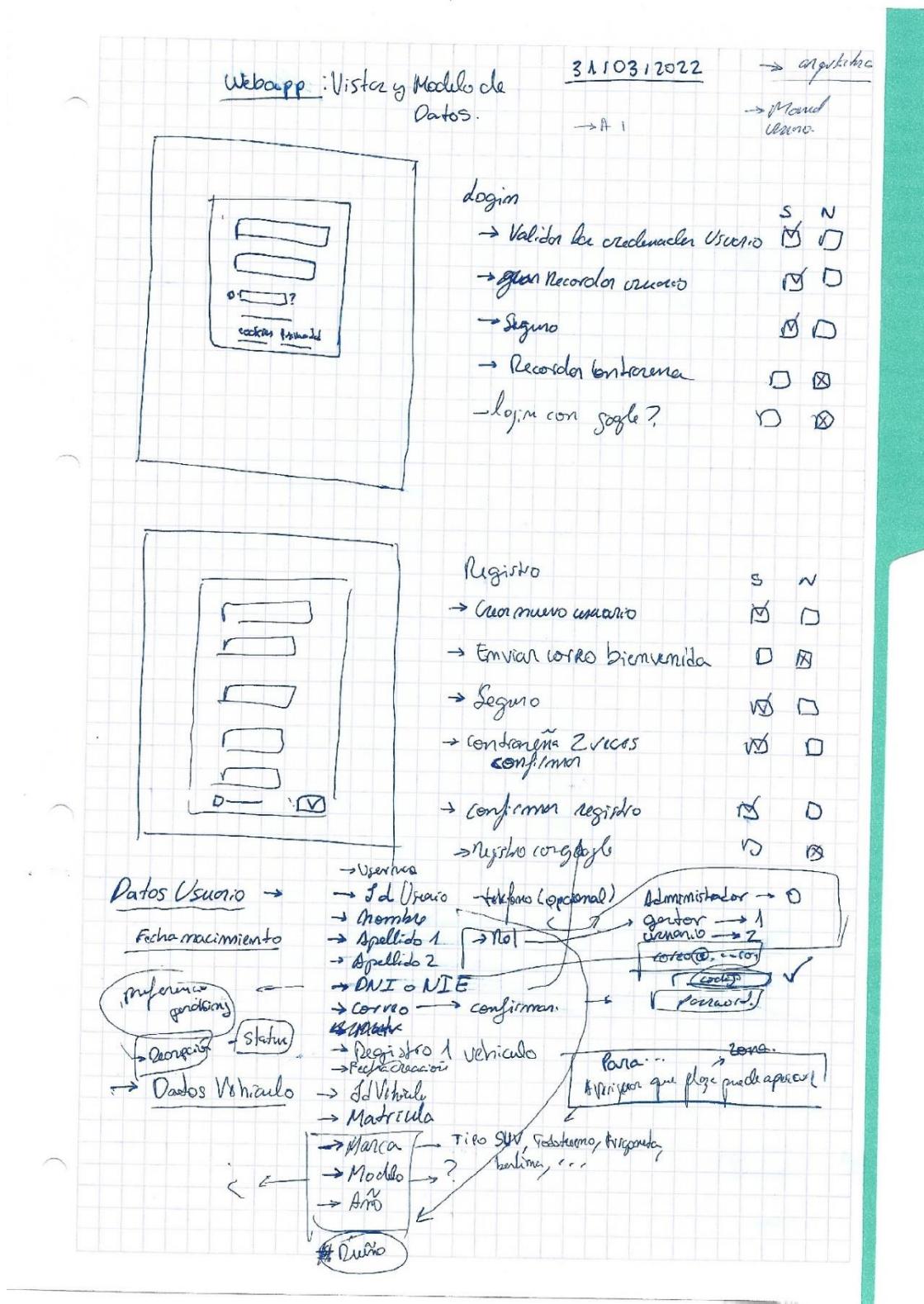


Figura 0.23: Diseño de las Vistas para WebApp UFV MyParking I – Elaboración Propia

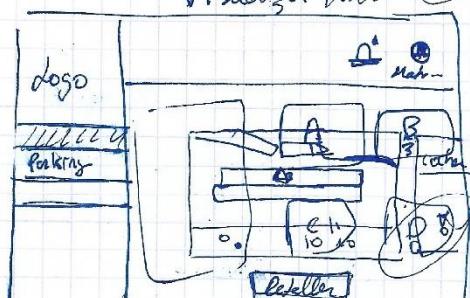
webapp

30/03/2022

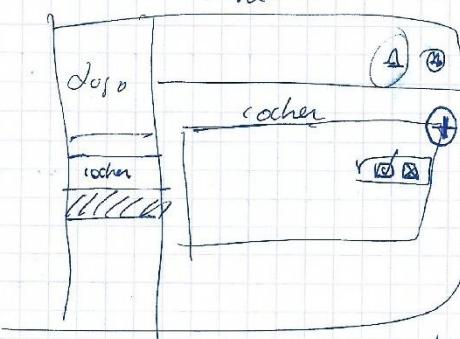
Área usuario

Visualizar Datos

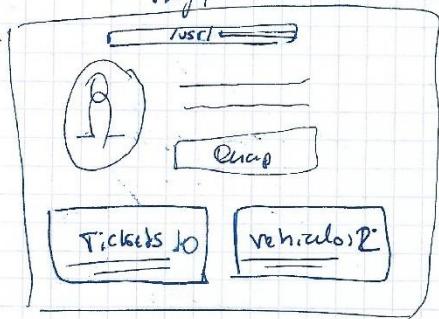
Optimal



coche Admin



perfil



- | | |
|---|---|
| s | n |
|---|---|
- Anadir vehiculos
 - Notificaciones
 - Visualizar Mapa parking
 - Anadir foto
 - Modificar datos per

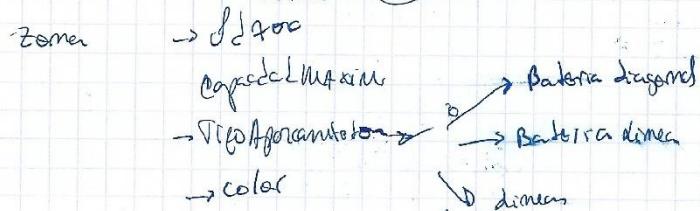
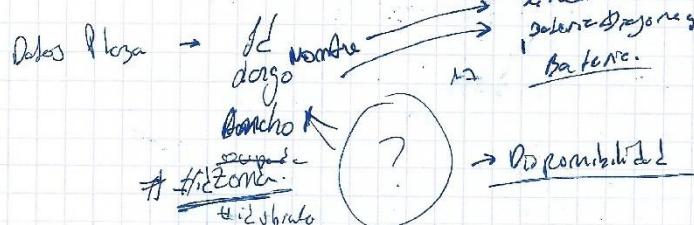


Figura 0.24 Diseño de las Vistas para WebApp UFV MyParking II – Elaboración Propia

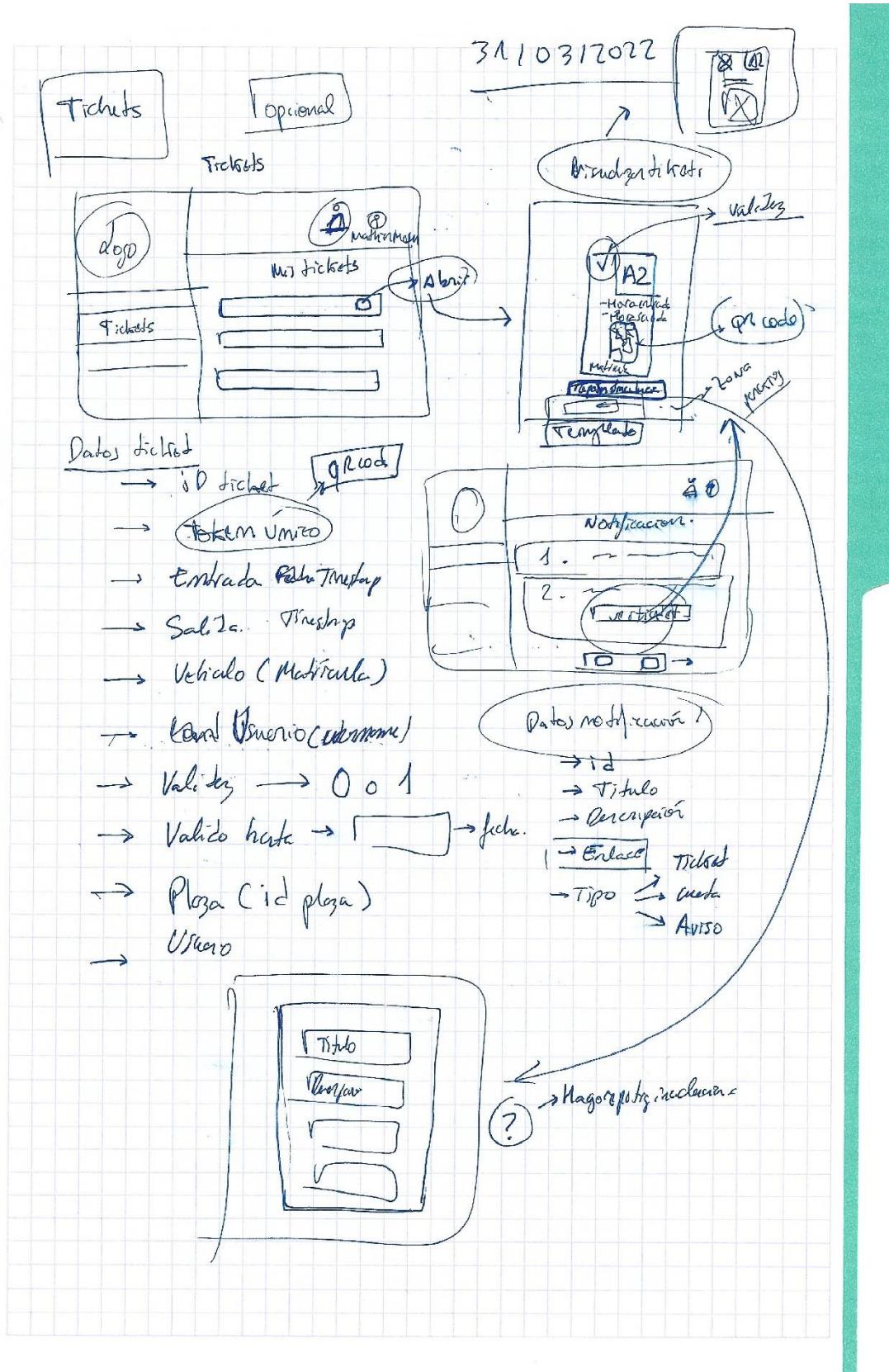


Figura 0.25: Diseño de las Vistas para WebApp UFV MyParking III – Elaboración Propia

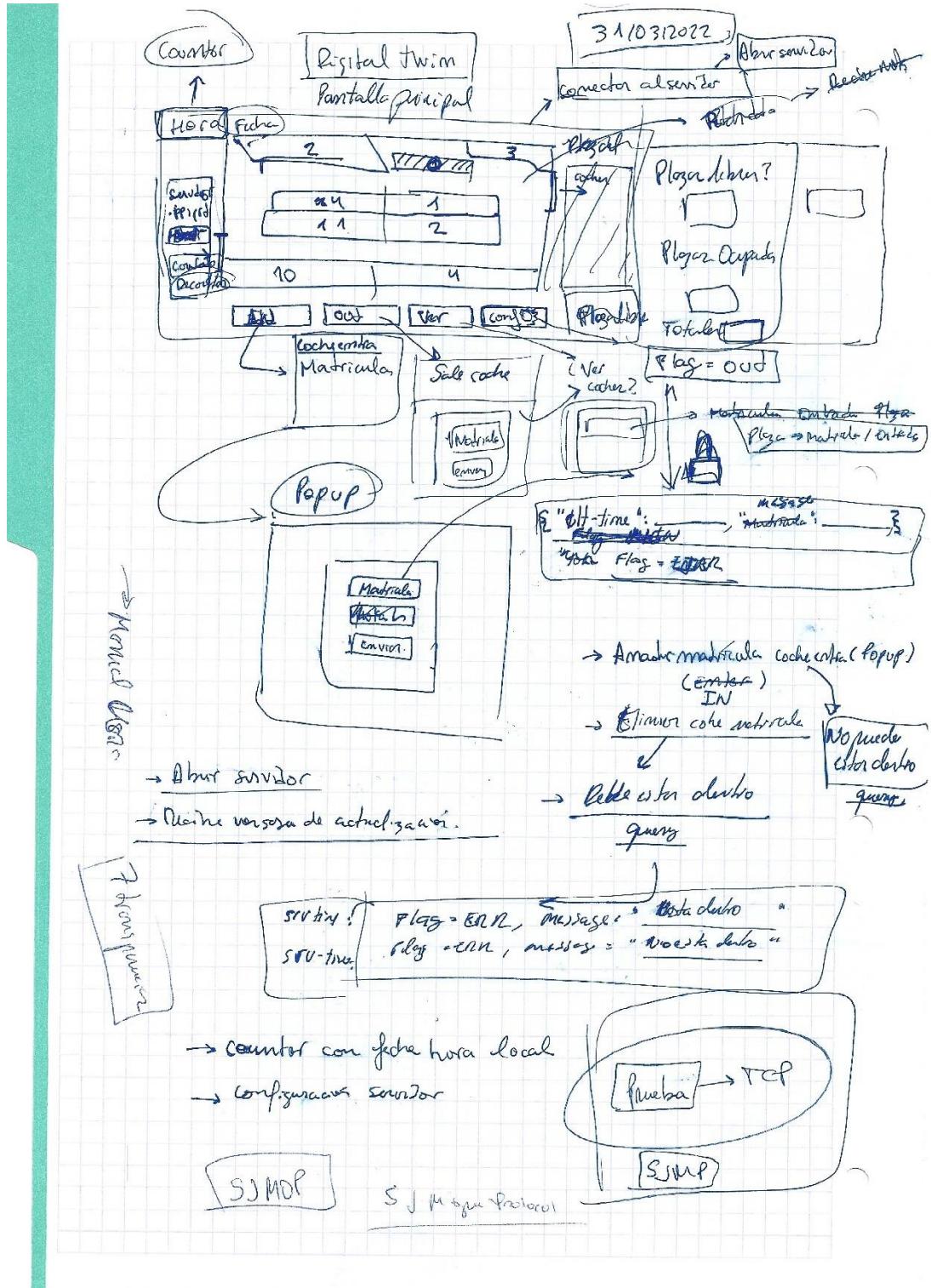


Figura 0.26: Diseño de las Vistas para Digital Twin I – Elaboración Propia

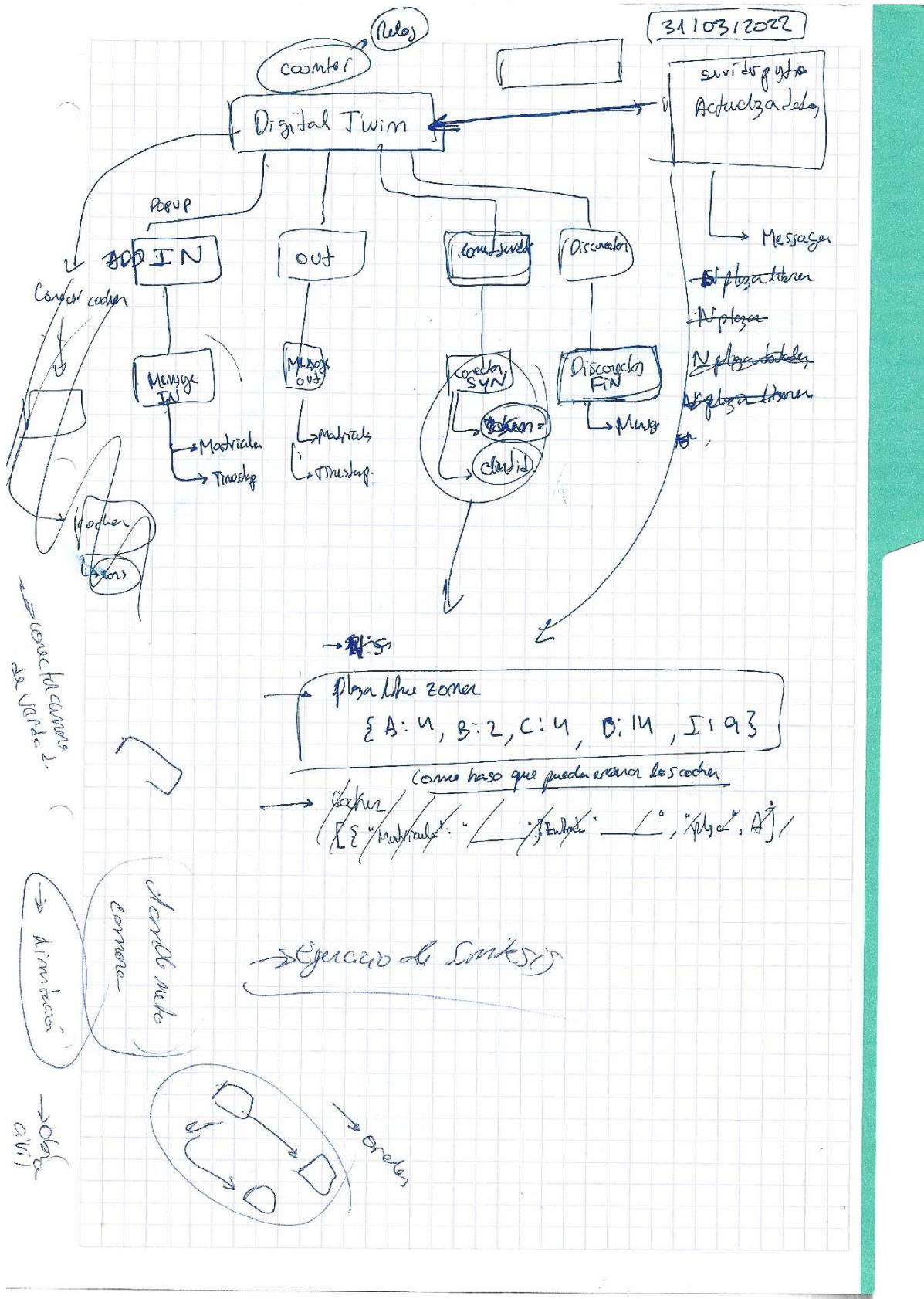


Figura 0.27: Diseño de las Vistas para Digital Twin II – Elaboración Propia

PLANIFICACIÓN

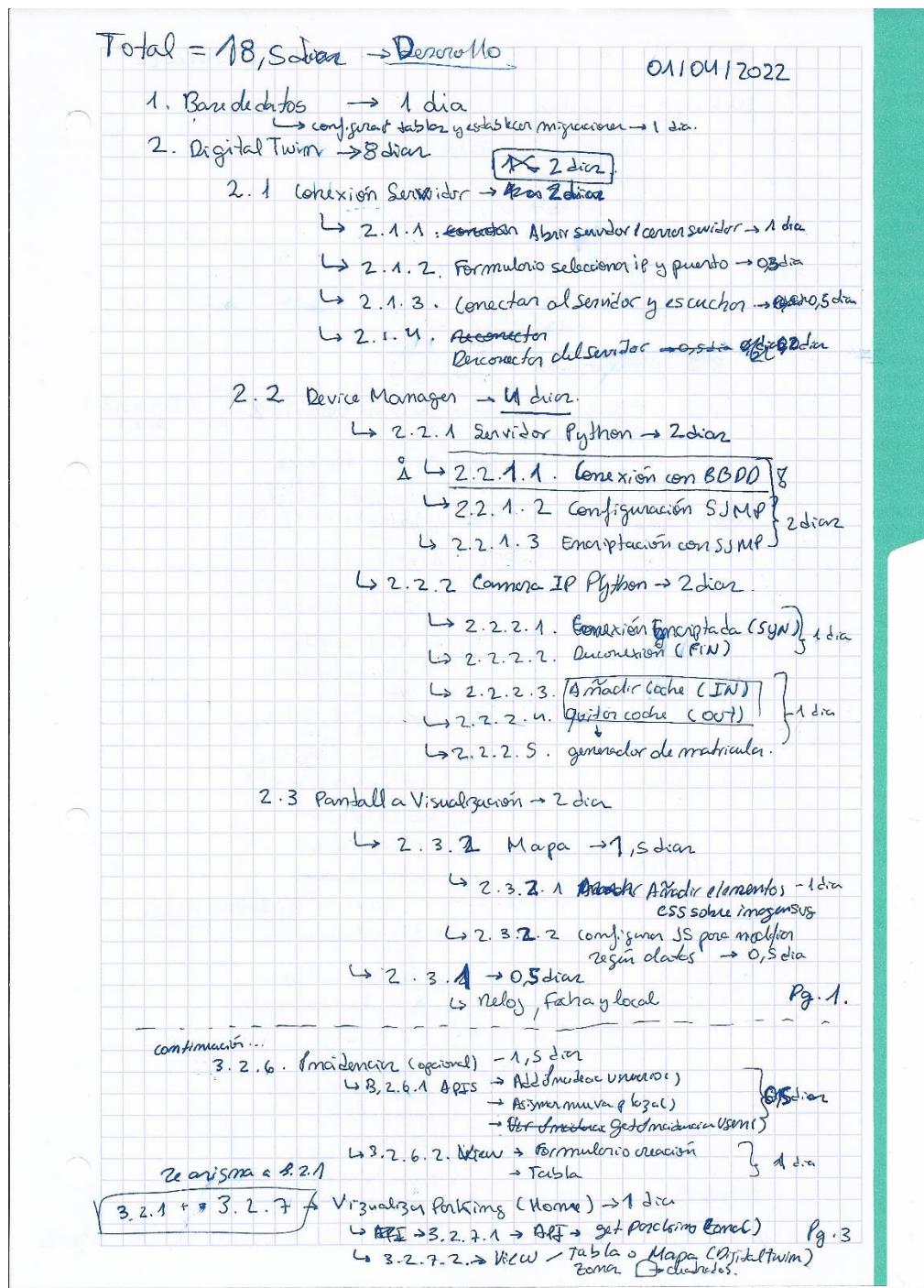


Figura 0.28: Boceto de planificación – Elaboración Propia

01/04/2022

3. Web App - 9,5 días

3.1. Autenticación → 2 días

3.1.1 ↳ API's → 1 día

3.1.1.1 ↳ login → 0,5 días

3.1.1.2 ↳ Registro → 0,5 días

3.1.2 ↳ View → 1 día

3.1.2.1 ↳ login → 0,5 días

3.1.2.2 ↳ Registro → 0,5 días

3.2. Área Usuario → 7 días

↳ 3.2.1 Esquema Dashboard (View) → 2 días (Dashboard)

↳ 3.2.2 Perfil → 1 día

↳ API's 3.2.2.1 API's → 0,2

3.2.2.2 API's ↳ getUSER → 0,2

↳ 3.2.3 Notificación → 1,5 días → modifyUSER

↳ API's 3.2.3.1 API's → 0,5 días

↳ getNotificacion → 0,2

↳ selectNotificacion

↳ 3.2.3.2 View → 1 día

↳ Tabla notificación
grid

↳ 3.2.4 Cacheo → 1,5 días

↳ 3.2.4.1 API's → 0,5 días

↳ getCacheoUsuario()

↳ addCacheoUsuario

↳ removeCacheoUsuario

↳ modificarCacheoUsuario

↳ 3.2.4.2 Views → 1,5 días → 1 día

↳ Tabla cacheo
grid

↳ Formulario Modificar

↳ Formulario Agregar

↳ 3.2.5. Tickets → 2 días

↳ 3.2.5.1 API's → 1 día

↳ getTicketsPendientes()

↳ openTicket()

↳ API QR code

↳ 3.2.5.2 View → 1 día

→ Tabla ticket → 0,5 días

→ Visualización ticket → 0,5

girar pagina

↳ 3.2.6.

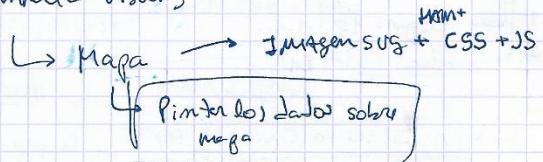
Pg. 2

Figura 0.29: Boceto de planificación II – Elaboración Propia

2 - Digital Twin

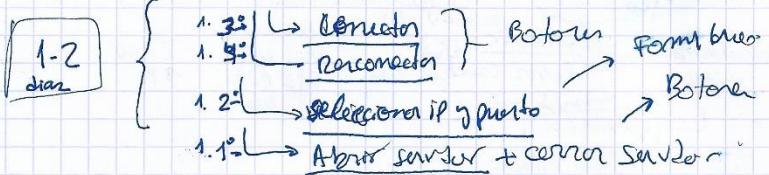
31/03/2022

↳ Pantalla Visualización

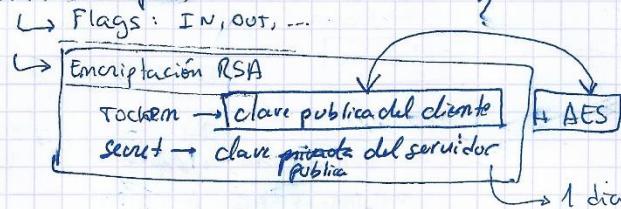


↳ Reloj, Fecha y local. → JS + HTML + CSS

1º → ↳ Conexión servidor → JS + PHP → Acceder a la máquina.



3 - ↳ SJMP (conceptual)



↳ Device Manager (servidor)

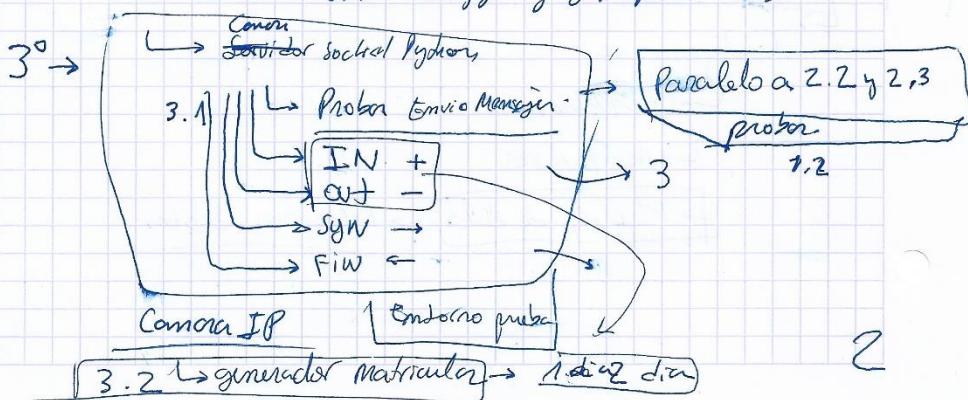
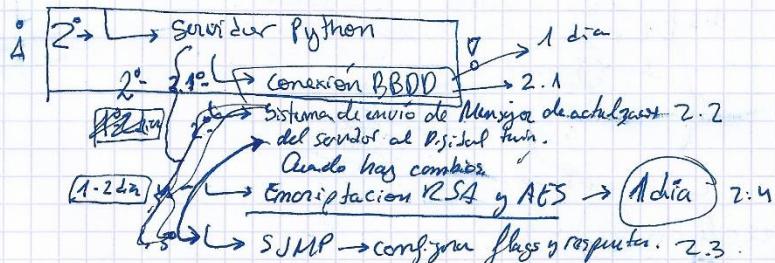


Figura 0.30: Boceto planificación III – Elaboración Propia



Figura 0.31: Boceto planificación IV – Elaboración Propia

MODELO DE DATOS

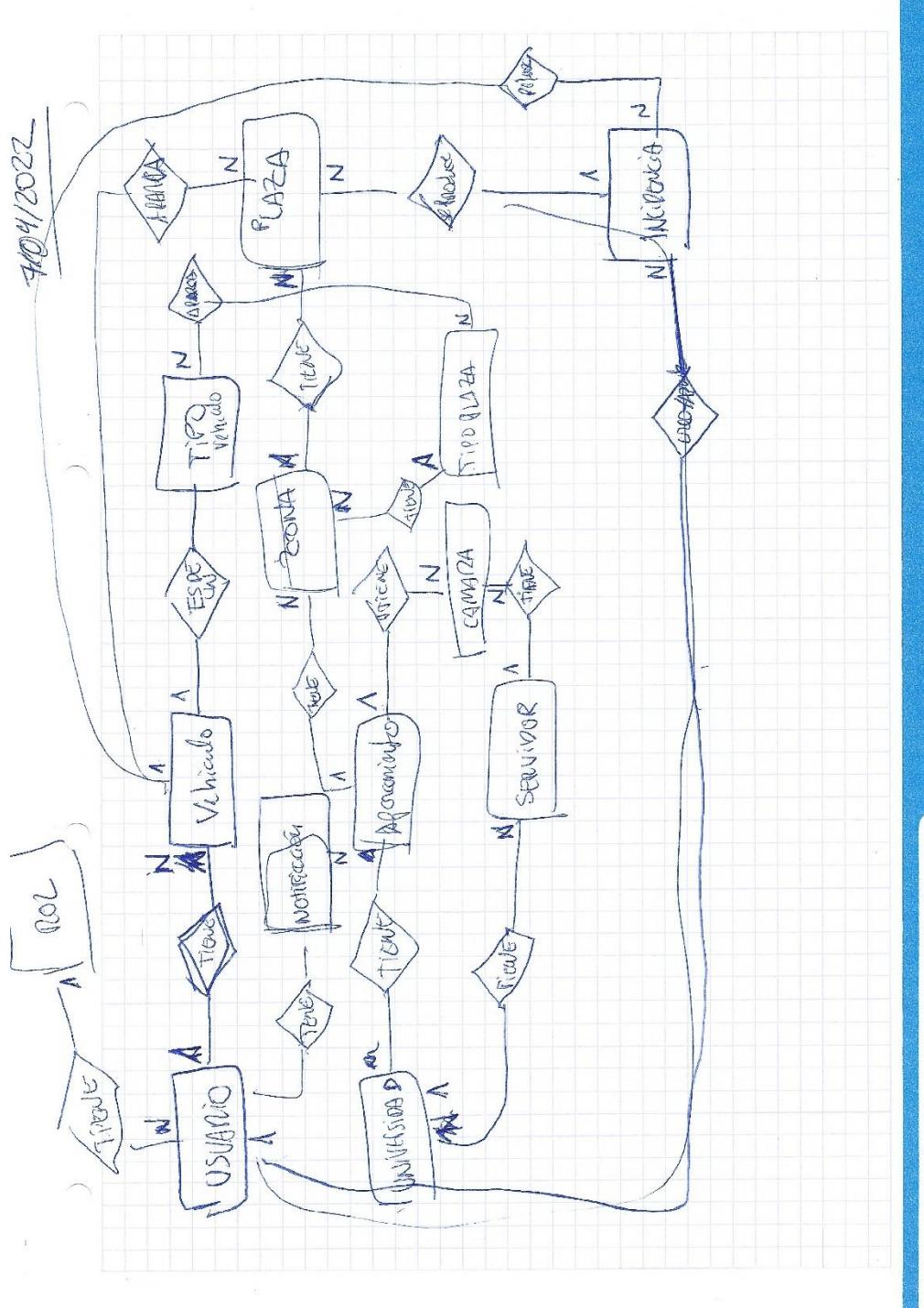


Figura 0.32: Modelo Relacional Entidad Relación - Elaboración Propia

ANEXO G: DETALLES PLANIFICACIÓN

Aquí se incluyen los detalles de la planificación realizada.

INVESTIGACIÓN PREVIA

La fase de investigación previa se pretende encontrar y validar la verdadera necesidad del proyecto. En esta fase se deberá recoger información de los diferentes usuarios del sistema, gestores y usuarios del *parking*. Se pretende también profundizar más en las tecnologías generadas en las soluciones del proyecto y en que Universidades españolas se aplican.

Tabla 50: Tarea PT01-T01: Segunda Reunión con Gestor – Elaboración Propia

Código	PT01-T01	Nombre	Segunda Reunión con Gestor
Descripción		Segunda Reunión con jefe de Servicios Generales.	
Entradas		N/A.	
Salidas		Problemas Actuales del Aparcamiento UFV y Soluciones Existentes.	
Actividades		Se deberán preguntar al gestor cuales son los problemas reales producidos en el aparcamiento, así como las soluciones existentes.	

Tabla 51: Tarea PT01-T02: Investigación del Estado del Arte – Elaboración Propia

Código	PT01-T02	Nombre	Investigación del Estado del Arte
Descripción		Investigar sobre el estado del arte de los Smart Campus en España.	
Entradas		N/A.	
Salidas		Estado del Arte.	
Actividades		Se deberán de investigar sobre las soluciones existentes y planes de Smart Campus en las diferentes Universidades Españolas, además de investigar sobre las <i>Smart Cities</i> y sus soluciones.	

Tabla 52: Tarea PT01-T03: Estudio del Parking UFV – Elaboración Propia

Código	PT01-T03	Nombre	Estudio del Parking UFV
Descripción	Investigar sobre la opinión de los alumnos, profesores y PAS		
Entradas	Problemas actuales del Aparcamiento UFV y soluciones existentes.		
Salidas	Opinión de los usuarios del aparcamiento y necesidad del sistema.		
Actividades	Se realizará una encuesta para poder conocer que piensan los usuarios a respecto de los problemas actuales y de la solución propuesta.		

DEFINICIONES INICIALES

En la fase de definiciones iniciales se definirá el problema, para ello se deberá acotar el alcance del proyecto definiendo objetivos y que tipo de sistema se desarrollará.

Tabla 53: Tarea PT02-T01: Introducción – Elaboración Propia

Código	PT02-T01	Nombre	Introducción
Descripción	Introducir el tema y poner en contexto el proyecto.		
Entradas	Investigación Previa en Anteproyecto.		
Salidas	Introducción del tema y puesta en contexto.		
Actividades	Se deberá de situar el problema dentro del contexto universitario propuesto para el proyecto.		

Tabla 54: Tarea PT02-T02: Definición de Objetivos y Alcance – Elaboración Propia

Código	PT02-T02	Nombre	Definición de Objetivos y Alcance
Descripción	Definir los objetivos y alcance del proyecto.		
Entradas	Investigación Previa en Anteproyecto.		
Salidas	Objetivos y alcances definidos.		
Actividades	Se deberán definir los objetivos que alcanzar con el proyecto y se definirá un alcance aceptable para el trabajo final de grado.		

PLANIFICACIÓN Y ESPECIFICACIÓN DEL SISTEMA

La primera fase de la metodología en cascada es la definición de los requisitos de usuario para poder definir de manera detallada lo que se pretende hacer. Una vez que tenemos los requisitos podemos empezar a realizar el plan d proyecto y definir las especificaciones.

Tabla 55: Tarea PT03-T01: Requisitos de Usuario – Elaboración Propia

Código	PT03-T01	Nombre	Requisitos de Usuario	
Descripción		Definir los requisitos de usuario del sistema.		
Entradas		Objetivos y alcances definidos.		
Salidas		Requisitos de Usuario.		
Actividades		Se deberán de definir los requisitos y sus módulos para poder conocer cómo será el sistema.		

Tabla 56: Tarea PT03-T02: Plan de Proyecto – Elaboración Propia

Código	PT03-T02	Nombre	Plan de Proyecto	
Descripción		Realizar el Plan de proyecto, definiendo los plazos y tareas a realizar.		
Entradas		Requisitos de Usuario.		
Salidas		Diagrama de Gantt y Tareas en plan de proyecto.		
Actividades		Realizar el diagrama de Gantt, además de definir las tareas a realizar y su orden de prioridad. Se deberán de definir las dependencias de cada tarea.		

Tabla 57: Tarea PT03-T03: Modelado 2D del Parking Seleccionado – Elaboración Propia

Código	PT03-T03	Nombre	Modelado 2D del Parking Seleccionado	
Descripción		Modelar el <i>parking</i> , tomando medidas de un aparcamiento real.		
Entradas		Requisitos de Usuario.		
Salidas		Mapa en 2D del aparcamiento, número de plazas y medidas de plaza.		
Actividades		Se deberá tomar las medidas de las plazas del aparcamiento de un aparcamiento de la Universidad Francisco de Vitoria. Se deberá de conocer el formato del aparcamiento y modelarlo en una imagen en 2D con vista aérea para conocer su formato y sus zonas.		

DISEÑO Y CONFIGURACIÓN DEL ENTORNO

Tabla 58: Tarea PT04-T01: Estudio de Tecnologías – Elaboración Propia

Código	PT04-T01	Nombre	Estudio de Tecnologías	
Descripción		Estudio detallado de las tecnologías a utilizar, buscando las diferentes opciones y eligiendo las correctas.		
Entradas		Requisitos de Usuario y Plan de Proyecto.		
Salidas		Listado de tecnologías a utilizar.		
Actividades		Se deberá buscar sobre las diferentes tecnologías que se pueden utilizar para poder preparar el entorno para el desarrollo.		

Tabla 59: Tarea PT04-T02: Configuración del Entorno – Elaboración Propia

Código	PT04-T02	Nombre	Configuración del Entorno
Descripción	Configuración del entorno usando las tecnologías seleccionadas.		
Entradas	Listado de tecnologías a utilizar.		
Salidas	Entorno configurado para empezar con el desarrollo.		
Actividades	Se deberán de realizar las siguientes tareas: ➤ <i>Configuración de Contenedores Docker</i> ➤ <i>Configuración de Repositorio</i>		

Tabla 60: Tarea PT04-T03: Diseño del sistema – Elaboración Propia

Código	PT04-T03	Nombre	Diseño del sistema
Descripción	Diseñar el sistema y sus vistas con detalles para guiar el desarrollo.		
Entradas	Requisitos de Usuario y Listado de Tecnologías.		
Salidas	Diseño del Interfaz, Componentes y SJMP.		
Actividades	Se deberán de realizar las siguientes tareas: ➤ <i>Diseño de Componentes</i> ➤ <i>Diseño Interfaz</i> ➤ <i>Diseño Protocolo SJMP</i>		

DESARROLLO DEL SOFTWARE

Base de Datos

Tabla 61: Tarea PT05-T01: Configuración de Base de Datos – Elaboración Propia

Código	PT05-T01	Nombre	Base de Datos
Descripción	Diseño del modelo de datos y configuración de la base de datos.		
Entradas	Requisitos de Usuario y Listado de Tecnologías.		
Salidas	Modelo de Datos y Base de Datos Configurada.		
Actividades	Se deberán de realizar las siguientes tareas: ➤ <i>Diseño del Modelo de Datos</i> ➤ <i>Configuración de la Base de Datos</i>		

Digital Twin

Tabla 62: Tarea PT05-T02: Conexión con Servidor – Elaboración Propia

Código	PT05-T02	Nombre	Conexión con Servidor	
Descripción		Configuración de la conexión e interacción con el servidor.		
Entradas		Modelo de Datos y Requisitos de Usuario.		
Salidas		Formulario de Conexión con Servidor y Controles del Servidor.		
Actividades		Se deberán de realizar las siguientes tareas: <ul style="list-style-type: none"> ➤ <i>Start/Stop Servidor</i> ➤ <i>Formulario Configuración Servidor IP:Puerto</i> ➤ <i>Conectar/Desconectar con</i> ➤ <i>Configuración de Servidor y Listener</i> 		

Tabla 63: Tarea PT05-T03: Device Manager – Elaboración Propia

Código	PT05-T03	Nombre	Device Manager	
Descripción		Configuración del Servidor de Cámaras y protocolo SJMP.		
Entradas		Conexión con Servidor.		
Salidas		Servidor <i>Device Manager</i> .		
Actividades		Se deberán de realizar las siguientes tareas: <ul style="list-style-type: none"> ➤ <i>Servidor Python</i> ➤ <i>Conexión con <u>Bases de Datos (BBDD)</u></i> ➤ <i>Configuración SJMP</i> ➤ <i>Encriptación con SJMP</i> 		

Tabla 64: Tarea PT05-T04: Cámara IP Python – Elaboración Propia

Código	PT05-T04	Nombre	Cámara IP Python	
Descripción		Desarrollo de la Cámara IP simuladora de la cámara real.		
Entradas		Servidor <i>Device Manager</i> .		
Salidas		Cámara IP simuladora de conexiones TCP con RTJP.		
Actividades		Se deberán de realizar las siguientes tareas: <ul style="list-style-type: none"> ➤ <i>Conexión Encriptada (SYN) (Handshake)</i> ➤ <i>Añadir Coche (IN)</i> ➤ <i>Generador de Matriculas</i> ➤ <i>Quitar Coche (OUT)</i> ➤ <i>Desconexión (FIN)</i> 		

Tabla 65: Tarea PT05-T05: Pantalla de Visualización – Elaboración Propia

Código	PT05-T05	Nombre	Pantalla de Visualización
Descripción	Se deberá hacer el login para la pantalla de visualización y visualizar el mapa con los datos obtenidos de la base de datos.		
Entradas	Cámara IP simuladora de conexiones TCP con RTJP y Base de Datos.		
Salidas	Digital Twin.		
Actividades	Se deberán de realizar las siguientes tareas: <ul style="list-style-type: none"> ➤ <i>Esqueleto HTML con Login</i> ➤ <i>Mapa</i> <ul style="list-style-type: none"> ○ <i>Añadir Elementos CSS sobre Imagen</i> ○ <i>Configurar JS para modificar datos</i> 		

WebApp

Tabla 66: Tarea PT05-T06: Autenticación – Elaboración Propia

Código	PT05-T06	Nombre	Autenticación
Descripción	Desarrollo de la Autenticación de la Aplicación Web.		
Entradas	Digital Twin.		
Salidas	Formularios de <i>Login</i> y Registro.		
Actividades	Se deberán de realizar las siguientes tareas: <ul style="list-style-type: none"> ➤ <i>API's</i> <ul style="list-style-type: none"> ○ <i>checkUser()</i> ○ <i>addUser()</i> ○ <i>sendEmail()</i> ➤ <i>Views</i> <ul style="list-style-type: none"> ○ <i>Form Login</i> ○ <i>Form Registro</i> 		

Área de Usuario

Tabla 67: Tarea PT05-T07: Esqueleto Dashboard y Home – Elaboración Propia

Código	PT05-T07	Nombre	Esqueleto Dashboard y Home
Descripción	Desarrollo del Esqueleto central utilizado en todas las vistas.		
Entradas	Formularios de <i>Login</i> y Registro.		
Salidas	Componente con Menú, Página de Inicio con Mapa <i>Parking</i> .		
Actividades	<p>Se deberán de realizar las siguientes tareas:</p> <ul style="list-style-type: none"> ➤ <i>API's</i> <ul style="list-style-type: none"> ○ <i>getParkingState()</i> ➤ <i>Views</i> <ul style="list-style-type: none"> ○ <i>Menú</i> ○ <i>Mapa</i> ○ <i>Cajas</i> 		

Tabla 68: Tarea PT05-T08: Perfil de Usuario – Elaboración Propia

Código	PT05-T08	Nombre	Perfil de Usuario
Descripción	Desarrollo del Perfil del Usuario con sus datos.		
Entradas	Componente con Menu, Página de Inicio con Mapa <i>Parking</i> .		
Salidas	Perfil de Usuario.		
Actividades	<p>Se deberán de realizar las siguientes tareas:</p> <ul style="list-style-type: none"> ➤ <i>API's</i> <ul style="list-style-type: none"> ○ <i>getUser()</i> ○ <i>updateUser()</i> ➤ <i>Views</i> <ul style="list-style-type: none"> ○ <i>Form Modificar Usuario</i> ○ <i>Template Perfil</i> 		

Tabla 69: Tarea PT05-T09: Notificaciones – Elaboración Propia

Código	PT05-T09	Nombre	Notificaciones
Descripción	Desarrollo Componente de Notificaciones.		
Entradas	Perfil de Usuario.		
Salidas	Componente de Notificaciones.		
Actividades	<p>Se deberán de realizar las siguientes tareas:</p> <ul style="list-style-type: none"> ➤ <i>API's</i> <ul style="list-style-type: none"> ○ <i>getNotifications()</i> ○ <i>deleteNotification()</i> ➤ <i>Views</i> <ul style="list-style-type: none"> ○ <i>Grid Notificaciones</i> ○ <i>Botón Borrado</i> 		

Tabla 70: Tarea PT05-T10: Área de Vehículos – Elaboración Propia

Código	PT05-T10	Nombre	Área de Vehículos
Descripción	Desarrollo del área de vehículos donde se podrán ver, añadir y borrar coches.		
Entradas	Componente de Notificaciones.		
Salidas	Área de Vehículos.		
Actividades	<p>Se deberán de realizar las siguientes tareas:</p> <ul style="list-style-type: none"> ➤ <i>API's</i> <ul style="list-style-type: none"> ○ <i>getUserCars()</i> ○ <i>addUserCar()</i> ○ <i>deleteUserCar()</i> ➤ <i>Views</i> <ul style="list-style-type: none"> ○ <i>Grid Coches</i> ○ <i>From Crear Coche</i> 		

Tabla 71: Tarea PT05-T11: Tickets – Elaboración Propia

Código	PT05-T11	Nombre	Tickets
Descripción	Desarrollo del área de tickets y la visualización de un ticket.		
Entradas	Área de Coches.		
Salidas	Área de Tickets y Visualización de un Ticket.		
Actividades	<p>Se deberán de realizar las siguientes tareas:</p> <ul style="list-style-type: none"> ➤ <i>API's</i> <ul style="list-style-type: none"> ○ <i>getTicketsFromUser()</i> ○ <i>openTicket()</i> ○ <i>isValidTicket()</i> ○ <i>genQRCode()</i> ➤ <i>Views</i> <ul style="list-style-type: none"> ○ <i>Grid Tickets</i> ○ <i>Template Ticket</i> 		

Tabla 72: Tarea PT05-T12: Incidencias – Elaboración Propia

Código	PT05-T12	Nombre	Incidencias
Descripción		Desarrollo del Portal de Incidencias y Denuncia de Incidencias dentro de los <i>Tickets</i> , así como la asignación de una nueva plaza.	
Entradas		Área de <i>Tickets</i> y Visualización de un <i>Ticket</i> .	
Salidas		Portal de Incidencias, Reporte de Incidencias en <i>Tickets</i> y WebApp.	
Actividades		<p>Se deberán de realizar las siguientes tareas:</p> <ul style="list-style-type: none"> ➤ <i>API's</i> <ul style="list-style-type: none"> ○ <i>addIncidencia()</i> ○ <i>getNewPlace()</i> ○ <i>getUserIncidencias()</i> ○ <i>solveIncidencia()</i> ➤ <i>Views</i> <ul style="list-style-type: none"> ○ <i>Grid Incidencias</i> ○ <i>Form Crear Incidencia</i> ○ <i>Form Resolver Incidencial</i> 	

INTEGRACIÓN Y PRUEBAS

Tabla 73: Tarea PT06-T01: Pruebas de Integración – Elaboración Propia

Código	PT06-T01	Nombre	Pruebas de Integración
Descripción		Realización de las pruebas para verificar que se ha realizado la integración de los dos sistemas correctamente.	
Entradas		Digital Twin y WebApp.	
Salidas		Pruebas de Integración.	
Actividades		<p>Se deberán de realizar las siguientes tareas:</p> <ul style="list-style-type: none"> - Probar Digital Twin y WebApp - Realizar pruebas interactuando los dos sistemas. 	

Tabla 74: Tarea PT06-T02: Pruebas de Verificación – Elaboración Propia

Código	PT06-T02	Nombre	Pruebas de Verificación
Descripción		Realización de las pruebas de verificación de los requisitos para comprobar que el sistema cumple con lo especificado.	
Entradas		Requisitos de Usuario, Digital Twin y WebApp.	
Salidas		Plan de Pruebas.	
Actividades		<p>Se deberán de realizar las siguientes tareas:</p> <ul style="list-style-type: none"> - Comprobar los requisitos - Crear plan de pruebas. 	

DIAGRAMA DE PERT

Aquí se encuentran los diagramas ampliados.

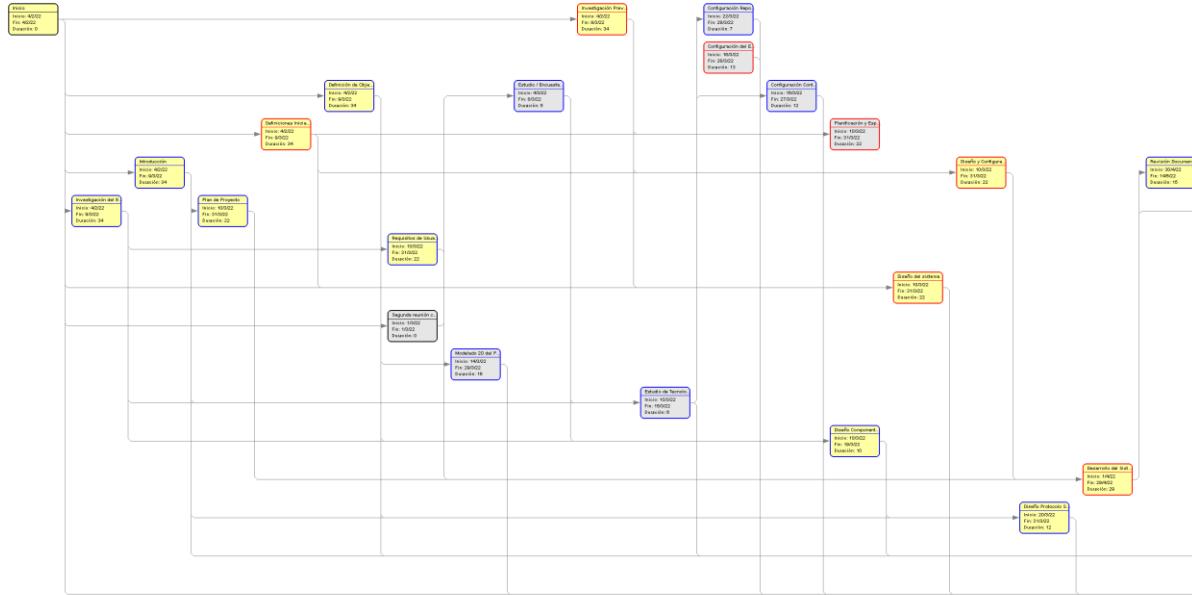


Figura 0.1: Diagrama de PERT – Parte 1, Fuente Elaboración Propia con GANTT Project [86]

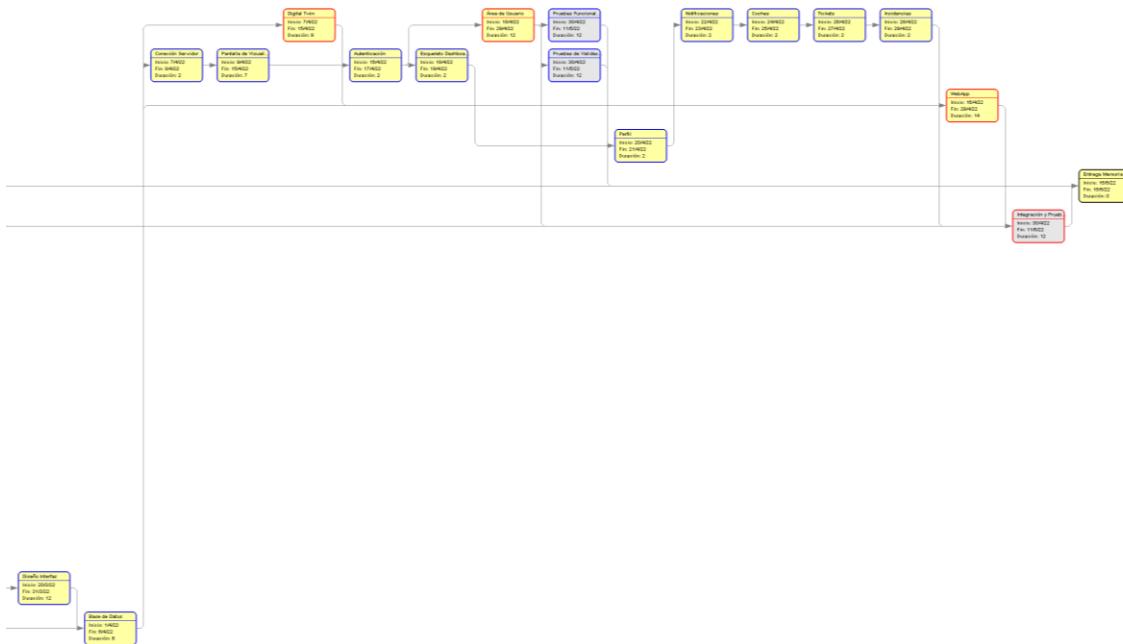


Figura 0.2: Diagrama de PERT – Parte 2, Fuente Elaboración Propia con GANTT Project [86]

ANEXO H: REGISTRO DE RIESGOS

TABLA DE RIESGOS

Tabla 75: Registro de Riesgos – Elaboración Propia

Riesgo	Descripción	Prob.	Acción	Imp.
<i>R1</i>	Pérdida de datos por fallo informático.	Media	Tener una copia de seguridad utilizando RAID (<i>Redundant Array of Independent Disks</i>) 5 si posible.	Alto
<i>R2</i>	Aparcamiento ilegal o en plaza equivocada.	Alta	Permitir el reporte de incidencias e infracciones.	Medio
<i>R3</i>	Caídas de Servicio.	Alta	Visualizar el estado del servidor en el Digital Twin. Posibilidad de apertura de otro servidor auxiliar.	Alto
<i>R4</i>	Filtración de datos en las comunicaciones o de la base de datos.	Media	Cifrado de las contraseñas utilizando un HASH fuerte como SHA-512	Alto
<i>R5</i>	Ataques tipo “ <i>Man in the Middle</i> ”, de visualización de comunicaciones en cualquier red utilizada (IP, LoRa, etc).	Alta	Crear un protocolo seguro con encriptación a nivel de dato, utilizando encriptación asimétrica RSA.	Alto
<i>R6</i>	Ataques SQL Injection en los servicios.	Alta	Filtrar los caracteres de los paquetes.	Alto
<i>R7</i>	Atascos en la entrada del <i>parking</i> .	Alta	Posicionamiento de la entrada al lado de un aparcamiento.	Alto
<i>R8</i>	Atascos en la salida del <i>parking</i> .	Alta	Posicionamiento de la salida al lado de calle principal del campus.	Medio
<i>R9</i>	Entrada ilegal en el aparcamiento.	Media	Colocación de barreras automáticas en la entrada y salida del aparcamiento.	Medio
<i>R10</i>	Envío de peticiones malintencionadas al sistema de terceros.	Media	Añadido un UUID único que deberá ser utilizado para enviar las peticiones al servidor web.	Alto
<i>R11</i>	Aparcamiento en otra plaza, no asignada.	Alta	Añadido reporte de incidencias para asignar nueva plaza.	Medio

Riesgo	Descripción	Prob.	Acción	Imp.
R12	Incidencias, como atropellamiento o choques con otros coches.	Media	Añadido al reporte de incidencias una gravedad de la situación donde un administrador puede borrar al usuario y contactar con las autoridades.	Alto
R13	Entornos climáticos diferentes en el campus pudiendo desactivar el sistema de cámaras.	Alta	Comprar una cámara de visión infrarroja utilizada en la universidad actualmente.	Alto
R14	Falta de conocimiento de la aplicación.	Media	Añadido un correo de soporte en caso de contacto y manual del usuario.	Bajo
R15	Entrada de camiones en el parking.	Baja	Probación de vehículos de grande porte como camiones.	Medio
R16	Mal funcionamiento de las barreras.	Baja	Reporte de incidencia al correo de soporte.	Alto
R17	Plazas no dimensionadas para motos y lectura de matrícula no posible.	Media	Prohibición de vehículos como motos o ciclomotores en el parking, para que se facilite la lectura y se evite problemas.	Medio
R18	Utilización del móvil para ver la plaza.	Alta	Añadir la posibilidad de colocar una pantalla TFT que indique en qué plaza debe aparcar.	Medio

MATRIZ DE RIESGOS

Tabla 76: Matriz de Riesgos – Elaboración Propia

Probabilidad	alta		R2, R8, R11	R3, R5, R6, R7, R13
	media	R14	R9, R17	R1, R4, R10, R12
	baja		R15	R16
		bajo	medio	alto
			Impacto	

ANEXO I: PLAN DE PRUEBAS

Tabla 77: Registro del Plan de Pruebas. – Elaboración Propia

Requirement No.	Módulo	Prioridad	Test Case Description	Input	Expected Result	Result	Status (OK / NOK)
1	Digital Twin	P1	Se debe poder visualizar el mapa del parking, (señalizado por zonas)	-	Visualizar Plazas	Visualizar Plazas	OK
2	Digital Twin	P2	Visualizar sobre el mapa el número de plazas en cada zona.	-	Se visualizan las plazas	Se visualizan las plazas	OK
3	Digital Twin	P1	Podrá abrir el servidor de Device Manager	Click sobre OpenServer	OpenServer Opens Server	OpenServer Opens Server	OK
4	Digital Twin	P1	Podrá cerrar el servidor de Device Manager	Clic sobre CloseServer	CloseServer Close Server	CloseServer Close Server	OK
5	Digital Twin	P1	Habrá un formulario para configurar los datos de conexión de la socket (IP:PUERTO) y el usuario al abrir el digital twin.	-	Existe	Existe	OK
6	Digital Twin	P1	Visualizar el total de plazas libres	botón Plazas Libres	Existe	Existe	OK
7	Digital Twin	P1	Visualizar el total de plazas ocupadas	botón Plazas Ocupadas	Existe	Existe	OK

Requirement No.	Módulo	Prioridad	Test Case Description	Input	Expected Result	Result	Status (OK / NOK)
8	Digital Twin	P1	Visualizar el total de plazas	botón Plazas Ocupadas y Libres	Existe	Existe	OK
9	Digital Twin	P1	Se podrá añadir un vehículo al parking indicando su matrícula.	Añadir Matricula	Añade	Añade	OK
10	Digital Twin	P1	Se podrá borrar un vehículo al parking indicando su matrícula.	Añadir Matricula	Borra	Borra	OK
11	Digital Twin	P1	Se podrá ver los vehículos dentro del parking en una lista con campos (Matricula, Plaza, Fecha Entrada, Tipo)	Dar a Botón Ver Plazas	Visualizar Plazas	Visualizar Plazas	OK
12	Digital Twin	P3	Al añadir un vehículo se podrá generar la matrícula de manera aleatoria.		NO IMPLEMENTADO		NOK
13	Digital Twin	P1	Se actualizará la información a cada 10 segundos.	AutoRefesh	Implementado	Implementado	OK
14	Digital Twin	P1	Se indicará la hora actual en el panel.	Reloj	Implementado	Implementado	OK
15	Digital Twin	P1	Se indicará la fecha actual en el panel.	Fecha	Implementado	Implementado	OK
16	Digital Twin	P1	Deberá incluir el username del usuario.	Username	Implementado	Implementado	OK
17	Digital Twin	P2	Se indicará la ip del cliente en el panel.	Ip Cliente	Implementado	Implementado	OK
18	Digital Twin	P1	Tendrá una pantalla de login.	Login	Implementado	Implementado	OK
19	Digital Twin	P1	El login será seguro	Tokenizado	Implementado	Implementado	OK
20	Digital Twin	P1	Solo se permitirá la entrada de roles autorizados.	Verificación de Rol	Implementado	Implementado	OK

Requirement No.	Módulo	Prioridad	Test Case Description	Input	Expected Result	Result	Status (OK / NOK)
21	Digital Twin	P1	Podrá levantar un servidor en un socket en su servidor local.	OpenServer	Implementado	Implementado	OK
22	Digital Twin	P1	Habrá un botón de refresh, para volver a pedir la información.	botón Refesh	Implementado	Implementado	OK
23	Servidor Device Manager	P1	Contará con un servidor hosteado en un socket	OpenServer	Implementado	Implementado	OK
24	Servidor Device Manager	P1	El servidor interpretará TCP	envío de Protocolo TCP	Implementado	Implementado	OK
25	Servidor Device Manager	P1	El servidor interpretará WebSocket	envío de Mensaje WebSocket	Implementado	Implementado	OK
26	Servidor Device Manager	P1	Se deberá rechazar cualquier conexión que no utilice SJMP.	envío de Mensaje Sin SJMP	Rechaza	Rechaza	OK
27	Servidor Device Manager	P1	Tendrá conexión con la base de datos	Conexión con BBDD en DBManager	Implementado	Implementado	OK
28	Servidor Device Manager	P1	El servidor interpretará SJMP.	SJMPHandler.py	Implementado	Implementado	OK
29	Protocolo SJMP	P1	Deberá ir sobre TCP.	TCPProtocol.py	Implementado	Implementado	OK
30	Protocolo SJMP	P1	Deberá ir sobre HTTP y WebSocket	HTTPProtocol.py y WebSocketProtocol.py	Implementado	Implementado	OK

Requirement No.	Módulo	Prioridad	Test Case Description	Input	Expected Result	Result	Status (OK / NOK)
31	Protocolo SJMP	P1	Su lógica de negocio será transmitida en paquetes	class packet()	Implementado	Implementado	OK
32	Paquete SJMP	P1	Deberá incluir campo "srv-time" con el timestamp de la hora de envío si sale del servidor.	Paquete con Server Time	Implementado	Implementado	OK
33	Paquete SJMP	P1	Deberá incluir campo "clt-time" con el timestamp de la hora de envío si sale de cualquier cliente.	Paquete con Client Time	Implementado	Implementado	OK
34	Paquete SJMP	P1	Deberá contener una flag.	Flags	Implementado	Implementado	OK
35	Flags Paquete SJMP	P1	La flag para añadir coches será "IN" y contendrá los siguientes campos: clt-time, flag, plate, sessionid.	Contiene los campos	Implementado	Implementado	OK
36	Flags Paquete SJMP	P1	La flag para quitar coches será "OUT" y contendrá los siguientes campos: clt-time, flag, plate, sessionid.	Contiene los campos	Implementado	Implementado	OK
37	Flags Paquete SJMP	P1	Se devolverá el flag "ACK" para confirmar que la conexión ha llegado al destino.: srv-time, flag, response.	Contiene los campos	Implementado	Implementado	OK
38	Flags Paquete SJMP	P1	Se devolverá el flag "ERR" para indicar que hubo un error en el servidor a raíz de la petición del cliente. srv-time, flag, response.	Contiene los campos	Implementado	Implementado	OK

Requirement No.	Módulo	Prioridad	Test Case Description	Input	Expected Result	Result	Status (OK / NOK)
39	Flags Paquete SJMP	P1	Se utilizará la flag "SYN" para realizar una petición de apertura de conexión por parte del cliente. Contendrá la siguiente estructura: clt-time, flag, cameraid, token.	Contiene los campos	Implementado	Implementado	OK
40	Flags Paquete SJMP	P1	Se utilizará la flag "OK" para indicar al cliente su sessionid y la clave pública secret del servidor. Este paquete irá encriptado con la clave publica del cliente desde el servidor, contiene: srv-time, flag, sessionid, secret.	Contiene los campos	Implementado	Implementado	OK
41	Paquete SJMP	P1	Cada cliente tendrá un "token" único que servirá de clave publica, que deberá estar relacionada matemáticamente con la clave privada. Solo se intercambiará en el mensaje de conexión.	Contiene los campos	Implementado	Implementado	OK
42	Paquete SJMP	P1	Cada servidor tendrá una clave publica llamada "secret" que permita encriptar las comunicaciones desde el cliente.	Contiene las claves	Implementado	Implementado	OK
43	Paquete SJMP	P1	Cada sesión iniciada por el cliente deberá tener un session id único.	Sessionid Añadido	Implementado	Implementado	OK
44	Paquete SJMP	P1	En la primera conexión se indicará un cameraid que permita identificar al gestor y se devolverá un OK.	Añadido	Implementado	Implementado	OK

Requirement No.	Módulo	Prioridad	Test Case Description	Input	Expected Result	Result	Status (OK / NOK)
45	Paquete SJMP	P1	Una vez encriptadas las sesiones el cliente encriptará los paquetes con las clave pública del servidor "secret" y el servidor encriptará los paquetes con la clave publica del cliente "token".	Encriptado en Python	Implementado	Implementado	OK
46	Parking	P1	El parking tendrá un aforo total de 182 plazas.	Aforo total	Implementado	Implementado	OK
47	Parking	P1	El parking tendrá 3 tipos de plazas (Linea, Bateria Diagonal, Bateria)	Tipo de plazas definidos	Implementado	Implementado	OK
48	Parking	P1	El parking estará dividido en 9 zonas nombradas del (A al I)	Dividido	Implementado	Implementado	OK
49	Parking	P1	Solo se permitirán entrar a los usuarios registrados en la Aplicación Web (MyParking)	El vehículo no puede entrar si no está registrado por un usuario.	Implementado	Implementado	OK
50	Parking	P1	Habrá una única entrada y una única salida con barreras. La zona estará cerrada alrededor.	Entradas y salidas únicas	Implementado	Implementado	OK
51	Parking	P1	Las barreras se abrirán si la matrícula leída cuadra con un usuario registrado y hay plazas disponibles.	Se abre cuando lee la matrícula y se asigna una plaza	Implementado	Implementado	OK
52	Parking	P1	Solo se permitirán coches y furgonetas.	Se definen las categorías	Implementado	Implementado	OK
53	Plazas Parking	P1	La zona A tiene 24 plazas de tipo batería	Establecido	Implementado	Implementado	OK

Requirement No.	Módulo	Prioridad	Test Case Description	Input	Expected Result	Result	Status (OK / NOK)
54	Plazas Parking	P1	La zona B tiene 25 plazas de tipo batería.	Establecido	Implementado	Implementado	OK
55	Plazas Parking	P1	La zona C tiene 22 plazas de tipo batería	Establecido	Implementado	Implementado	OK
56	Plazas Parking	P1	La zona D tiene 21 plazas de tipo batería	Establecido	Implementado	Implementado	OK
57	Plazas Parking	P1	La zona E tiene 21 plazas de tipo batería.	Establecido	Implementado	Implementado	OK
58	Plazas Parking	P1	La zona F tiene 21 plazas de tipo batería.	Establecido	Implementado	Implementado	OK
59	Plazas Parking	P1	La zona G tiene 14 plazas de tipo batería diagonal.	Establecido	Implementado	Implementado	OK
60	Plazas Parking	P1	La zona H tiene 4 plazas de tipo línea.	Establecido	Implementado	Implementado	OK
61	Plazas Parking	P1	La zona I tiene 23 plazas de tipo batería.	Establecido	Implementado	Implementado	OK
62	Plazas Parking	P1	Una plaza de tipo batería tiene: Largo: 515 cm, Ancho: 250 cm. Puede albergar todos los tipos menos Pequeños	Definido	Implementado	Implementado	OK
63	Plazas Parking	P1	Una plaza de tipo batería diagonal tiene: Largo: 555cm, Ancho: 300 cm, y puede todos los tipos.	Definido	Implementado	Implementado	OK
64	Plazas Parking	P1	Una plaza de tipo línea tiene: Largo:500 cm, Ancho: 200cm y puede albergar vehículos tipos Pequeño y Medianos.	Definido	Implementado	Implementado	OK

Requirement No.	Módulo	Prioridad	Test Case Description	Input	Expected Result	Result	Status (OK / NOK)
65	Plazas Parking	P1	Se deberá priorizar a los tipos de vehículos que permita la plaza.	Prioriza el vehículo por tipo de plaza	Implementado	Implementado	OK
66	Plazas Parking	P1	La plaza será asignada según el tipo de vehículo.	Prioriza el vehículo por tipo de plaza y tipo vehículo	Implementado	Implementado	OK
67	WebApp	P1	El nombre de la aplicación web será: "UFV MyParking".	Configurado	Implementado	Implementado	OK
68	WebApp	P1	Tendrá un menú que permita navegar entre vistas.	Dashboard Menú	Implementado	Implementado	OK
69	WebApp	P1	Tendrá una vista de área de usuario	Dashboard Mapa	Implementado	Implementado	OK
70	WebApp	P1	Tendrá una vista de perfil	Perfil Usuario	Implementado	Implementado	OK
71	WebApp	P3	Tendrá una vista de notificaciones	No configurado	NO IMPLEMENTADO		NOK
72	WebApp	P1	Tendrá una vista de para gestionar los coches del usuario.	Mis Coches	Implementado	Implementado	OK
73	WebApp	P1	Tendrá una vista para visualizar los tickets.	Mis Tickets	Implementado	Implementado	OK
74	WebApp	P3	Tendrá una vista para visualizar y resolver las incidencias.	Mis Incidencias y Admin incidencias	Implementado	Implementado	OK
75	WebApp	P3	Tendrá una vista para visualizar todos los tickets y un solo ticket.	Open Ticket y Mis Tickets	Implementado	Implementado	OK
76	WebApp Autenticación	P1	Para acceder a la web app se deberá insertar el correo al loguearse.	Login Con Correo	Implementado	Implementado	OK
77	WebApp Autenticación	P1	Para acceder a la web app se deberá insertar una contraseña al loguearse.	Login con Contraseña	Implementado	Implementado	OK

Requirement No.	Módulo	Prioridad	Test Case Description	Input	Expected Result	Result	Status (OK / NOK)
78	WebApp Autenticación	P1	Si el usuario no está registrado habrá un formulario de registro.	Formulario de Registro	Implementado	Implementado	OK
79	WebApp Autenticación	P3	Se enviará un correo al usuario al registrarse para confirmar el correo.	No configurado	NO IMPLEMENTADO		NOK
80	WebApp Área de Usuario	P1	Tendrá un dashboard que permita visualizar el estado del parking.	Dasboard Parking	Implementado	Implementado	OK
81	WebApp Área de Usuario	P1	El dashboard tendrá el número de plazas libres por zona.	Dashboard Parking	Implementado	Implementado	OK
82	WebApp Área de Usuario	P1	Se enseñarán las diferentes zonas en cajas.	Dashboard Parking	Implementado	Implementado	OK
83	WebApp Menu	P1	El menú será accesible desde cualquier página.	Dashboard Menu	Implementado	Implementado	OK
84	WebApp Menu	P1	El menú permitirá navegar a todas las vistas.	Dashboard Menu	Implementado	Implementado	OK
85	WebApp Menu	P1	Tendrá un icono de campana como botón para ver las notificaciones	botón Implementado	Implementado	Implementado	OK
86	WebApp Menu	P1	Tendrá un icono de usuario como botón para ver el perfil.	Icono Implementado	Implementado	Implementado	OK
87	WebApp Menu	P1	Será variable según el rol, permitiendo más o menos funcionalidades.	Variable según rol, enseña admin portal	Implementado	Implementado	OK

Requirement No.	Módulo	Prioridad	Test Case Description	Input	Expected Result	Result	Status (OK / NOK)
88	WebApp Notificaciones	P3	Permitirá visualizar las notificaciones en una tabla	No configurado	NO IMPLEMENTADO		NOK
89	WebApp Notificaciones	P3	Permitirá borrar las notificaciones	No configurado	NO IMPLEMENTADO		NOK
90	WebApp Notificaciones	P3	Una notificación tendrá un título	Configurado	Implementado	Implementado	OK
91	WebApp Notificaciones	P3	Una notificación tendrá una descripción.	Configurado	Implementado	Implementado	OK
92	WebApp Notificaciones	P3	Tendrá un botón y un enlace de manera opcional.	Configurado	Implementado	Implementado	OK
93	WebApp Notificaciones	P3	Tendrá un botón para borrar notificación	Configurado	Implementado	Implementado	OK
94	WebApp Notificaciones	P3	El usuario recibirá una notificación cuando se inicialice el aparcamiento incluyendo el ticket.	No configurado	NO IMPLEMENTADO		NOK
95	WebApp Notificaciones	P3	El usuario recibirá una notificación cuando se termine el aparcamiento incluyendo el ticket.	No configurado	NO IMPLEMENTADO		NOK
96	WebApp Notificaciones	P3	El usuario recibirá una notificación cuando se envíe la incidencia.	No configurado	NO IMPLEMENTADO		NOK

Requirement No.	Módulo	Prioridad	Test Case Description	Input	Expected Result	Result	Status (OK / NOK)
97	WebApp Notificaciones	P3	El usuario recibirá una notificación cuando se resuelva la incidencia, incluyendo la nota de cierre.	No configurado	NO IMPLEMENTADO		NOK
98	WebApp Coches	P1	Se podrán añadir coches insertando la matrícula y el tipo de coche.	botón Añadir coches	Implementado	Implementado	OK
99	WebApp Coches	P1	Un coche debe pertenecer a un solo usuario.	Añadir Coches Usuario	Implementado	Implementado	OK
100	WebApp Coches	P1	Se podrá borrar un coche.	Borrar Coche	Implementado	Implementado	OK
101	WebApp Coches	P1	Se visualizarán los coches en una tabla	Mis Vehículos	Implementado	Implementado	OK
102	WebApp Perfil	P1	Se visualizarán los datos introducidos por el usuario	Mi perfil	Implementado	Implementado	OK
103	WebApp Perfil	P3	El usuario podrá modificar su contraseña	Editar Perfil	Implementado	Implementado	OK
104	WebApp Perfil	P3	Se indicará el número de incidencias	No configurado	NO IMPLEMENTADO		NOK
105	WebApp Perfil	P3	Se indicará el número de tickets	No configurado	NO IMPLEMENTADO		NOK
106	WebApp Perfil	P3	Se indicará el número de coches	No configurado	NO IMPLEMENTADO		NOK
107	WebApp Ticket	P1	Habrá una opción para abrir un ticket	Open Ticket	Implementado	Implementado	OK
108	WebApp Ticket	P1	Tendrá un QR que podrá ser validado.	QR con Token Único	Implementado	Implementado	OK
109	WebApp Ticket	P1	Se indicará si el ticket es válido o no.	Verificación de validez	Implementado	Implementado	OK

Requirement No.	Módulo	Prioridad	Test Case Description	Input	Expected Result	Result	Status (OK / NOK)
110	WebApp Ticket	P1	Se podrán visualizar todos los tickets en una tabla.	Mis Tickets	Implementado	Implementado	OK
111	WebAPP Ticket	P1	Un ticket quedará invalidado cuando el vehículo haya salido del parking.	Invalidación de Ticket	Implementado	Implementado	OK
112	WebApp Ticket	P1	Se asignará una plaza al usuario cuando entre en el parking.	Se asigna al entrar	Implementado	Implementado	OK
113	WebApp Ticket	P1	Habrá un botón para crear una incidencia.	El botón está implementado	Implementado	Implementado	OK
114	WebApp Ticket	P1	Se visualizarán los datos en un <i>template</i> predefinido.	<i>Template</i> de Ticket Definido (VerTicket.vue)	Implementado	Implementado	OK
115	WebApp Incidencias	P2	Solo un usuario gerente o administrador podrá resolver las incidencias.	Definido en la BBDD	Implementado	Implementado	OK
116	WebApp Incidencias	P3	Se podrán visualizar las incidencias	Configurado	Implementado	Implementado	OK
117	WebApp Incidencias	P3	Una vez rellenado el formulario de incidencia, se asignará una nueva plaza.	No configurado	NO IMPLEMENTADO		NOK
118	WebApp Incidencias	P3	Se podrá subir una foto, enseñando la matricula.	Configurado	Implementado	Implementado	OK
119	WebApp Incidencias	P3	Se podrá resolver una incidencia indicando una nota de cierre.	Configurado, pero no funcional	NO FUNCIONAL	NO FUNCIONAL	OK
120	WebApp Incidencias	P3	Todos los usuarios podrán visualizar sus propias incidencias	Configurado	Implementado	Implementado	OK
121	Base de Datos	P1	La base de datos deberá ser segura	Configurado	Implementado	Implementado	OK

Requirement No.	Módulo	Prioridad	Test Case Description	Input	Expected Result	Result	Status (OK / NOK)
122	<i>Base de Datos</i>	P1	La base de datos tendrá un usuario de administración	Configurado	Implementado	Implementado	OK
123	<i>Base de Datos</i>	P1	La base de datos deberá poder ser administrada utilizando un administrador de base de datos como PhpMy Admin	Configurado	Implementado	Implementado	OK
124	<i>Base de Datos</i>	P1	La base de datos deberá ser construida utilizando una sola migración en Laravel.	Configurado	Implementado	Implementado	OK
125	<i>Base de Datos</i>	P1	La base de datos deberá contener un usuario de administración por defecto	Configurado dbadmin	Implementado	Implementado	OK
126	<i>Base de Datos</i>	P1	Se deberá cargar las configuraciones iniciales de la base de datos con datos prefijados al iniciar una migración.	Configurado	Implementado	Implementado	OK