

CICLO FORMATIVO DE GRADO SUPERIOR EN

Parking Shield

SERVICIO DE GESTIÓN DE APARCAMIENTO E INCIDENTES



Parking Shield

Guillermo Herreros Peláez

Francisco Javier Seleiro Borrero



Índice

1- Introducción	3
2- Resumen del proyecto y justificación.....	3
a) ¿Qué queremos hacer?	3
b) ¿Qué podemos hacer?	3
c) ¿Para qué sirve nuestra aplicación?.....	3
d) ¿Por qué elegimos este proyecto?	3
3- Estudio de mercado y ventajas.....	4
a) Análisis de aplicaciones con funciones similares a la aplicación a desarrollar.....	4
b) Análisis DAFO	4
4- Antecedentes y Estudio del Arte	4
5- Descripción detallada del proyecto.....	8
a) Secciones	8
b) Funcionalidades	8
c) Modelo Entidad Relación.....	8
d) Diagrama de casos de uso.....	8
e) Codificación del proyecto.....	8
6- Metodología	8
7- Herramientas utilizadas	9
a) Android Studio.....	9
b) Visual Studio Code.....	10
c) Java.....	10
d) Python	10
e) Django	10

1-Introducción

Parking Shield tiene como objetivo ser capaz de gestionar zonas de aparcamiento restringidas, salvo autorización eventual. Este proyecto pretende gestionar de forma eficaz los incidentes que puedan ocurrir dentro de estos espacios, así como gestionar las plazas ocupadas y libres. Para ello contará con dos aplicaciones, una móvil y otra web.

La aplicación móvil la usarán los guardias que estén vigilando estos espacios y podrán reportar los vehículos que entran y salen o que están aparcados sin tener permiso para ello, también podrás reportar incidentes en una plaza en concreto o un vehículo en particular.

Por otro lado, la aplicación web es el centro de control, donde se recogen e interpretan estos resultados y se toman decisiones con respecto a estos. Se podrán visualizar toda esta información en tiempo real según los vigilantes vayan reportando con su aplicación móvil.

Toda esta información se envía y se recoge de una base de datos intermedia a la que las dos aplicaciones están conectadas. De esta forma, si se quiere ampliar el servicio con otra aplicación o crear una API en un futuro esta información pueda ser consultada independientemente de la situación de las otras aplicaciones ya implantadas.

2-Resumen del proyecto y justificación

a) ¿Qué queremos hacer?

Aplicación/Web para gestionar las incidencias con los vehículos en un área determinada de responsabilidad de la empresa, pero con acceso público.

Además de las incidencias también debe registrar los vehículos autorizados a aparcar en alguno de los lugares y el horario de la autorización.

b) ¿Qué podemos hacer?

Realizaremos una App/Web que, de acceso a una BBDD, donde se podrán registrar las incidencias, tener acceso a las mismas, comprobar si existen incidencias pasadas con el mismo vehículo y proporcionar estadísticas útiles para mejorar la eficiencia.

Además, proporciona la información de que si un vehículo está autorizado lo muestre.

c) ¿Para qué sirve nuestra aplicación?

Da soporte a los trabajadores de seguridad que controlan las zonas brindándoles de forma sencilla información extra y da acceso rápido a la central para saber las incidencias que han ocurrido y estadísticas para ver las problemáticas crónicas.

d) ¿Por qué elegimos este proyecto?

La elección de este proyecto nos permite desarrollar los conocimientos adquiridos y ponerlos en práctica.

Este proyecto desarrolla todos los aspectos que se requieren y otros extra. Contiene el desarrollo de la aplicación tanto como App nativa como en Web, el uso de BBDD y el acceso a servidores.

Una razón importante para la elección de este proyecto es que se trata de proponer una solución para una problemática real de una empresa.

3- Estudio de mercado y ventajas

Realizar una breve conclusión sobre:

- a) Análisis de aplicaciones con funciones similares a la aplicación a desarrollar

	Parking Shield	ParkOffice
Gestión de aparcamientos	✓	✓
Asignación automática(algoritmo) de plazas	✗	✓
Reasignación de plazas	✓	✓
Gestión de incidencias	✓	✗
Panel de actividades	✓	✓
Creación de informes y estadísticas	✓	✓
Actualizaciones en tiempo real	✓	✓
Gestión de usuarios	✓	✓
Acceso móvil	✓	✗
Portal de autoservicio	✗	✓

- b) Análisis DAFO

Debilidades <ul style="list-style-type: none"> Falta de experiencia comercial Empresa no conocida <p>Añadir Debilidad +</p>	Amenazas <ul style="list-style-type: none"> Mucha competencia en el desarrollo de software Tendencia a infravalorar las herramientas de software Leyes de protección de datos <p>Añadir Amenaza +</p>
Fortalezas <ul style="list-style-type: none"> Dispongo de un buen servicio de atención al cliente Desarrollo personalizado Precio ajustado <p>Añadir Fortaleza +</p>	Oportunidades <ul style="list-style-type: none"> Reducción en los costes tecnológicos Crecimiento de la implantación de soluciones en la nube <p>Añadir Oportunidad +</p>

4- Antecedentes y Estudio del Arte

a) Desarrollo web: Fundamentos y herramientas

Desde la comercialización de Internet a principios de los años 90, el desarrollo web ha estado en constante evolución. Mientras que anteriormente las páginas web eran un cúmulo de campos de texto sobrios que hoy en día la presentación de contenidos multimedia ocupa un lugar privilegiado. La que fue en su momento una plataforma informativa se ha convertido en un medio de entretenimiento. En ella, los usuarios se decantan por páginas web interactivas que convengan a través de un diseño atractivo y que se puedan manejar de manera puramente intuitiva.

Los desarrolladores web cuentan con distintas herramientas que facilitan la integración de contenidos dinámicos y crear las páginas web a medida para los usuarios. En este sentido el requisito es, crear un alojamiento web dinámico que sea capaz de ofrecer los recursos necesarios favoreciendo los estándares actuales.

b) Fundamentos técnicos de una página web

Uno de los requisitos principales para crear una página web desde cero, es tener capacidad de almacenamiento en un servidor, se trata de un servicio que ofrecen los proveedores de internet especializados, es decir, los proveedores de alojamiento web. Por lo general, la oferta comprende packs completos que además del espacio en disco necesario también incluyen prestaciones como la memoria RAM, un dominio propio, bases de datos y las herramientas pertinentes para el desarrollo web. En general, el conocimiento de los dispositivos físicos que se ocupan de alojar los recursos en su aspecto que permanece oculto para los usuarios de los packs de alojamiento web. En el caso de los modelos de alojamiento alternativos como los servidores web dedicados ocurre lo inverso, ya que estos se alquilan en un centro de datos como componentes de hardware independientes.

c) Dominio

Cualquier proyecto de Internet se debe mostrar con un nombre exclusivamente único. El llamado **dominio** es uno de los componentes básicos de los packs de alojamiento web. Los dominios siguen estrictamente la estructura jerárquica del sistema de nombres de dominio y constan de dominios de nivel superior (p. ej., .es), de dominios libres (dominios de segundo nivel) y de subdominios facultativos. El registro de dominios se realiza a través de un proveedor de Internet que remite la solicitud a la autoridad competente. Para la elección del dominio están disponibles, desde formulaciones breves y concisas, hasta dominios de nivel superior comunes y corrientes.

d) Espacio web

Cada página web contiene datos que el servidor web pone a disposición de los usuarios. Por ello, los packs de alojamiento web siempre cuentan con un determinado contingente de almacenamiento para documentos HTML, hojas de estilo, imágenes, vídeos o bases de datos. La transferencia de datos se ejecuta generalmente a través del protocolo FTP y de un programa cliente como Filezilla

o la aplicación WinSCP. Para la elección de un pack de alojamiento web debe tenerse en cuenta que el espacio web pueda ampliarse si es necesario.

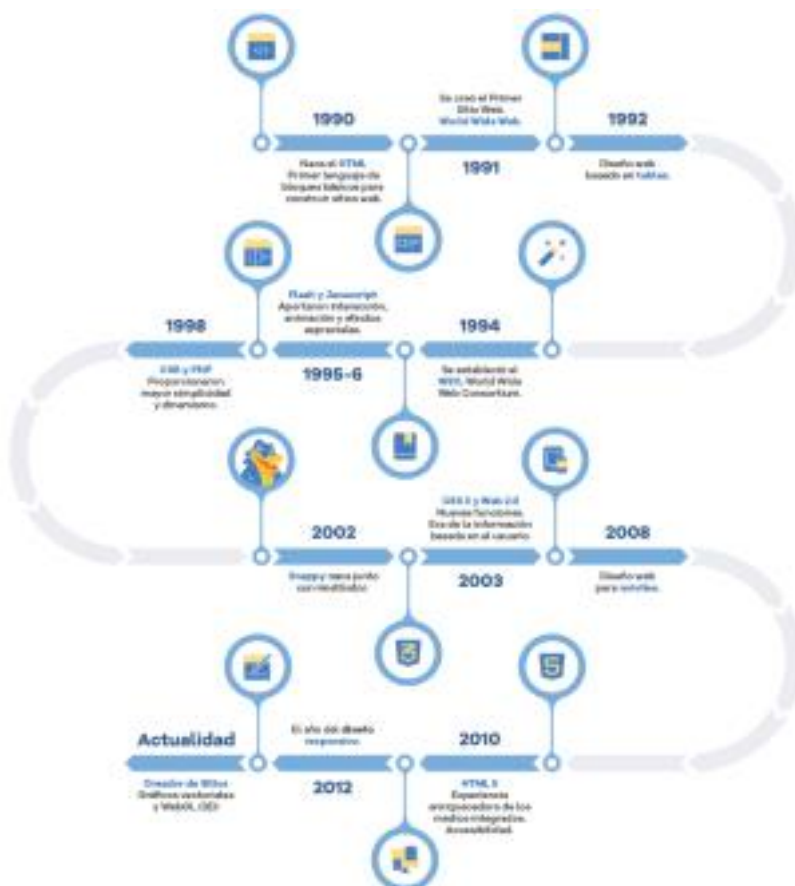
e) Base de datos

Si las páginas web no se ponen a disposición de manera estática en el servidor, sino que estas se generan en función de las necesidades que surgen con cada visita, los desarrolladores web recurren en la mayoría de los casos a bases de datos relacionales. Con esta modalidad de gestión de datos, estos se estructuran por medio de tablas y se establecerá una relación con otros datos a través de identificadores. La transmisión y la gestión de estructuras de datos se realiza, en general, a través del lenguaje para bases de datos SQL. Por otro lado, MySQL, software de código abierto, es una de las bases de datos relacionales para páginas web dinámicas a nivel mundial.

f) Servidor web

Los packs de alojamiento web para principiantes se llevan a cabo a través del llamado alojamiento compartido. Con este tipo de alojamiento, son varias las cuentas de hosting las que pueden compartir los recursos de un mismo servidor. Para los proyectos profesionales de gran envergadura, los proveedores de alojamiento web ofrecen diferentes posibilidades de servidores web que brindan a los clientes componentes de hardware de manera exclusiva. A las empresas grandes se les recomienda, además del alquiler, la adquisición de equipamiento propio.

g) Avance de Internet, década de los 90:



En 1990 nace el lenguaje **HTML**, primer lenguaje para páginas webs. Su objetivo era construir bloques a partir de códigos. Estos códigos de las páginas estaban escritos con el lenguaje HTML, que indicaba dónde colocar cada texto, imágenes o vídeos. También se especificaba la forma en que tenían que ser colocados. Como el HTML se creó con un objetivo divulgativo, no daba respuesta a todos los usos que tendría posteriormente en materia de ocio.

En **1991**, **Tim Berners-Lee** creó la primera página web, y su función fue informar sobre la nueva y emocionante World Wide Web.

En **1995** se crea MySQL, cuyo objetivo principal era ofrecer opciones eficientes y fiables de gestión de datos para los usuarios domésticos y profesionales. Además, se puede utilizar en diferentes sistemas operativos como son Linux, Windows, Solaris etc.

Los programadores web utilizan con gran frecuencia MySQL para poder realizar cambios en los sitios web de manera simple sin tener que modificar el código web. Combinado con PHP se convierte en una poderosa herramienta para realizar aplicaciones que requieran el uso de una base de datos rápida, segura y potente.

En **1996**, aparecieron el **Flash & JavaScript**. Fueron la introducción para llegar a conseguir una web mucho más interactiva ya que supusieron toda una explosión en temas de diseño y de animación pues eran páginas basadas en elementos móviles, animaciones e imágenes.

Las páginas web modernas interactúan con sus usuarios a través de la integración de casillas interactivas, menús desplegables o diapositivas. JavaScript entra en acción en caso de que los contenidos dinámicos de esta índole se integren en páginas HTML y ello supone dar un paso más allá de HTML y CSS, puesto que permite que se evalúen las interacciones de los usuarios y que los contenidos de las páginas web se recarguen, se actualicen o se modifiquen.

En **1998** se introdujeron dos novedades: el **CSS Design** y el **PHP Websites**. El CSS es un lenguaje de hojas de estilo estructurado y escrito en un lenguaje de marcado. Muy usado para el diseño visual de las páginas web, pero también para estilos no visuales como las hojas de estilo auditivas.

Mientras que las páginas web estáticas están disponibles en el servidor web, las dinámicas se generan, tras su solicitud, a través de dicho servidor, lo que es posible gracias a lenguajes de programación como Python. La programación con Python es interpretada por el servidor y, en lugar de que el código fuente de las páginas web dinámicas se le entregue directamente a dicho servidor, este transmite los datos a un intérprete de Python. Dicho intérprete genera la página web solicitada, a menudo tomando como base los datos depositados en las bases de datos, y los devuelve al servidor web, desde donde se enviará la edición del intérprete al servidor. Actualmente Python se encuentra en su tercera versión, siendo compatible con versiones anteriores mejorando el rendimiento con **CMS** y la gestión de errores. **Perl** o **PHP** son otros lenguajes de programación utilizados para el desarrollo de páginas web dinámicas.

h) Evolución del diseño web: telefonía, web 2.0 y diseño responsive.

En **2003** se introdujo el **CSS3** y la **Web 2.0**. El CSS3 es una evolución mucho más mejorada del CSS. Es una tecnología desarrollada para separar la presentación de la estructura HTML. Este lenguaje se ha ido haciendo importante por su facilidad de uso y los resultados flexibles. En este año, también apareció por primera vez el concepto de Web 2.0 o Web Social. Son los sitios web cuyo diseño está centrado en el usuario, para que comparta información y pueda interactuar con la página web.

En **2010** aparece el **HTML5**, la quinta revisión del lenguaje HTML. Cambia los paradigmas de diseño web pues introduce herramientas como la publicación de archivos de audio y vídeo con soportes diferentes, etiquetas para que los usuarios dibujen en 2D y 3D, cambios en formularios y, además, se aprovecha más la web semántica.

A raíz de la popularidad de los teléfonos móviles, en **2012** surge la web **responsive**. Se trata de una técnica de diseño web que busca la buena visualización junto a la adaptación de la web en distintos dispositivos: ordenadores, tablets y teléfonos móviles de diversos tamaños. Ahora ya no solo se visualizan por el ordenador sino por cualquier dispositivo.

5- Descripción detallada del proyecto

- a) Secciones
- b) Funcionalidades
- c) Modelo Entidad Relación
- d) Diagrama de casos de uso
- e) Codificación del proyecto

6- Metodología

a) Planificación del proyecto

A continuación, podemos diferenciar las 3 fases de desarrollo del proyecto:

FASE INICIAL	FASE INTERMEDIA	FASE FINAL
1. Elección del tema a tratar.	1. Dibujo boceto de la web.	1. Esquema de la memoria.

2. Planificación de:	2. Comienzo del diseño.	2. Reparto partes memoria.
2.1 Back-end.	3. 1º Puesta en común	3. Comienzo memoria.
2.2 Front-end.	4. 1º Corrección de errores	4. Correcciones aplicación.
2.3 Base de datos.	5. Avance en el desarrollo	5. Consenso de la memoria.
3. Elección de lenguajes.	6. 2º Puesta en común	6. Correcciones memoria.
4. Revisión bibliográfica.	7. Prueba de funcionalidad	7. Repaso global aplicación.
5. Reparto de tareas.	8. 2º Corrección de errores	8. Repaso global memoria.
6. Aprendizaje nuevas técnicas. (Python, PHP,...)	9. Avance en el desarrollo y prueba de la aplicación.	9. Unión de todo el trabajo listo para su entrega.

b) Método: Revisión bibliográfica

Se ha realizado una investigación documental previa al diseño del proyecto, ya que era muy necesario consultar el estado actual del tema a tratar, viendo así las carencias actuales, así como las posibles necesidades descubiertas en cuanto al usuario se refiere.

Partiendo de esta base, las carencias más comunes y destacadas son las siguientes:

1. Diseño desestructurado y poco intuitivo.
2. Disparidad de colores.
3. Falta o escasez de funcionalidad.
4. Falta de información para el usuario.

Contrastando con estas necesidades se ha continuado con la búsqueda de información, pero esta vez sobre todo lo contrario; páginas bien desarrolladas con gran intuición para el usuario y con un gran índice de demanda online relacionadas con la **restauración**.

Esta búsqueda nos permite contrastar los puntos débiles de la búsqueda inicial con los puntos fuertes de ésta para poder tener diversos ejemplos de inspiración a la vez que se eliminan los errores y se perfeccionan o potencian dichos puntos fuertes.

7- Herramientas utilizadas

El hardware utilizado para desarrollar este proyecto han sido nuestros propios ordenadores y teléfonos móviles.

En cuanto al software, hemos necesitado lo siguiente:

- a) Android Studio
Android Studio para como IDE de la aplicación de Android.

- b) Visual Studio Code
Visual Studio Code como IDE para el desarrollo de la web que monitoriza la actividad.
- c) Java
Java como lenguaje para el desarrollo Android.
- d) Python
Python como lenguaje para el desarrollo de la web.
- e) Django
Django como framework de Python.