

CICLO FORMATIVO DE GRADO SUPERIOR EN

# Desarrollo Aplicaciones Multiplataforma

SERVICIO DE GESTIÓN DE APARCAMIENTO E INCIDENTES

## Parking Shield



## Parking Shield

Equipo 3

Guillermo Herreros Peláez

Francisco Javier Seleiro Borrero

Curso 21/22 Noviembre 2021



## Índice

<b>1- Introducción .....</b>	<b>3</b>
<b>2- Resumen del proyecto y justificación.....</b>	<b>3</b>
a) ¿Qué queremos hacer? .....	3
b) ¿Qué podemos hacer? .....	3
c) ¿Para qué sirve nuestra aplicación?.....	3
d) ¿Por qué elegimos este proyecto? .....	3
<b>3- Estudio de mercado y ventajas.....</b>	<b>4</b>
a) Análisis de aplicaciones con funciones similares a la aplicación a desarrollar.....	4
b) Análisis DAFO .....	4
<b>4- Antecedentes y Estudio del Arte .....</b>	<b>5</b>
<b>5- Descripción detallada del proyecto.....</b>	<b>8</b>
a) Secciones .....	8
b) Funcionalidades .....	9
Web: .....	14
c) Modelo Entidad Relación.....	16
.....	16
d) Diagrama de casos de uso.....	17
App:.....	17
e) Codificación del proyecto.....	19
<b>6- Metodología .....</b>	<b>22</b>
<b>7- Herramientas utilizadas .....</b>	<b>24</b>
• Android Studio .....	24
• Visual Studio Code .....	24
• Java.....	24
• Python .....	24
• Django .....	24
• PHPMyAdmin .....	24
• HTML Hel Workshop .....	24
<b>8- Conclusiones.....</b>	<b>24</b>
a) Trabajo realizado .....	24
b) Lecciones aprendidas .....	25
c) Trabajo pendiente .....	25
d) Opinión personal .....	25

# 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	✓	✓
Gestion 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

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

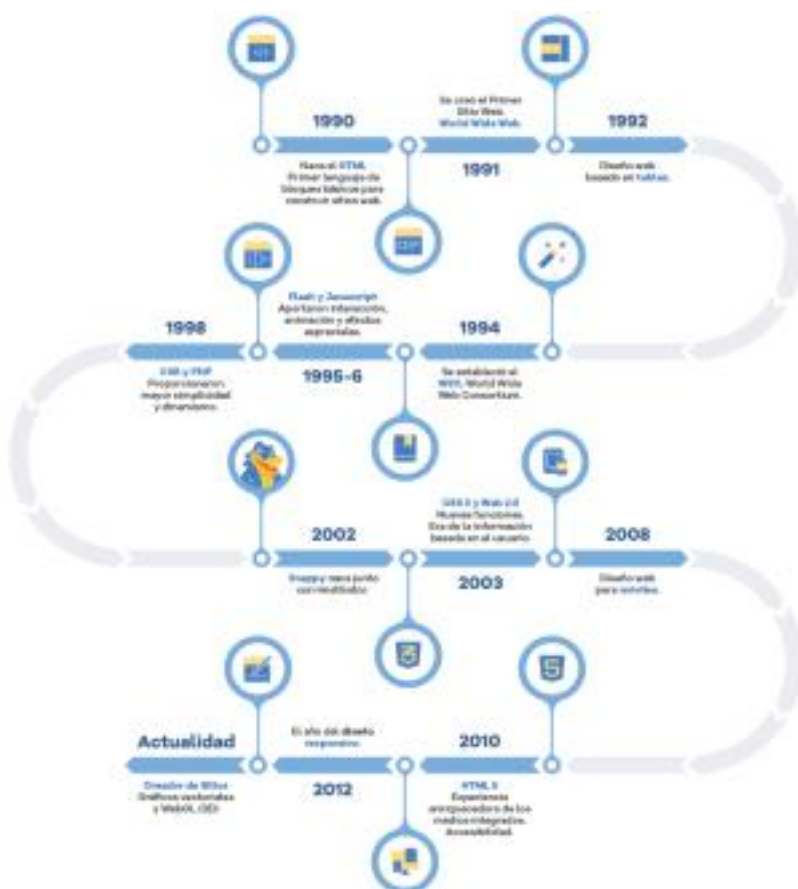
estilo, imágenes, vídeos o bases de datos. La transferencia de datos se ejecuta generalmente a través del protocolo FTP y de un programa cliente como Filezilla o la aplicación WinSCP. Para la elección de un pack de alojamiento web debe tenerse en cuenta que el espacio web pueda ampliarse si es necesario.

### 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 estructurarán 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 desplegable 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**

El proyecto pretende dar solución a la problemática organizativa de las zonas de aparcamiento permitido y restringido, acelerando la comunicación y permitiendo el acceso a la información entre el personal de seguridad y los responsables de zona. Para ello se divide en dos grandes secciones, una app para el personal de seguridad y un acceso desde web para los responsables de zona.

### **a) Secciones**

#### **App:**

En la app nos encontramos con una serie de “activities” a través de la cual se desarrolla el programa. Estas se dividen en 3 secciones a su vez. La primera introduce a la aplicación a través de un “splash” y está compuesto por “activity main”.

La segunda sección esta compuesta por “activity\_login” y “activity\_registro” y hacen de acceso al programa.

La tercera sección estaría compuesta “activity\_second”, “activity\_third” y “activity\_update” y son en las que accedemos a las funcionalidades principales de la App.



## Web:

Lo primero que nos encontramos en la web es el login, ya que no se puede acceder a ninguna otra sección de la web sin estar logeado. Una vez logeado podemos ver la página principal donde se mostrarán todos los incidentes, cada incidente dispone de un botón “Ver” y “Editar” para ver en más detalle el incidente o editarlo.

En la vista ver\_placa.php veremos un formulario con los campos deshabilitados, pero veremos más detalles que en la vista home.php.

En la vista editar\_placa.php veremos el mismo formulario que en la vista ver\_placa.php pero con los campos habilitados y podremos modificarlos para después pulsar editar.

Por último, hay una sección donde hay un gráfico con el número de incidentes por mes.

## b) Funcionalidades

### App:

**Login:** Tenemos una pantalla para iniciar sesión en la aplicación.

**Registro:** Podemos registrar un nuevo usuario y su contraseña que deberá ser validado por los administradores.

**Uso:** Una vez ya estamos registrados desde la pantalla de inicio introducimos nuestro usuario y contraseña y presionamos el botón Acceder. En caso de error nos pondremos en contacto con nuestro CIS-POC.



A screenshot of a mobile application login and registration screen. The screen is white with a black border. It features two input fields: the top one is labeled 'usuario' and the bottom one is labeled 'Clave'. Below the 'Clave' field, there are two blue buttons with white text: 'ACCEDER' and 'REGISTRARSE'.

**Registrar vehículo:** En esta sección podemos introducir los datos de un nuevo vehículo como matricula, modelo, color, ubicación (Donde está o debería estar aparcado), intervención(Si es necesario que realice alguna acción) y comentarios(Datos que pueda ser de interés recordar).

Uso: Cuando accedemos a la aplicación después del splash tenemos acceso a los botones de ACCEDER y REGISTRARSE.

Pulsamos el botón de REGISTRARSE.

A screenshot of a mobile application's login and registration screen. The screen is white with a thin black border. It features two input fields: the top one is labeled 'usuario' and the bottom one is labeled 'Clave'. Below the 'Clave' field, there are two blue buttons with white text. The first button is labeled 'ACCEDER' and the second button, positioned below the first, is labeled 'REGISTRARSE'.

Una vez en la pantalla de registro debemos introducir los datos que deseemos sean nuestro usuario y contraseña y presionamos el botón REGISTRAR.

usuario

clave

REGISTRAR

VOLVER

---

Habremos creado nuestra información de acceso y solo tendremos que presionar VOLVER para volver a la pantalla anterior, desde la que podremos acceder.

**Buscar vehículo:** Insertando en la aplicación la matrícula podemos buscar la información sobre ese vehículo que haya sido registrada con anterioridad. Esta información consta de la matrícula, modelo, color, ubicación(Donde está o debería estar aparcado), intervención(Si es necesario que realice alguna acción) y comentarios(Datos que pueda ser de interés recordar).

Uso:

Dentro de la aplicación ya hecho el Login nos encontramos con la pantalla de búsqueda, introduciremos la matrícula que queremos buscar y le daremos al botón BUSCAR.



Si la búsqueda encontrara resultados aparecerían en pantalla, en caso contrario nos saldría un pop-up de que no se han encontrado resultados.

**Actualizar vehículo:** Una vez hemos buscado un vehículo y estamos viendo sus datos podemos cambiar a la pantalla de actualizar datos para registrar nuevos datos o actualizar los existentes.

Uso:

Si al ver los datos de un vehículo que hemos buscado queremos modificar algo solo tenemos que hacer clic sobre el campo. Nos aparecerá la siguiente pantalla, y solo tendremos que introducir o modificar los campos que queramos actualizar y darle al botón de ACTUALIZAR.



The screenshot shows a mobile application interface for updating vehicle data. It features a vertical list of input fields, each with a placeholder text 'Introduce los datos...' and a label below it. The labels are 'Color', 'Ubicación', 'Intervención', and 'Comentarios'. At the bottom of the form is a blue button with the text 'ACTUALIZAR'.

## Web:

Buscar placas: buscará todas las placas que se encuentren almacenadas en la base de datos y las mostrará en la página principal en una tabla.

### Welcome Guillermo

[Logout](#)

Matrícula	Modelo	Color	Ubicación	Intervención	Solucionado	Ver	Editar
1234CHT	BMW	Azul	A12	Multado	Sí	<a href="#">Ver</a>	<a href="#">Editar</a>
1234a	mod	col	ubi	int	Sí	<a href="#">Ver</a>	<a href="#">Editar</a>
4321	b	b	b	b	Sí	<a href="#">Ver</a>	<a href="#">Editar</a>
1639HTM	Peugeot 508.	negro	casarrubios del monte (Toledo)	grua no se lleva cacamovil nunca	No	<a href="#">Ver</a>	<a href="#">Editar</a>
8474LBP	BMW 335D	gris	RR6	Ivan	No	<a href="#">Ver</a>	<a href="#">Editar</a>
7912HGY	piaggio liberty 125	Amarilla	princesa 94	nacional localizar propietario	No	<a href="#">Ver</a>	<a href="#">Editar</a>

© 2021 Copyright: [f](#) [t](#) [i](#) [c](#)

Ver placa: buscará una placa por el id y mostrará los detalles en un formulario con los campos deshabilitados.

### Solucionar

Matrícula

Modelo

Color

Ubicación

Intervención

Comtenarios

Solucionado

[Volver](#)



Editar placa: buscará una placa por el id y mostrará los detalles en un formulario con los campos habilitados, si el usuario pulsa en “Editar” se hará una consulta a la base de datos para actualizar ese registro.

## Editar Placa

Matrícula

Modelo

Color

Ubicación

Intervención

Comtenarios

Solucionado




Login: hace una consulta a la base de datos para comprobar que el usuario existe y que además es administrador.

## Login

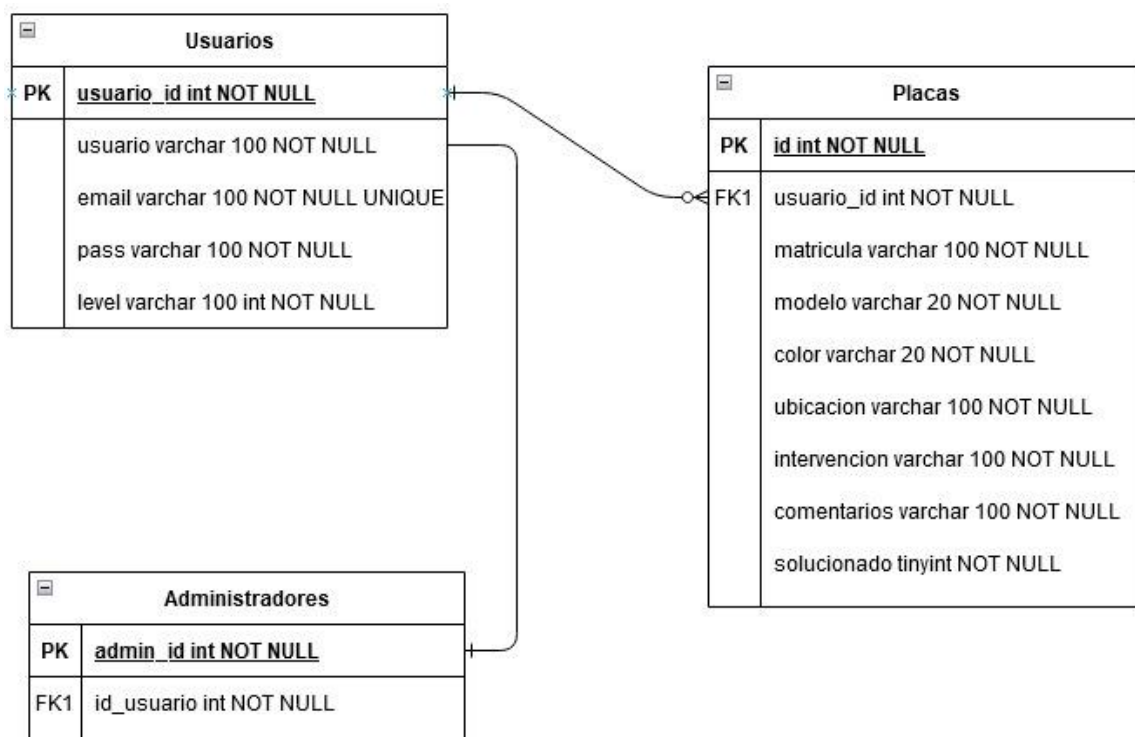
Email

Password



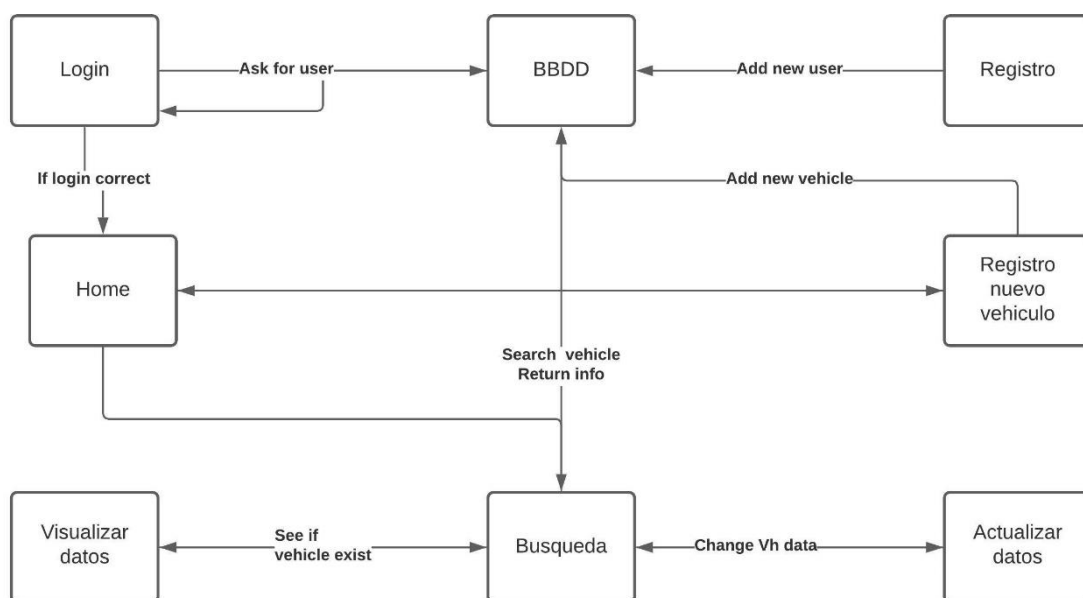
Los administradores no podrán eliminar ningún registro, solamente podrán validar si se ha solucionado o no.

### c) Modelo Entidad Relación

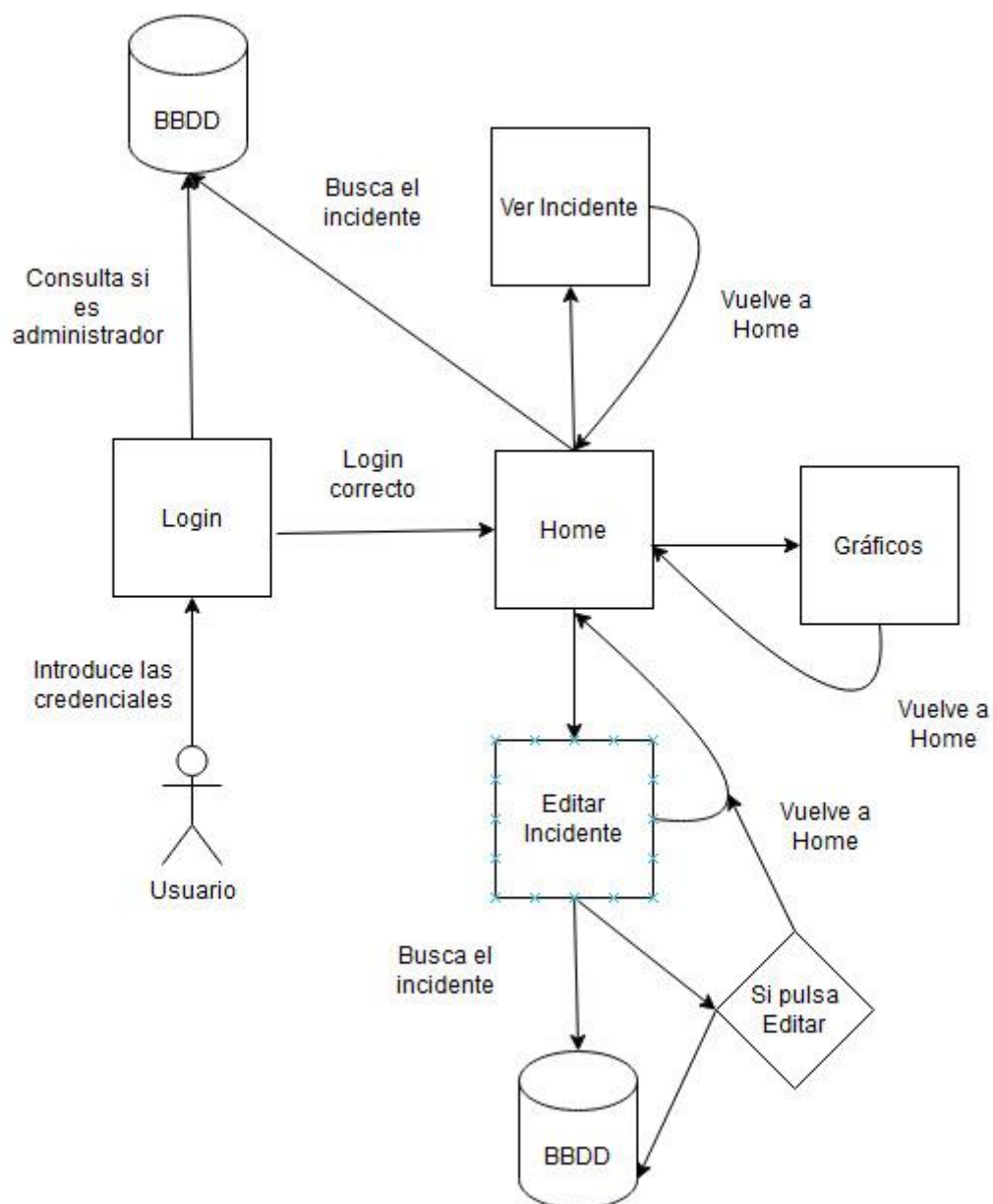


## d) Diagrama de casos de uso

**App:**



**Web:**



## e) Codificación del proyecto

### App:

Conexión a la BBDD:

```
package es.ifp.splash;

import ...

public class DBConnection extends StringRequest {

    private static final String ruta = "https://splashplacas.000webhostapp.com/androidstudio/SplashPlacas/registro.php";
    private final Map<String, String> parametros;

    public DBConnection(String usuario, String clave, String level, Response.Listener<String> listener) {
        super(Request.Method.POST, ruta, listener, errorListener: null);
        parametros = new HashMap<>();
        parametros.put("usuario", usuario + "");
        parametros.put("clave", clave + "");
        parametros.put("level", level + "");
    }

    @Override
    protected Map<String, String> getParams() { return parametros; }
}
```

Control de acceso por permisos(level):

```
@Override
public void onResponse(String response) {
    try{
        JSONObject jsonRespuesta = new JSONObject(response);
        boolean ok = jsonRespuesta.getBoolean( name: "success");
        if(ok==true)
        {
            String usuario= jsonRespuesta.getString( name: "usuario");
            int level= jsonRespuesta.getInt( name: "level");
            if(level==1 || level==0)
            {
                Intent secondActivity = new Intent ( packageContext LoginActivity.this, SecondActivity.class);
                secondActivity.putExtra( name: "usuario", usuario);
                secondActivity.putExtra( name: "level", level);

                LoginActivity.this.startActivity(secondActivity);
                LoginActivity.this.finish();
            }
            if(level==2)
            {
                Toast.makeText( context LoginActivity.this, text "Cuenta aún no activada, contacte con el administrador", Toast.LENGTH_LONG).show();
            }
        }
    }
}
```

Botón de búsqueda y devolución del objeto json y creación del ListView:

```

boton1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String matricula = caja1.getText().toString();
        String URL = "https://splashplacas.000webhostapp.com/androidstudio/SplashPlacas/search.php?busqueda="+matricula;
        JSONArrayRequest jsonArrayRequest=new JSONArrayRequest(URL, new Response.Listener<JSONArray>() {
            @Override
            public void onResponse(JSONArray response) {
                JSONObject jsonObject = null;
                for (int i = 0; i < response.length(); i++) {
                    try {
                        jsonObject = response.getJSONObject(i);
                        adaptador=new ArrayAdapter<String>(context: SecondActivity.this, android.R.layout.simple_list_item_1, placas);
                        lista1.setAdapter(adaptador);
                        adaptador.clear();
                        placas.add(jsonObject.getString( name: "matricula"));
                        String modelo=jsonObject.getString( name: "modelo");
                        placas.add(modelo);
                        String color=jsonObject.getString( name: "color");
                        placas.add(color);
                        String ubicacion=jsonObject.getString( name: "ubicacion");
                        placas.add(ubicacion);
                        String intervencion=jsonObject.getString( name: "intervencion");
                        placas.add(intervencion);
                        String comentarios=jsonObject.getString( name: "comentarios");
                        placas.add(comentarios);
                        String escolta=jsonObject.getString( name: "escolta");
                        placas.add(escolta);
                        String idDB=jsonObject.getString( name: "id");
                    } catch (JSONException e) {
                        e.printStackTrace();
                    }
                }
            }
        });
    }
});

```

Método tras el botón guardar, muy similar a actualizar:

```

private void guardarPlaca(String URL){
    StringRequest stringRequest=new StringRequest(Request.Method.POST, URL, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            Toast.makeText(context: ThirdActivity.this, text: "Guardado correcto", Toast.LENGTH_SHORT).show();
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Toast.makeText(getApplicationContext(), error.toString(), Toast.LENGTH_SHORT).show();
        }
    });
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String,String> parametros=new HashMap<>();
        parametros.put("matricula",caja1.getText().toString());
        parametros.put("modelo",caja2.getText().toString());
        parametros.put("color",caja3.getText().toString());
        parametros.put("ubicacion",caja4.getText().toString());
        parametros.put("intervencion",caja5.getText().toString());
        parametros.put("comentarios",caja6.getText().toString());
        parametros.put("escolta",label7.getText().toString());
        return parametros;
    }
};
RequestQueue requestQueue= Volley.newRequestQueue(context: this);
requestQueue.add(stringRequest);

```

**WEB:**



## Conexión a la BBDD:

```
<?php
//import Placa model
require_once 'Placa.php';
//create model database
class Database{
    private $host = "localhost";
    private $db_name = "id16377610_logindb";
    private $username = "root";
    private $password = "";
    public $conn;
    // get the database connection
    public function getConnection(){
        $this->conn = null;
        try{
            $this->conn = new PDO("mysql:host=" . $this->host . ";dbname=" . $this->db_name, $this->username, $this->password);
            $this->conn->exec("set names utf8");
        }catch(PDOException $exception){
            echo "Connection error: " . $exception->getMessage();
        }
        return $this->conn;
    }
}
?>
```

## Login:

```
public function login(){
    //if user is not logged in
    echo "entra en el login";
    if(!isset($_SESSION['username'])){
        echo "no hay usuario";
        //if user submits form
        if($_SERVER['REQUEST_METHOD'] == 'POST'){
            //get email and password from form
            $email = $_POST['email'];
            $pass = $_POST['password'];
            $admin_controller = new AdminController();
            //get admin from database
            $admin = $admin_controller->getAdminByEmail($email);

            //if admin exists
            if($admin){
                echo "hay admin";
                //if password is correct
                if ($pass == $admin->getPass()){
                    //set session variables
                    session_start();
                    $_SESSION['user_id'] = $admin->getId();
                    $_SESSION['email'] = $admin->getEmail();
                    $_SESSION['username'] = $admin->getUsuario();
                    //redirect to dashboard
                    header('Location: home.php');
                }else{
                    //set error message
                    echo '<div class="alert alert-danger">Wrong password</div>';
                }
            }else{
                //set error message
                echo '<div class="alert alert-danger">Admin does not exist</div>';
            }
        }
    }else{
        //redirect to dashboard
        header('Location: home.php');
    }
}
```

Control de inicio de sesión:

```
session_start();
if(!isset($_SESSION['username'])){
    header("Location: login.php");
}
```

Select de todas las placas a la base de datos:

```
//get all placas
public function getPlacas(){
    $db = new Database();
    $db->query('SELECT * FROM placa');
    $placas = $db->resultSet();
    return $placas;
}
```

Update de una palca a la base de datos:

```
//update placa
public function updatePlaca($id, $matricula, $modelo, $color, $ubicacion){
    $db = new Database();
    $db->query('UPDATE placa SET matricula = :matricula, modelo = :modelo, color = :color, ubicacion = :ubicacion WHERE id = :id');
    $db->bind(':id', $id);
    $db->bind(':matricula', $matricula);
    $db->bind(':modelo', $modelo);
    $db->bind(':color', $color);
    $db->bind(':ubicacion', $ubicacion);
    $db->execute();
}
```

<https://github.com/Laetian/ProyectoDAM>

## 6- Metodología

### a) Tipo de metodología

Metología Agil:

Por definición, las metodologías ágiles son aquellas que permiten adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo flexibilidad e inmediatez en la respuesta para amoldar el proyecto y su desarrollo a las circunstancias específicas del entorno.

## b) Planificación del proyecto

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

Cronograma:

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.

## c) 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:

- **Android Studio**  
Android Studio como IDE de la aplicación de Android.
- **Visual Studio Code**  
Visual Studio Code como IDE para el desarrollo de la web que monitoriza la actividad.
- **Java**  
Java como lenguaje para el desarrollo Android.
- **Python**  
Python como lenguaje para el desarrollo de la web.
- **Django**  
Django como framework de Python.
- **PHPMyAdmin**  
Utilizada para las bases de datos.
- **HTML Hel Workshop**  
Utilizada para la documentación

## 8-Conclusiones

### a) Trabajo realizado

El proyecto se encuentra ya en su fase final, el código tanto de la App móvil como de la web se encuentra casi terminado a falta de posibles retoques fundamentalmente estéticos.

En la parte de la App tenemos ya todas las clases definidas con las diferentes “activities”, la interacción entre ellas es correcta y el paso de una a otra es el adecuado. Se han corregido errores encontrados durante las pruebas que producían el error de funcionamiento de la App y se ha variado el estilo para que sea más acorde con el Logo y la imagen que queremos que dé.

En la parte de la base de datos el trabajo está también terminado, los campos definidos y las relaciones entre la aplicación y la BBDD es el adecuado.

## **b) Lecciones aprendidas**

Durante la realización del proyecto hemos aprendido a superar numerosas dificultades técnicas, durante el desarrollo de la App me encontré con numerosos escollos a la hora de trabajar con una base de datos en la nube, que te obliga a aprender métodos diferentes a los aprendidos durante el curso. También la rápida evolución de las tecnologías, al realizar partes con métodos aprendidos de tutoriales que aún siendo relativamente recientes ya estaban “deprecated”, teniendo que buscar otros nuevos y la lectura durante horas de documentación técnica.

Por todo ello indicaría que la principal lección aprendida ha sido la de que en este mundo hay que estar siempre actualizándose y aprendiendo, que siempre vas a tener que aplicar nuevas tecnologías.

## **c) Trabajo pendiente**

Los trabajos pendientes en la App son principalmente los de actualización y mantenimiento del funcionamiento del programa. El sistema está en un estado funcional, a la espera de más pruebas no hay fallos detectados que corregir.

Se esperan también trabajos que realizar a petición del cliente, que desee modificaciones o ajustes.

Otro trabajo a realizar sería la refactorización de parte del código para mejorar su mantenimiento a futuro.

## **d) Opinión personal**

En mi opinión personal trabajando con la App ha sido un trabajo desafiante que ha pedido utilizar mucho de lo que se de programación y aplicar muchas cosas nuevas, investigación constante y un desafío ante el que no nos podemos rendir. Darme cuenta de que lo que sé no es nada comparado con todo lo que tengo que aprender todavía.