



2025

# RENT CAR VIR

I.E.S Fuengirola N°1

DESARROLLO DE  
APLICACIONES WEB



PROYECTO INTEGRADO

VIRGINIA PEREZ TOLEDO

## ÍNDICE DE CONTENIDOS

1 - Sobre éste proyecto	2
1.1 - Control de versiones	2
1.2 - Licencia de uso	2
2 - Análisis del problema	3
2.1 - Introducción al problema	3
2.2 - Objetivos	3
2.3 - Recursos	4
2.3.1 - Software	4
3 - Diseño de la solución software	4
3.1 - Analysis	4
3.2 – Estructura de la navegacion	5
3.3 - Diseño	5
3.3.1 – Estructura de ficheros y carpetas	6
3.4 – Base de datos	11
3.4.1 – Diseño entidad-realacion	11
3.4.2 – Diseño logico	12
4 - Implementación	13
4.1 - Codificación	14
4.2 – Back-end	14
4.3 – Front-end	14
5 – Distribucion	15
5.1 – Posibles pasos para la distribucion	15
6 - Bibliografía	15

## 1 - Sobre éste proyecto

En este primer apartado me gustaría presentar mi proyecto, al cual he llamado **Rent Car Vir**.

En breves palabras, **Rent Car Vir** es una aplicación web que facilita el alquiler de vehículos en la localidad de Fuengirola, mejorando la experiencia del usuario y modernizando la presencia digital de las empresas del sector.

Fuengirola es una de las localidades más turísticas de la Costa del Sol, especialmente en temporada alta, cuando miles de visitantes buscan movilidad para desplazarse por la zona. Sin embargo, a diferencia de otras ciudades cercanas como Málaga o Marbella, la oferta de empresas de alquiler de coches en Fuengirola es limitada, y muchas de ellas carecen de una presencia digital moderna que facilite las reservas online.

Mi proyecto plantea el desarrollo de una **aplicación web para una empresa de rent-a-car en Fuengirola**, con el fin de cubrir esa carencia en el mercado local. Esta herramienta permitirá tanto la **gestión interna de vehículos, reservas y usuarios**, como una **plataforma accesible para clientes**, donde podrán alquilar coches de forma rápida y sencilla desde cualquier dispositivo.

### 1.1 - Control de versiones

Rent Car Vir cuenta con un **control de versiones** mediante Git, una herramienta ampliamente utilizada en el desarrollo de software para gestionar cambios en el código fuente de manera eficiente.

El proyecto dispone de un **repositorio remoto** alojado en GitHub, lo que permite a los desarrolladores mantener un historial de modificaciones, colaborar con otros miembros del equipo y garantizar la integridad del código. Este control de versiones facilita el trabajo en equipo, la detección de errores y la implementación de nuevas características de manera estructurada.

Si deseas acceder al repositorio de Rent Car Vir aquí está el enlace:  
<https://github.com/virginiapereztoledo/ProyectoFinalRentCarVir.git>

### 1.2 Licencia de uso

Rent car Vir es una aplicación gratuita para todo el mundo, pero posee una licencia muy restrictiva, licencia concedida por Creative Commons.

Nuestra licencia es la siguiente :

Reconocimiento-NoComercial-SinObraDerivada CC BY-NC-ND

Rent Car Vir © 2025 by Virginia Perez is licensed under [CC BY-NC-ND 4.0](#)

Rent Car Vir © 2025 by Virginia Perez is licensed under [Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International](#)

## 2. Análisis del programa

En los siguientes apartados analizaremos los requisitos de nuestra aplicación, además de plantear los objetivos que se persiguen y cómo resolveremos estos desafíos en Rent Car Vir.

### 2.1 Introducción al problema

Como se mencionó anteriormente, **Rent Car Vir** nace de la necesidad de modernizar el alquiler de vehículos en Fuengirola, ofreciendo una solución accesible y eficiente tanto para clientes como para administradores.

Actualmente, muchas empresas de alquiler de coches en la zona **carenecen de una presencia digital moderna**, lo que dificulta la gestión de reservas y reduce la accesibilidad para los usuarios. Además, la falta de un sistema centralizado de control interno puede generar problemas en la administración de vehículos y clientes.

Nuestra aplicación busca **resolver estos problemas** proporcionando una plataforma digital intuitiva y funcional que permita a las empresas gestionar sus servicios de manera eficiente, mejorando la experiencia de los clientes y la operatividad del negocio.

### 2.2 Objetivos

El objetivo principal de Rent Car Vir es **crear una aplicación web** que permita a los usuarios **alquilar coches de forma sencilla** y a los administradores **gestionar vehículos, reservas y clientes** de manera eficiente.

#### Objetivos específicos:

- **Facilitar el acceso a los vehículos disponibles** mediante una plataforma digital clara y organizada.
- **Optimizar la gestión interna** de las empresas de rent-a-car en Fuengirola.
- **Permitir reservas online** sin necesidad de trámites físicos o llamadas telefónicas.
- **Establecer roles específicos** (Administrador, Empleado y Cliente) para mejorar la administración de la plataforma.
- **Garantizar la escalabilidad del sistema**, permitiendo futuras mejoras y adaptaciones.
- **Implementar un sistema de rastreo GPS** en los vehículos alquilados para una mejor gestión de la flota.

## 2.3 Recursos

Para el desarrollo de **Rent Car Vir**, se han seleccionado diversas herramientas de software que permitirán implementar todas las funcionalidades de la aplicación de manera eficiente.

### 2.3.1 Software

Necesidad	Herramienta elegida
Sistema operativo	Windows 11
Front-end	PHP, Laravel, Bootstrap, JavaScript, CSS, HTML
Back-end	PHP, Laravel
Base de datos	MySQL
Control de versiones	Git
Repositorio remoto	GitHub
Editor de código	Visual Studio Code, XAMPP

## 3. Diseño de la solución de software

### 3.1 Análisis

Analicemos en profundidad nuestra aplicación, por qué la necesitamos y qué problema intenta resolver.

El problema principal que **Rent Car Vir** busca solucionar es la **falta de presencia digital moderna** en las empresas de alquiler de coches en la localidad, lo que dificulta la accesibilidad y gestión del servicio tanto para clientes como para administradores.

Actualmente, el proceso de alquiler de vehículos en Fuengirola es **poco eficiente y poco digitalizado**, con muchas empresas que dependen de métodos tradicionales para gestionar reservas y vehículos. Esto genera **procesos lentos, menor alcance de clientes y problemas administrativos** que pueden afectar la experiencia del usuario.

Nuestra solución consiste en una **plataforma web moderna y responsive** que permite tanto la gestión interna de los vehículos como la interacción con los clientes, asegurando un proceso de alquiler **rápido, seguro y accesible desde cualquier dispositivo**.

Además, se implementará un **sistema de rastreo GPS** que permitirá a la empresa ubicar los coches alquilados en tiempo real, optimizando la gestión de flota y aumentando la seguridad del servicio.

### 3.2 Estructura de la navegación

La aplicación **Rent Car Vir** presenta una interfaz clara y organizada, con una división de secciones mediante pestañas para facilitar la navegación de los usuarios.

Para usuarios no registrados, la barra de navegación incluirá:

- **Home:** Página principal con información general sobre el servicio.
- **Condiciones:** Términos y condiciones del alquiler.
- **Contacto:** Formulario de contacto y datos de la empresa.
- **Catálogo:** Lista de vehículos disponibles con detalles.
- **Quiénes somos:** Información sobre la empresa.
- **Registrarse:** Opción para crear una cuenta en la plataforma.
- **Iniciar sesión:** Botón para que los usuarios accedan a sus cuentas.

Para usuarios registrados, la barra de navegación mantendrá las mismas pestañas, pero con modificaciones:

- Se **eliminarán los botones de "Iniciar sesión" y "Registrarse"**, ya que el usuario ya estará autenticado.
  - Se añadirá una nueva pestaña:
    - **Zona personal:** Espacio privado donde los clientes podrán gestionar sus reservas, modificar datos personales y consultar el estado de alquiler de sus vehículos.
- Salir:** Opción para cerrar sesión y salir de la cuenta.

### 3.3 Diseño

#### Dependencias del programa

Para desarrollar **Rent Car Vir**, se han seleccionado diversas herramientas y tecnologías que garantizan una gestión eficiente de la plataforma y una experiencia óptima para los usuarios.

Las herramientas y dependencias clave son:

- **PHP y Laravel:** Base del backend para gestionar la lógica del negocio y la interacción con la base de datos.

- **HTML, CSS y Bootstrap:** Estructura y diseño responsive para una interfaz amigable.
- **JavaScript:** Funcionalidad interactiva en el front-end.
- **Git y GitHub:** Control de versiones y almacenamiento de código en un repositorio remoto.
- **XAMPP:** Entorno de desarrollo local con Apache y MySQL.
- **phpMyAdmin:** Herramienta para gestionar la base de datos MySQL de forma gráfica.
- **API de Geolocalización:** Integración para el rastreo en tiempo real de los vehículos alquilados.

### 3.3.1 Estructura de ficheros y carpetas

RentCarVir/

```
|—— app/
|   |—— http/
|   |   |—— controllers/
|   |   |   |—— AlquilerController.php
|   |   |   |—— ClienteController.php
|   |   |   |—— EmpleadoController.php
|   |   |   |—— LocalizacionController.php
|   |   |   |—— StorageController.php
|   |   |   |—— UsuarioController.php
|   |   |   |—— VehiculoController.php
|   |   |—— middleware/
|   |   |—— models/
|   |   |   |—— Alquiler.php
|   |   |   |—— Cliente.php
|   |   |   |—— Empleado.php
|   |   |   |—— Localizacion.php
|   |   |   |—— Usuario.php
|   |   |   |—— Vehiculo.php
|—— bootstrap/
|—— config/
|—— database/
|—— public/
|   |—— css/
|   |   |—— app.css
```

```
|   └── js/
|       └── catalogo/
|           ├── video/
|           ├── about.blade.php
|           ├── catalogo.blade.php
|           ├── condiciones.blade.php
|           ├── contacto.blade.php
|           ├── home.blade.php
|           ├── login.blade.php
|           ├── register.blade.php
|           └── vehiculo-mostrar.blade.php
|   └── resources/
|       └── views/
|           └── admin/
|               ├── client-table.blade.php
|               ├── cliente-index.blade.php
|               ├── empleado-create.blade.php
|               ├── empleado-edit.blade.php
|               ├── empleado-index.blade.php
|               ├── estadisticas.blade.php
|               ├── localizacion.blade.php
|               └── layout/
|                   └── admin-layout.blade.php
|               └── navbar/
|                   └── admin-navbar.blade.php
|           └── cliente/
|               ├── cliente-alquiler.blade.php
|               ├── cliente-edit-password.blade.php
|               ├── cliente-edit-profile.blade.php
|               └── layout/
|                   └── cliente-layout.blade.php
|               └── navbar/
|                   └── cliente-navbar.blade.php
|   └── empleado/
```

```
|   |   |   ├── alquiler-index.blade.php  
|   |   |   ├── vehiculo-create.blade.php  
|   |   |   ├── vehiculo-edit.blade.php  
|   |   |   ├── vehiculo-index.blade.php  
|   |   |   └── layout/  
|   |   |       ├── empleado-layout.blade.php  
|   |   |       └── navbar/  
|   |   └── components/  
|   |       ├── footer.blade.php  
|   |       ├── image-item.blade.php  
|   |       └── pagination.blade.php  
|   └── routes/  
|       ├── console.php  
|       └── web.php
```

## Descripción de la organización

- **app/**: Carpeta principal con la lógica de negocio de la aplicación.
- **http/**: Contiene los **controllers** (controladores de las diferentes funcionalidades) y **middleware** (gestión de seguridad y permisos).
- **models/**: Define las entidades y su relación con la base de datos.
- **bootstrap, config, database**: Configuración y estructura del proyecto.
- **public/**: Archivos accesibles como **CSS, JavaScript, páginas principales** (como **home, login, catálogo**) y **videos**.
- **resources/views/**: Contiene la estructura de la interfaz de usuario.
  - **admin/**: Sección de administración con archivos para gestión de clientes, empleados y estadísticas.
  - **cliente/**: Sección de clientes con archivos para alquiler, edición de perfil y contraseña.
  - **empleado/**: Sección de empleados con archivos para gestión de alquileres y vehículos.
  - **components/**: Componentes reutilizables como **footer, elementos de imagen y paginación**.
  - **layout/**: Plantillas base para cada vista (**admin, cliente y empleado**).
  - **navbar/**: Barras de navegación personalizadas para cada sección.
- **routes/**: Define la estructura de rutas del proyecto.
  - **console.php**: Contiene comandos de consola para tareas específicas.
  - **web.php**: Maneja las rutas accesibles desde la aplicación web.

## Templates HTML y recursos para el cliente en RentCarVir

Nuestra aplicación **RentCarVir** no será REST y contará con archivos **CSS, JavaScript y otros recursos** que deben estar organizados correctamente. Cada página HTML se nombrará según su apartado correspondiente (**home, catálogo, contacto, login, etc.**) y los archivos **CSS** asociados llevarán el mismo nombre para evitar confusiones.

Para evitar errores, es importante asegurarnos de que todos estos recursos estén accesibles a los clientes. La configuración de Laravel debe permitir el acceso a archivos estáticos, garantizando que **CSS y JavaScript** se carguen correctamente.

## Diseñando la aplicación web - Menú

El menú de la aplicación será **un archivo HTML independiente**, del cual el resto de las páginas heredarán su estructura. Esto permitirá reutilizar el menú en todas las secciones sin necesidad de programarlo en cada página individualmente.

En Laravel, utilizaremos **Blade Templates** para la herencia de contenido HTML. La barra de navegación estará ubicada en `resources/views/components/navbar.blade.php`, y las vistas principales extenderán este archivo.

## Diseñando la aplicación web - Home

La página principal **Home** contará con un controlador que responderá a la petición y devolverá la correspondiente vista **HTML**.

- **Ruta de acceso:** /

## Diseñando la aplicación web - Quienes somos

Similar a la sección anterior, esta página responderá a la petición del usuario y mostrará una vista HTML con información sobre la empresa.

- **Ruta de acceso:** /about

## Diseñando la aplicación web - Catálogo

En esta sección, se deben considerar varios aspectos:

- El número de vehículos que se cargarán estará **limitado a 10 por página**, y se podrán cargar más conforme el usuario lo solicite.
- Se debe incluir una **barra de búsqueda**, que permitirá filtrar por **categoría o nombre de vehículo**.

El **controlador** de esta sección se encargará de procesar las solicitudes y recuperar los datos de la base de datos según los filtros aplicados. Para esto, se implementarán dos métodos en el servicio:

1. **Buscar por nombre del vehículo**
2. **Buscar por categoría**

Estos métodos devolverán **una lista con los vehículos disponibles** y se utilizarán objetos **Model** para enviar esta información a las vistas. La búsqueda se estructurará siguiendo buenas prácticas:

```
/catalogo?type=[nombre o categoría]&value=[valor elegido]
```

## Diseñando la aplicación web - Login

La autenticación de usuarios se implementará usando **Laravel Auth**. Por defecto, Laravel permite **autenticación basada en sesiones**, pero en futuras versiones podríamos integrar **Laravel Passport** para autenticación mediante tokens.

Aspectos clave de seguridad:

- **Encriptación de contraseñas:** Se utilizará **bcrypt** para almacenar contraseñas de forma segura.
- **Middleware de autenticación:** Se restringirá el acceso a ciertas secciones de la aplicación.

Las páginas restringidas solo serán accesibles a **usuarios registrados** y el sistema deberá **redireccionar a la página de inicio de sesión** si el usuario no está autenticado.

## Diseñando la aplicación web - Perfil de usuario

Esta sección estará **protegida y restringida** para aquellos usuarios que no estén autenticados.

Una vez que el usuario acceda correctamente, el controlador cargará **sus datos personales** desde la base de datos y le permitirá:

- **Actualizar su información**
- **Eliminar su cuenta**
- **Gestionar sus reservas de alquiler**

## Diseñando la aplicación web - Alquiler de vehículos

Esta sección permitirá a los usuarios **realizar reservas** directamente desde la aplicación.

- Se utilizará **JavaScript** para mejorar la experiencia de usuario.

## Diseñando la aplicación web - Localización de vehículos

La funcionalidad de **geolocalización** en **RentCarVir** estará **restringida exclusivamente para administradores**. A través del **panel de gestión**, el administrador podrá consultar la ubicación **actual** de los vehículos en el mapa sin que la aplicación guarde un historial de posiciones.

### Tecnologías utilizadas:

- **OpenStreetMap** para visualizar la ubicación de los coches en tiempo real.
- **Laravel** en el backend para verificar permisos y asegurar que solo el administrador tenga acceso.
- **AJAX con JavaScript** para obtener dinámicamente las coordenadas y mostrarlas en el mapa.

### Restricción de acceso:

- Solo los administradores autenticados podrán ver la ubicación de los vehículos desde la sección de **gestión**.
- Los clientes y empleados **NO tendrán acceso** a esta funcionalidad.
- No se guardarán registros históricos de la ubicación de los coches.

## 3.4 Base de datos en RentCarVir

La base de datos que utilizaremos en **RentCarVir** será **MySQL**, donde almacenaremos **información de usuarios, vehículos, alquileres y empleados**. Este sistema garantizará un acceso rápido y estructurado a los datos de la aplicación.

### 3.4.1 Diseño entidad-relación

El diseño **Entidad-Relación** de **RentCarVir** se ha optimizado para garantizar una **gestión eficiente** de la información. A continuación, presentamos el esquema general:

#### Entidades principales

- **Usuario** → Representa los clientes y empleados registrados en la aplicación.
- **Vehículo** → Contiene datos de cada coche disponible para alquiler.
- **Alquiler** → Registra las reservas de vehículos por parte de los clientes.
- **Empleado** → Gestiona las operaciones internas del sistema.
- **Localización** → Muestra la ubicación en tiempo real de los vehículos (solo accesible por el administrador).

#### Relaciones

- **Un usuario puede tener múltiples alquileres** (Relación 1:N).
- **Un alquiler pertenece a un solo vehículo** (Relación N:1).
- **Un vehículo puede estar disponible para varios alquileres a lo largo del tiempo** (Relación 1:N).
- **Un empleado puede gestionar varios alquileres** (Relación 1:N).

- Solo los administradores pueden visualizar las ubicaciones de los vehículos (N:1).

### 3.4.2 Diseño lógico

Tables_in_nueva
alquiler
cliente
empleado
migrations
sessions
usuario
vehiculo

mysql> describe alquiler;						
Field	Type	Null	Key	Default	Extra	
id	bigint(20) unsigned	NO	PRI	NULL	auto_increment	
fechaRecogida	date	NO		NULL		
lugarRecogida	varchar(100)	NO		NULL		
horaRecogida	varchar(5)	NO		NULL		
fechaEntrega	date	NO		NULL		
lugarEntrega	varchar(100)	NO		NULL		
horaEntrega	varchar(5)	NO		NULL		
importe	decimal(9,2)	NO		NULL		
activo	tinyint(1)	NO		NULL		
clienteID	bigint(20) unsigned	NO	MUL	NULL		
vehiculoID	bigint(20) unsigned	NO	MUL	NULL		

mysql> describe cliente;						
Field	Type	Null	Key	Default	Extra	
id	bigint(20) unsigned	NO	PRI	NULL	auto_increment	
nombre	varchar(30)	NO		NULL		
apellidos	varchar(30)	NO		NULL		
domicilio	varchar(50)	NO		NULL		
ocupacion	varchar(30)	NO		NULL		
fechaNacimiento	date	NO		NULL		
foto	varchar(255)	NO		NULL		

mysql> describe empleado;						
Field	Type	Null	Key	Default	Extra	
id	bigint(20) unsigned	NO	PRI	NULL		auto_increment
nombre	varchar(30)	NO		NULL		
apellidos	varchar(30)	NO		NULL		
foto	varchar(255)	NO		NULL		

  

mysql> describe usuario;						
Field	Type	Null	Key	Default	Extra	
id	bigint(20) unsigned	NO	PRI	NULL		auto_increment
username	varchar(30)	NO	UNI	NULL		
password	varchar(255)	NO		NULL		
utenteable_id	bigint(20) unsigned	YES		NULL		
utenteable_type	varchar(25)	NO		NULL		
remember_token	varchar(100)	YES		NULL		

  

mysql> describe vehiculo;						
Field	Type	Null	Key	Default	Extra	
id	bigint(20) unsigned	NO	PRI	NULL		auto_increment
matricula	char(7)	NO	UNI	NULL		
modelo	varchar(100)	NO		NULL		
marca	varchar(30)	NO		NULL		
motor	varchar(20)	NO		NULL		
cambio	varchar(20)	NO		NULL		
equipamiento	varchar(100)	NO		NULL		
puertas	varchar(10)	NO		NULL		
asientos	char(1)	NO		NULL		
autonomia	decimal(8,2)	NO		NULL		
color	varchar(30)	NO		NULL		
foto	varchar(255)	YES		NULL		
descripcion	text	YES		NULL		
disponible	tinyint(1)	NO		1		
emision	date	NO		NULL		
vencimiento	date	NO		NULL		
costoDiario	decimal(6,2)	NO		NULL		
created_at	timestamp	YES		NULL		
updated_at	timestamp	YES		NULL		
lat	decimal(10,7)	YES		NULL		
lng	decimal(10,7)	YES		NULL		

#### 4. Implementación en RentCarVir

La implementación de **RentCarVir** es un proceso técnico que se lleva a cabo en distintas etapas y con una variedad de herramientas especializadas. Debido a la complejidad de esta fase, el desarrollo se gestiona de manera separada y organizada.

## 4.1 Codificación

Para el desarrollo de **RentCarVir**, utilizamos los siguientes lenguajes y tecnologías:

### Lenguajes de programación:

- **PHP** → Para la lógica de negocio y gestión del backend.
- **JavaScript** → Para mejorar la interactividad en el front-end.
- **HTML5** → Estructura de la interfaz de usuario.
- **CSS3** → Diseño visual y adaptación responsiva.

### Herramientas utilizadas:

- **Bootstrap 4** → Framework de CSS para estilos y diseño responsive.
- **Laravel** → Framework de PHP para estructurar el backend.
- **Visual Studio Code** → Editor de código con soporte para extensiones avanzadas.
- **XAMPP** → Entorno de desarrollo local para ejecutar Apache y MySQL.
- **phpMyAdmin** → Gestión visual de la base de datos MySQL.

Desde el principio, se consideró el uso de otros entornos, pero **VSCode** ofrece extensiones que facilitan la gestión del cliente y permite una integración más flexible con **Laravel** y **Bootstrap**.

## 4.2 Back-end

La elección de **Laravel** como framework para el backend se basa en su **estructura robusta, seguridad y escalabilidad**. Laravel proporciona:

- **Autenticación integrada** → Fácil gestión de usuarios.
- **ORM (Eloquent)** → Manejo eficiente de bases de datos sin necesidad de consultas SQL manuales.
- **Controladores y rutas organizadas** → Mejora la estructura del código y su mantenimiento.
- **Middleware de seguridad** → Protección contra accesos no autorizados.

A diferencia de otros frameworks más ligeros, **Laravel** ofrece una **comunidad extensa** y múltiples paquetes que facilitan el desarrollo. Además, se complementa con **Bootstrap 4** en el front-end para garantizar una experiencia fluida.

## 4.3 Front-end

Para el **front-end**, se han utilizado las siguientes tecnologías:

- **Blade Templates (Laravel)** → Sistema de plantillas que permite reutilizar código sin redundancia.
- **Bootstrap 4** → Diseño responsive para adaptar la página a diferentes dispositivos.
- **JavaScript + jQuery** → Mejora la interactividad y la experiencia del usuario.

La **responsividad** es esencial en **RentCarVir**. Gracias al uso de **Bootstrap**, los elementos de la interfaz se ajustan automáticamente, eliminando la necesidad de diseñar una versión móvil aparte. Por ejemplo:

- La **barra de navegación** se transforma en un menú desplegable en pantallas pequeñas.
- La **visualización de vehículos** se adapta dinámicamente a los tamaños de pantalla.

## 5. Distribución en RentCarVir

Actualmente, **RentCarVir** se encuentra en fase de desarrollo y pruebas, sin estar aún desplegado en un entorno de producción. Sin embargo, en el futuro, **si se decide convertir la aplicación en un negocio real**, será necesario implementar ciertas estrategias para su distribución en la web.

### 5.1 Posibles pasos para la distribución

Si **RentCarVir** evoluciona hacia una plataforma pública, estos serán los aspectos clave a considerar:

1. **Registro de un dominio web**
  - Se necesitará adquirir un dominio personalizado para la aplicación, permitiendo un acceso fácil y profesional.
2. **Contratación de un servidor web**
  - Para alojar la aplicación, será necesario seleccionar un servicio de hosting confiable.
3. **Configuración del balanceador de carga**
  - En caso de recibir un gran volumen de usuarios, un **servidor NGINX** podría ayudar a distribuir el tráfico de manera eficiente.
4. **Implementación de medidas de seguridad**
  - Será esencial configurar **certificados SSL** para garantizar conexiones seguras.
  - Se puede utilizar **Let's Encrypt** para obtener certificados gratuitos.
5. **Optimización para acceso global**
  - Se deberá garantizar una **correcta configuración de servidores** para permitir acceso desde cualquier parte del mundo.
  - Configurar un **CDN (Red de distribución de contenido)** para mejorar el rendimiento en distintas ubicaciones.

## 6. Bibliografía

- **Creative Commons:** [https://creativecommons.org/licenses/?lang=es\\_ES](https://creativecommons.org/licenses/?lang=es_ES) → Para entender licencias abiertas y derechos de uso.
- **Laravel Framework:** <https://laravel.com/docs> → Documentación oficial del framework utilizado en el backend.
- **Bootstrap 4:** <https://getbootstrap.com/docs/4.0/getting-started/introduction/> → Framework de CSS utilizado en el front-end.
- **PHP y MySQL:** <https://www.php.net/manual/en/> y <https://dev.mysql.com/doc> → Guías oficiales de los lenguajes base para la aplicación.

- **JavaScript y jQuery:** <https://developer.mozilla.org/es/docs/Web/JavaScript> → Referencia de JavaScript para la interactividad del proyecto.
- **Oracle Cloud:** <https://www.oracle.com/cloud/> → Información sobre las máquinas virtuales utilizadas en la posible distribución futura.
- **NGINX:** <https://www.nginx.com/resources/wiki/> → Servidor web recomendado para futuras implementaciones.
- **Let's Encrypt:** <https://letsencrypt.org/es/about/> → Sistema de certificación SSL gratuito para garantizar la seguridad del sitio.
- **phpMyAdmin:** <https://www.phpmyadmin.net/docs/> → Herramienta de gestión de base de datos utilizada en el proyecto.