

Fernando Batista
Villarrubia

Trabajo fin de grado

Plataforma para el Alquiler de
Coches a través de Tecnologías
Web, On-Premise y Móvil: parte Web

Índice

- 1. Introducción**
2. Estado del arte
3. Diseño
4. Implementación
5. Demostración
6. Líneas futuras
7. Conclusión



Introducción

Este proyecto consiste en una aplicación web que permita a sus usuarios lo siguiente:

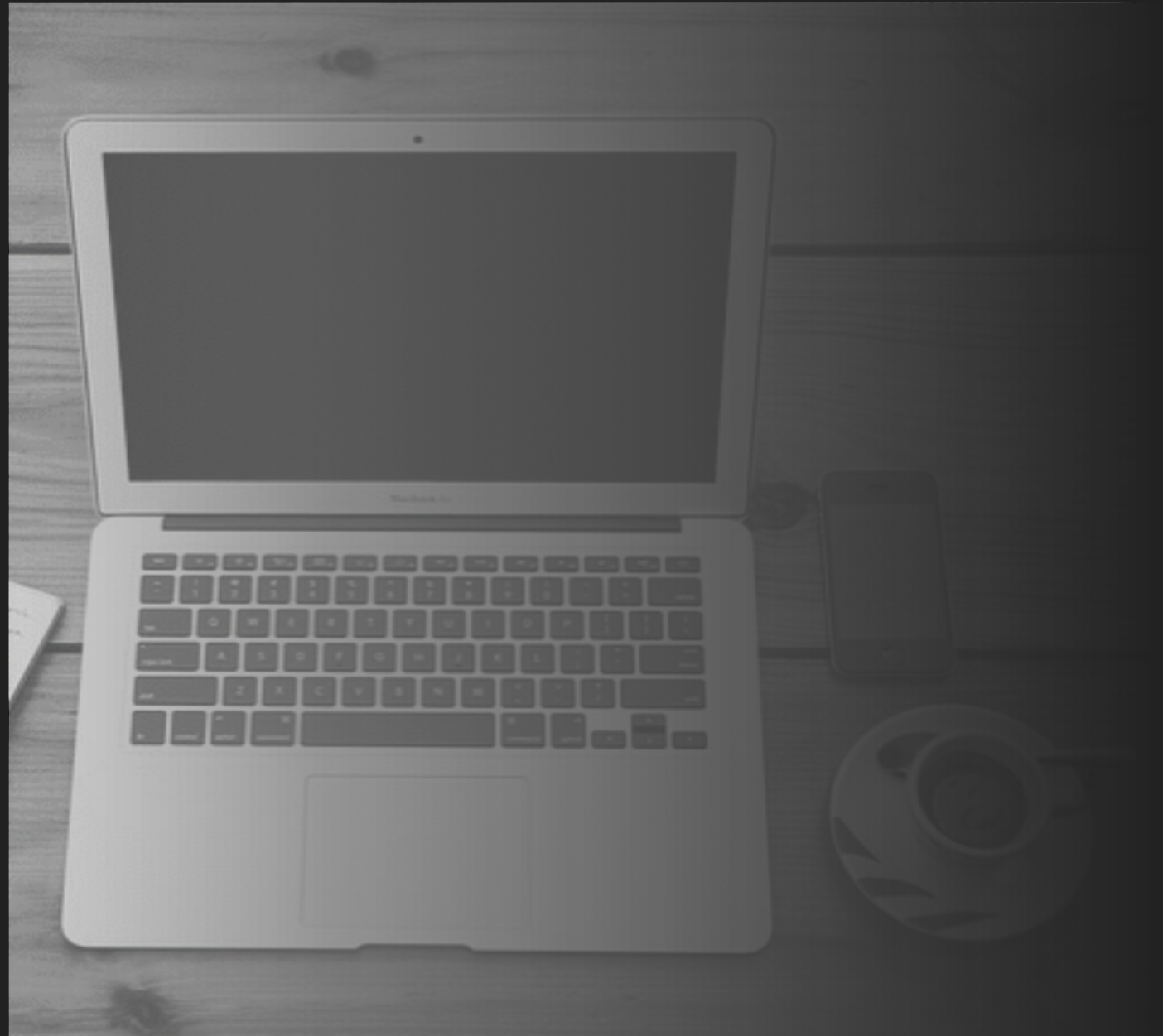
- Registrarse en el sistema e iniciar sesión
- Realizar reservas de coches
- Ver las reservas que has realizado
- Cancelar una reserva
- Modificar una reserva

Objetivos iniciales:

- Establecimiento de la tecnología y las herramientas a utilizar
- Definición y especificación de los requisitos
- Diseño e implementación del front-end
- Diseño e implementación de la base de datos
- Diseño e implementación de la API REST

Índice

1. Introducción
- 2. Estado del arte**
3. Diseño
4. Implementación
5. Demostración
6. Líneas futuras
7. Conclusión



Estado del arte



CHECK24

Transición
ecológica

Multiplataforma



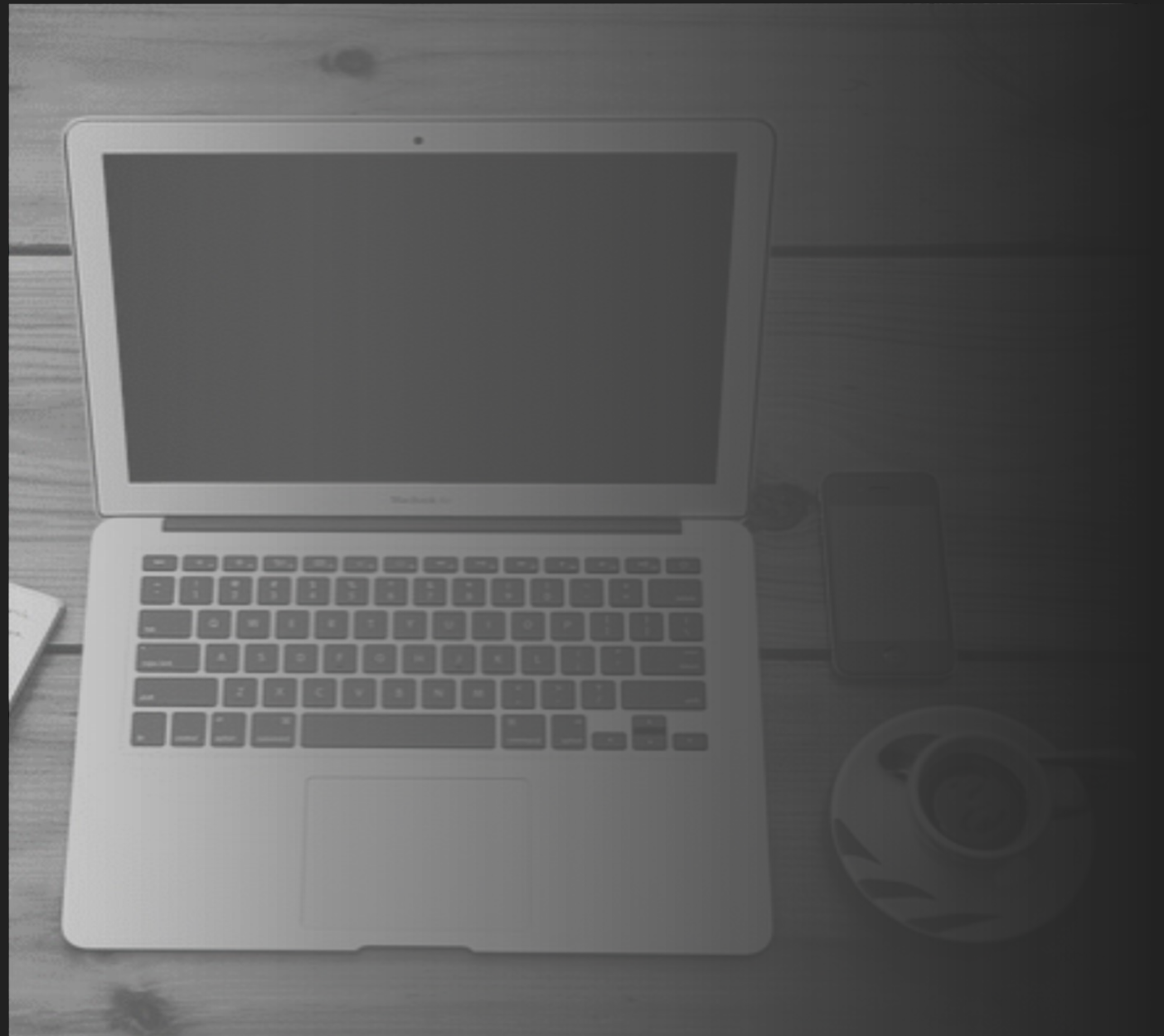
centauro
rent a car



SIXT
rent a car

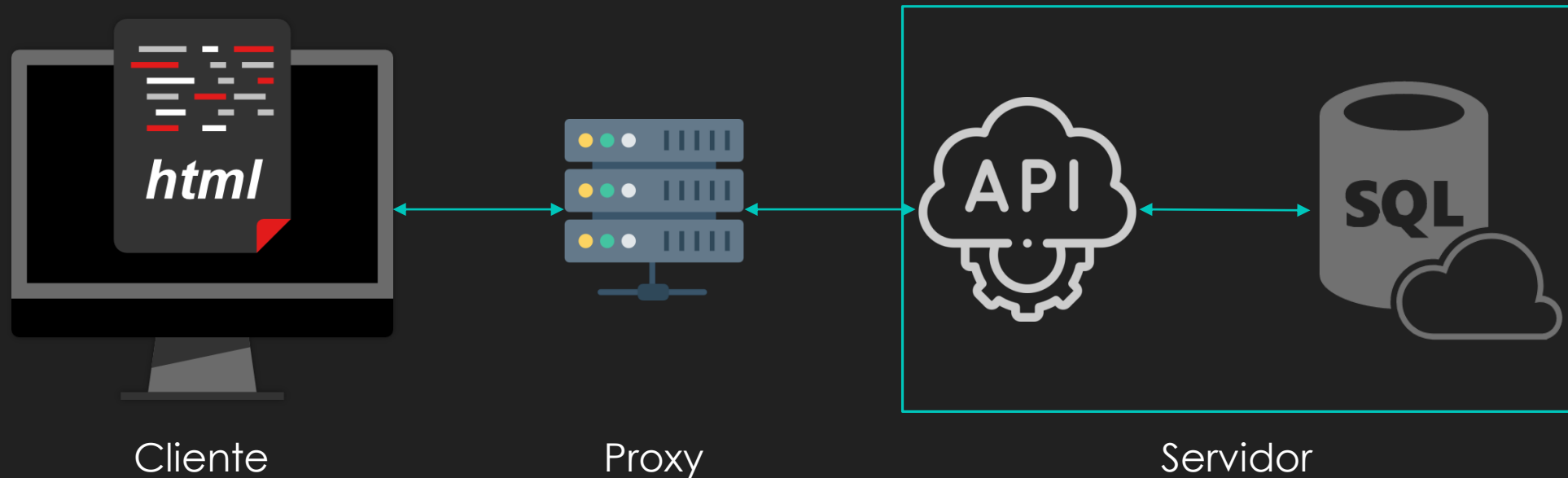
Índice

1. Introducción
2. Estado del arte
- 3. Diseño**
4. Implementación
5. Demostración
6. Líneas futuras
7. Conclusión



Diseño

El proyecto se divide en diferentes subsistemas, la conexión entre estos sigue el modelo de cliente-servidor, donde la pagina web representa el primero de estos, y la API el segundo, junto con esta está la base de datos, a la que ha de acceder para llevar a cabo las funcionalidades requeridas. A su vez, está el proxy que es un servidor intermedio necesario para la conexión.



Diseño: Pagina web

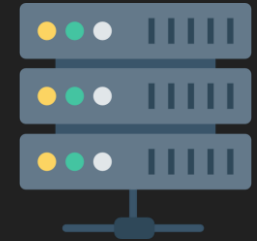


La aplicación web funciona como front-end del sistema para uso del cliente, del mismo modo, actúa como cliente en nuestro modelo de cliente servidor. En esta se podrá realizar las siguientes funciones:

- Registro de nuevo usuario
- Inicio de sesión
- Búsqueda de ofertas
- Consulta de reservas
- Modificación de reservas
- Cancelación de reservas
- Realización de una reserva

Diseño: Servidor proxy y API REST

Como cualquier proxy, este actúa como intermediario en la conexión entre la web y la API



Esta se encarga de gestionar las operaciones a la base de datos mediante peticiones a distintas URI. Estas las hemos agrupado de la siguiente manera:

- Usuarios: <http://localhost:8080/apirest/api/usuario>
 - Pagos: <http://localhost:8080/apirest/api/usuario/{idUsuario}/pago>
 - Reserva: <http://localhost:8080/apirest/api/usuario/{idUsuario}/reserva>
- Filiales: <http://localhost:8080/apirest/api/filial>
 - Coches ofertados: <http://localhost:8080/apirest/api/filial/{idFilial}/oferta>

Diseño: Base de datos



En esta, se guardan todos los datos relevantes para el correcto funcionamiento de la aplicación web, así como también para el correcto funcionamiento de la aplicación de escritorio, explicada en la presentación de mi compañero, Ricardo López Crespo.

La organización en tablas se distribuye de la siguiente manera:

- Usuario
- Reserva
- Pago
- CocheOfertado
- Disponibilidad
- Trabajadores
- Filial

Índice

1. Introducción
2. Estado del arte
3. Diseño
- 4. Implementación**
5. Demostración
6. Líneas futuras
7. Conclusión



Implementación: Pagina web

○ Lenguajes:



HTML



CSS



JavaScript

○ Herramientas:



Visual Studio Code

○ Problemas encontrados:

La programación fue lenta debido a mis escasos conocimientos en esta materia, ya que nunca antes había estudiado estos lenguajes hasta ahora.

Implementación: Servidor proxy

○ Lenguajes:



JavaScript

○ Herramientas:



Visual Studio Code



Node

○ Problemas encontrados:

La falta de información en internet, ya que muchos recomendaban un proxy ajeno de uso general, en vez de crear el tuyo propio.

Tras buscar, encontré un tutorial de como hacer el proxy para un método GET, y extrapolando, pude hacerlo para el resto de métodos.

Implementación: API REST

○ Lenguajes:



Java

○ Herramientas:



Visual Studio Code



Apache tomcat server

○ Problemas encontrados:

El mayor problema fue la configuración, tras varias alternativas intentando hacer uso del gestor de servidores de Eclipse sin resultado positivo, busque la manera de hacerlo sin ningún IDE.

El resultado final fue hacer uso del kernel de Ubuntu en Windows y ejecutar todo mediante comandos.

Implementación: Base de datos

○ Lenguajes:



SQL

○ Herramientas:



MySQL



XAMPP



MySQL Workbench 8.0 CE

○ Problemas encontrados:

El mayor inconveniente, mas que problema, fueron los cambios sufridos en el esquema inicial ya que fue lo primero que diseñamos e implementamos.

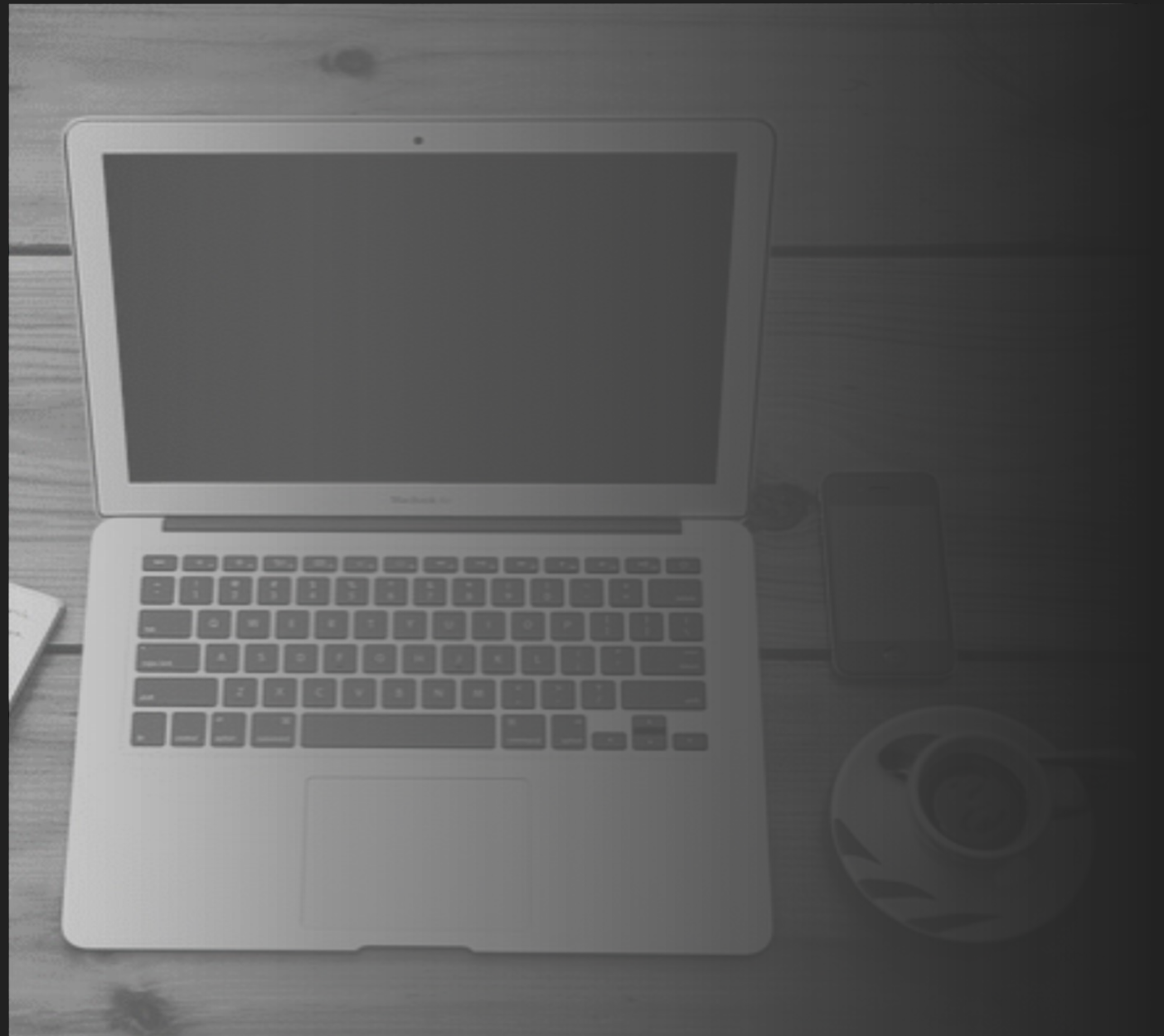
Implementación: Problemas de conexión

Tras la creación de la API, al intentar realizar una conexión desde la pagina web, esta devolvía un error que consistía en la falta de cabecera CORS (Cross-Origin Resource Sharing). Tras investigar, descubrí que se podía solucionar esto mediante el uso de un proxy que añadiera dicha cabecera. El proceso al que se somete la llamada es el siguiente:

- El servidor proxy recibe la llamada y la analiza.
- Añade la cabecera CORS a la petición
- Llama a la api con la URL correspondiente
- Añade la cabecera CORS a la respuesta
- Devuelve los datos solicitados por el cliente

Índice

1. Introducción
2. Estado del arte
3. Diseño
4. Implementación
- 5. Demostración**
6. Líneas futuras
7. Conclusión





Car Rental

La mejor aplicación de alquiler de coches

[Inicio de sesión](#)

[Resgistro](#)

Lugar de recogida

Madrid, Calle Ardemans



Recogida

15/06/2021



Devolucion

dd/mm/aaaa



Buscar

Filtros adicionales:

Numero de plazas:



Numero de puertas:



Cambio de marchas:



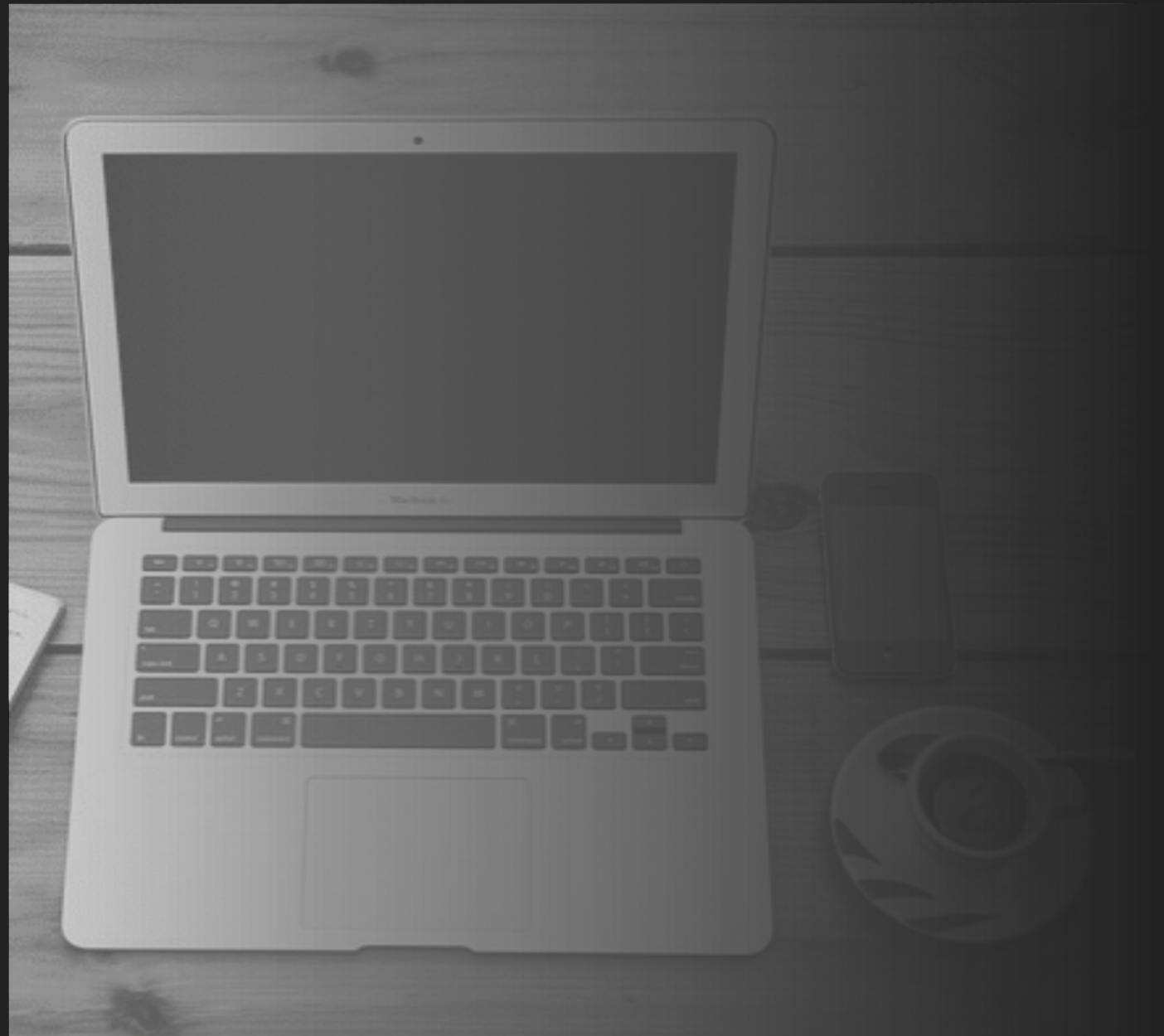
Otros filtros:



Contactanos a traves de:

Índice

1. Introducción
2. Estado del arte
3. Diseño
4. Implementación
5. Demostración
- 6. Líneas futuras**
7. Conclusión

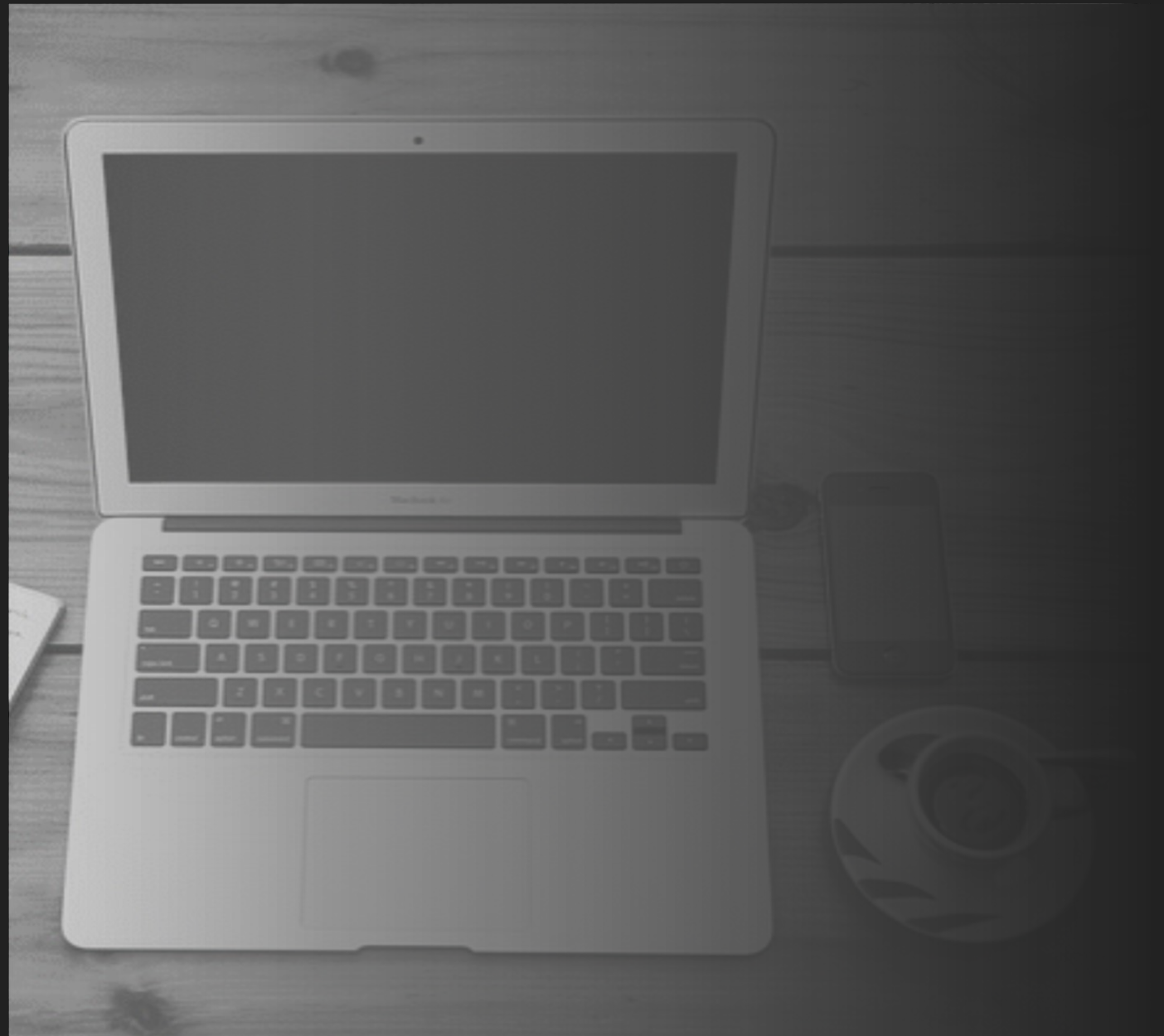


Líneas futuras

- Codificación de las contraseñas para mayor seguridad
- Mejora del diseño grafico de la pagina web
- Implementación de un mapa interactivo donde ver la localización de las filiales
- Implementación de verificación por correo electrónico al crear una cuenta
- Implementación de envíos de correo con la factura y los datos de la reserva
- Hacer publico el servidor proxy para poder acceder a la API desde cualquier lugar

Índice

1. Introducción
2. Estado del arte
3. Diseño
4. Implementación
5. Demostración
6. Líneas futuras
- 7. Conclusión**



Conclusión

La experiencia ha sido frustrante en ciertos momentos, debidos a numerosos problemas sufridos a lo largo del proyecto, sin embargo, superar esos desafíos por méritos propios es algo gratificante.

También es cierto que contando con la ayuda de mi compañero Ricardo López Crespo, la carga es menor, ya que siempre podía contar con su ayuda en caso de quedarme atascado durante largos periodos de tiempo. Además, durante los años previos de carrera hemos trabajado juntos en más de un proyecto, de tal manera estamos acostumbrados a trabajar en grupo.

Fin

Gracias por su atención

¿Alguna pregunta?