

DOCUMENTO DE DISEÑO SOFTWARE

SDD (Software Design Document)

Nombre del documento	Documento de Diseño Software
Nombre del fichero:	<i>ISST-SSD-V19.docx</i>
Versión:	3 (Versión final)
Fecha de entrega:	<i>06/05/2024</i>
Caso de estudio:	Ocio con mascotas
Entrega:	Sprint 3

Autores/as: Alejandro Muñoz, María Lozano, Marina Lumbreras, Javier González, Ignacio Díez

Revisor/a: Marina Lumbreras

INDICE

1 INTRODUCCIÓN.....	2
2.ANÁLISIS DE LA ARQUITECTURA.....	2
1.1 Requisitos y arquitectura.....	2
1.2 Patrón de arquitectura seleccionado.....	3
1.3 Otros criterios y restricciones.....	3
3.DESCRIPCIÓN DE LA ARQUITECTURA	4
3.1 Vista lógica de la arquitectura.....	4
3.2 Vista de despliegue de la arquitectura.....	4
4.MODELO DE DATOS	7
5.INTERFAZ DE USUARIO	8
5.1 Estructura y contenido.....	8
5.2 Flujos de navegación	13

1 INTRODUCCIÓN

Este documento permite servir como guía para el diseño de nuestra aplicación. Para ello vamos a indagar en la arquitectura de esta, valiéndonos de diagramas para una mejor representación. También se mostrará un esquema de los datos a utilizar en nuestra aplicación, así como la relación entre ellos. Finalmente, podremos ver un primer boceto de la interfaz de usuario de nuestra aplicación.

2. ANÁLISIS DE LA ARQUITECTURA

La arquitectura que hemos decidido seguir es la que se basa en la propuesta de la asignatura, una arquitectura Cliente-Servidor en tres niveles. La elección de esta arquitectura se basa en cumplir los requisitos y objetivos de la aplicación. Consideramos que la separación de la aplicación en los niveles de interfaz de usuario, nivel de procesamiento de datos y almacenamiento y administración de datos es el que más se ajusta a nuestra visión y el que mejor facilitará el desarrollo simultáneo de las diferentes partes.

1.1 Requisitos y arquitectura

Los requisitos establecidos en el documento de visión fueron los relacionados con seguridad, portabilidad, recursos y escalabilidad y estas son las razones por las que consideramos que la arquitectura seleccionada es la que mejor se ajusta a cada una de ellas:

- **Escalabilidad:** La arquitectura de tres niveles permite escalar los diferentes niveles de manera independiente. De tal manera, si el número de establecimientos o de usuarios aumenta, cualquiera de los tres niveles puede ser modificado para que sea capaz de soportar la nueva cantidad de datos sin que los otros niveles sean afectados.
- **Recursos:** Permite gestionar cada componente de manera independiente. Tanto aplicando balanceo de carga como implementando redundancia en cada una de las capas. Además, la separación permitirá la ejecución de mantenimiento en cada capa sin que afecte a la totalidad del sistema.
- **Portabilidad:** La separación en tres niveles permite que en la capa de interfaz de usuario la adaptabilidad a diferentes tipos de dispositivos y navegadores sea mayor.
- **Seguridad:** Esta arquitectura, ayudará a que a cada una de las capas se le apliquen medidas de seguridad específicas, para poder así asegurar la protección de datos sensibles y la gestión centralizada de la autenticación y autorización. Además, permitirá el seguimiento de una manera más sencilla de un registro de actividades.

1.2 Patrón de arquitectura seleccionado

Hemos analizado tres tipos posibles de patrones arquitectónicos para poder establecer cuál sería el elegido para implementar en nuestra aplicación.

- **Peer-to-peer:** Esta arquitectura permite compartir una gran cantidad de datos. Se diferencia de la arquitectura Cliente-Servidor en que es una red descentralizada de clientes y servidores. Una de las ventajas de este tipo de arquitectura es que el gasto de recursos es menor. Pero el desarrollo de la aplicación sería más complicado y difícil de coordinar entre los diferentes desarrolladores. Además, la gestión de la escalabilidad y seguridad podría verse comprometida teniendo en cuenta que muchos de los nodos de la aplicación estarían más expuestos
- **EDA:** Es una arquitectura orientada a eventos. Ofrece beneficios siendo uno de ellos la capacidad de reaccionar en tiempo real y la desvinculación de componentes. Pero la complejidad que esto añadiría al desarrollo es innecesaria y no se alinea con el tipo de servicio que queremos ofrecer.
- **Cliente-Servidor de dos niveles:** Es una arquitectura común en la que aplicación se separa en dos niveles: el cliente y el servidor. Normalmente, suelen ser aplicaciones síncronas y sencillas en las que el cliente envía una petición al servidor y espera su respuesta. Esta simplicidad hace que la aplicación pueda tener problemas en cumplir los requisitos de escalabilidad y mantenimiento establecidos.

Después de analizar varios tipos de patrones arquitectónicos, hemos decidido optar por usar una arquitectura Cliente-Servidor de tres niveles. Consideramos que esta arquitectura es la que más se ajusta a las necesidades, requisitos y objetivos establecidos anteriormente además de permitirnos comodidad a la hora de desarrollarla

1.3 Otros criterios y restricciones

La herramienta que usaremos como nuestro marco de trabajo para el desarrollo del backend de MascotMercio será SpringBoot para poder así tener una infraestructura ágil y que nos permita desarrollar aplicaciones Java de una manera sencilla. Usaremos también Java Persistence API, es una API standard que proporciona un modelo objeto-relacional para la persistencia y el acceso a datos relacionales. Para la comunicación eficaz entre capas de la aplicación usaremos los servicios REST para poder acceder y modificar datos basándonos en los protocolos HTTP.

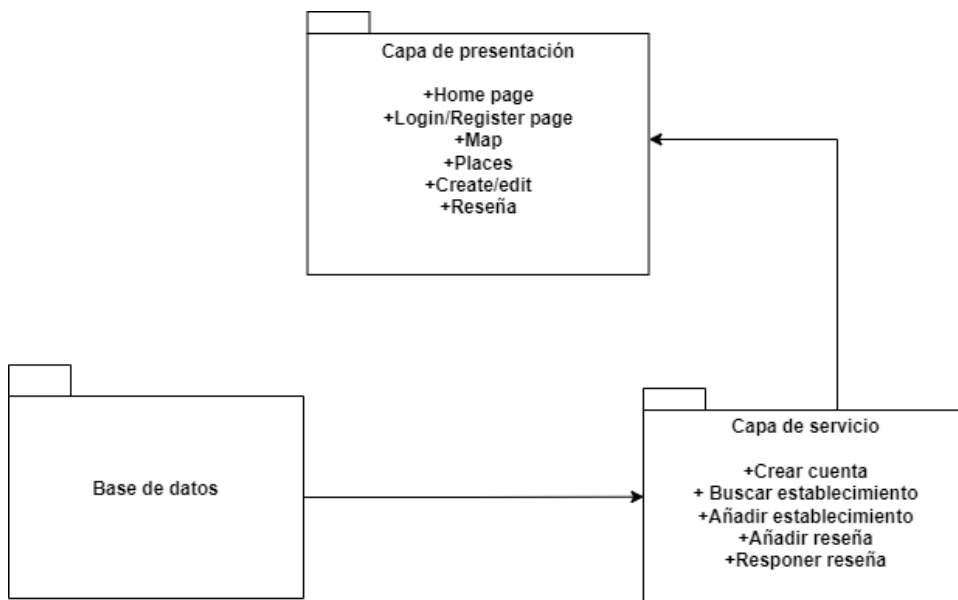
A través de las llamadas de estos servicios REST accederemos a la información en el frontend, el cual estará construido también con SpringBoot al ser el recomendado en la asignatura.

3.DESCRIPCIÓN DE LA ARQUITECTURA

Para el desarrollo de la aplicación Mascotmercio hemos optado por implementar una arquitectura del tipo cliente-servidor en tres niveles: presentación, servicio y base de datos. Esto nos permite dividir las distintas partes de la aplicación, permitiendo mejor escalabilidad y mantenimiento del código.

3.1 Vista lógica de la arquitectura

En este diseño de la arquitectura cada elemento tiene una responsabilidad distinta: la capa de presentación está relacionada con la interacción con el cliente y la interfaz de usuario, la capa de servicio será el núcleo funcional de la aplicación, y la información se guardará en la base de datos.

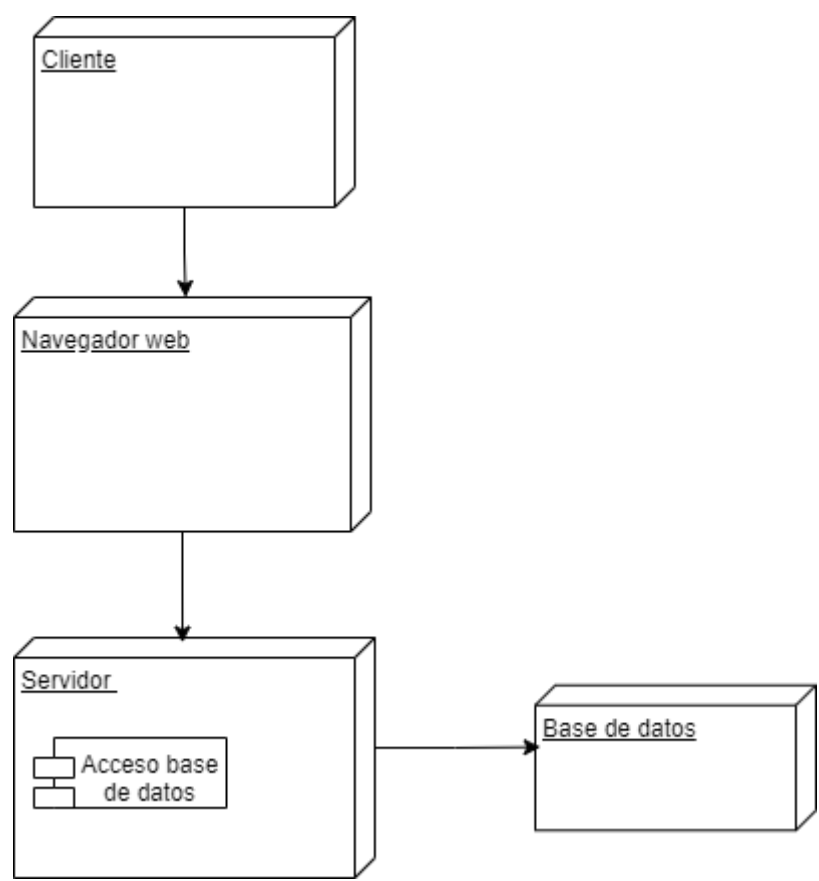


3.2 Vista de despliegue de la arquitectura

Para el despliegue de nuestra aplicación utilizaremos una configuración basada en distintos componentes:

- Un servidor web para alojar la capa de presentación
- Un servidor de aplicaciones que ejecuta la capa de servicio
- Un servidor de bases de datos

La comunicación entre nodos será la siguiente: el servidor web actuará como punto de entrada para las solicitudes de los usuarios, que se conectará con el servidor para realizar las funciones de la aplicación. A su vez, el servidor se comunicará con el servidor de bases de datos para acceder y manipular los datos almacenados.



Definición de la interfaz API-REST

Ruta	Función
@GetMapping("api/cliente/findAll")	Leer todos los clientes
@PostMapping("/api/cliente")	Crear un cliente
@PutMapping("/api/cliente/{usuario}")	Editar un cliente con cierto usuario
@DeleteMapping("api/cliente/{usuario}")	Borrar un cliente con cierto id

Ruta	Función
@GetMapping("api/propietario/findAll")	Leer todos los propietarios
@PostMapping("/api/propietario")	Crear un propietario
@PutMapping("/api/propietario/{usuario}")	Editar un propietario con cierto usuario
@DeleteMapping("api/propietario/{usuario}")	Borrar un propietario usuario

Ruta	Función
@GetMapping("/api/resena/findAll")	Leer todas las reseñas
@PostMapping("/api/resena")	Crear una reseña
@PutMapping("/api/resena/{id}")	Editar una reseña con cierta reseña
@DeleteMapping("/api/resena/{id}")	Borrar una reseña con cierta reseña

Ruta	Función
@GetMapping("/api/servicio/findAll")	Leer todos los servicios
@PostMapping("/api/servicio")	Crear un servicio
@PutMapping("/api/servicio/{id}")	Editar un servicio con un cierto id
@DeleteMapping("/api/servicio/{id}")	Borrar un servicio con un cierto id

Ruta	Función
@GetMapping("/api/tienda/findAll")	Leer todas las tiendas
@PostMapping("/api/tienda")	Crear una tienda
@PutMapping("/api/tienda/{id}")	Editar una tienda con cierto id
@DeleteMapping("/api/tienda/{id}")	Borrar una tienda con cierto id

4.MODELO DE DATOS

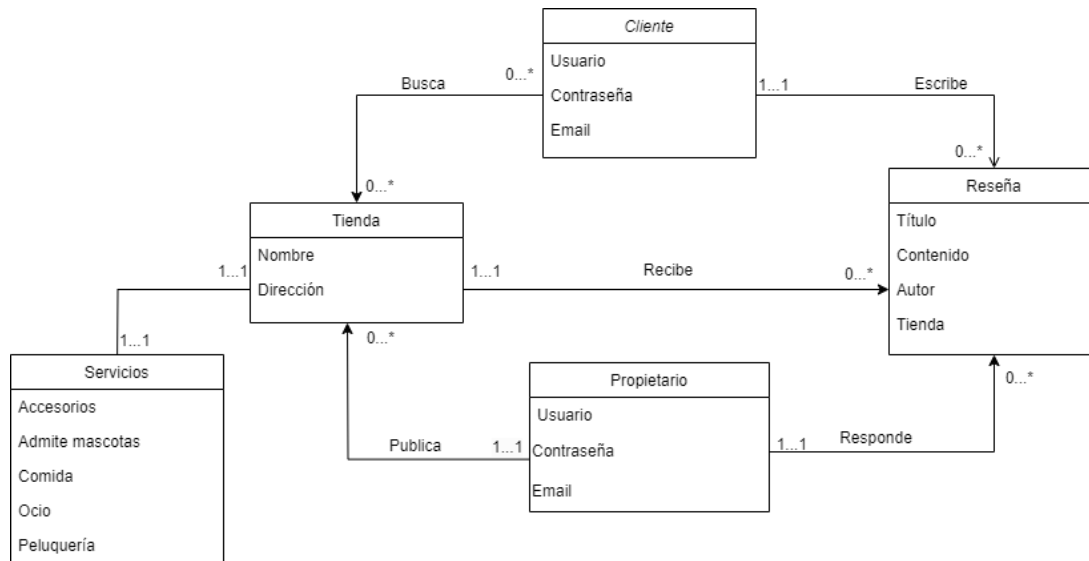


Figura 1. Modelo de datos

Entidad		Atributo						
Nombre	Descripción	Nombre	Descripción	Tipo (Tamaño)	PK	Obligatorio	Unicidad	Verificación
Cliente	Un cliente es un usuario que utiliza la plataforma para buscar establecimientos que aceptan mascotas	Nombre	Nombre con el que se ha registrado el cliente	VARCHAR (255)	S	S	S	N
		Contraseña	Contraseña con la que se ha registrado el cliente	VARCHAR (255)	N	S	N	N
		Email	Dirección de correo electrónico asociada a una cuenta de usuario	VARCHAR (255)	N	S	S	N
Propietario	Un propietario es un usuario que publica la información de su comercio en la web	Nombre	Nombre con el que se ha registrado el titular de una tienda	VARCHAR (255)	S	S	S	N
		Contraseña	Contraseña con la que se ha registrado el titular de una tienda	VARCHAR (255)	N	S	N	N
		Email	Dirección de correo electrónico asociada a la cuenta de propietario	VARCHAR (255)	N	S	S	N
Tienda	Una tienda es un establecimiento que se registra en la aplicación	Id	Número de identificación única de la tienda	LONG	S	S	S	N
		Nombre	Nombre de la tienda	VARCHAR (255)	N	S	N	N
		Dirección	Ubicación del establecimiento	VARCHAR (255)	N	S	S	N
Servicios	Los servicios son las ofertas de actividades o	Accesorios	El establecimiento vende accesorios para mascotas	BOOLEAN	N			
		Admite mascotas	El establecimiento admite mascotas	BOOLEAN	N			
		Comida	El establecimiento ofrece comida para mascotas	BOOLEAN	N			

		Ocio	El establecimiento ofrece algún tipo de ocio para mascotas	BOOLEAN	N			
		Peluquería	El establecimiento ofrece el servicio de peluquería para mascotas	BOOLEAN	N			
Reseña	Una reseña es un comentario que realiza un cliente acerca de su experiencia en la tienda. Los propietarios pueden responder a las reseñas de su establecimiento	Id	Número de identificación unívoca de una reseña	LONG	S	S	S	N
		Autor	Autor de la reseña o comentario. Puede ser un cliente o un propietario	VARCHAR (255)	S	S	S	N
		Título	Breve descripción del comentario	VARCHAR (255)	N	S	N	N
		Contenido	Reseña con detalle de la estancia en la tienda de un cliente o respuesta de su propietario	VARCHAR (255)	N	S	N	N
		TiendaId	Establecimiento sobre el que se está realizando el comentario	LONG	N	S	N	N

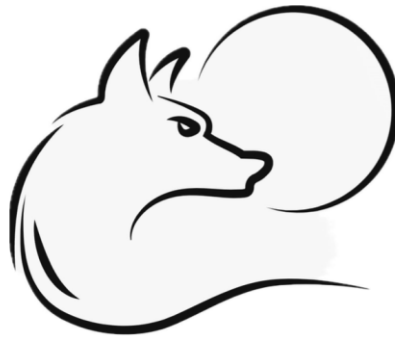
5.INTERFAZ DE USUARIO

La interfaz de usuario es el elemento que conectará las funcionalidades de la aplicación con el usuario, por lo que es de vital importancia que esta se adapte lo mejor posible al cliente para que se haga un correcto uso de la misma. Más adelante, se explica tanto su estructura como el flujo de navegación.

5.1 Estructura y contenido

El contenido de la aplicación tiene una estructura web con barra de navegación superior que permitirá acceder a cada una de las páginas que tanto cliente como propietario de establecimiento necesitan para los casos de uso mencionados en el documento visión.

A continuación, se proporciona una imagen con una breve explicación de cada una de las páginas principales a diseñar. Todas las páginas tendrán acceso en el backend a una base de datos SQL que proporcionará la información necesaria para cada tabla de establecimientos que se muestre, información de servicios que ofrecen, reseñas que se realizan y usuarios que accedan a la aplicación web.



ACCEDE

"El objetivo de este proyecto es desarrollar un servicio que permita por una parte que los establecimientos de ocio muestren su disposición a recibir clientela con animales de compañía, incluyendo la posibilidad de ofrecer productos o servicios específicos para mascotas."

HomePage esta es la página de inicio, requerirá de Registro o Log In para poder tener acceso al resto de funcionalidades de la web. Necesitará acceder a los datos de usuario de la base de datos para poder saber si se ha iniciado sesión.



Please sign in

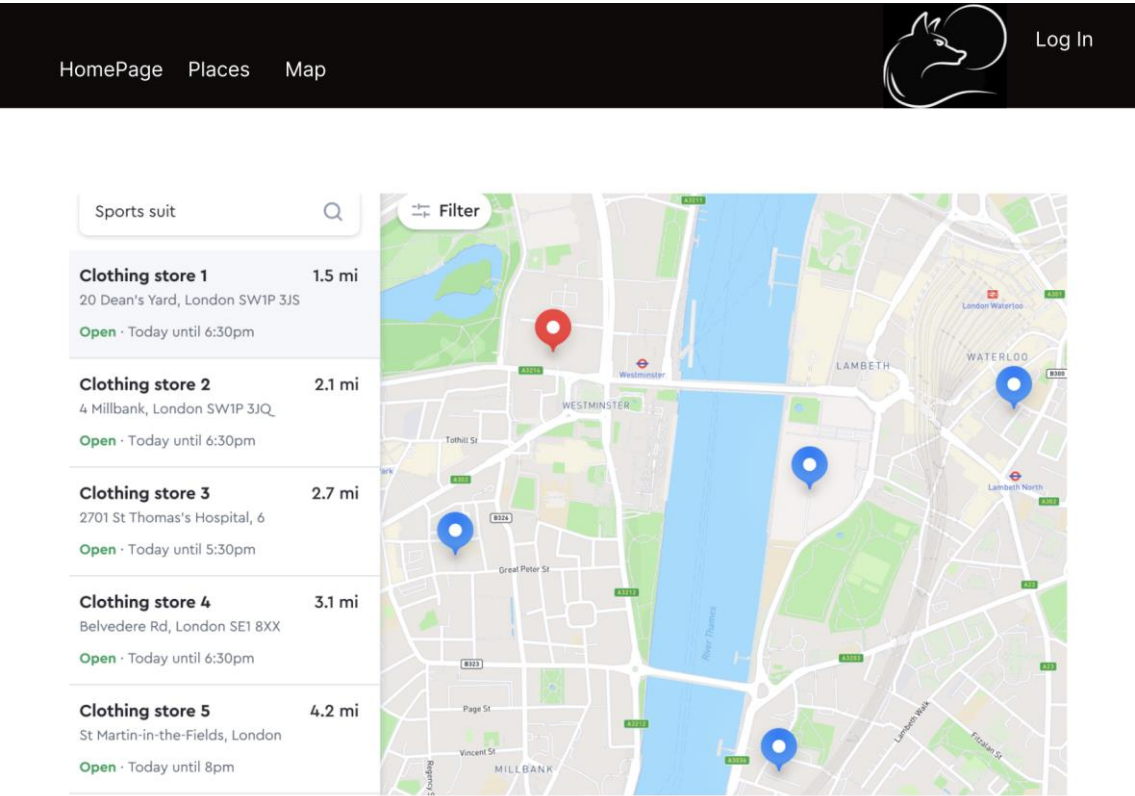
Username

Password

☐ Remember me on this computer.

Sign in

Log In o Registro esta página permitirá hacer tanto el log in o registro que deberán ser realizados para acceder a todas las funcionalidades. En esta página se permite la creación de una entidad de usuario para poder acceder a la web, que quedará registrada en la base de datos.



Map esta página proporcionará un mapa al usuario en el que este podrá localizar cada uno de los comercios y ver los servicios que ofrecen para su mascota. Es importante que desde el mapa se tenga completo acceso al conjunto de datos que forman la entidad de establecimientos, para que así, se puedan mostrar todos los que están disponibles con sus correspondientes servicios.



Place Name	Mascotas Aceptadas	Servicios	Actions
			Update Delete
			Update Delete
			Update Delete
			Update Delete
			Update Delete

Places página en la que se desplegará una lista con todos los comercios disponibles, será de

gran utilidad para los propietarios, ya que tendrán solo ellos acceso para poder modificar su comercio dentro de la web. Los usuarios solo podrán ver los comercios disponibles. Aquí es muy importante conocer el tipo de usuario que ha iniciado la sesión, esto sirve para activar unas funcionalidades u otras, también se necesita poder acceder a todos los datos de establecimientos para poder visualizarlos.

Edita tu Negocio


Name	<input type="text" value="Spring Boot Thymeleaf"/>
Description	<input type="text" value="Writing web app with Spring Boot & Thymeleaf"/>
Price	<input type="text" value="9.9"/>
Enabled	<input checked="" type="checkbox"/>
<div><input type="button" value="Save"/> <input type="button" value="Cancel"/></div>	

Añade tu Negocio

Name	<input type="text"/>
Description	<input type="text"/>
Price	<input type="text"/>
Enabled	<input checked="" type="checkbox"/>
<div><input type="button" value="Save"/> <input type="button" value="Cancel"/></div>	

Create/Edit son las páginas en las que los propietarios podrán añadir o editar sus comercios. En ambas pestañas accederemos a una entidad establecimiento concreta y se permitirá añadir o modificar sus datos en la base de datos, donde quedarán registrados para que se puedan acceder desde el resto de vistas.

[Home](#)[Page](#)[Places](#)[Map](#)

[Log In](#)



Añade Tu Reseña

Name

Description

Rate

Save

Cancel

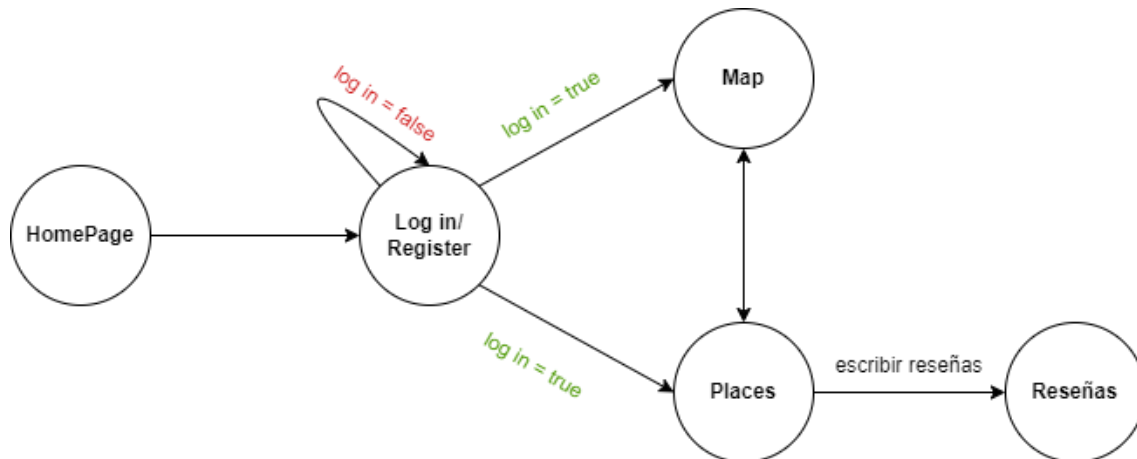
Reseña esta página permitirá al usuario añadir reseñas a cada establecimiento proporcionando una descripción y puntuación. Las reseñas quedarán registradas en la base de datos haciendo referencia a un establecimiento, también, se podrán acceder desde otras páginas de la web.

La conexión constante con la base de datos es fundamental para el correcto funcionamiento de la web, ya que la interfaz gráfica no sirve de nada si no hay una constante actualización de la información a proporcionar al cliente. Por esto el backend estará constantemente sincronizado con la interfaz gráfica para proporcionar datos con máxima fiabilidad.

DISCLAIMER: *Todas las imágenes aquí proporcionadas se han realizado intentando acercarse lo máximo posible al producto que se va a diseñar, pero hay que remarcar que se han empleado formularios de creación y edición, tablas y otras interfaces que no son de este proyecto concreto. Aun así estas sirven como referencia gráfica, ya que se han diseñado usando SpringBoot y sus herramientas por lo que proporcionan una similitud gráfica a lo que será la aplicación Mascotmercio.*

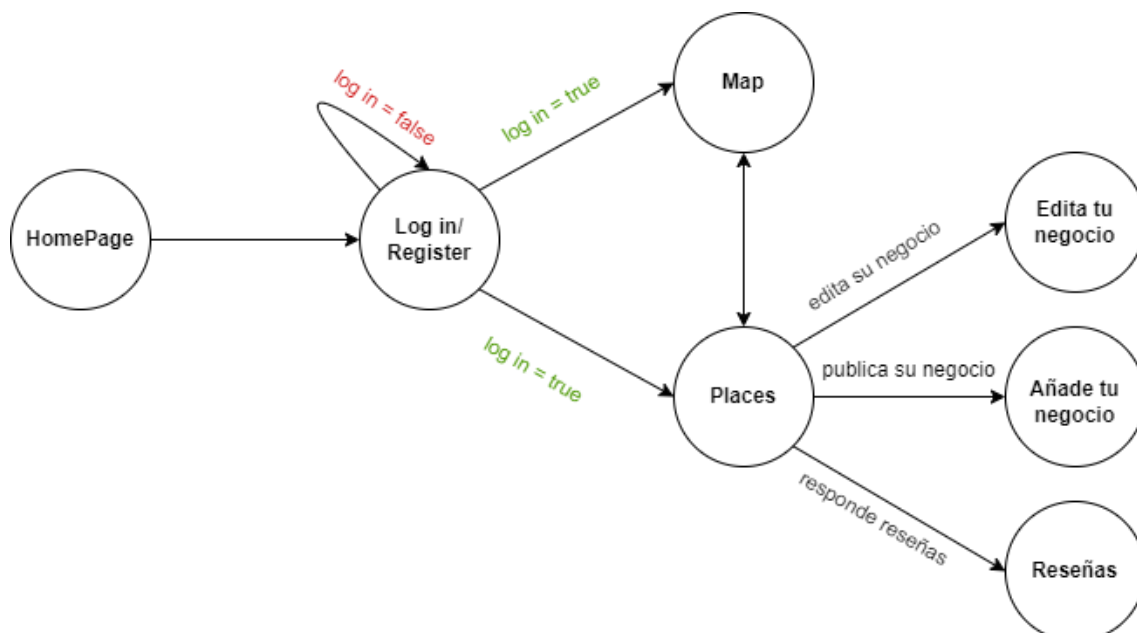
5.2 Flujos de navegación

Navegación del cliente



Lo primero que ve que el cliente es la página de inicio donde hay una breve explicación de lo que hay en la página web. A continuación, se tendrá que registrar o iniciar sesión, para poder acceder a los diferentes servicios. Entonces tendrá la posibilidad de navegar por el mapa viendo donde están localizados los diferentes establecimientos o acceder a la vista “places” donde podrá ver las especificaciones de los distintos locales y escribir reseñas.

Navegación del propietario



En el caso del propietario, cuando ya esté registrado, deberá publicar su negocio y en caso de que en algún momento lo desee, editarlo. Además, podrá responder a las reseñas de los clientes sobre su local.