

# Diseño CashMe



Carlos Moragón

Orianna Milone

Daniel Fernández

Javier Martínez

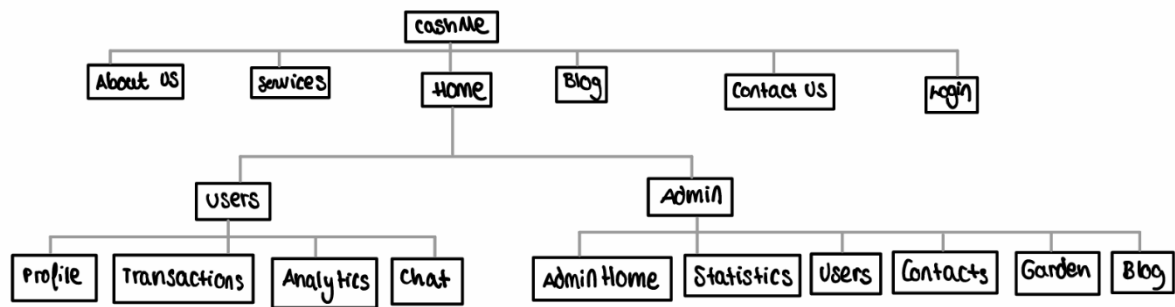
Ángel Castillo

Paloma Pérez de Madrid

## Índice

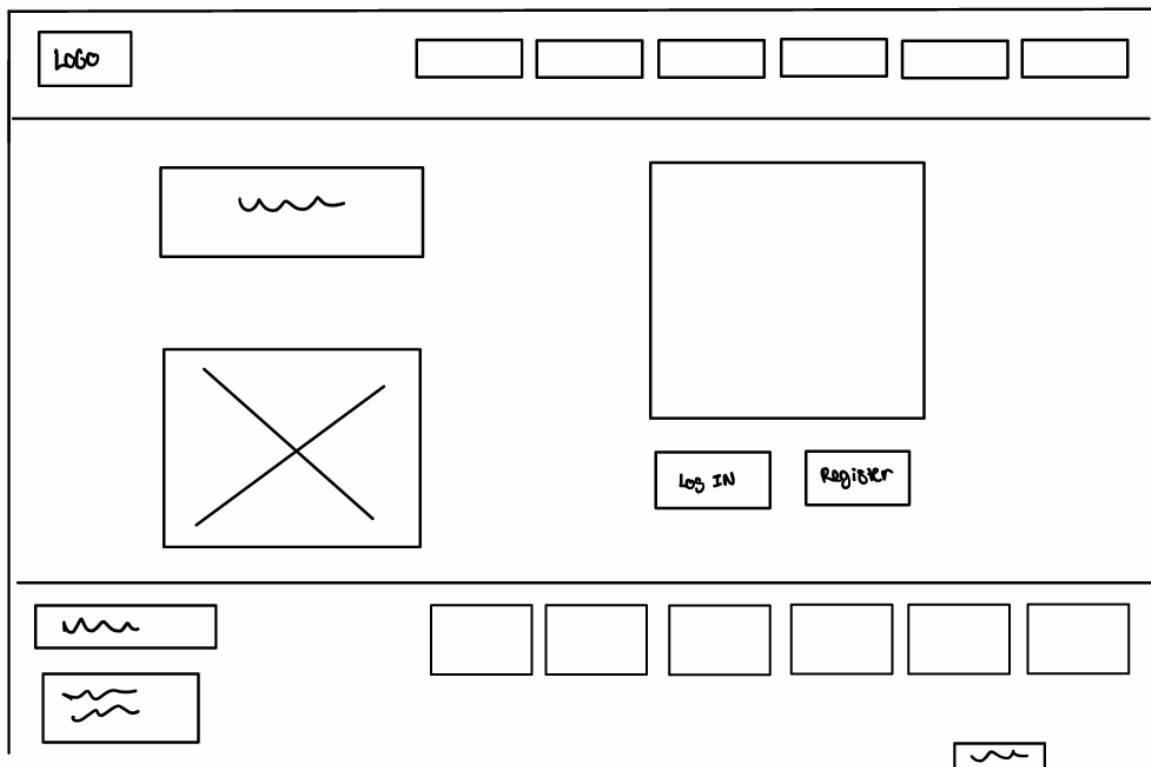
<b>SiteMap</b> .....	3
<b>Wireframe</b> .....	3
- Home .....	3
- About Us .....	3
- Services .....	5
- Blog .....	7
- Login .....	8
- Contact Us .....	8
- Profile .....	9
- Transactions .....	10
- Analytics .....	12
- Chat.....	13
- AdminHome & Statistics & Users.....	13
- AdminContact .....	15
- AdminBlog.....	15
- Garden .....	16
<b>Arquitectura</b> .....	18
Ejemplo de flujo en la arquitectura: .....	19
Estructura del proyecto: .....	20
Diagrama Entidad-Relación: .....	21

## SiteMap

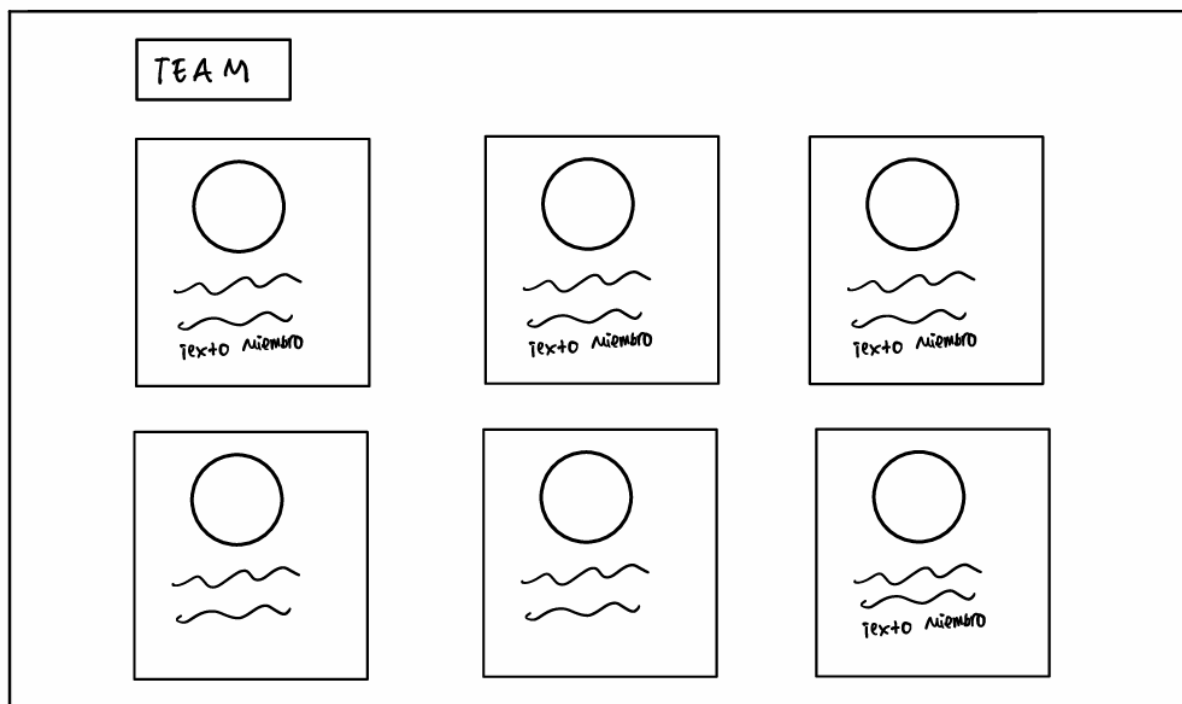
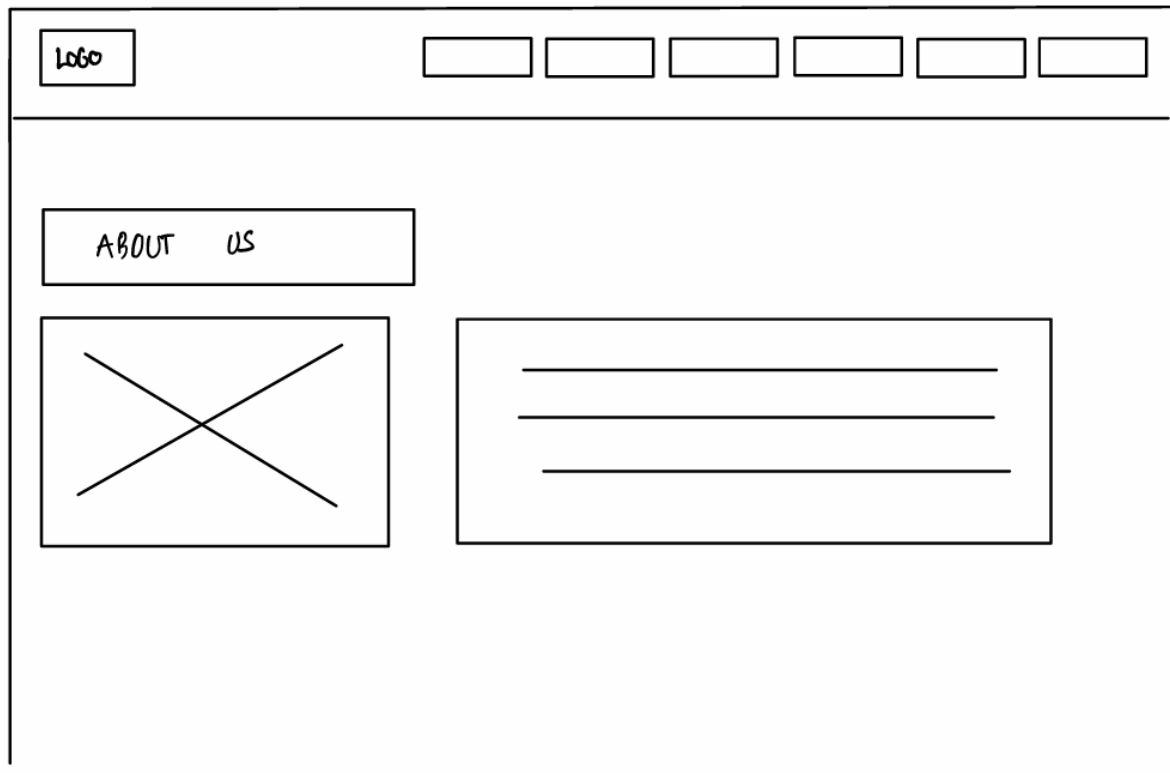


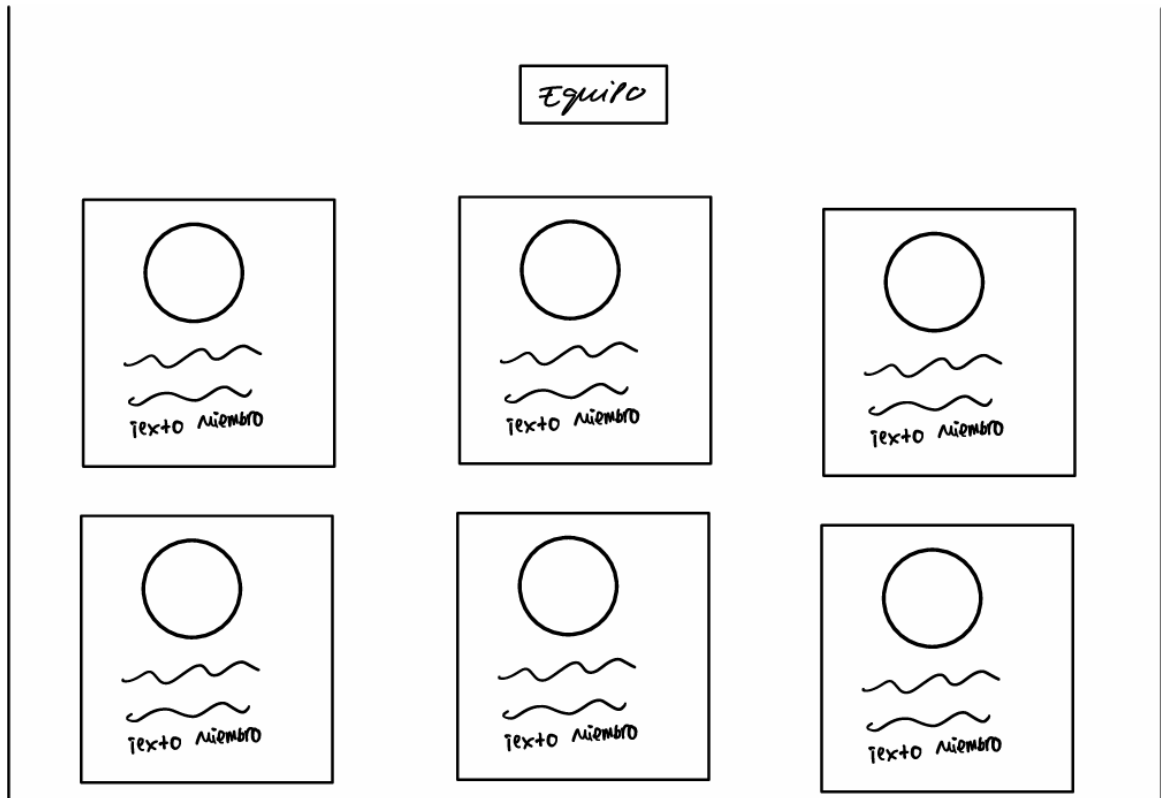
## Wireframe

- Home:

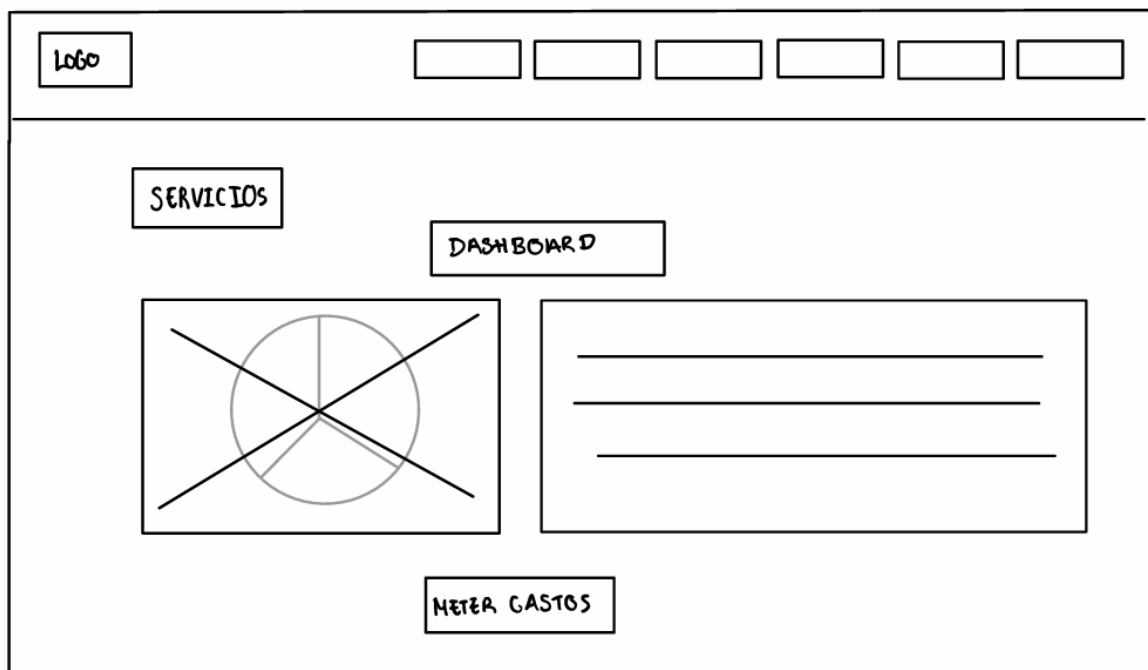


- About Us





- Services

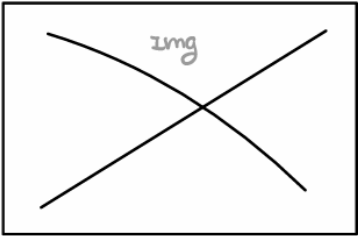


MEER GASTOS

---

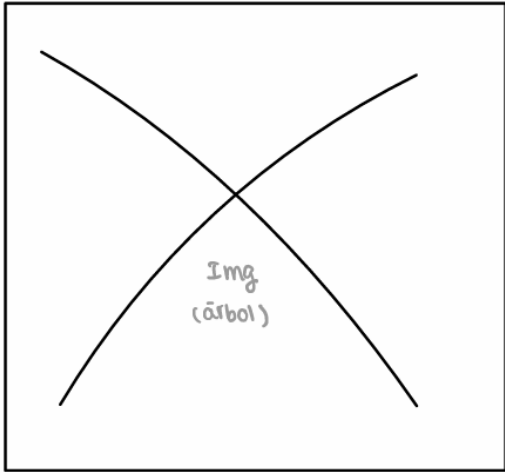
---

---



RECOMPENSAS

RECOMPENSAS

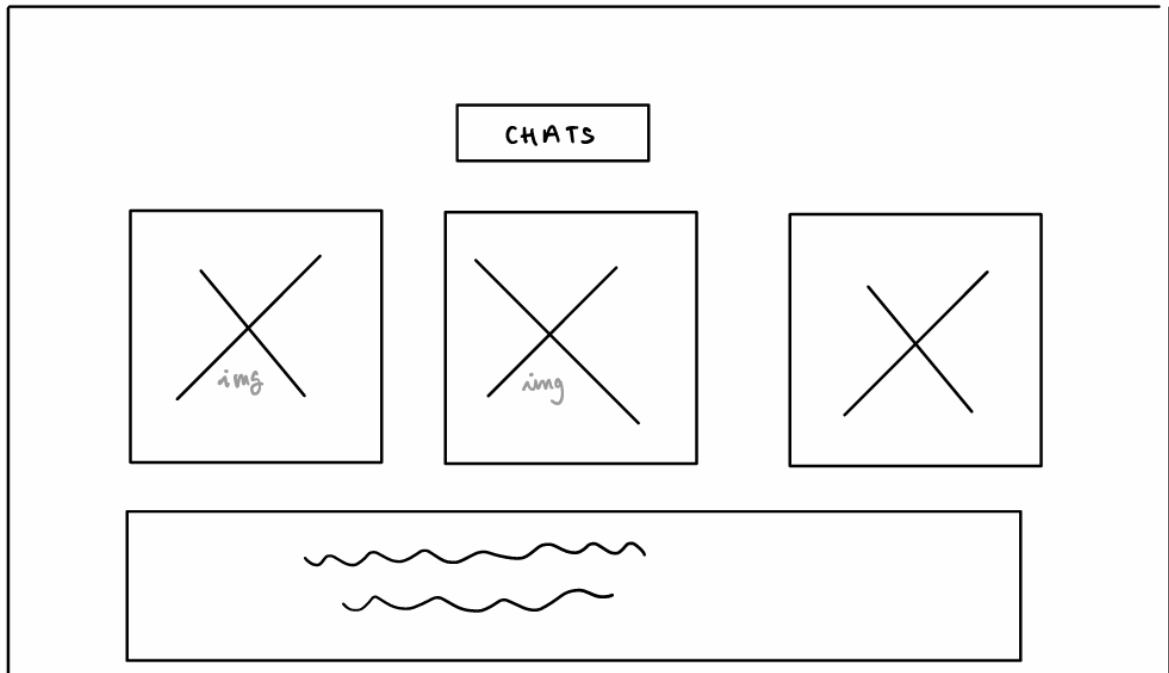


---

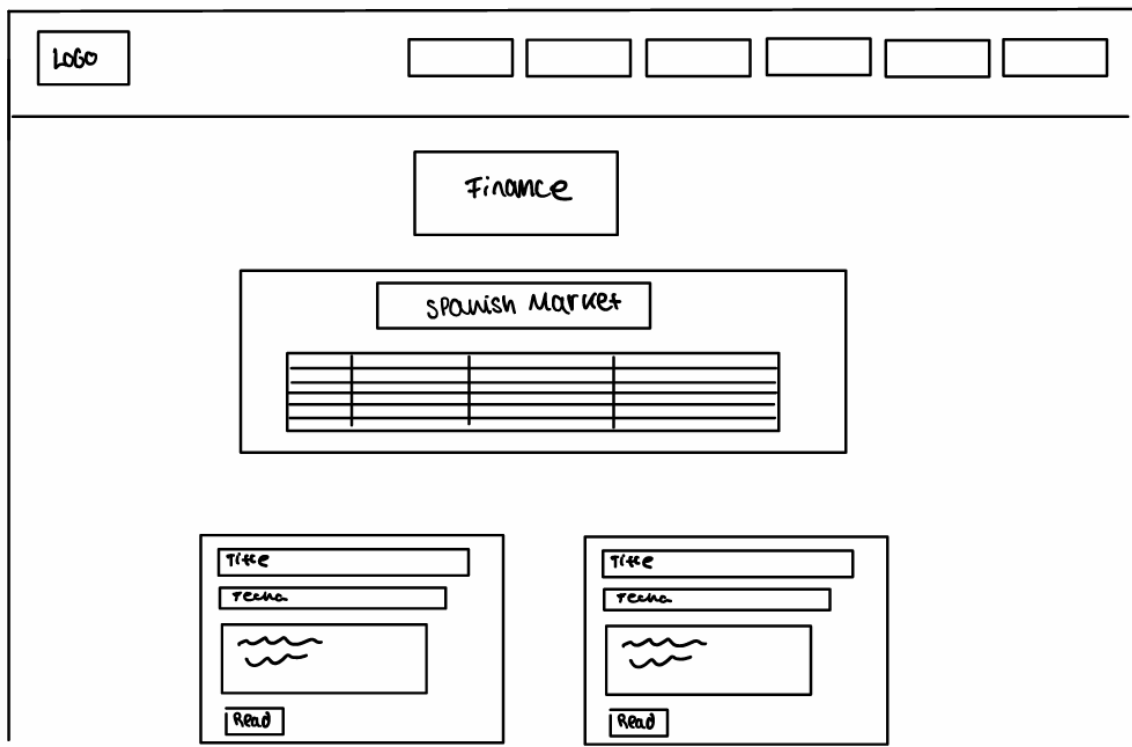
---

---

---



- Blog



- Contact Us

Logo

contact

name

email

Subject

Message

Send

- Login

Logo

log IN

log in

register



Logo

registrier

Submit

login

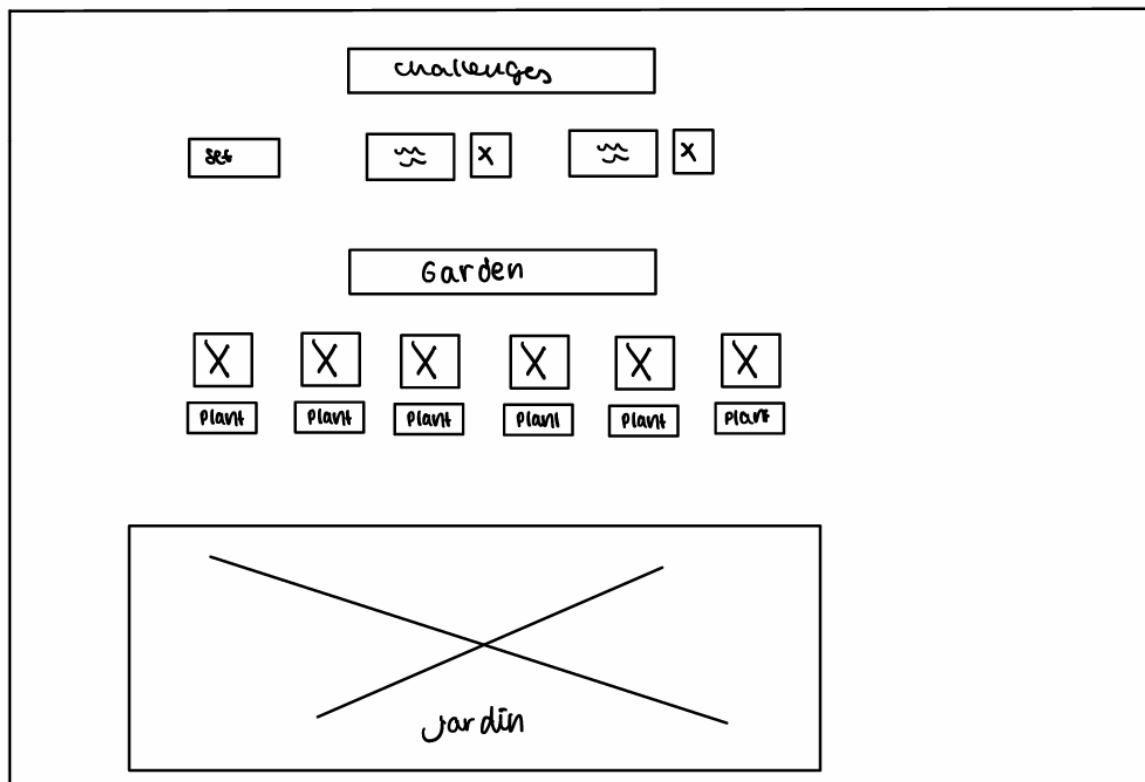
- Profile

Logo

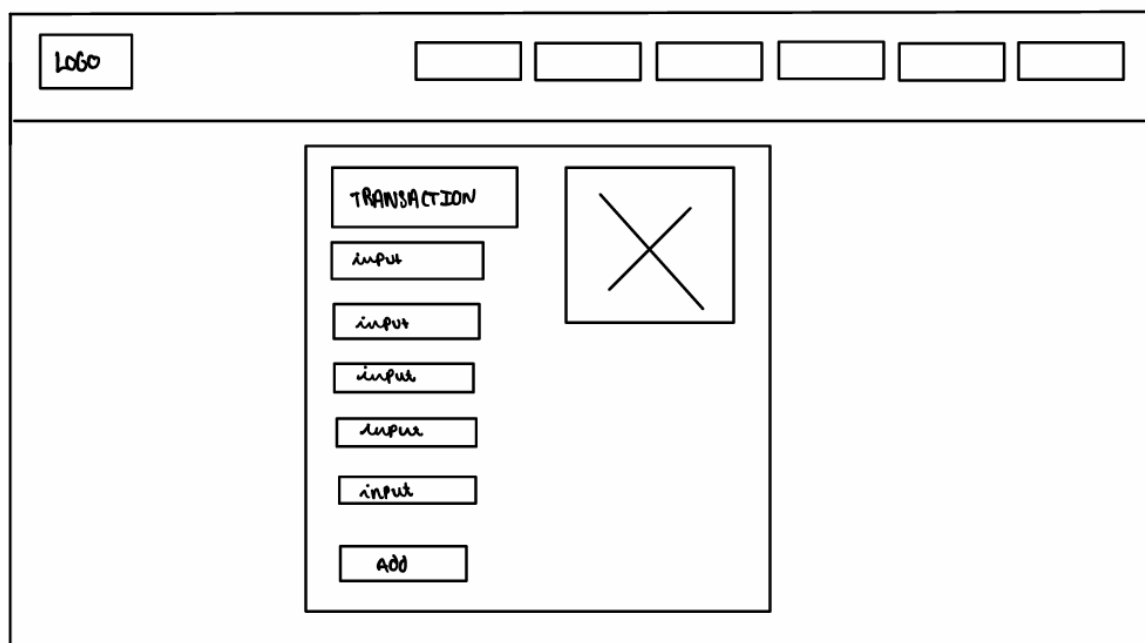
Avatar

client Data


Accounts

- Transactions



transaction history

export

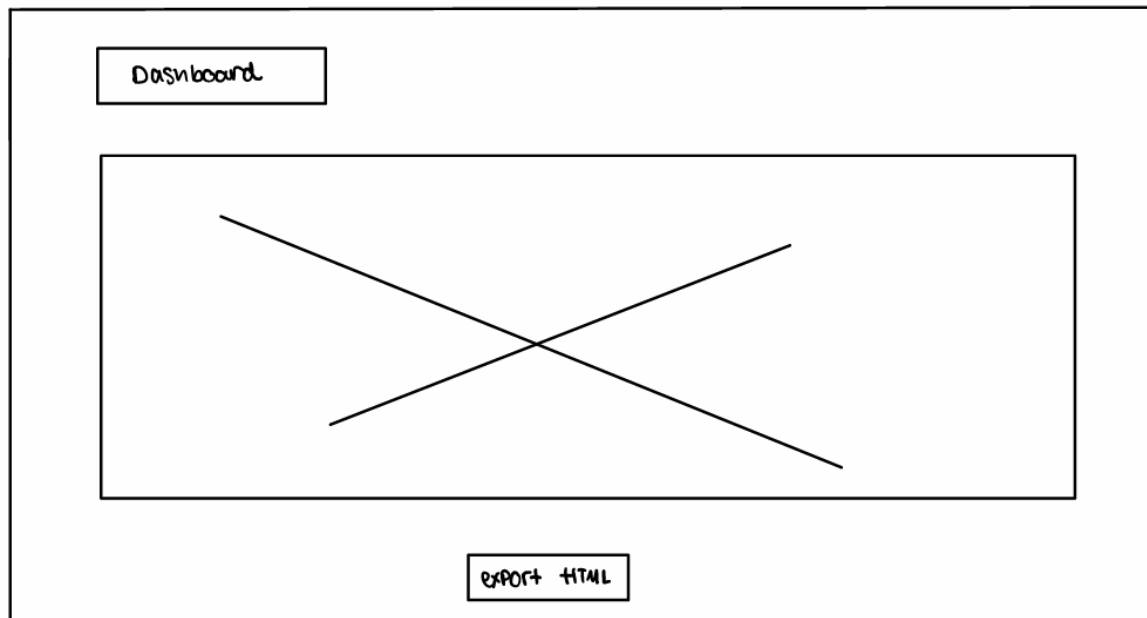
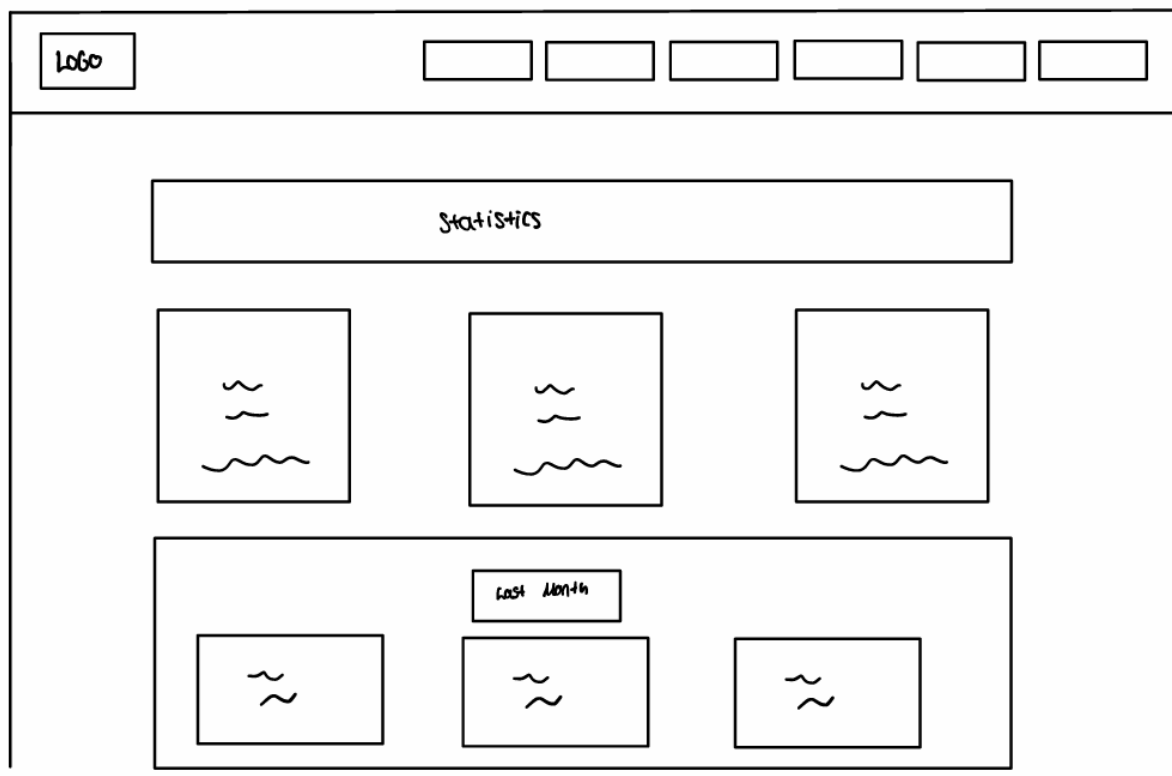
export

Import

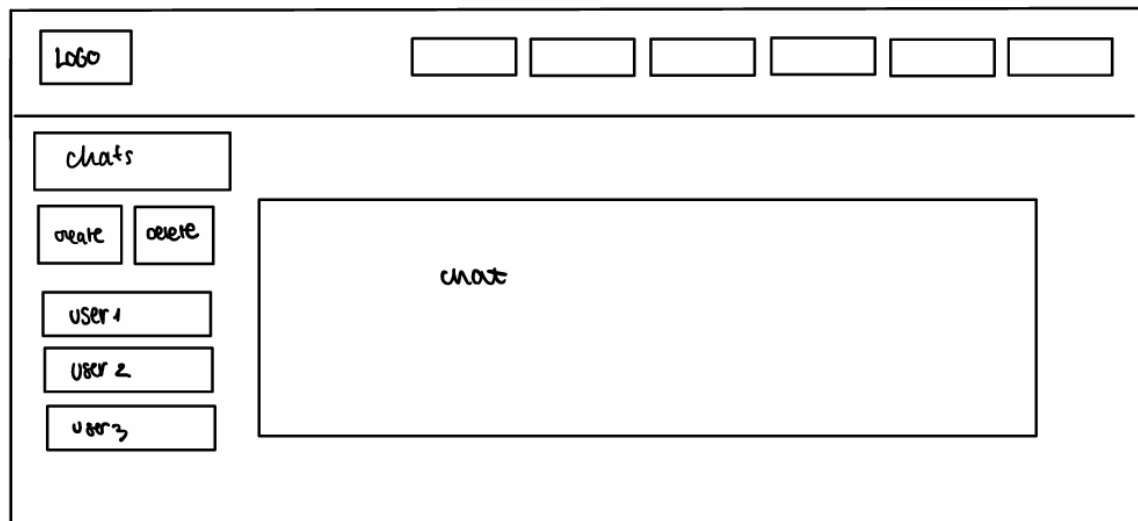
archivo

Import

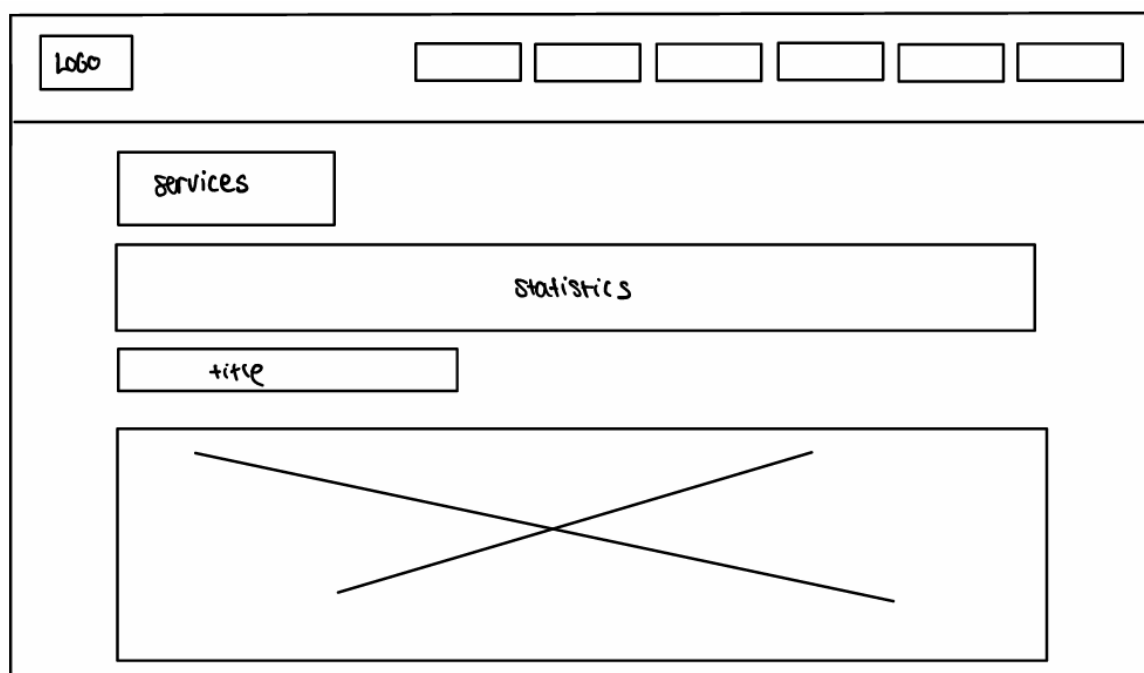
- Analytics



- Chat



- AdminHome & Statistic & Users



User List

⊗

~~~~~

⊗

~~~~~

⊗

~~~~~

⊗

~~~~~

Garden

⊗

~~~~~  
~~~~~

- AdminContact

Logo

contact Messages

~	~	~	~	~	~
~	~	~	~	~	delete
~	~	~	~	~	delete
~	~	~	~	~	delete
~	~	~	~	~	delete
~	~	~	~	~	delete
~	~	~	~	~	delete

- AdminBlog

Logo

add Entry

~

input

~

input

~

input

~

input

add Entry

edit/delete

~~~~~	~~~~~	~~~~~
~~~~~	~~~~~	<div>Edit</div> <div>modal</div>
~~~~~	~~~~~	<div>Edit</div> <div>modal</div>
~~~~~	~~~~~	<div>Edit</div> <div>modal</div>
~~~~~	~~~~~	<div>Edit</div> <div>modal</div>

- Garden

logo

Garden

~~~~~

new Activity

~~~~~

X

~~~~~

X

~~~~~

X

~~~~~

X

~~~~~

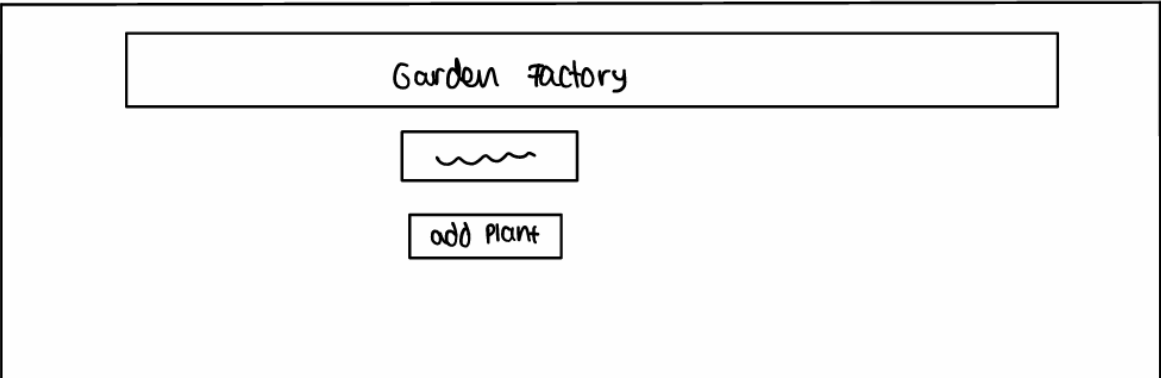
X

~~~~~

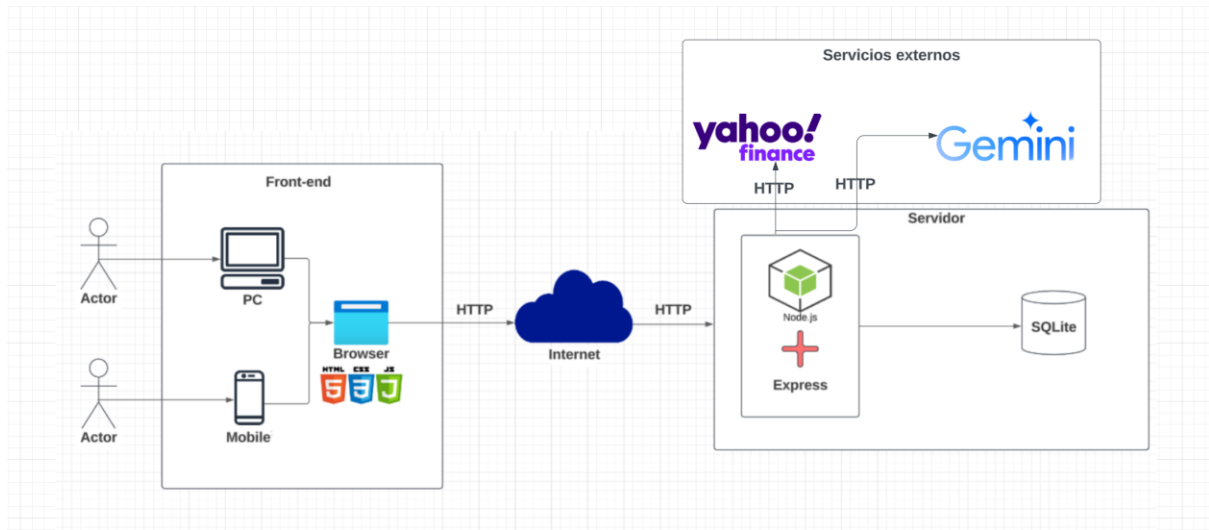
X

~~~~~





## Arquitectura



CashMe es una aplicación web que sigue una arquitectura de Cliente-Servidor. En esta arquitectura podemos identificar 4 elementos principales:

- Cliente:

Los actores utilizan dispositivos como ordenadores o móviles para interactuar con nuestro servidor a través de un navegador compatible con html, css y javascript.

El navegador actúa como cliente de la aplicación web, interactuando con ella a través de solicitudes HTTP.

- Internet:

Internet actúa como el medio de transporte de nuestras peticiones HTTP entre el cliente y el servidor.

Al utilizar Internet como medio de transporte, CashMe se hace accesible desde cualquier ubicación física.

- Servidor:

El servidor está compuesto por un servidor desarrollado en Node.js y el framework Express.

La función del servidor es manejar las peticiones enviadas desde el cliente, procesarlas y devolver una respuesta adecuada.

El servidor interactúa con una base de datos para asegurar la persistencia de los datos.

- Base de datos:

La base de datos es una instancia de SQLite, una base de datos liviana que no requiere un servidor de base de datos externo, siendo perfecto para CashMe, una aplicación de pequeña escala.

- Servicios externos:
  - Yahoo Finance:

Este servicio externo es accesible por el protocolo HTTP.

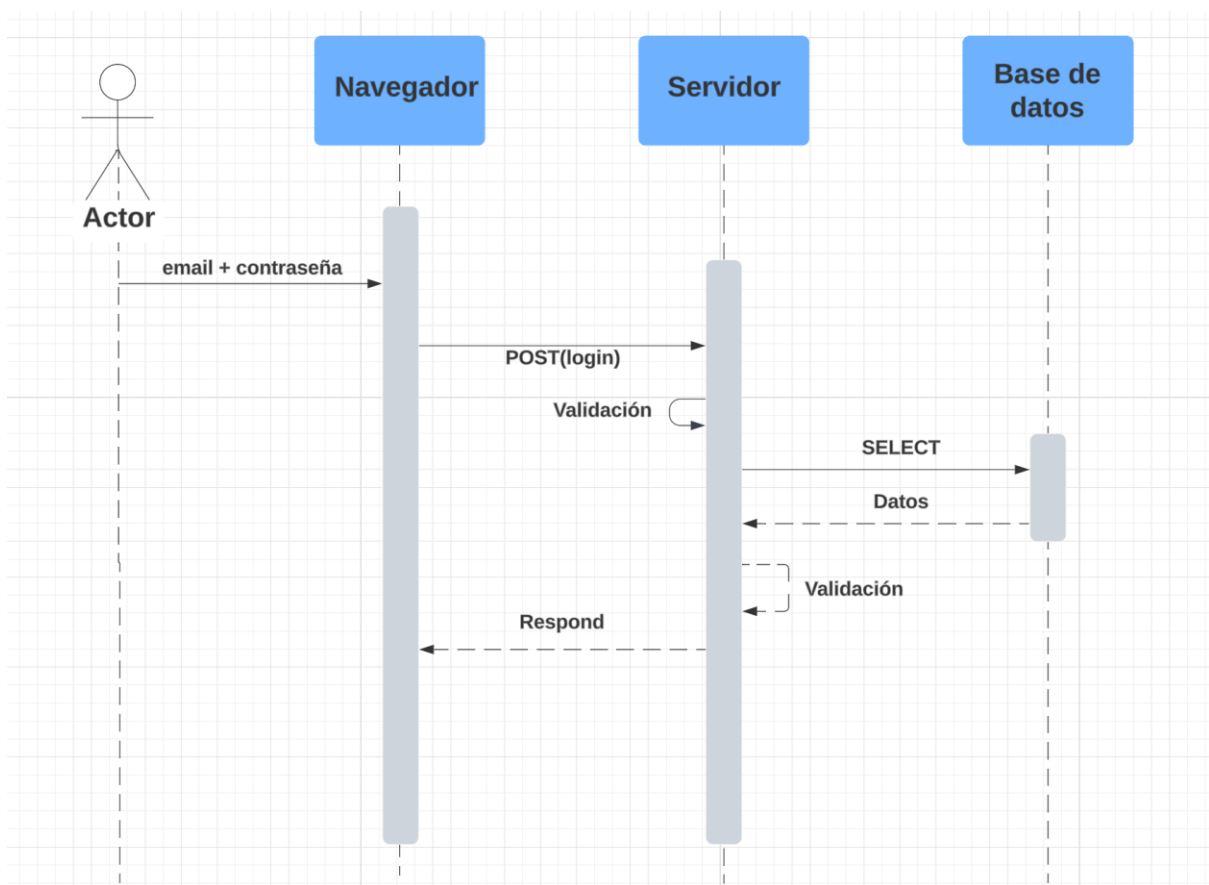
Utilizamos este servicio para obtener los índices bursátiles de forma actualizada.

- Gemini:

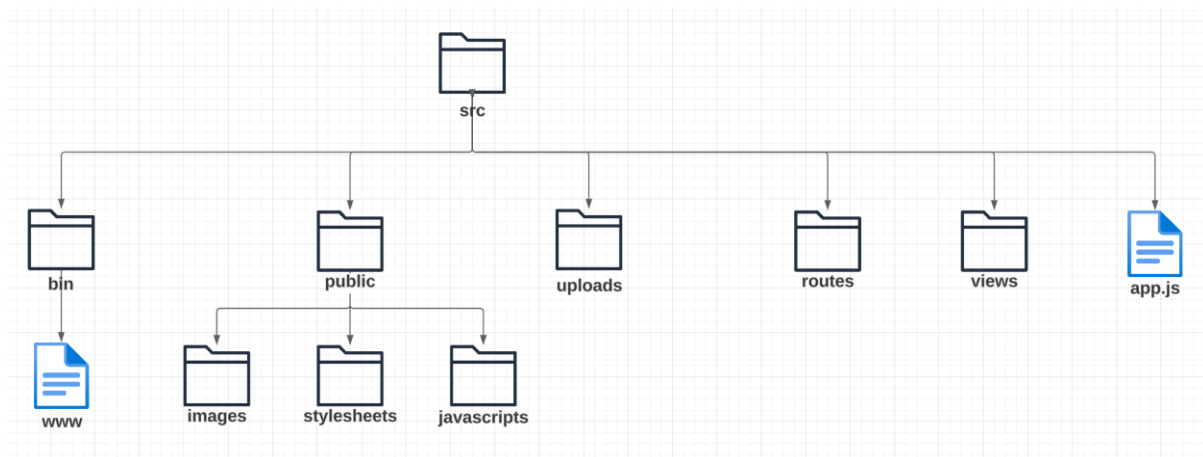
Este servicio externo es accesible por el protocolo HTTP.

El servicio lo proporciona la empresa Google, donde nos da acceso a una inteligencia artificial que funciona como chatbot en CashMe.

Ejemplo de flujo en la arquitectura:



## Estructura del proyecto:



- **app.js:**

App.js es el archivo principal de la aplicación. Es el punto de entrada donde se inicializa el servidor Express.

Aquí se configuran los middlewares, las rutas y otros componentes importantes como la configuración de plantillas o la conexión a bases de datos.

- **bin:**

Contiene scripts de inicialización, como el archivo `www`.

- **www:**

Se usa para arrancar el servidor. Este archivo gestiona la configuración del puerto y otros detalles específicos del entorno del servidor.

- **public:**

Esta carpeta es usada para almacenar recursos estáticos accesibles públicamente, como imágenes, hojas de estilos y archivos JavaScript.

- **Images:**

Contiene imágenes que serán usadas en la aplicación.

- **Stylesheets:**

Aquí se ubican las hojas de estilos CSS.

- **Javascripts:**

Archivos JavaScript que son usados del lado del cliente.

- **uploads:**

Esta carpeta se utiliza para almacenar archivos subidos por los usuarios.

- routes:

Aquí se almacenan los archivos que definen las rutas de la aplicación. Cada archivo representa un conjunto de rutas para un recurso específico.

- views:

Contiene las plantillas EJS que la aplicación renderiza.

Estas plantillas generan contenido HTML dinámico basado en los datos enviados desde el backend.

## Diagrama Entidad-Relación:

