

Proyecto Final de Grado (Fintrack)



Fintrack

Nombre del autor Victor Pinero Ollero

Fecha 30/04/2025

Nombre del tutor Fernando

Índice

1.Introducción de proyecto.....	3
1.2 Objetivos / Motivación.....	3
1.3 Problemas a Solucionar.....	3
2. Estudio del Mercado y Justificación.....	4
2.3 Situación actual y análisis de necesidades.....	4
2.4 Estudio de viabilidad.....	4
2.5 Análisis Dafo.....	5
3.Arquitectura de proyecto.....	6
3.1 Tecnologías empleadas.....	6
3.3 Base de datos Estructura.....	7
3.4 Envío de correos y notificaciones (Resend & Jobs).....	9
3.4.1 Factor autenticación en 2 pasos.....	10
3.4.5 Resumen Mensual por Email.....	10
3.4.5 IA para prompts.....	11
3.4.5 Gestor de Gastos divididos.....	11
4. Requisitos Funcionales y No Funcionales.....	12
4.1 Autenticación.....	12
4.2 Perfil de Usuario.....	13
4.3 Dashboard Financiero.....	13
4.4 Suscripción.....	14
4.5 Chatbot.....	14
4.6 Sistema de Gestión autenticación.....	15
4.7 Generación de Documentos.....	15
Requisitos No Funcionales.....	16
01: Rendimiento.....	16
02: Usabilidad.....	16
03: Seguridad.....	16
04: Compatibilidad.....	16
05: Mantenibilidad.....	16
5.Casos de uso y Escenarios.....	17
6. Presupuesto del proyecto y tiempo estimado.....	24
7.Mejoras Futuras.....	29

1.Introducción de proyecto

1.2 Objetivos / Motivación

El objetivo principal es desarrollar una aplicación financiera , que te permite administrar tus ingresos y gastos con herramientas inteligente , diseño moderno y experiencia intuitiva de usuario

1.3 Problemas a Solucionar

También pretende aplicar soluciones a problemas reales , como la falta de educación económica , para ello aportamos un sistema de newsletter que se podrá configurar en la sección del perfil de usuario , donde semanalmente enviaremos un correo con los artículos más destacados , así como noticias importantes . Otra de las propuestas que aportamos es la famosa IA , que hoy en día la tenemos presente en casi todo , en nuestro caso , es una herramienta fundamental para manejar datos numéricos como ingresos y gastos , queremos que el usuario no tenga que estar rompiéndose la cabeza con preguntas como : ¿Donde gastó más dinero? , ¿Como puedo ahorrar más? , ¿Cuánto tendría que ahorrar para comprarme x? ...

Y a quien no le ha pasado ir a cenar , hacer un regalo o comprar los vuelos para un viaje y tener que dividir los gastos entre amigos o familiares... Los famosos “paga tu y te hacemos bizum” , una estrategia peligrosa para el que paga . Es muy agotador tener que estar recordado fulanito me debe x , Juan me debe y . Para ello cuando se produzca un gasto con un concepto relacionado (“configurable desde el perfil de usuario”) podremos utilizar un modal que te irá acompañando con preguntas del tipo : ¿Este es un gasto compartido ? ¿Con quienes comparte el gasto ? De esta manera si no te acuerdas quien ha pagado que o si te deben dinero podras comprobarlo rápido sin tener que estresarte.

2. Estudio del Mercado y Justificación

2.3 Situación actual y análisis de necesidades

Actualmente, las aplicaciones de gestión financiera como Mint y Goodbudget ofrecen soluciones valiosas pero presentan limitaciones significativas para ciertos usuarios. Mint, aunque automatizada y con múltiples integraciones, está restringida geográficamente y contiene publicidad que puede afectar la experiencia. Goodbudget, en cambio, ofrece un método tradicional efectivo, pero carece de automatización, lo que puede resultar poco práctico para usuarios con menos tiempo o experiencia.

Existe una necesidad clara de una aplicación que combine lo mejor de ambos mundos: una plataforma intuitiva, automatizada, sin restricciones geográficas ni publicidad intrusiva, y que además permita personalización y educación financiera integrada. Esto responde a una demanda creciente de usuarios que buscan gestionar sus finanzas de manera sencilla, eficiente y educativa.

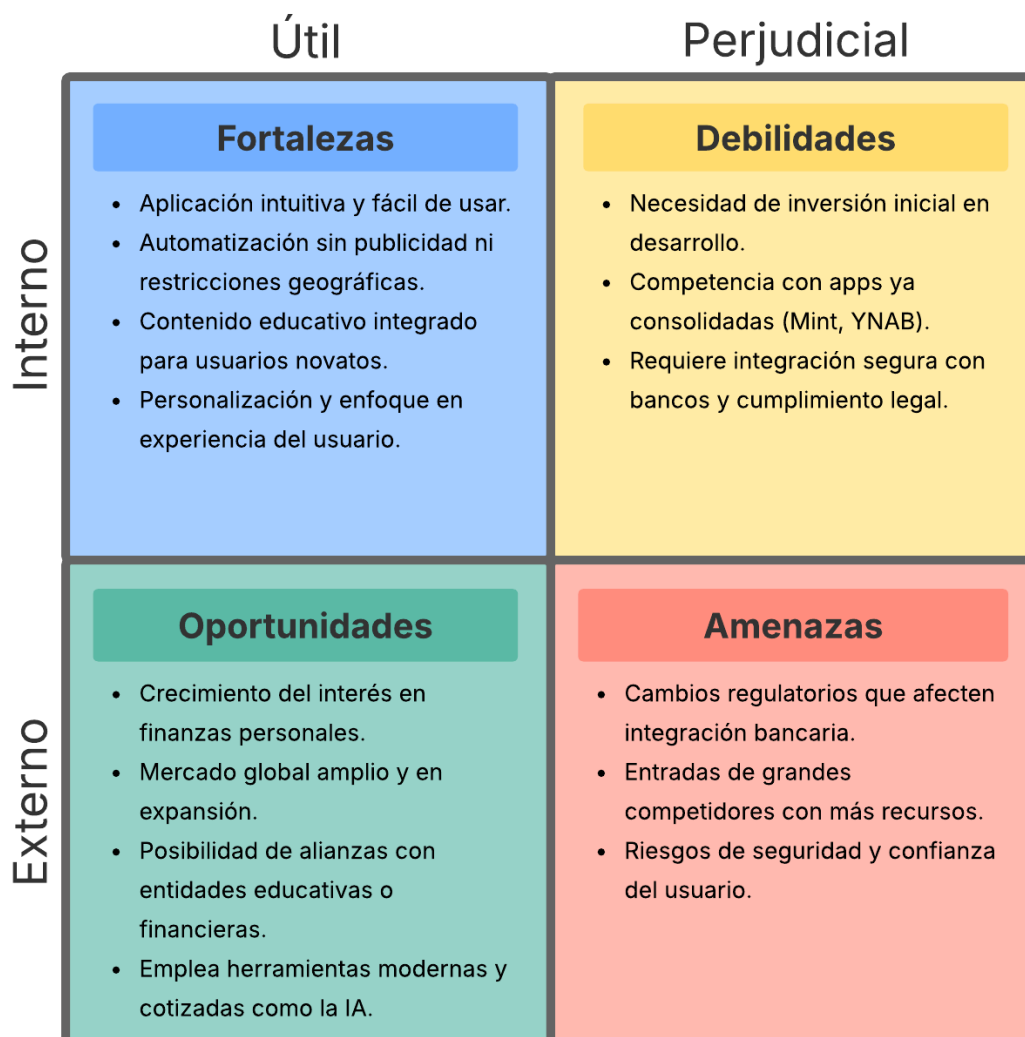
2.4 Estudio de viabilidad

La propuesta de desarrollar una nueva aplicación financiera es viable dado el creciente interés en la gestión personal del dinero y la educación financiera. Técnicamente, las tecnologías actuales permiten integrar múltiples cuentas bancarias, ofrecer análisis personalizados y notificaciones en tiempo real, todo dentro de una experiencia amigable y sin publicidad intrusiva.

Económicamente, el mercado global de apps financieras sigue en expansión, con oportunidades para innovar en funcionalidad y accesibilidad. Además, la incorporación de contenido educativo dentro de la app puede diferenciarse y aportar un valor añadido que atraiga y retenga usuarios.

Por tanto, existe un nicho claro y favorable para lanzar una aplicación que supere las limitaciones de las opciones actuales, enfocándose en automatización, usabilidad, educación financiera y experiencia libre de distracciones comerciales.

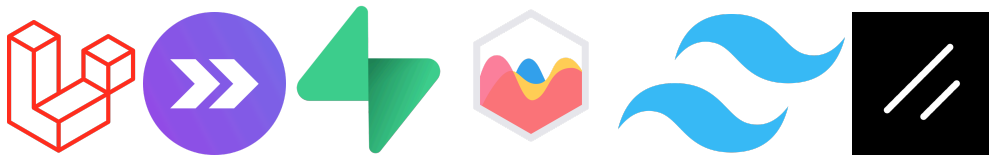
2.5 Análisis Dafo



3.Arquitectura de proyecto

3.1 Tecnologías empleadas

Para el desarrollo de **Fintrack**, se ha seleccionado un conjunto de tecnologías modernas y bien soportadas, que permiten crear una aplicación web robusta y eficiente. Las tecnologías elegidas son:

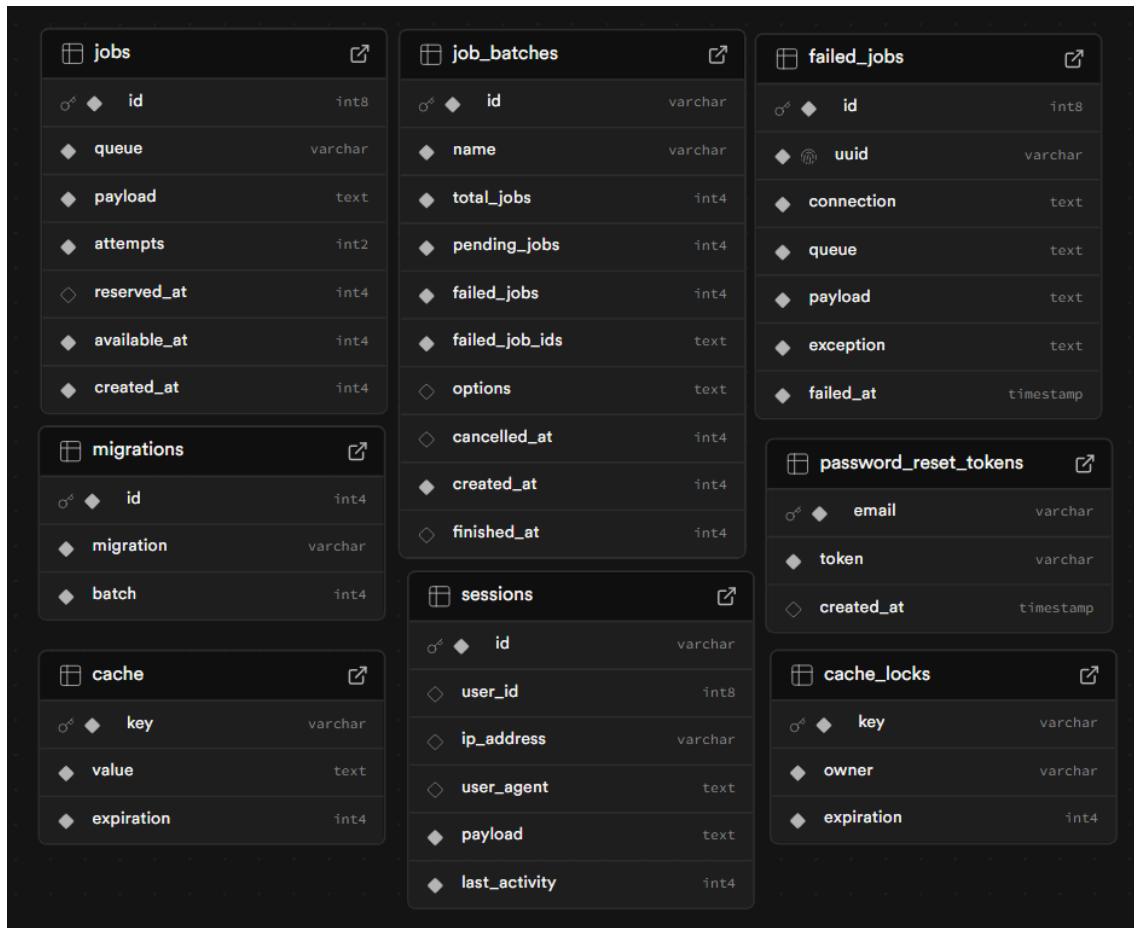


- Laravel (Backend): Un framework PHP moderno y eficiente que gestiona la lógica de la aplicación, las bases de datos y la seguridad.
- Inertia.js (Frontend): Una herramienta que facilita la creación de aplicaciones SPA (Single Page Application) manteniendo la simplicidad en el desarrollo.
- Supabase (Base de Datos): Plataforma de código abierto basada en PostgreSQL que facilita la gestión de bases de datos, autenticación y funciones en tiempo real.
- Chart.js (Visualización de Datos): Biblioteca de JavaScript para crear gráficos interactivos y dinámicos.
- Tailwind CSS (Estilo y Diseño): Framework CSS que permite crear interfaces de usuario personalizadas y adaptativas.
- Shad Cn/ui: Plataforma de componentes modernos Ui que facilitan la integración con las vistas de React

3.3 Base de datos Estructura

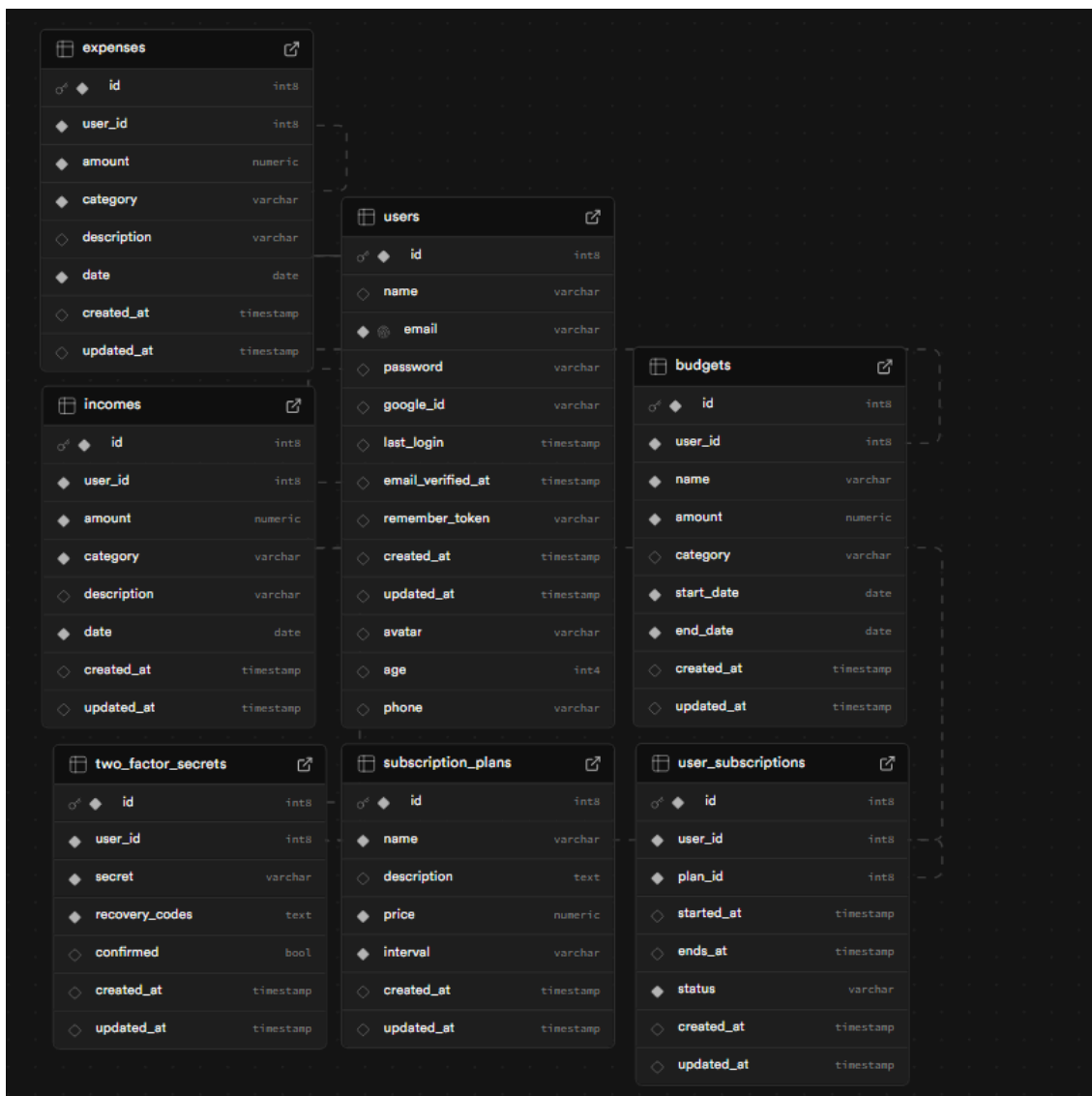
<https://dbdiagram.io/d>

Estas son las tablas que vienen por defecto en Laravel cuando ejecutamos las migraciones pero en ellas podremos almacenar los jobs programados , las migraciones ejecutadas , sesiones etc



Proyecto Final de Grado (Fintrack)

Estas son las tablas funcionales , es decir , donde vamos a almacenar lo más importante de la app , usuarios , gastos , ingresos , 2fa , suscripciones etc ..



3.4 Envío de correos y notificaciones (Resend & Jobs)

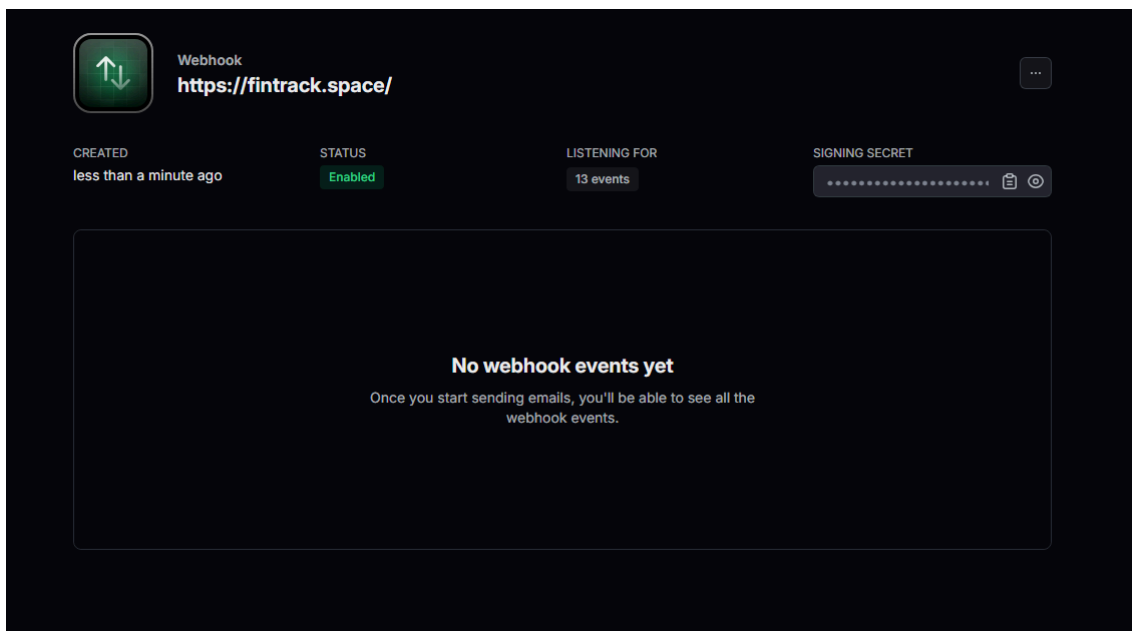
Para enviar correos utilizaremos la librería resend , la cual hace poco ha sido añadida ya a laravel como método de envío de correos , por lo que no tendremos que cambiar mucha cosa.

Solo un par de keys en nuestro .env y en config.mail cambiar esta linea

```
'default' => env('MAIL_MAILER', 'smtp'),  
//Cambiamos el método smtp por resend
```

¿El porqué de elegir renders? Es sencillo , primero porque es muy sencillo de utilizar y lo segundo es por la gran facilidad que nos aporta a los desarrolladores . Esta plataforma cuenta ya con algún hooks de gran interés como los siguientes =>





Como vemos en la imagen podemos ver métricas y datos de interés como cuantos usuarios han hecho click en nuestro email , si han dado click a un enlace interno del email etc . Lo que nos dará un gran feedback de estrategias a la hora de elegir qué tipo de contenido enviar a nuestro usuarios .

3.4.1 Factor autenticación en 2 pasos

Para la seguridad de autenticación usaremos el 2fa , que no es otra cosa que cuando un usuario se registra por primera vez , se le redirigirá a una ventana nueva donde tendrá un tiempo , en nuestro caso de 5 min , para que pueda meter un numero de 5 dígitos que se le enviará con el correo que se ha intentado loguear con el objetivo de evitar suplantación de identidad , en términos general , aportar más seguridad a la aplicación.

3.4.5 Resumen Mensual por Email

- Mini resumen de lo más importante cada domingo.
- 💡 Valor: ofrece información relevante sin saturar.
- 🎯 Técnica: ejecuta un Job cada final de mes con un resumen breve (total, categoría top, aviso si superó su promedio).

3.4.5 IA para prompts

En la ruta chatbot encontraremos un chatbot , con la librería langchain , que nos permite guardar en un buffer de memoria los props anteriores todo esto para poder simular una conversación con un bot que actúa como gestor de finanzas el cual recibe antes de empezar a hablar contigo un contexto de tu edad , una media de ingresos y gastos para poder responder mejor . Que te ayudará con planes de ahorro , inversiones etc

3.4.5 Gestor de Gastos divididos

Ahora quiero implementar un gestor de gastos divididos como tricount , pero que al recibir un ingreso con un concepto que esté dentro de palabras claves nos permita eliminar a esa gente automáticamente , es decir vamos a imaginar que salimos con nuestros amigos a cenar .

Y que pagas tu , si el gasto se registra como un gasto alto y de tipo Comidas , viajes o algo así .

Podremos crear un gestor de gastos , que nos permita dividir el gasto entre los usuarios que queremos , y que si recibimos un ingreso de alguno de esas personas con el concepto Viaje o cena , esa persona se elimine de la lista de deudores o se pongo como pagado , para que no tengas que estar recordando quién te ha pagado y quien no .

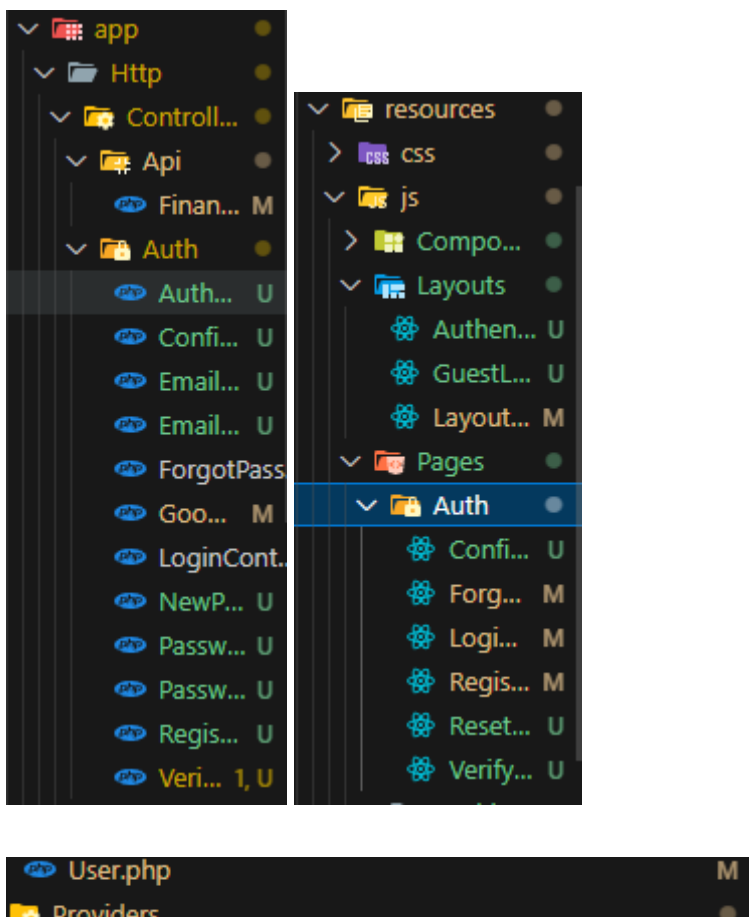
4. Requisitos Funcionales y No Funcionales

4.1 Autenticación

Gestiona el registro, login, recuperación de contraseña y seguridad.

Funciones:

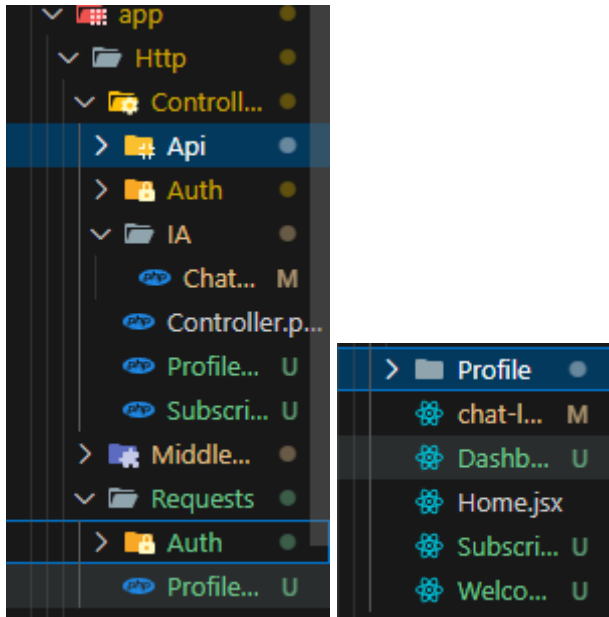
- Registro de nuevos usuarios
- Autenticación mediante email y contraseña
- Gestión de sesiones seguras
- Control de acceso según roles (Jefe y Empleado)



4.2 Perfil de Usuario

Permite ver y actualizar información personal de los usuarios.

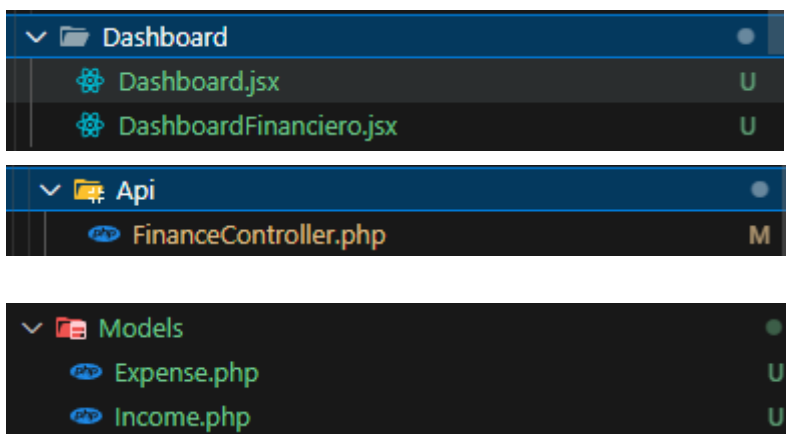
Funciones: Editar datos personales, cambiar contraseña, subir avatar, actualizar teléfono, email y otros datos relevantes



4.3 Dashboard Financiero

Permite ver y comparar la información personal del usuario de sus gastos e ingresos


Funciones: Ver y comparar gastos en función del tipo y fecha .




4.4 Suscripción

Permite mostrar los planes de suscripción que tiene la app .

Funciones: El usuario podrá consultar en su perfil el tipo de suscripción .

 SubscriptionPlanController.php U

 SubscriptionPlans.jsx U

4.5 Chatbot

Chatbot financiero que acepta prompts de consultas financieras

Funciones: Hablar con un chat financiero que te ayuda con consultas tipo plan de ahorros , gestión de activos etc.

▼ langchain-bot
 > node_modules
 chat.js
 index.js
 package-lock.json
 package.json













▼ app
 ▼ Http
 ▼ Controllers
 ▼ Api
 FinanceController.php M
 > Auth
 ▼ IA
 ChatController.php M
 Controller.php
 ProfileController.php U
 SubscriptionPlanController.php U

 chat-longchain.jsx M

4.6 Sistema de Gestión autenticación

Plan de autenticación sólido con sesiones , recuperación de contraseña , email de verificación etc.

Funciones: Esto garantiza rutas protegidas y un sistema de login sólido .

Auth		
	AuthenticatedSessionController.php	U
	ConfirmablePasswordController.php	U
	EmailVerificationNotificationController.php	U
	EmailVerificationPromptController.php	U
	ForgotPasswordController.php	
	GoogleController.php	M
	LoginController.php	
	NewPasswordController.php	U
	PasswordController.php	U
	PasswordResetLinkController.php	U
	RegisteredUserController.php	U
	VerifyEmailController.php	1, U

4.7 Generación de Documentos

Creación automatizada de contratos y documentos oficiales.

Funciones: Generar contratos PDF listos para imprimir o descargar según plantilla y datos ingresados.

Para ello empleados la librería jspdf para imprimir en nuestro caso la conversación con el chatbot

Requisitos No Funcionales

01: Rendimiento

- Carga inicial < 3 segundos
- Respuesta API < 500 ms

02: Usabilidad

- Interfaz intuitiva y responsive
- Navegación clara y coherente
- Mensajes de error comprensibles

03: Seguridad

- Contraseñas encriptadas con bcrypt
- Validación exhaustiva de entrada de datos
- Protección de sesiones y tokens

04: Compatibilidad

- Soporte para Chrome, Firefox, Safari, Edge
- Compatibilidad con móviles y tablets

05: Mantenibilidad

- Código modular y documentado
- Arquitectura escalable

5. Casos de uso y Escenarios

CU-02: Registrar Usuario

Campo	Descripción
ID	CU-02
Actor	Usuario nuevo
Descripción	Permite a un nuevo usuario registrarse en el sistema
Precondiciones	El usuario no debe estar registrado previamente
Flujo Principal	<ol style="list-style-type: none"> 1. El usuario accede a la página de registro 2. Completa el formulario 3. El sistema valida la información 4. Se crea la cuenta
Postcondiciones	El usuario queda registrado en el sistema

CU-03: Iniciar Sesión (Login)

Campo	Descripción
ID	CU-03
Actor	Usuario registrado
Descripción	El usuario debe estar registrado
Precondiciones	El usuario debe haber iniciado sesión
Flujo Principal	<ol style="list-style-type: none"> 1. Usuario accede al formulario de login 2. Ingresa email y contraseña 3. El sistema valida 4. Accede al dashboard
Postcondiciones	El usuario accede al sistema y se inicia su sesión

CU-04: Ver Dashboard Financiero

Campo	Descripción
ID	CU-04
Actor	Usuario autenticado
Descripción	Permite visualizar ingresos, gastos y métricas financieras
Precondiciones	El usuario debe haber iniciado sesión
Flujo Principal	1. Usuario accede al dashboard 2. El sistema carga datos financieros 3. Se muestran métricas y gráficos
Postcondiciones	El usuario tiene una visión clara de su estado financiero

CU-05: Interactuar con el Chatbot Financiero

Campo	Descripción
ID	CU-05
Actor	Usuario autenticado
Descripción	Permite hacer preguntas o recibir recomendaciones del asistente inteligente
Precondiciones	Usuario con sesión iniciada
Flujo Principal	<ol style="list-style-type: none"> 1. Usuario abre el chatbot 2. Envía consulta 3. El chatbot analiza y responde
Postcondiciones	El usuario recibe respuestas útiles y asistencia

CU-06: Suscribirse a un Plan Premium

Campo	Descripción
ID	CU-06
Actor	Usuario registrado
Descripción	Permite contratar una suscripción con beneficios adicionales
Precondiciones	El usuario debe tener cuenta
Flujo Principal	<ol style="list-style-type: none"> 1. Usuario accede a la sección de planes 2. Selecciona uno 3. Realiza el pago 4. El sistema activa los beneficios
Postcondiciones	El usuario accede a funcionalidades premium

CU-07: Activar o Configurar Doble Autenticación (2FA)

Campo	Descripción
ID	CU-07
Actor	Usuario registrado
Descripción	Permite activar la autenticación en dos pasos para mayor seguridad
Precondiciones	El usuario debe haber iniciado sesión
Flujo Principal	<ol style="list-style-type: none"> 1. Usuario accede a configuración de seguridad 2. Selecciona activar 2FA 3. Escanea el QR en la app autenticadora 4. Ingresa el código de verificación 5. El sistema activa el 2FA
Postcondiciones	2FA queda configurado en la cuenta del usuario

CU-8: Elegir Plan de Suscripción

Campo	Descripción
ID	CU-8
Actor	Usuario registrado
Descripción	Permite al usuario recuperar el acceso en caso de olvidar su contraseña
Precondiciones	El usuario debe estar registrado y haber olvidado su contraseña
Flujo Principal	<ol style="list-style-type: none"> 1. Usuario accede a "¿Olvidaste tu contraseña?" 2. Ingresa su email 3. El sistema envía un enlace de recuperación por correo 4. Usuario accede al enlace y establece una nueva contraseña
Postcondiciones	El usuario puede iniciar sesión con su nueva contraseña

6. Presupuesto del proyecto y tiempo estimado

Presupuesto detallado para la app financiera

1. APIs

OpenAI (ChatGPT API)

- Modelo: GPT-4 Turbo (precio aproximado \$0.03/1000 tokens)
- Supongamos un uso promedio diario de 20,000 tokens (entrada + salida) para generación de contenido, consultas y respuestas.
- Tokens mensuales: $20,000 \times 30 = 600,000$ tokens
- Coste mensual = $600 \times 0.03 = \$18$ (≈ 17 €)

Open Banking API

- Proveedor internacional: **Plaid** (muy popular en USA)
 - Plan básico: \$500/mes (con acceso a cuentas, transacciones y balances)
 - Plan startups o más pequeños pueden costar alrededor de \$200/mes
- En Europa, proveedores como **TrueLayer** o **Tink** ofrecen planes similares.
 - TrueLayer: plan básico gratuito limitado, luego desde €100 al mes.

Referencias de precios →

1. [Referencia precios OpenAI API](#)
2. [Truelayer](#)
3. [Plaid](#)

2. Desarrollo

Fases y estimación

- Diseño UI/UX: 1,000 – 2,500 €
- Desarrollo frontend y backend (integración APIs + app móvil/web): 7,000 – 12,000 €
- Pruebas y ajustes: 1,000 – 2,000 €

Total desarrollo: 9,000 – 16,500 €

3. Mantenimiento mensual

- Actualizaciones, servidores, seguridad, soporte: 400 – 1,000 €
 - Hosting cloud (AWS, Azure o Google Cloud): 100 – 300 €
 - Costes API (OpenAI + Open Banking): ~120 – 130 €
 - Legal, GDPR y auditorías: 100 – 300 € (coste recurrente)
-

4. Marketing y difusión

- Campañas iniciales en redes sociales, Google Ads, colaboraciones: 500 – 1,500 €
- Creación de contenido (blog, videos, newsletter): 200 – 500 € mensuales

Proyecto Final de Grado (Fintrack)

Concepto	Coste inicial	Coste mensual
Diseño y desarrollo	9,000 – 16,500	
Apis (OpenAI , Openbank)		120 – 130
Hosting y mantenimiento		500 – 1,300
Legal y seguridad	1,000 – 3,000	100 - 300
Marketing	500 – 1,500	200 - 500

Proyecto Final de Grado (Fintrack)
















En cuanto al tiempo estimado ...

Módulo	Descripción	Tiempo Estimado
1. Autenticación básica (Login/Registro)	Formularios, validaciones, verificación email, recuperar contraseña	10–14 h
2. Perfil de usuario	Editar nombre, edad, teléfono, subir avatar, validaciones	6–8 h
3. 2FA (Autenticación de dos factores)	Implementación de códigos temporales por TOTP (ej: Google Authenticator)	6–10 h
4. Gestión de ingresos/gastos	CRUD completo (formulario, listado, filtros, validación, DB)	12–16 h
5. Dashboard financiero	Resumen de movimientos, estadísticas, gráficos	8–12 h
6. Suscripciones y planes	Elección de plan, pago simulado o real (ej: Stripe), límites por plan	10–14 h
7. Chatbot financiero	Interfaz de chat + lógica base (sin IA compleja, tipo FAQ o info guía)	6–8 h
8. Historial y sesiones activas	Listar sesiones, ubicación, fecha, posibilidad de cerrar sesiones	4–6 h
9. Recuperar contraseña	Envío de email, token, formulario de nueva contraseña	3–4 h
10. Configuración y tests básicos	Rutas, permisos, middleware, validaciones, pruebas manuales	6–10 h
11. UI y mejoras visuales	TailwindCSS, responsive, UX básica	6–8 h


Proyecto Final de Grado

(Fintrack)

[Diagrama link](#)

	Semana 1	Semana 2	Semana 3	Semana 4	Semana 5	Semana 6	Semana 7	Semana 8	Semana 9
1. Autenticación básica									
2. Perfil usuario									
3. 2FA									
4. Gestión ingresos/gasto									
5. Dashboard financiero									
6. Suscripciones y planes									
7. Chatbot financiero									
8. Historial y sesiones									
9. Recuperar contraseña									
10. UI y mejoras visuales									

Legenda:  Auth  Dashboard  Chatbot

Cada  indica aproximadamente medio día a un día de trabajo.

7. Mejoras Futuras

Integración de Inteligencia Artificial avanzada

Incorporar modelos más sofisticados que permitan ofrecer recomendaciones personalizadas basadas en el comportamiento financiero del usuario, así como análisis predictivos que anticipen necesidades o riesgos económicos.

Creación de comunidad y funciones sociales

Desarrollar espacios dentro de la aplicación donde los usuarios puedan compartir consejos, logros y retos financieros, fomentando una comunidad activa de apoyo y aprendizaje colaborativo.

Soporte multi-idioma y expansión geográfica

Adaptar la aplicación para soportar múltiples idiomas y cumplir con las regulaciones financieras de diferentes regiones, con el fin de ampliar el alcance a nivel global.

Integración con más servicios financieros

Añadir funcionalidades para gestionar inversiones, seguros, préstamos y otros productos financieros, ofreciendo así una solución integral para la gestión del patrimonio personal.

Desarrollo de la aplicación móvil

Crear una versión móvil nativa de la app, dado que este tipo de software es cada vez más demandado en dispositivos móviles, gracias a su fácil accesibilidad, notificaciones a nivel sistema y mayor interacción con el usuario.

Integración real con OpenBank

La api de openbank devolverá datos reales a tiempo real cada vez que haya un movimiento y se insertará en la base de datos por lo que los datos serán realistas y se podrá implementar nuevas funcionalidades

Bibliografía

Fuentes utilizadas para el desarrollo, investigación y documentación del proyecto.

1. Gráficos utilizados

<https://www.lucidchart.com>

2. Factor de doble autenticación y librería Resend

<https://www.youtube.com/watch?v=xUTeII982w>

<https://www.youtube.com/watch?v=yxAxvFhf8ro>

<https://resend.com/docs/send-with-laravel>

3. Laravel Breeze para login y Forgot password

<https://laravel.com/docs/10.x/starter-kits#laravel-breeze>

<https://www.youtube.com/watch?v=3hRkE64YIgg>

4. Laravel + inertia.js

<https://www.youtube.com/watch?v=yxAxvFhf8ro>

<https://inertiajs.com/>

<https://www.youtube.com/watch?v=qBxo6hW83jU&list=PL38wFHH4qYZVOnXxcS0NMGGmUsZky6JNG>

https://www.youtube.com/watch?v=mziz-YOo5Nw&list=PLWiQT7FWaG1i-MbUTm1_x2LgIuEM_VRpv

5. Oauth2 con Google

<https://www.youtube.com/watch?v=FLsSEV5uID4>