

Pontificia Universidad Javeriana Cali

Facultad de Ingeniería y Ciencias
Ingeniería de Sistemas y Computación

Trabajo de Grado

**FinVise: Desarrollo de una Plataforma Web de Educación
Financiera
con Inteligencia Artificial Adaptativa**



Nicolás Valencia Payán

Código: 8968552

Director: MSc. Luis Gonzalo Noreña

Santiago de Cali, 2026

FICHA RESUMEN DEL TRABAJO DE GRADO

ÁREA DE TRABAJO: Ingeniería de Software, Inteligencia Artificial Aplicada, Educación Financiera

TIPO DE PROYECTO: Desarrollo de Software / Innovación Tecnológica

ESTUDIANTE: Nicolás Valencia Payán

CORREO ELECTRÓNICO: nvalenciapayan@javerianacali.edu.co

DIRECTOR: Luis Gonzalo Noreña Agudelo

PALABRAS CLAVE: Educación financiera, Inteligencia artificial, Sesgos cognitivos, Algoritmo bandit, Personalización adaptativa, Plataforma web, Finanzas conductuales, Colombia

DURACIÓN ESTIMADA: 6 meses

RESUMEN

El presente trabajo de grado describe el diseño y desarrollo de FinVise, una plataforma web educativa que emplea inteligencia artificial adaptativa para personalizar el aprendizaje financiero de sus usuarios. A diferencia de los robo-advisors tradicionales, cuya finalidad es ejecutar operaciones financieras reales, FinVise opera exclusivamente como herramienta pedagógica, enmarcando cada recomendación como un ejemplo contextualizado al perfil de vida del usuario y no como asesoramiento financiero regulado. El sistema integra cinco fuentes de datos de mercado en tiempo real, un motor de recomendaciones basado en el modelo de lenguaje Claude Sonnet de Anthropic, un algoritmo de aprendizaje Multi-Armed Bandit (UCB1) para personalización adaptativa, detección automática de cinco sesgos cognitivos respaldados por la literatura de finanzas conductuales (Kahneman y Tversky), y un diseño de encuesta pre/post que permite medir el impacto educativo cuantitativamente. La plataforma fue construida con NestJS en el backend y React con Vite en el frontend, con Supabase (PostgreSQL) como capa de persistencia. Este documento presenta la identificación de requisitos funcionales y no funcionales del sistema, el diseño de su arquitectura, la implementación del prototipo y los resultados de las pruebas de validación.

Palabras clave: Educación financiera, inteligencia artificial, sesgos cognitivos, algoritmo bandit UCB1, personalización adaptativa, finanzas conductuales, plataforma web.

ABSTRACT

This thesis presents the design and development of FinVise, an educational web platform that uses adaptive artificial intelligence to personalize financial literacy for its users. Unlike traditional robo-advisors focused on executing real financial operations, FinVise operates exclusively as a pedagogical tool, framing every recommendation as an example contextualized to the user's life profile rather than regulated financial advice. The system integrates five real-time market data sources, a recommendation engine powered by Anthropic's Claude Sonnet language model, a Multi-Armed Bandit (UCB1) learning algorithm for adaptive personalization, automatic detection of five cognitive biases supported by behavioral finance literature (Kahneman and Tversky), and a pre/post survey design that quantitatively measures educational impact. The platform was built with NestJS on the backend and Angular on the frontend, with Supabase (PostgreSQL) as the persistence layer. This document covers the identification of functional and non-functional requirements, system architecture design, prototype implementation, and validation test results.

Keywords: Financial education, artificial intelligence, cognitive biases, UCB1 bandit algorithm, adaptive personalization, behavioral finance, web platform.

TABLA DE CONTENIDO

INTRODUCCIÓN.....	10
1. DEFINICIÓN DEL PROBLEMA.....	11
1.1 Planteamiento del Problema	11
1.2 Formulación del Problema.....	12
1.3 Sistematización	12
2. OBJETIVOS DEL PROYECTO	13
2.1 Objetivo General	13
2.2 Objetivos Específicos.....	13
2.2.1 Objetivo Específico 1	13
2.2.2 Objetivo Específico 2	13
2.2.3 Objetivo Específico 3	13
2.2.4 Objetivo Específico 4	13
2.2.5 Objetivo Específico 5	13
2.3 Resultados Obtenidos / Esperados	14
3. ALCANCE	15
4. JUSTIFICACIÓN.....	17
5. MARCO DE REFERENCIA.....	18
5.1 Marco Teórico.....	18
5.1.1 Plataformas de asesoría financiera impulsadas por IA.....	18
5.1.2 Finanzas Conductuales y Sesgos Cognitivos	19
5.1.3 Aprendizaje por Refuerzo y Algoritmos Bandit	21
5.1.4 Modelos de Lenguaje Generativos en Educación	22
5.1.5 Personalización.....	23
5.2 Trabajos Relacionados	25
6. OBTENCIÓN Y ANÁLISIS DE REQUISITOS	27
6.1 Estrategias de Elicitación de Requisitos	27
6.2 Trabajo de Campo y Síntesis de Hallazgos.....	30
6.3 Formulación de Requisitos Funcionales	32
6.4 Formulación de Requisitos No Funcionales	35
6.5 Características Clave del Sistema	36
6.6 Onboarding como punto de partida de la personalización.....	37
6.6.1 ¿Qué construye el onboarding?	38

6.6.2 El primer nivel: ¿Qué activos ve el usuario?.....	38
6.6.3 El segundo nivel: ¿Cómo se explica cada activo?.....	39
7. DISEÑO DE ARQUITECTURA Y SOFTWARE.....	40
7.1 Estructura General del Sistema.....	40
7.2 Arquitectura del Software.....	44
7.2.1 Stack Tecnológico.....	44
7.2.2 Módulos del Backend (NestJS).....	44
7.2.3 Justificación de elección de tecnologías usadas.....	45
7.2.4 Arquitectura General del sistema.....	50
7.3 Modelo de Datos.....	51
7.4 Diseño de Interfaz Gráfica.....	52
8. DESARROLLO DEL PROTOTIPO.....	60
8.1 Configuración del Entorno de Desarrollo.....	60
8.2 Implementación del Backend.....	61
8.3 Implementación del Frontend.....	62
8.4 Integración de Fuentes de Datos Externas.....	63
8.5 Implementación del Motor de IA y Algoritmo UCB1.....	63
8.5.1 Flujo del Motor de Recomendaciones.....	63
8.5.2 Perfil Conservador.....	64
8.5.3 Perfil Balanceado.....	64
8.5.4 Perfil de Crecimiento.....	65
8.5.5 Perfil Agresivo.....	65
8.5.6 Implementación del Algoritmo UCB1.....	65
8.5.7 Sistema de Detección de Sesgos Cognitivos.....	67
9. EVALUACIÓN DEL SISTEMA.....	69
9.1 Plan de Pruebas.....	69
9.2 Pruebas Funcionales.....	69
9.2.1 Pruebas de Caja Negra.....	69
9.2.2 Pruebas de Caja Blanca.....	80
9.3 Pruebas de Rendimiento.....	87
9.4 Pruebas de Usabilidad.....	89
9.5 Medición del Impacto Educativo.....	95
10. CONSIDERACIONES ÉTICAS.....	98

11. CONCLUSIONES Y TRABAJOS FUTUROS	99
11.1 Conclusiones	99
11.2 Trabajos futuros	100
12. REFERENCIAS BIBLIOGRÁFICAS	102
13. ANEXOS	106

TABLA DE FIGURAS

Figura 1: Diagrama de Arquitectura del Frontend. Elaboración propia	40
Figura 2: Diagrama de Arquitectura del Backend. Elaboración propia	42
Figura 3: Diagrama de Arquitectura general del sistema. Elaboración propia.....	50
Figura 4: Esquema de la Base de Datos. Elaboración propia	51
Figura 5: Dashboard principal. Elaboración propia.....	52
Figura 6: Portafolio educativo (Recomendaciones) Elaboración propia.....	53
Figura 7: Simulador de portafolio. Elaboración propia	54
Figura 8: Asistente IA Conversacional. Elaboración propia	55
Figura 9: Página de sesgos. Elaboración propia	56
Figura 10: Trayectoria de aprendizaje (Perfil UCB1). Elaboración propia.....	57
Figura 11: Trayectoria de aprendizaje (Rendimiento histórico, sesgos y gráfico de línea de evolución de intereses por sector). Elaboración propia.....	57
Figura 12: Encuesta Pre/Post. Elaboración propia	59
Figura 13: n8n workflow (Fetch Prices) Elaboración propia	61
Figura 14: Sesgo Activo Elaboración propia.....	84
Figura 15: Test tiempo de respuesta del endpoint. Elaboración propia.....	87
Figura 16: Timeouts internos del endpoint de recomendaciones. Elaboración propia.....	87
Figura 17: Test cache de precios. Elaboración propia.....	88
Figura 18: Test actualización tira de precios. Elaboración propia	88
Figura 19: Prueba de usabilidad 1 (conocimiento financiero de los usuarios). Elaboración propia.....	90
Figura 20: Prueba de usabilidad 2 (facilidad de uso de la plataforma). Elaboración propia	90
Figura 21: Prueba de usabilidad 3 (utilidad de la plataforma para mejorar educación financiera). Elaboración propia	91
Figura 22: Prueba de usabilidad 4 (dificultades en el onboarding). Elaboración propia....	91
Figura 23: Prueba de usabilidad 5 (mejoras para entender los activos recomendados). Elaboración propia.....	92
Figura 24: Prueba de usabilidad 6 (utilidad del portafolio simulado). Elaboración propia.	93
Figura 25: Prueba de usabilidad 7 (calificación lenguaje usado por asistente IA). Elaboración propia.....	93
Figura 26: Prueba de usabilidad 8 (relevancia de la alerta del sesgo). Elaboración propia	94
Figura 27: Prueba de usabilidad 9 (mejoras para la alerta del sesgo). Elaboración propia	94
Figura 28: Medición de impacto educativo en plataforma web. Elaboración propia.....	106

ÍNDICE DE TABLAS

Tabla 1: Comparación de frameworks de Frontend	Elaboración propia.....	45
Tabla 2: Comparación de frameworks del Backend.	Elaboración propia	46
Tabla 3: Comparación de modelos de IA.	Elaboración propia.....	47
Tabla 4: Comparación de fuentes de datos del mercado.	Elaboración propia.....	49
Tabla 5: Distribución de perfiles de riesgo	Elaboración propia	64
Tabla 6: Risk score: perfil mínimo	Elaboración propia	85
Tabla 7: Risk score: perfil máximo.	Elaboración propia.....	86
Tabla 8: Ganancia de conocimiento.	Elaboración propia.....	96
Tabla 9: Incremento de confianza.	Elaboración propia.....	97
Tabla 10: Incremento de reconocimiento de sesgos.	Elaboración propia	97

INTRODUCCIÓN

Manejar bien las finanzas personales es difícil para la mayoría de colombianos, y eso no ha cambiado mucho a pesar de que hoy hay más información disponible que nunca. Los precios cambian constantemente, los productos financieros son cada vez más complejos y casi nadie tiene acceso a una orientación personalizada. En ese panorama, la educación financiera tradicional se queda corta, y aprender sobre el tema sigue siendo algo fragmentado y, para muchos, simplemente inalcanzable.

Ahí es donde la inteligencia artificial empieza a tener sentido como herramienta. Su capacidad para dar respuestas contextualizadas y adaptarse al nivel de cada persona abre una oportunidad real para que más gente pueda acceder a conocimientos financieros sólidos y útiles en su día a día.

FinVise nace de esa idea. Es una plataforma web educativa que usa inteligencia artificial para hacer que el aprendizaje financiero sea más adaptativo, más dinámico y, sobre todo, más medible. Incluye simulaciones interactivas, pero no realiza transacciones bancarias ni da recomendaciones bajo ningún marco regulatorio. Su propósito es estrictamente formativo: ayudar al usuario a entender principios económicos, reconocer sus propios sesgos cognitivos y pensar con más claridad cuando se enfrenta a decisiones financieras.

La plataforma no se queda en explicar qué es un ETF o para qué sirve diversificar. También trabaja la dimensión más humana de las decisiones económicas, esa parte en la que entran en juego los miedos, los atajos o bloqueos mentales y los errores de percepción. A través de simulaciones con datos de mercado en tiempo real, el usuario puede explorar distintos escenarios sin arriesgar dinero real. Por debajo, el sistema combina varias fuentes de datos financieros internacionales, el modelo de lenguaje de Anthropic para generar respuestas adaptadas al nivel del usuario, y un algoritmo UCB1 que va afinando las recomendaciones con cada interacción. A todo eso se suma un módulo que detecta patrones de comportamiento asociados a sesgos como la subestimación del riesgo u otras distorsiones comunes en el razonamiento financiero.

Este trabajo documenta el diseño, desarrollo y evaluación de FinVise. Se analizan las decisiones técnicas detrás de cada componente, se exploran los fundamentos teóricos que justifican su enfoque y se evalúa si la plataforma logra lo que se propone: mejorar la comprensión financiera de sus usuarios de forma medible. Los resultados sugieren que combinar IA conversacional, simulación con datos reales y detección de sesgos es una alternativa viable frente a los modelos de educación financiera convencionales, aunque con limitaciones que el propio trabajo discute abiertamente.

1. DEFINICIÓN DEL PROBLEMA

1.1 Planteamiento del Problema

Que haya más información financiera disponible que antes no significa que la gente tome mejores decisiones. El problema de fondo no es la falta de datos, sino saber qué hacer con ellos. Invertir requiere entender de finanzas, evaluar riesgos y pensar a largo plazo, y esas son habilidades que la mayoría de las personas, especialmente quienes están empezando, simplemente no tienen.

En Colombia esa brecha es especialmente visible. Según la Encuesta de Medición de Capacidades Financieras realizada por CAF y la Superintendencia Financiera (2019), solo el 14% de los adultos colombianos declara hacer algún tipo de inversión más allá de pensiones o planes de jubilación. Las razones se repiten; no conocen los instrumentos disponibles, los perciben como algo arriesgado, nunca tuvieron educación financiera formal y no encuentran herramientas que les ayuden a aprender sin sentirse perdidos desde el primer paso.

De manera general, los robo-advisors, que son los sistemas automatizados de asesoría financiera que existen hoy, tampoco resuelven el problema desde una perspectiva educativa. La mayoría clasifica a los usuarios en perfiles de riesgo genéricos y les entrega recomendaciones estándar, sin considerar que dos personas con metas completamente distintas pueden terminar recibiendo exactamente la misma sugerencia. Además, funcionan como cajas negras: el usuario recibe una recomendación pero no entiende por qué, lo que hace imposible que aprenda algo en el proceso.

A eso se suma que casi ninguna plataforma presta atención a los sesgos cognitivos. La economía conductual, impulsada por los trabajos de Kahneman y Tversky (1979) y (1974), lleva décadas demostrando que los inversores no toman decisiones de forma racional, sino que se dejan llevar por atajos mentales que los llevan a errores sistemáticos. El sesgo de recencia los hace sobrevalorar lo que acaba de pasar en el mercado, la aversión a la pérdida los paraliza ante cualquier caída, y el exceso de confianza los lleva a asumir riesgos que no han calculado bien. Sin embargo, las plataformas actuales raramente detectan estos patrones ni ofrecen algo concreto para trabajarlos.

Un ejemplo concreto: un universitario que quiere empezar a invertir se enfrenta a plataformas llenas de gráficos, métricas y terminología técnica que nadie le explica. Sin una guía adaptada a su nivel, lo más probable es que tome una mala decisión o que simplemente abandone. Y mientras no exista un espacio seguro donde pueda experimentar y equivocarse sin perder dinero real, la distancia entre quienes tienen acceso a asesoría financiera de calidad y quienes no la tienen va a seguir creciendo.

1.2 Formulación del Problema

¿Cómo desarrollar una plataforma web educativa que utilice inteligencia artificial adaptativa para personalizar el aprendizaje financiero, detectar sesgos cognitivos y medir el impacto educativo en los usuarios?

1.3 Sistematización

Para resolver esta pregunta, se deben abordar las siguientes subpreguntas:

- ¿Cuáles son los requisitos funcionales y no funcionales que debe cumplir la plataforma para garantizar un sistema eficiente, seguro y educativamente efectivo?
- ¿Cómo diseñar un motor de recomendaciones educativas basado en IA que se personalice al perfil de vida y al nivel de conocimiento financiero de cada usuario?
- ¿Cómo implementar el algoritmo Multi-Armed Bandit UCB1 para que el sistema aprenda de las interacciones del usuario y mejore progresivamente la pertinencia de las recomendaciones?
- ¿Cómo detectar automáticamente sesgos cognitivos a partir del comportamiento del usuario en la plataforma e intervenir con educación específica para cada sesgo?
- ¿Cómo medir el impacto educativo de la plataforma de forma cuantificable mediante un diseño de encuesta pre/post?

2. OBJETIVOS DEL PROYECTO

2.1 Objetivo General

Desarrollar una software web responsive que utilice inteligencia artificial con el fin de proporcionar recomendaciones personalizadas de inversión, ayudando a la toma de decisiones financieras de los usuarios:

2.2 Objetivos Específicos

2.2.1 Objetivo Específico 1: Identificar los requisitos funcionales y no funcionales del software, así como las características clave que el sistema debe tener.

2.2.2 Objetivo Específico 2: Diseñar un sistema basado en IA que pueda analizar datos financieros en tiempo real y aprender de las preferencias del usuario, con el fin de apoyarlo en la toma de decisiones financieras futuras.

2.2.3 Objetivo Específico 3: Implementar en el software técnicas de aprendizaje automático y aprendizaje por refuerzo para mejorar la precisión y adaptabilidad de las recomendaciones que se le dará al usuario.

2.2.4 Objetivo Específico 4: Desarrollar un software web responsive con una interfaz intuitiva que permita a los usuarios interactuar con el software, ajustar parámetros y evaluar el impacto de distintas estrategias de inversión.

2.2.5 Objetivo Específico 5: Evaluar la funcionalidad del sistema mediante simulaciones y pruebas con datos históricos y en tiempo real

2.3 Resultados Obtenidos / Esperados

Al finalizar el proyecto, se entregará un prototipo funcional de la plataforma FinVise que cumple con los siguientes resultados:

- Un sistema web responsive con autenticación, onboarding de perfil de riesgo y vida, y todas las funcionalidades del sistema operativas.
- Un motor de recomendaciones educativas personalizado que genera portafolios de 5–7 activos con explicaciones contextualizadas al perfil del usuario.
- Un algoritmo UCB1 implementado, con trazabilidad completa del proceso de aprendizaje por activo.
- Detección automática y activa de cinco sesgos cognitivos con intervenciones educativas específicas.
- Un sistema de medición de impacto educativo pre/post con resultados cuantitativos de ganancia de conocimiento, confianza y autoconciencia.
- Pruebas funcionales, de rendimiento y de usabilidad documentadas, con análisis de resultados.
- Un documento técnico completo que integra todos los artefactos del proceso de desarrollo.

3. ALCANCE

El presente proyecto contempla el diseño, desarrollo y evaluación de un prototipo funcional de la plataforma FinVise. El alcance incluye los siguientes módulos y capacidades:

Módulos incluidos:

- **Autenticación y gestión de usuarios:** Registro, inicio de sesión, persistencia de sesión y protección de rutas mediante Supabase Auth.
- **Onboarding y perfil del usuario:** Cuestionario de 11 pasos que genera un risk_score (1–10) y un perfil de vida con 6 dimensiones.
- **Motor de recomendaciones educativas:** Generación de portafolios de 5–7 activos con Claude Sonnet 4.6, datos de mercado en tiempo real y tres escenarios simultáneos (perfil adyacente inferior, perfil actual, perfil adyacente superior), con tabs de labels dinámicos que reflejan el nombre real de cada perfil.
- **Algoritmo UCB1 (Multi-Armed Bandit):** Personalización adaptativa de recomendaciones basada en el historial de interacciones del usuario.
- **Simulador de portafolio:** Cálculo de rendimiento hipotético a 1 día con distintos capitales predefinidos o personalizados.
- **Asistente IA conversacional:** Chat multi-turno con Claude en español, con renderizado de Markdown completo.
- **Detección de sesgos cognitivos:** Identificación automática de cinco sesgos: recencia, familiaridad, aversión a la pérdida, exceso de confianza y efecto disposición.
- **Encuesta pre/post de impacto educativo:** Cuatro secciones: conocimiento conceptual (8 preguntas), confianza (5 Likert), autoconciencia (4 Likert) y valoración general.
- **Trayectoria de aprendizaje e historial:** Registro de sesiones, evolución de intereses por sector, rendimiento histórico de recomendaciones y métricas UCB1.
- **Fuentes de datos de mercado:** TwelveData (42 tickers), CoinGecko (criptomonedas), Alternative.me (Fear & Greed), FRED API (Tesoro 10A), NewsAPI.

Delimitaciones:

- La plataforma no ejecuta operaciones financieras reales ni está conectada a cuentas o carteras de inversión de los usuarios.
- Las recomendaciones generadas por la IA se utilizan únicamente con fines académicos y de aprendizaje. No constituyen asesoramiento financiero regulado.
- No se abordan en esta fase temas relacionados con regulación financiera, cumplimiento normativo (MiFID II, Ley 964 de 2005) ni seguridad avanzada de datos personales más allá de las capacidades nativas de Supabase.
- La evaluación de impacto educativo se realiza con una muestra de usuarios de conveniencia, no con una muestra representativa de la población general colombiana.
- El sistema opera con datos de mercado de Estados Unidos (NYSE, NASDAQ) dado que los datos de la Bolsa de Valores de Colombia presentan limitaciones de acceso en las APIs seleccionadas.
- Los montos del simulador están denominados en Pesos Colombianos (COP) pero son hipotéticos, sin conexión a mercados reales.

4. JUSTIFICACIÓN

FinVise parte de una idea simple pero ambiciosa: que cualquier colombiano, sin importar su nivel de conocimiento o sus ingresos, pueda aprender a relacionarse mejor con el mundo de las inversiones. Para lograrlo, la plataforma usa inteligencia artificial que se adapta a cada persona según su edad, ocupación, nivel educativo, metas y tolerancia al riesgo. Esa personalización no es un detalle simple, es lo que hace que el usuario sienta que la plataforma le habla a él y no a un inversor genérico, y eso, con el tiempo, se traduce en más confianza y más autonomía para tomar decisiones.

Todo empieza con el onboarding. Antes de que el usuario haya hecho cualquier cosa en la plataforma, ya existe un perfil construido a partir de sus respuestas iniciales. Esto es fundamental en sistemas adaptativos, porque sin esa información de partida no hay forma de personalizar nada: las recomendaciones serían iguales para todos y la experiencia perdería su valor pedagógico. En la literatura esto se conoce como el problema del cold start, y el onboarding es precisamente la solución.

Una de las apuestas más importantes de FinVise es su enfoque educativo. A diferencia de los robo-advisors convencionales, que simplemente ejecutan estrategias sin explicar nada, FinVise dedica esfuerzo a que el usuario entienda por qué se le recomienda lo que se le recomienda. El modelo de lenguaje genera explicaciones en español que conectan cada activo con el perfil del usuario, señalan el riesgo asociado y describen qué rol cumple ese activo dentro de una estrategia de diversificación. Cada sesión se convierte, en ese sentido, en una clase práctica.

El proyecto también tiene una dimensión de democratización que vale la pena mencionar. Herramientas sofisticadas de análisis financiero han estado históricamente reservadas para instituciones o para personas con recursos suficientes para pagar asesoría profesional. FinVise busca romper esa lógica ofreciendo, de forma gratuita y con una interfaz accesible, el tipo de apoyo que antes solo estaba disponible para unos pocos.

Desde el punto de vista académico, el proyecto aporta algo concreto al campo de las finanzas conductuales aplicadas a la educación. Detectar automáticamente cinco sesgos cognitivos, como la aversión a la pérdida o el exceso de confianza, y acompañar esa detección con intervenciones educativas específicas, es un enfoque que va bastante más allá de lo que suele entenderse por alfabetización financiera.

Por último, el hecho de medir el impacto con una encuesta en formato pre/post permite generar evidencia real sobre si la plataforma funciona o no. Eso no solo sirve para este proyecto, sino que sienta una base para investigaciones futuras que quieran validar este tipo de enfoques a mayor escala. En conjunto, FinVise tiene el potencial de contribuir de forma genuina a mejorar la cultura financiera en Colombia, poniendo herramientas inteligentes y formativas al alcance de quienes más las necesitan.

5. MARCO DE REFERENCIA

5.1 Marco Teórico

5.1.1 Plataformas de asesoría financiera impulsadas por IA

La inteligencia artificial aplicada a las finanzas personales ha dado lugar a una nueva categoría de herramientas conocidas como robo-advisors, que son plataformas que gestionan inversiones de forma automática sin intervención humana directa. Wealthfront y Betterment son las más representativas de esta tendencia y usan algoritmos de optimización de carteras basados en la Teoría Moderna de Portafolios (Markowitz, 1952) para construir y gestionar portafolios diversificados sin que el usuario tenga que tomar cada decisión manualmente. Betterment gestiona actualmente más de 33 mil millones de dólares en activos mediante inteligencia artificial, mientras que Wealthfront supera los 27 mil millones bajo gestión (HashStudioz, 2025).

La Teoría Moderna de Portafolios parte de una idea que parece simple pero tiene implicaciones profundas: no existe una cartera universalmente óptima, sino una cartera óptima para cada nivel de tolerancia al riesgo. Su aporte central es demostrar matemáticamente que combinar activos con comportamientos distintos, es decir, activos que no suben y bajan al mismo tiempo ni por las mismas razones, reduce el riesgo total del portafolio sin necesariamente sacrificar el retorno esperado. Ese principio es el que los robo-advisors implementan de forma automatizada cuando distribuyen el capital de un usuario entre acciones, bonos y otros instrumentos según su perfil.

Formalmente, dado un portafolio compuesto por n activos, el retorno esperado del portafolio se calcula como:

$$E(R_p) = \sum_i w_i \cdot E(R_i)$$

donde w_i es el peso del activo i dentro del portafolio y $E(R_i)$ su retorno esperado, con la restricción de que $\sum_i w_i = 1$.

La varianza del portafolio, que es la medida de riesgo que Markowitz usa, no es simplemente el promedio ponderado de las varianzas individuales. Depende también de las covarianzas entre activos:

$$\sigma_p^2 = \sum_i \sum_j w_i \cdot w_j \cdot \sigma_{ij}$$

donde σ_{ij} es la covarianza entre los retornos de los activos i y j . Cuando σ_{ij} es baja o negativa, es decir, cuando los activos no se mueven en la misma dirección al mismo tiempo, la varianza total del portafolio resulta menor que la suma de las varianzas individuales ponderadas. Ese es el mecanismo matemático detrás del principio de diversificación.

En FinVise, esta lógica se refleja en la construcción de los perfiles de riesgo (Tabla 5): los pesos asignados a cada clase de activo dentro de cada perfil buscan producir combinaciones

donde la varianza conjunta sea coherente con el nivel de tolerancia al riesgo declarado por el usuario.

Sin embargo, estas plataformas tienen una limitación importante que la propia industria ha empezado a reconocer. El Instituto CFA encontró en 2025 que la falta de explicabilidad era la segunda barrera más citada para la adopción de IA entre profesionales de inversión. En otras palabras, el problema no es que los sistemas no funcionen, sino que el usuario no entiende por qué hacen lo que hacen. Esa brecha entre lo que el algoritmo calcula y lo que la persona comprende es el punto de partida que ha motivado el desarrollo de una nueva rama dentro de la inteligencia artificial.

La Inteligencia Artificial Explicable, conocida por sus siglas en inglés como XAI, surge precisamente como respuesta a ese problema. Su objetivo no es hacer los modelos más simples, sino hacerlos más transparentes. Arrieta et al. (2020) establecen que un sistema de IA se considera explicable si cumple tres condiciones. La primera es la interpretabilidad, que significa que el modelo puede ser comprendido directamente por una persona sin necesitar herramientas adicionales que lo traduzcan. La segunda es la explicabilidad propiamente dicha, que se refiere a la capacidad del sistema de justificar sus decisiones en lenguaje que cualquier usuario pueda entender, incluso después de que la decisión ya fue tomada. La tercera es la trazabilidad, que implica que el proceso de decisión puede reconstruirse paso a paso y auditarse si es necesario. En aplicaciones de alto impacto como la evaluación de crédito o la gestión de portafolios, estas tres condiciones no son opcionales: reguladores y usuarios las exigen al mismo tiempo (Kalasampath et al., 2025).

En la práctica, los métodos de XAI se dividen en dos grandes tipos según el alcance de la explicación que generan. Las explicaciones globales describen cómo se comporta el modelo en términos generales, respondiendo preguntas como qué variables son las más influyentes en sus decisiones de forma habitual. Las explicaciones locales, en cambio, justifican una decisión específica para un caso concreto, respondiendo preguntas como por qué el modelo recomendó este activo a este usuario en este momento. Técnicas como LIME, que construye modelos simples y comprensibles alrededor de cada decisión individual, y SHAP, que calcula la contribución de cada variable a un resultado particular, representan el estado del arte en explicaciones locales y han sido adoptadas en sistemas financieros precisamente porque le permiten al usuario entender la lógica detrás de cada recomendación en su situación particular.

5.1.2 Finanzas Conductuales y Sesgos Cognitivos

Las finanzas conductuales combinan psicología y economía para explicar por qué los inversores toman cierto tipo de decisiones, identificando los factores cognitivos y emocionales que afectan el comportamiento financiero y cómo generan ineficiencias en los mercados (Kahneman y Tversky, 1979; Barberis y Thaler, 2005). Este campo parte de una premisa incómoda para la economía tradicional: las personas no toman decisiones de forma racional y consistente, sino que siguen patrones sistemáticos de error que se repiten de forma predecible.

El trabajo de Kahneman y Tversky, culminado en la Teoría Prospectiva (1979), es la contribución central de este campo y una de las bases más sólidas de toda la economía del comportamiento. La teoría nació de una observación aparentemente sencilla: cuando las personas toman decisiones financieras, no evalúan los resultados en términos de cuánta riqueza total tendrían al final, sino en términos de cuánto ganan o pierden respecto a un punto de referencia, que suele ser su situación actual o el precio al que compraron algo. Esa distinción, que puede parecer menor, tiene consecuencias enormes sobre cómo se comportan los inversores.

Para describir ese comportamiento matemáticamente, los autores propusieron una función de valor, donde las pérdidas tienen más peso psicológico que las ganancias de la misma magnitud.

$$v(x) = \begin{cases} x^\alpha & \text{si } x \geq 0 \\ -\lambda \cdot (-x)^\beta & \text{si } x < 0 \end{cases}$$

donde x representa la ganancia o pérdida relativa al punto de referencia, α y β son parámetros prácticamente alrededor de 0.88 (Tversky y Kahneman, 1992), y λ es el coeficiente de aversión a la pérdida, estimado en aproximadamente 2.25. En términos simples, perder duele psicológicamente más del doble de lo que satisface ganar la misma cantidad.

La teoría también incluye una función de ponderación de probabilidades:

$$w(p) = \frac{p^\gamma}{(p^\gamma + (1-p)^\gamma)^{\frac{1}{\gamma}}}$$

donde $\gamma \approx 0.69$. Esta función hace que las personas sobrestimen las probabilidades bajas y subestimen las altas, lo que explica por qué alguien compra una lotería sabiendo que casi con certeza va a perder, o por qué se resiste a contratar un seguro cuando la probabilidad de pérdida ya es alta.

De todos los sesgos documentados en la literatura, cinco tienen especial relevancia en decisiones de inversión digital. El sesgo de recencia lleva a darle demasiado peso a lo que acaba de pasar, lo que empuja a comprar lo que está subiendo y vender lo que está cayendo, como si la tendencia reciente fuera a durar para siempre. En la práctica, esto amplifica los ciclos del mercado porque cuando todo sube todos compran y cuando todo cae todos venden, justo en el peor momento. El exceso de confianza hace que los inversores creen que saben más de lo que realmente saben, llevándolos a asumir riesgos que su información no justifica. No es que actúen de mala fe, sino que sobreestiman la precisión de sus propios juicios, un error que Barber y Odean (2000) documentaron extensamente al mostrar que los inversores más activos, quienes más confianza tienen en sus decisiones, tienden a obtener peores resultados que los pasivos. El sesgo de familiaridad genera una preferencia por activos conocidos, no porque sean mejores, sino simplemente porque resultan más cómodos.

Huberman (2001) mostró que los inversores tienden a concentrar sus carteras en empresas locales o de sectores que conocen, renunciando a los beneficios de la diversificación sin ser conscientes de ello. El efecto de disposición, documentado por Shefrin y Statman (1985), describe algo que muchos inversores reconocerán, la tendencia a vender demasiado rápido los activos que van bien y a quedarse demasiado tiempo con los que van mal, como si vender con pérdida fuera admitir un fracaso. Es una consecuencia directa de la aversión a la pérdida descrita por la Teoría Prospectiva, donde el dolor de cristalizar una pérdida es tan grande que el inversor prefiere esperar aunque racionalmente ya no tenga sentido. Y el sesgo de anclaje ocurre cuando un número inicial, como el precio al que se compró un activo, se convierte en una referencia tan fuerte que el inversor no logra actualizarla aunque la situación haya cambiado por completo. Ese número original ancla las expectativas y distorsiona la lectura de la información nueva que llega después.

Estos cinco sesgos comparten tres características que los hacen especialmente relevantes para el diseño de sistemas de educación financiera. Tienen sólida evidencia empírica en la literatura de finanzas conductuales, están bien documentados en el contexto específico de plataformas de inversión digital, y pueden inferirse a partir de patrones de comportamiento observables sin necesidad de aplicar instrumentos psicológicos externos al usuario (Gupta et al., 2025).

5.1.3 Aprendizaje por Refuerzo y Algoritmos Bandit

El aprendizaje por refuerzo es una rama del aprendizaje automático en la que un agente aprende a tomar mejores decisiones a través de la experiencia. No parte de un conjunto de datos etiquetados sino que aprende ensayo a error, probando opciones, observando qué resultados producen y ajustando su comportamiento en consecuencia (Sutton y Barto, 2018).

El problema del Multi-Armed Bandit es un caso particular de ese marco y su nombre viene de una analogía con las máquinas tragamonedas. Imagina que estás frente a varias máquinas, cada una con una palanca y una probabilidad de premio diferente, pero sin que nadie te diga cuál paga mejor. La única forma de averiguarlo es jugar, y cada intento tiene un costo. Si dedicas demasiados intentos a explorar máquinas desconocidas, pierdes dinero probando opciones que quizás no valen la pena. Pero si te quedas solo con la que ya conoces, nunca sabes si existe una mejor. Ese es precisamente el dilema que el problema formaliza, cómo distribuir los intentos entre explorar opciones nuevas y explotar las que ya demostraron funcionar, de manera que la recompensa acumulada sea la mayor posible.

En cada ronda el agente elige una opción, recibe una recompensa y acumula información. Con el tiempo va aprendiendo qué opciones tienden a dar mejores resultados, pero siempre bajo incertidumbre porque las recompensas son variables y ninguna opción garantiza el mismo resultado cada vez. El reto no es encontrar la opción perfecta de inmediato, sino aprender de forma eficiente cometiendo el menor número posible de errores en el camino. Esa pérdida acumulada por no conocer de antemano cuál es la mejor opción se llama regret, y es la métrica estándar para comparar qué tan bien resuelve este problema cada algoritmo.

El algoritmo UCB1, propuesto por Auer, Cesa-Bianchi y Fischer (2002), resuelve esa tensión con una idea elegante. Le asigna a cada opción una puntuación que combina dos cosas al mismo tiempo: qué tan bien ha funcionado en el pasado y qué tan poco se sabe sobre ella. Las opciones poco exploradas reciben un bonus que las hace más atractivas, bonus que va disminuyendo a medida que se acumula experiencia con ellas. De esta forma el sistema nunca descarta una opción solo por falta de información, pero tampoco ignora lo que ya aprendió. Los autores demostraron que este mecanismo garantiza que el sistema converge hacia las mejores opciones de forma progresiva y eficiente, con una pérdida acumulada que crece de forma logarítmica en el tiempo. En la práctica eso significa que el costo de aprender se vuelve cada vez menor a medida que el sistema madura, lo que lo hace especialmente adecuado para entornos donde se requiere aprendizaje progresivo con recursos limitados.

En el contexto de los sistemas de recomendación financiera, los brazos del algoritmo corresponden a los distintos instrumentos disponibles para recomendar. La recompensa se define como el nivel de engagement del usuario con cada opción, medido a través de sus reacciones explícitas y sus interacciones con las herramientas de simulación. Con cada sesión el algoritmo aprende qué tipo de contenidos generan mayor comprensión e interés para ese usuario en particular, sin necesitar datos previos para arrancar. Investigaciones recientes han refinado este mecanismo con técnicas de promedio ponderado que permiten al algoritmo adaptarse mejor cuando las preferencias del usuario cambian con el tiempo, lo que lo hace útil no solo para perfiles estáticos sino para personas cuyo conocimiento e intereses evolucionan a lo largo del proceso de aprendizaje (Manome et al., 2025).

5.1.4 Modelos de Lenguaje Generativos en Educación

Los grandes modelos de lenguaje son uno de los avances más significativos en inteligencia artificial de los últimos años. Para entender qué los hace especiales, conviene partir de lo que son. Son redes neuronales artificiales, es decir, sistemas computacionales inspirados vagamente en el funcionamiento del cerebro humano, entrenadas sobre enormes volúmenes de texto en múltiples idiomas y dominios. Ese entrenamiento masivo les permite aprender patrones del lenguaje tan profundos que pueden generar texto coherente, resumir documentos, responder preguntas y razonar sobre problemas complejos sin haber recibido instrucciones explícitas para cada tarea (Zhao et al., 2023).

Lo que los hace cualitativamente distintos de los sistemas de inteligencia artificial anteriores es su capacidad de generalización. Un sistema basado en reglas necesita que alguien programe explícitamente qué hacer en cada situación, lo que lo hace rígido y limitado a los escenarios que sus creadores anticiparon. Un modelo de lenguaje grande, en cambio, aprende patrones tan fundamentales del lenguaje que puede transferirlos a dominios que no fueron parte explícita de su entrenamiento. Con las instrucciones adecuadas, el mismo modelo puede comportarse como un tutor financiero, un asistente de programación o un redactor, sin haber sido diseñado específicamente para ninguno de esos roles.

Una capacidad especialmente relevante en contextos educativos es el aprendizaje en contexto. Este término se refiere a la habilidad de estos modelos de ajustar su

comportamiento a una tarea específica simplemente leyendo las instrucciones que se le dan al inicio de cada conversación, lo que en el campo se llama prompt. No necesitan ser modificados internamente para cambiar cómo responden, basta con decirles cómo hacerlo. Esto contrasta con el fine-tuning, que es el proceso de reentrenar un modelo sobre datos específicos de un dominio para que aprenda a comportarse de cierta manera. El fine-tuning es computacionalmente costoso, requiere grandes volúmenes de datos curados y puede hacerle olvidar al modelo capacidades que ya tenía, un fenómeno conocido como olvido catastrófico porque el aprendizaje nuevo sobrescribe el conocimiento anterior. El prompt engineering, en cambio, consiste en diseñar cuidadosamente las instrucciones que se le dan al modelo en cada conversación para guiar su tono, profundidad y restricciones temáticas, sin modificar nada dentro del modelo. Es la estrategia más accesible para adaptar estos sistemas a dominios especializados porque no requiere infraestructura costosa ni grandes volúmenes de datos propios.

Esa flexibilidad tiene implicaciones directas en educación. Una revisión sistemática de 55 estudios publicados entre 2020 y 2024 encontró que los sistemas basados en modelos de lenguaje ofrecen rutas de aprendizaje adaptadas a cada estudiante, favoreciendo mayor compromiso y retención frente a enfoques tradicionales (Springer Nature, 2025). La razón es bastante intuitiva. Cuando el contenido responde al nivel y al contexto de quien aprende, la experiencia se siente relevante en lugar de genérica, y eso sostiene la motivación de una forma que la instrucción uniforme no logra. Su capacidad para mantener coherencia a lo largo de conversaciones extensas los hace especialmente útiles para el aprendizaje activo, un enfoque pedagógico en el que el estudiante construye conocimiento a través de la interacción y el diálogo, en lugar de recibirlo de forma pasiva como ocurre en la instrucción tradicional donde el profesor habla y el estudiante escucha.

Desde la perspectiva de la Educación 4.0, término que designa el conjunto de enfoques pedagógicos y tecnológicos que responden a las demandas de la Cuarta Revolución Industrial (World Economic Forum, 2020), estos modelos tienen el potencial de cambiar quién tiene acceso a educación de calidad. Hasta hace poco, recibir explicaciones adaptadas al propio nivel y contexto era algo reservado para quienes podían pagar asesoría personalizada. Un modelo de lenguaje bien configurado puede aproximar esa experiencia a escala y de forma accesible, lo que abre posibilidades importantes especialmente en áreas como la educación financiera, donde la brecha entre quienes tienen acceso a orientación profesional y quienes no es especialmente pronunciada (Frontiers in Education, 2024).

5.1.5 Personalización

La personalización en sistemas digitales es, en esencia, la diferencia entre una experiencia que se siente hecha para alguien en particular y una que se siente hecha para cualquiera. En sistemas educativos y financieros, esa diferencia tiene consecuencias reales. El contenido relevante se procesa mejor, se retiene más y genera mayor compromiso que el contenido genérico (Lusardi y Mitchell, 2014).

Un sistema de recomendación es el mecanismo técnico que hace posible esa personalización. En términos simples, es un algoritmo que estima qué tan relevante es cada opción disponible para una persona específica y las ordena en consecuencia. La literatura clasifica estos sistemas en tres tipos fundamentales (Ricci et al., 2011).

El filtrado colaborativo infiere las preferencias de un usuario a partir del comportamiento de usuarios similares, bajo el supuesto de que personas con historiales parecidos tenderán a valorar las mismas cosas. Es el principio detrás de recomendaciones como "otros usuarios como tú también vieron esto", y funciona bien cuando existe una base amplia de usuarios con historiales ricos. El filtrado basado en contenido, por su parte, recomienda opciones con características similares a las que el usuario ya valoró positivamente, sin necesitar compararse con nadie más. Analiza los atributos de lo que le gustó y busca más de lo mismo. Los sistemas híbridos combinan ambos enfoques para aprovechar sus fortalezas y compensar sus limitaciones individuales. Y una cuarta categoría, los sistemas basados en aprendizaje por refuerzo, prescinden del historial previo y aprenden directamente de las interacciones en tiempo real, tratando cada sesión como una nueva oportunidad de entender mejor al usuario y ajustar las recomendaciones en consecuencia.

Una limitación clásica de todos estos sistemas es el problema del cold start, que podría traducirse como el problema del arranque en frío. Ocurre cuando un usuario nuevo entra al sistema sin ningún historial de interacciones. Sin saber nada sobre esa persona, el sistema no tiene base para recomendar nada con precisión, y cualquier sugerencia sería tan arbitraria como elegir al azar. La solución más documentada es construir un perfil explícito desde el inicio, lo que Burke (2002) llama knowledge-based recommendation o recomendación basada en conocimiento declarado. En lugar de esperar a que el comportamiento del usuario revele sus características con el tiempo, el sistema se las pregunta directamente a través de un cuestionario o formulario de bienvenida. Ese perfil inicial permite generar recomendaciones contextualizadas desde la primera interacción, reduciendo la fase de exploración y acelerando la convergencia hacia las opciones más adecuadas para cada persona.

La personalización también tiene implicaciones pedagógicas profundas. Bloom (1984) demostró en su famoso problema de los dos sigma que la instrucción personalizada uno a uno produce mejoras de dos desviaciones estándar por encima de la instrucción estándar en grupo. Para ponerlo en términos concretos, una desviación estándar equivale aproximadamente a la diferencia entre un estudiante promedio y uno que está en el tercio superior de su clase, así que dos desviaciones estándar representan una mejora enorme. Expresado de otra forma, un estudiante promedio que recibe tutoría personalizada supera al 98% de los estudiantes que reciben instrucción tradicional. Los sistemas adaptativos no pueden replicar exactamente ese efecto, pero intentan aproximarse ajustando el contenido al perfil de quien aprende. Lusardi y Mitchell (2014) refuerzan esa idea desde el campo de la educación financiera específicamente. Las intervenciones genéricas han dado resultados decepcionantes en múltiples estudios, y la razón casi siempre es la misma. No se adaptan al nivel de conocimiento, los objetivos ni la situación real de cada persona.

En últimas, adaptar el material a quien aprende no es un refinamiento opcional, es una condición para que el aprendizaje ocurra de verdad. Esa es la premisa sobre la que se construyen los sistemas de recomendación educativa modernos. No se trata solo de mostrar contenido relevante, sino de ajustar el vocabulario, el tono, la profundidad de las explicaciones y el tipo de retroalimentación según lo que se sabe de cada usuario, y de seguir refinando ese conocimiento con cada nueva interacción.

5.2 Trabajos Relacionados

El uso de inteligencia artificial en el sector financiero ha dado lugar a múltiples plataformas y estudios que buscan optimizar la toma de decisiones de inversión. A continuación, se presentan los trabajos que sirvieron como referencia conceptual y técnica:

- Wang et al. (2023) describen EduChat, un sistema conversacional construido sobre un modelo de lenguaje de gran escala y ajustado con datos educativos para funciones como evaluación de ensayos, enseñanza socrática y soporte emocional. Lo que lo distingue de un asistente genérico es un módulo de perfil de estudiante que persiste entre sesiones y orienta el tono y la profundidad de las respuestas. FinVise comparte ese principio básico, aunque difiere en dominio y en propósito: no busca reproducir la dinámica de un tutor humano sino contextualizar información financiera de mercado en tiempo real. La comparación es útil porque EduChat demuestra que los modelos de lenguaje pueden operar con lógica pedagógica explícita sin requerir un fine-tuning masivo, siempre que el sistema de prompts esté bien diseñado.
- Gupta et al. (2025) publican en Springer una revisión de 30 estudios sobre el uso de inteligencia artificial para identificar y mitigar sesgos cognitivos en decisiones financieras. Sus hallazgos confirman que los sesgos más detectables con modelos de machine learning son la aversión a la pérdida, el exceso de confianza, el anclaje y el comportamiento de manada, precisamente los que concentra la literatura de Kahneman y Tversky. El artículo también señala una tensión que FinVise enfrenta de frente: los sistemas que solo detectan el sesgo sin retroalimentarlo al usuario en lenguaje comprensible no logran ningún cambio de comportamiento. La detección automatizada tiene valor pedagógico únicamente cuando va acompañada de una explicación accesible, lo que justifica la decisión de integrar el módulo de sesgos con el asistente conversacional en lugar de presentarlo como una métrica aislada.
- Alsabab et al. (2021) proponen en Management Science un modelo de robo-asesoría que incorpora interacción activa con el cliente para refinar las estimaciones de tolerancia al riesgo a lo largo del tiempo, en lugar de fijarlas de una vez en el onboarding inicial. Sus experimentos muestran que la precisión del perfil de riesgo mejora significativamente cuando el sistema puede actualizar sus creencias con retroalimentación implícita derivada del comportamiento del usuario. Hay una similitud estructural clara con FinVise: el algoritmo UCB1 cumple una función análoga al actualizar los pesos de los activos según el historial de interacciones. La diferencia es de propósito: Alsabab et al. optimizan para maximizar el retorno de cartera, mientras que FinVise optimiza para maximizar el aprendizaje del usuario, lo

que cambia de forma importante cómo se interpreta el feedback y qué señales se consideran válidas.

- Tavares y Salgado (2025) publican en Neurocomputing un trabajo que aborda uno de los problemas más concretos del cold start en sistemas de recomendación financiera: cómo generar recomendaciones útiles para usuarios sin ningún historial de inversión. La solución combina aprendizaje en línea con componentes de explicabilidad que le permiten al usuario entender por qué el sistema sugiere determinados activos. Que el trabajo priorice la explicabilidad por encima de la precisión de la cartera refleja una tendencia reciente en la literatura: los usuarios novatos confían más en sistemas que justifican sus sugerencias que en los que simplemente las presentan. FinVise parte de esa misma premisa, aunque la extiende al incorporar datos de mercado en tiempo real como insumo de las explicaciones, lo que le permite ofrecer contexto dinámico sobre por qué un activo se comporta de determinada manera en el momento exacto en que el usuario lo consulta.

6. OBTENCIÓN Y ANÁLISIS DE REQUISITOS

6.1 Estrategias de Elicitación de Requisitos

Para identificar los requisitos del sistema se recurrió a cuatro estrategias complementarias. Cada una aportó un tipo de evidencia distinta y, en conjunto, permitieron tomar decisiones de diseño con respaldo concreto en lugar de suposiciones.

- **Análisis comparativo de plataformas existentes:** Se estudió un conjunto de plataformas de asesoría financiera automatizada, específicamente Wealthfront, Betterment, Robinhood y M1 Finance, junto con herramientas de educación financiera como Khan Academy Finance e Investopedia Simulator, con el objetivo de entender qué hacen bien, qué hacen mal y qué no hacen del todo. Este análisis fue el punto de partida para identificar las brechas que una plataforma de educación financiera adaptativa debería cubrir.

Wealthfront y Betterment son los referentes del modelo robo-advisor. El primero se destaca por su automatización fiscal y resulta especialmente útil para inversores con cuentas impositivas, mientras que Betterment combina gestión automatizada con acceso opcional a asesores humanos (Finance Monthly, 2025). Sin embargo, los dos comparten un problema que desde el punto de vista pedagógico resulta crítico: los usuarios no mejoran su comprensión financiera con el tiempo. Toman mejores decisiones gracias a la automatización, pero no necesariamente entienden por qué (Gambit Finance, 2025). La plataforma decide, el usuario obedece, y el conocimiento nunca se transfiere. El problema de fondo es que estos sistemas trasladan la responsabilidad evaluativa al usuario, muchos de los cuales no tienen el conocimiento necesario para cuestionar lo que la plataforma les sugiere (Back et al., citado en ResearchGate, 2025).

El caso de Robinhood es quizás el más ilustrativo de los riesgos que puede traer un diseño centrado en la retención por encima del aprendizaje. La plataforma transformó la inversión en una experiencia visualmente atractiva y ágil usando gamificación, es decir, mecánicas propias de los videojuegos aplicadas a contextos financieros, lo que le permitió llegar a 25,9 millones de cuentas activas en abril de 2025, aunque también acumuló un historial regulatorio considerable (EngineerBabu, 2026). En enero de 2024, la División de Valores de Massachusetts la demandó por el uso de animaciones de confeti, recompensas tipo lotería, listas de acciones populares y notificaciones push, argumentando que estas mecánicas trivializaban la inversión y empujaban a los usuarios a operar con más frecuencia de la que era prudente (Berkeley Technology Law Journal, 2026). La investigación reciente muestra que estas estrategias funcionan especialmente bien en inversores nuevos con poco conocimiento financiero (The Conversation, 2024), lo que las convierte en un riesgo real de profundizar desigualdades cognitivas en lugar de reducirlas.

M1 Finance ocupa un lugar diferente en el espectro. Incorpora contenido educativo con artículos, blogs y videos para nuevos inversores, pero carece de herramientas de análisis o filtrado (Tokenist, 2025) y tiene restricciones en las clases de activos disponibles sin mecanismos de cosecha de pérdidas fiscales (HiCapitalize, 2025), lo que limita su utilidad para enseñar estrategias de optimización de carteras.

En cuanto a las herramientas educativas, Khan Academy Finance es el referente no comercial más sólido. Su enfoque en lecciones en video y ejercicios prácticos sobre ahorro, inversión y crédito lo hace accesible para cualquier persona sin importar su situación económica (Business Money, 2024). El problema es que fue diseñado pensando en el sistema financiero estadounidense (Khan Academy Help Center, 2022), lo que limita su utilidad en contextos latinoamericanos donde los instrumentos, las regulaciones y las convenciones son distintas. A eso se suma que varios usuarios describieron su tono como demasiado escolar, lo que puede alejar a adultos que buscan una experiencia más práctica y menos parecida a un aula.

El Simulador de Investopedia, por su parte, ofrece un entorno de práctica con dinero virtual que resulta útil para principiantes. Sin embargo, sus tipos de órdenes son reducidos y el retraso de 15 minutos en las cotizaciones hace inviable practicar operaciones en tiempo real (TraderHQ, 2026), lo que limita significativamente la utilidad del simulador para quien quiere entender cómo reacciona el mercado en el momento en que ocurren las cosas.

En conjunto, este análisis permitió identificar un patrón claro. Las plataformas más poderosas desde el punto de vista financiero son las menos efectivas desde el punto de vista educativo, y las más educativas carecen de la conexión con datos reales que haría que el aprendizaje se sintiera relevante. Esa brecha es precisamente el espacio que una plataforma de educación financiera adaptativa tiene la oportunidad de ocupar.

• **Definición de casos de uso:** Se construyeron casos de uso para los flujos principales del sistema: registro, onboarding, generación de recomendaciones, interacción con el simulador, uso del chat con inteligencia artificial, consulta de sesgos cognitivos y revisión de trayectoria financiera. Esta etapa fue útil no solo para documentar lo que el sistema debía hacer, sino para identificar qué requisitos eran realmente necesarios y cuál era el comportamiento esperado en cada punto de contacto con el usuario.

El onboarding recibió atención especial, dado que es el momento donde más usuarios abandonan una plataforma. El 73% de los usuarios deja aplicaciones financieras durante el registro por culpa de un diseño deficiente (SaasFactor, 2026), y un onboarding bien diseñado debe entregar valor rápidamente sin abrumar al usuario con información o pasos innecesarios. Dividir el proceso en partes pequeñas y bien explicadas ha demostrado reducir ese abandono. La solución fue estructurar el cuestionario en 11 pasos progresivos, separando el quiz de riesgo del perfil de vida y recopilando la información de forma gradual para que el usuario no sienta que está llenando un formulario interminable antes de poder usar la plataforma.

Para el flujo de recomendaciones y sesgos, los casos de uso mostraron algo que parece obvio pero que muchas plataformas ignoran: presentar datos financieros complejos sin contexto genera más confusión que aprendizaje. La evidencia muestra que presentar esa información de manera accesible y dar retroalimentación inmediata mejora significativamente la experiencia del usuario (Design Studio UIUX, 2026). Eso llevó a incluir una explicación educativa contextualizada y un concepto clave de riesgo por cada

activo, en lugar de limitarse a mostrar números que el usuario promedio no sabe cómo interpretar.

Para el simulador y el chat, los casos de uso evidenciaron la necesidad de un tono pedagógico accesible desde el primer momento. Revelar la información de forma progresiva, es decir, mostrar primero lo esencial y añadir complejidad a medida que el usuario avanza, reduce el abandono y mejora la activación de los usuarios en este tipo de plataformas (G&Co., 2026). El flujo de trayectoria financiera, finalmente, se construyó con una premisa distinta a la de los módulos anteriores: que la experiencia debe sentirse más como un acompañamiento que como una herramienta, lo que orientó tanto el vocabulario usado como la forma en que se visualiza el progreso del usuario a lo largo del tiempo.

• **Consulta e investigación en finanzas conductuales:** Se consultaron los estudios de Kahneman y Tversky, Thaler y trabajos aplicados recientes sobre sesgos en plataformas digitales, con el objetivo de que los mecanismos de detección de sesgos estuvieran respaldados por evidencia científica y no fueran decisiones arbitrarias. Esta investigación fue clave para decidir qué sesgos incluir, cómo detectarlos y cómo presentarlos al usuario de una manera que invite a reflexionar en lugar de generar defensividad.

La Teoría Prospectiva de Kahneman y Tversky (1979) es el punto de partida obligado. La teoría explica cómo las personas toman decisiones bajo incertidumbre y ha sido confirmada por trabajos recientes que documentan comportamientos irracionales en inversores durante periodos de crisis. La aversión a las pérdidas hace que los inversores retengan activos con rendimiento negativo porque el impacto psicológico de perder supera al de ganar. El sesgo de recencia lleva a sobrevalorar eventos recientes del mercado ignorando la tendencia histórica de largo plazo. Y el sesgo de familiaridad hace que las personas concentren su atención en activos conocidos, perdiendo oportunidades de diversificación sin ser conscientes de ello.

El diseño de interfaces financieras digitales, incluyendo banca móvil, plataformas de inversión y robo-advisors, es en sí mismo una forma de arquitectura que moldea el comportamiento financiero a escala. Lo que una plataforma muestra primero, cómo presenta las pérdidas y las ganancias, qué tan fácil hace ciertas acciones y qué tan difícil hace otras, todo eso influye en cómo decide el usuario, muchas veces sin que este lo note. Incluso los asesores financieros humanos son vulnerables a los mismos sesgos que afectan a sus clientes, lo que refuerza el argumento de que la inteligencia artificial puede aportar valor real al identificar esos patrones de forma sistemática y sin el componente emocional que afecta al juicio humano.

Esta investigación permitió construir un mapa de los sesgos con mayor presencia documentada en contextos de inversión digital. La aversión a la pérdida, el sesgo de recencia, el sesgo de familiaridad, el exceso de confianza y el efecto de disposición fueron seleccionados por dos razones concretas: tienen respaldo empírico sólido en la literatura y es posible detectarlos a partir del comportamiento observable del usuario dentro de una plataforma, sin necesidad de aplicar instrumentos psicológicos externos.

• **Diseño emergente durante el desarrollo:** La interfaz y los flujos de interacción no se definieron en una fase de diseño separada antes de programar, sino que fueron tomando forma durante el desarrollo mismo. Este enfoque, conocido como diseño emergente, parte de reconocer que en proyectos de alcance acotado muchas decisiones de interfaz resultan más claras al enfrentar la implementación real que al intentar anticiparlas todas por adelantado (Beck et al., 2001). Las pantallas se construyeron y ajustaron en función de lo que el sistema requería técnicamente y de lo que resultaba más comprensible al usarlo.

Este proceso fue el que reveló varios requisitos que no habrían aparecido en ningún documento de análisis previo. La necesidad de mensajes de estado intermedios mientras el sistema realiza cálculos, por ejemplo, no es algo que se vea en un diagrama sino algo que se siente cuando el usuario espera una respuesta y no sabe si el sistema está procesando o simplemente se quedó quieto. La importancia de tener puntos claros de salida y retorno en flujos de varios pasos tampoco es evidente hasta que se navega el flujo completo y se nota cuándo el usuario se siente atrapado. Y la conveniencia de mostrar el perfil de sesgos de forma narrativa en lugar de puramente numérica fue una conclusión que llegó al ver que los números solos, sin contexto, generaban más confusión que comprensión. Construir directamente también permitió tomar decisiones de vocabulario, jerarquía de información y densidad de contenido de forma contextualizada, ajustando cada pantalla a lo que el flujo completo exigía.

La combinación de estas cuatro estrategias apunta hacia una misma conclusión. Una plataforma de educación financiera adaptativa necesita ser personalizable no como característica opcional sino como condición estructural. El análisis comparativo mostró que las plataformas existentes fallan porque tratan a todos los usuarios igual. Los casos de uso mostraron que flujos como el onboarding o la retroalimentación conductual no funcionan si se diseñan de forma genérica. La investigación en finanzas conductuales demostró que los sesgos varían en tipo e intensidad según cada persona, lo que hace inviable un modelo de intervención único. Y el desarrollo mismo confirmó que ajustar el nivel de complejidad, el vocabulario y la densidad de información según el contexto no es un detalle estético sino una necesidad funcional. Las cuatro fuentes apuntan en la misma dirección, y eso es lo que fundamenta el carácter adaptativo del sistema.

6.2 Trabajo de Campo y Síntesis de Hallazgos

Para orientar el diseño de FinVise, se realizaron consultas informales con usuarios potenciales, en las que se indagó sobre sus hábitos financieros, dificultades con el manejo del dinero y expectativas frente a una herramienta de este tipo. Los resultados de estas consultas sirvieron como base para definir los requerimientos funcionales del sistema.

La frustración más repetida fue no entender el razonamiento detrás de las recomendaciones. Frases como "entiendo que me están recomendando algo, pero no sé por qué" o "la app me dice que invierta en esto, pero no me explica qué significa" dejaron claro que las plataformas actuales comunican resultados pero no construyen comprensión. Eso derivó directamente en uno de los requisitos centrales del sistema: que cada recomendación venga acompañada de

una explicación en lenguaje cotidiano, adaptada a lo que el usuario ya sabe y a su situación particular. No alcanza con mostrar un número; hay que ayudar a entender qué significa ese número para esa persona.

A eso se sumó otro problema igual de concreto: varias personas describieron haberse rendido antes de empezar. "Al principio me confundí con tanto término técnico" y "abandoné el registro porque pedía demasiada información de una sola vez" señalan que la primera experiencia con la plataforma termina expulsando a quienes más la necesitan. Esto orientó la decisión de que el proceso de ingreso a FinVise sea gradual, que pida solo lo necesario en cada momento, y que evite el lenguaje especializado cuando no es imprescindible.

Más allá de la entrada al sistema, los participantes también expresaron algo que las plataformas actuales casi nunca ofrecen: querer entender si habían razonado bien, no solo si les había ido bien. "Quiero saber si estoy tomando buenas decisiones, no solo si gané o perdí dinero" captura esa necesidad con precisión. Y "me gustaría que alguien me explicara mis errores sin juzgarme" la completa: no se trata solo de recibir retroalimentación, sino de recibirla en un ambiente donde equivocarse no se sienta como un fracaso. A partir de ahí se definió que el sistema debe evaluar la calidad del proceso de decisión del usuario y comunicar esa evaluación con un tono neutro, sin condescendencia.

Relacionado con esto, una frase apareció de forma espontánea en varias conversaciones y terminó siendo la que más directamente fundamentó el módulo de simulación: "necesito practicar antes de arriesgar plata real." No fue una decisión que vino de la teoría, sino de escuchar a personas que querían aprender pero no querían pagar el costo de aprender con dinero real. Que esta necesidad emergiera de manera independiente y recurrente confirmó que el simulador no es un componente opcional: es una respuesta a algo que los usuarios ya estaban buscando sin encontrar.

El trabajo de campo también reveló algo sobre cómo ciertas plataformas distorsionan la percepción del riesgo. Algunos participantes describieron haber sentido emoción o urgencia al operar en entornos con elementos de juego, sin conectar esas sensaciones con las consecuencias reales de sus decisiones. Eso confirmó lo que la literatura ya advertía sobre la gamificación mal aplicada, y orientó un requisito específico: que FinVise pueda identificar cuando un usuario está mostrando ese tipo de comportamiento e intervenir con una alerta que invite a la reflexión, sin interrumpir bruscamente la experiencia.

La desconfianza fue otro tema que apareció con más fuerza de lo esperado. "No sé si confiarle mis datos a una app que no conozco" y "prefiero algo que sienta que fue hecho para alguien como yo" no son solo preferencias: son condiciones para que la persona siquiera considere usar la plataforma. En el contexto latinoamericano, donde hay una historia de crisis financieras y poca familiaridad con productos de inversión digital, esa desconfianza tiene raíces concretas. Por eso se definió que FinVise debe ser transparente sobre el manejo de datos desde el primer contacto, y que su lenguaje, ejemplos y referencias deben sentirse cercanos a la realidad del usuario, no importados de otro contexto.

En esa misma línea, varios participantes con poca experiencia financiera compartieron creencias que no son solo percepciones equivocadas, sino barreras reales: "siempre pensé que invertir era para gente con mucha plata" y "nadie me enseñó esto en el colegio ni en la universidad" señalan que el problema no es falta de interés sino falta de acceso a una educación que se sienta relevante y alcanzable. A eso se sumó la insatisfacción con las opciones disponibles: quienes habían intentado educarse por su cuenta chocaron con plataformas que los trataban como niños o con tanto contenido que no sabían por dónde entrar. Ambas experiencias generaban abandono. Eso llevó a definir que FinVise debe partir de cero sin que eso resulte condescendiente, y ajustar su nivel de complejidad según lo que cada usuario ya sabe y cómo va avanzando.

Finalmente, una de las tensiones más interesantes que emergió del trabajo de campo fue la que existe entre querer sentir que hay una persona del otro lado y preferir la privacidad que da hablar con una máquina. "Me daría más confianza saber que hay una persona detrás" y "prefiero que sea una IA porque no me da pena preguntar cosas básicas" son posturas opuestas, y ambas aparecieron en las conversaciones. Lejos de ser una contradicción, esa tensión definió el carácter del asistente conversacional de FinVise: debe tener la calidez de un trato humano, pero también la disponibilidad y la ausencia de juicio que solo un sistema automatizado puede ofrecer.

6.3 Formulación de Requisitos Funcionales

Los requisitos funcionales se derivaron siguiendo la notación estándar IEEE.

RF-01 · Autenticación y gestión de usuarios

El sistema permite registro e inicio de sesión con correo y contraseña mediante Supabase Auth. El acceso a todas las rutas protegidas requiere autenticación activa.

RF-02 · Aceptación de aviso académico

Antes de cualquier interacción, el sistema presenta un aviso obligatorio que declara que FinVise es material educativo de la Pontificia Universidad Javeriana Cali, no asesoramiento financiero. El sistema registra la aceptación con marca de tiempo y bloquea el acceso hasta que sea aceptada.

RF-03 · Perfil de riesgo y perfil de vida (Onboarding)

El sistema ejecuta un cuestionario de 11 pasos compuesto por: (a) Quiz de riesgo (5 preguntas): horizonte de inversión, reacción ante pérdidas, estabilidad de ingresos, conocimiento financiero previo y capacidad de ahorro mensual. Calcula un `risk_score` de 1 a 10, (b) Perfil de vida (6 pasos): rango de edad, tipo de ocupación, nivel educativo, contexto familiar, meta financiera principal y motivación para usar la plataforma. Todos los datos se persisten en Supabase.

RF-04 · Motor de recomendaciones educativas

El sistema genera un portafolio educativo personalizado compuesto entre 5 y 7 activos, incluyendo: ticker, nombre, clase de activo, sector, precio actual, cambio en 1 día (%), porcentaje de asignación sugerido, explicación educativa contextualizada al perfil del

usuario y concepto clave de riesgo. Las recomendaciones se generan usando Claude Sonnet (Anthropic) con datos de mercado en tiempo real. El sistema presenta tres escenarios simultáneos: el perfil adyacente inferior al del usuario (tab izquierdo), el perfil actual y el perfil adyacente superior (tab derecho). Los cuatro perfiles y sus scores representativos son: Conservador (rango 1–3), Equilibrado (rango 4–6), Crecimiento (rango 7–8) y Agresivo (rango 9–10). Si el usuario ya se encuentra en el extremo de la escala, el tab correspondiente muestra su mismo perfil.

RF-05 · Reemplazo de activos

El usuario puede reemplazar cualquier activo de su portafolio. El sistema excluye el activo descartado y todos los activos actualmente visibles, y retorna un activo genuinamente nuevo desde el pool de candidatos.

RF-06 · Simulador de portafolio

El sistema calcula el rendimiento hipotético a 1 día de un capital ingresado por el usuario (valores predefinidos de \$500K, \$1M, \$5M, \$10M COP o monto personalizado). Muestra por activo: capital asignado, ganancia/pérdida en pesos y porcentaje de cambio. El resultado total se presenta en verde (ganancia) o rojo (pérdida).

RF-07 · Asistente IA conversacional

El sistema ofrece un chat multi-turno con Claude que responde cualquier pregunta sobre finanzas, inversión y economía en español. Incluye 14 sugerencias de preguntas rápidas predefinidas. Las respuestas se renderizan con Markdown completo (tablas, encabezados, listas, código). El historial de conversación se envía en cada mensaje para mantener el contexto entre turnos.

RF-08 · Detección y mitigación de sesgos cognitivos

El sistema detecta automáticamente 5 sesgos conductuales basados en los patrones de eventos del usuario: Sesgo de recencia: preferencia por activos con rendimiento reciente positivo, Sesgo de familiaridad: sobreconcentración en un único sector, Aversión a la pérdida: reacciones desproporcionadas ante pérdidas en el simulador, Exceso de confianza: discrepancia entre `risk_score` declarado y calculado por perfil de vida, Efecto disposición: intención de venta detectada en mensajes de chat. Cuando un sesgo supera el umbral de activación (0.45), el sistema muestra en el Dashboard una alerta con nombre, intensidad, evidencia, explicación investigativa y consejo práctico.

RF-09 · Registro de eventos de comportamiento

El sistema registra en Supabase cada interacción del usuario con los siguientes tipos de evento: `view`, `dwell` (con segundos transcurridos), `save`, `rate_up`, `rate_down`, `dismiss`, `chat_message` (con detección de intención de venta/retención), `simulator_used` (con resultado %), `recommendation_requested` y `bias_acknowledged`. Todos incluyen timestamp, ID de usuario, ticker, clase de activo y sector.

RF-10 · Algoritmo de aprendizaje UCB1 (Multi-Armed Bandit)

El sistema aplica el algoritmo UCB1 para ordenar los candidatos de recomendación por clase de activo. Cada ticker es un 'brazo', los eventos de feedback son recompensas. Los activos nunca vistos reciben puntaje infinito y se exploran primero.

RF-11 · Encuesta de impacto educativo (pre/post)

El sistema aplica dos encuestas en momentos distintos del uso. Sección A (8 preguntas): conocimiento conceptual con retroalimentación inmediata, diversificación, ETFs, índice Fear & Greed, riesgo en bonos, horizonte de inversión, volatilidad y sesgo de recencia. Sección B (5 ítems Likert 1–5): confianza en decisiones financieras. Sección C (4 ítems Likert 1–5): autoconciencia de sesgos cognitivos. Sección D: valoración general (0–10), aprendizaje declarado y disposición a recomendar. La encuesta pre se aplica al inicio; la post, después de usar la plataforma. El sistema calcula el delta de cada sección para medir el impacto educativo.

RF-12 · Historial y trayectoria de aprendizaje

El sistema almacena cada sesión de recomendaciones con: fecha, risk_score utilizado, snapshot de mercado (índice Fear & Greed, rendimiento del Tesoro 10A) y la composición completa del portafolio. Muestra el precio al momento de la recomendación versus el precio actual para calcular el rendimiento histórico. Incluye gráfico de evolución de intereses por sector a lo largo del tiempo.

RF-13 · Métricas de efectividad del sistema

La página Trayectoria presenta tres paneles de evidencia cuantitativa. Perfil UCB1: activos con recompensa positiva y negativa, nivel de personalización (%), tasa de feedback positivo y activos explorados. Impacto educativo: ganancia de conocimiento en %, confianza post, autoconciencia post y valoración de la plataforma. Rendimiento histórico: rendimiento promedio de recomendaciones pasadas, tasa de acierto (%) y cantidad de activos evaluados.

RF-14 · Datos de mercado en tiempo real

El sistema obtiene datos actualizados de múltiples fuentes: TwelveData (44 tickers de acciones y ETFs), CoinGecko (BTC, ETH, SOL), Alternative.me (índice Fear & Greed 0–100), FRED API (rendimiento del Tesoro EE.UU. 10 años), NewsAPI (titulares financieros de contexto). La tira de precios en vivo se actualiza cada 60 segundos. Se aplica arquitectura de fallback multicapa: datos externos → caché Supabase → precios estáticos de respaldo.

RF-15 · Alertas de mercado

El sistema muestra notificaciones clasificadas por severidad (info, advertencia, peligro) con badge de conteo de no leídas.

RF-16 · Glosario financiero

El sistema incluye un glosario con búsqueda por término o definición, filtrado por categorías con conteo, y términos expandibles con definición completa, ejemplo de uso y términos relacionados.

RF-17 · Perfil y preferencias del usuario

El usuario puede visualizar su risk_score con clasificación (Conservador/Balanceado/Crecimiento/Agresivo) y editar: presupuesto mensual, horizonte de inversión y sectores excluidos (energía, cripto, tabaco, armas).

6.4 Formulación de Requisitos No Funcionales

RNF-01 · Rendimiento y tiempos de respuesta

El endpoint de recomendaciones apunta a responder en máximo 28 segundos, por debajo del timeout de 35 segundos configurado en el cliente. El sistema aplica timeouts internos escalonados: 10s para datos de precios de mercado, 5s para datos de sentimiento de mercado, 5s para el algoritmo bandit y 20s para el modelo de IA. Los datos de precios se cachean en memoria por 5 minutos para evitar llamadas externas repetidas en solicitudes consecutivas.

Nota: Los valores de timeout son decisiones de diseño propias, fundamentadas en el patrón de deadline propagation (Beyer et al., 2016) y los umbrales de respuesta percibida de Nielsen (1993).

RNF-02 · Disponibilidad y resiliencia

Cada servicio externo tiene manejo de fallback independiente. Si la IA no responde en tiempo, el sistema retorna datos del pool de candidatos calculado cuantitativamente. Si los precios externos fallan, se usan los precios almacenados en la caché de Supabase. Si Supabase no está configurado o su tabla está vacía, el sistema consulta directamente las APIs externas (Yahoo Finance y CoinGecko); si estas también fallan, el sistema continúa operando sin datos de precio.

RNF-03 · Responsividad de la interfaz

La interfaz se adapta a escritorio, tableta y móvil mediante grids con minmax adaptable y layouts flexibles. Las tarjetas de recomendación se reorganizan automáticamente según el ancho disponible del dispositivo.

RNF-04 · Seguridad y aislamiento de datos

Todas las rutas del backend están protegidas con guard de autenticación. Las consultas a Supabase están filtradas por user_id, impidiendo acceso cruzado entre usuarios. El token JWT se adjunta automáticamente a cada solicitud HTTP del frontend.

RNF-05 · Integridad académica

Cada página del sistema incluye el banner académico de la Pontificia Universidad Javeriana Cali. El lenguaje del sistema enmarca toda la información como material educativo, no como asesoramiento financiero regulado. Las recomendaciones incluyen disclaimer explícito en cada respuesta generada por la IA.

RNF-06 · Persistencia de datos

Todos los eventos, encuestas, historial de recomendaciones, preferencias y snapshots de mercado se persisten en Supabase (PostgreSQL). Los datos de sesión de autenticación persisten entre recargas del navegador.

RNF-07 · Idioma y localización

La interfaz, las recomendaciones generadas por IA y las respuestas del asistente están en español colombiano (es-CO). Los formatos de fecha, moneda y números siguen la convención colombiana (COP).

RNF-08 · Mantenibilidad y arquitectura

El backend utiliza arquitectura modular de NestJS (un módulo por dominio de negocio). El frontend separa páginas, componentes reutilizables y capa de servicios. Todo el código está en TypeScript.

6.5 Características Clave del Sistema

A partir de la síntesis de requisitos, se identificaron ocho características clave que definen la propuesta de valor diferencial de FinVise frente a soluciones existentes:

- **Personalización adaptativa:** El sistema aprende de cada interacción mediante el algoritmo UCB1 y ajusta el orden de las recomendaciones futuras. Cuantos más eventos registra el usuario, más personalizado es el resultado.
- **Marco educativo, no especulativo:** Todas las recomendaciones están enmarcadas como ejemplos educativos contextualizados al perfil de vida del usuario. El sistema nunca instruye a 'invertir \$X'; siempre explica conceptos y principios.
- **Detección conductual de sesgos:** El sistema detecta automáticamente 5 sesgos cognitivos respaldados por la literatura de finanzas conductuales (Kahneman y Tversky) e interviene con educación específica para cada uno.
- **Medición de impacto cuantificable:** El diseño pre/post de la encuesta permite medir el cambio real en conocimiento, confianza y autoconciencia del usuario, constituyendo la base de validación empírica del sistema.
- **Datos reales de mercado:** El sistema integra 5 fuentes de datos externas con arquitectura de fallback multicapa. Los precios, el índice de sentimiento y el contexto macroeconómico son datos reales, no simulados.
- **IA generativa contextualizada:** Claude genera explicaciones en español adaptadas al perfil específico de cada usuario (edad, ocupación, meta financiera, nivel educativo), no respuestas genéricas.
- **Trazabilidad completa del comportamiento:** Cada acción del usuario queda registrada con metadatos estructurados, permitiendo análisis conductual, detección de sesgos y generación de evidencia cuantitativa del uso del sistema.
- **Comparación de escenarios de inversión:** El usuario puede evaluar simultáneamente tres perfiles de riesgo, el perfil adyacente inferior, su perfil actual y el perfil adyacente superior, en una interfaz de tabs con labels dinámicos que reflejan el nombre real de cada perfil, desarrollando comprensión del espectro completo de estrategias de inversión.

6.6 Onboarding como punto de partida de la personalización

Para que un sistema pueda adaptarse a quien lo usa, primero necesita saber algo sobre esa persona. En la literatura de sistemas de recomendación, el momento en que alguien llega por primera vez a una plataforma sin ningún historial previo tiene un nombre: cold start. Sin datos de comportamiento, el sistema no tiene forma de distinguir entre usuarios y termina ofreciendo exactamente lo mismo a todos, sin importar quiénes son ni qué necesitan. La solución más documentada para este problema es construir un perfil explícito desde el inicio, lo que Burke (2002) llama knowledge-based recommendation: en lugar de esperar a que el comportamiento del usuario revele sus características con el tiempo, el sistema se las pregunta directamente.

Bajo ese principio funciona el onboarding de FinVise. No es un formulario de registro, es lo que le permite a la plataforma saber, desde el primer momento, con quién está hablando.

Esa personalización importa especialmente en un contexto financiero y educativo. Lusardi y Mitchell (2014) encontraron que las intervenciones genéricas de educación financiera han dado resultados decepcionantes en múltiples estudios, y la razón es casi siempre la misma: no se adaptan al nivel de conocimiento, los objetivos ni la situación real de cada persona. Bloom (1984) ya había llegado a una conclusión similar décadas antes con lo que se conoce como el "problema de los dos sigma", la instrucción personalizada uno a uno produce mejoras de dos desviaciones estándar por encima de la instrucción estándar en grupo. Los sistemas adaptativos no pueden replicar exactamente ese efecto, pero intentan aproximarse ajustando el contenido al perfil de quien aprende. Sin los datos que recoge el onboarding, FinVise no puede hacer nada de eso, y la experiencia de un estudiante universitario de 22 años sería idéntica a la de un empresario de 50.

El onboarding tiene dos partes. La primera construye el perfil de riesgo a través de cinco preguntas de selección múltiple. Horizonte de inversión, reacción ante pérdidas, estabilidad de ingresos, nivel de conocimiento financiero y capacidad de inversión mensual. Las respuestas se combinan en una puntuación del 1 al 10 que clasifica al usuario en uno de cuatro perfiles: conservador, equilibrado, de crecimiento o agresivo.

La segunda parte construye el perfil de vida con seis preguntas sobre grupo de edad, ocupación, nivel educativo, contexto familiar, objetivo financiero principal y motivación para usar la plataforma.

Al terminar el onboarding, el usuario completa una encuesta de línea base con veinte preguntas distribuidas en cuatro secciones: conocimiento conceptual, con ocho preguntas de opción múltiple y retroalimentación inmediata; autoconfianza en decisiones financieras, con cinco ítems Likert; autoconciencia sobre sesgos cognitivos, con cuatro ítems Likert; y percepción general de la experiencia. Esa misma encuesta se aplica al final, lo que permite comparar el estado del usuario antes y después de haber usado la plataforma.

6.6.1 ¿Qué construye el onboarding?

Las respuestas del onboarding construyen lo que en sistemas adaptativos se conoce como un modelo de usuario, que es una representación estructurada de quién es la persona, y que el sistema usa como base para tomar decisiones. Ese modelo tiene dos partes.

La primera se construye a partir de un quiz de riesgo que recoge el horizonte de inversión, la reacción ante pérdidas, la estabilidad de ingresos, el nivel de conocimiento financiero y la capacidad de ahorro mensual. Esta información es necesaria porque la tolerancia al riesgo no es un rasgo uniforme. Varía según los recursos disponibles, el conocimiento de cada persona y su horizonte temporal, y diseñar recomendaciones sin medirla produce orientaciones que no corresponden a la situación real de quien las recibe (Grable, 2000).

La segunda parte recoge seis variables de contexto vital: etapa de vida, situación laboral, nivel educativo, dependientes a cargo, objetivo financiero y motivación para usar la plataforma. Su inclusión responde a que las decisiones financieras están condicionadas por el momento del ciclo de vida en que se encuentra cada persona, y a que las necesidades y prioridades cambian de forma sistemática con el tiempo (Modigliani, 1986).

Juntos, estos dos perfiles permiten que el sistema adapte tanto el contenido como el tono de sus respuestas a quien está del otro lado. Lo que hace que todo esto sea útil en la práctica es que ninguno de esos datos queda guardado como información sin importancia. Cada campo del onboarding está conectado directamente a un comportamiento concreto del sistema. La personalización ocurre en dos niveles al mismo tiempo: uno algorítmico, que define qué activos se le muestran al usuario, y otro de contenido, que define cómo se le explica cada uno de ellos.

6.6.2 El primer nivel: ¿Qué activos ve el usuario?

Antes de generar cualquier explicación, el sistema ya filtró qué activos tiene sentido mostrarle al usuario. Cada instrumento financiero de la plataforma tiene asignado un nivel de riesgo, y el sistema solo considera los activos que corresponden al perfil que el usuario obtuvo en la evaluación inicial. Un perfil conservador no va a encontrarse con criptomonedas de alto riesgo, y un perfil agresivo no va a recibir principalmente bonos de gobierno a corto plazo. Este filtro no cambia cómo se ve la plataforma. Cambia qué instrumentos existen para esa persona.

La lógica detrás de ese filtrado tiene raíces tanto en la teoría financiera como en la regulación. Markowitz (1952) ya lo planteaba en la Teoría Moderna de Portafolio. No existe una cartera universalmente óptima, sino una cartera óptima para cada nivel de tolerancia al riesgo. La ESMA (2022) convierte ese principio en una obligación regulatoria concreta: las recomendaciones deben corresponder al perfil del cliente, y presentar activos inadecuados para ese perfil es un fallo de idoneidad. El filtrado por `risk_score` implementa ese principio de forma computacional, y solo es posible porque el usuario respondió el quiz al inicio.

6.6.3 El segundo nivel: ¿Cómo se explica cada activo?

El segundo nivel de personalización opera sobre el contenido de cada explicación que genera la plataforma. Cuando el usuario explora un activo, el sistema no recupera un texto predefinido sino que lo genera en tiempo real, incorporando todo lo que sabe sobre esa persona desde el onboarding. Los campos recopilados se convierten en instrucciones concretas que modifican el vocabulario, el tono, la profundidad y el enfoque de lo que se le dice.

El nivel educativo define el registro lingüístico de toda la experiencia. Quien declaró nivel bachillerato recibe explicaciones sin terminología técnica sin definir, con analogías cotidianas. Quien declaró posgrado recibe el mismo concepto con una terminología acorde a su formación. Esa adaptación no es cosmética. Van Rooij, Lusardi y Alessie (2011) demostraron que las personas con baja alfabetización financiera no comprenden las explicaciones formuladas con terminología estándar, independientemente de su disposición a aprender. Presentarle a alguien que empieza desde cero una explicación diseñada para un analista financiero no produce aprendizaje, produce abandono.

El objetivo financiero encaja todo el contenido dentro del problema concreto que el usuario está tratando de resolver. Quien declaró que su prioridad es construir un fondo de emergencia recibe explicaciones donde cada activo se presenta en función de su liquidez y seguridad. Quien está ahorrando para comprar vivienda recibe esos mismos activos contextualizados en términos de preservación de capital a corto plazo. Shefrin y Statman (2000) demostraron que los inversores organizan mentalmente su dinero en cuentas separadas por objetivo y evalúan cada instrumento según su utilidad para ese objetivo específico, no para el portafolio en abstracto.

La motivación ajusta el nivel de profundidad y directividad de las respuestas. Quien ya invierte y quiere refinar su criterio no necesita que le expliquen conceptos básicos que probablemente domina, así que el sistema enfoca el contenido en lo que puede ser nuevo para alguien con su experiencia. Por otro lado, quien llega por curiosidad recibe un tono más exploratorio, sin la presión de quien va a tomar decisiones de inmediato. Deci y Ryan (1985), en la Teoría de la Autodeterminación, establecen que la motivación se sostiene cuando el contenido respeta el estado actual y la autonomía de quien aprende.

La ocupación tiene el papel de modificar las advertencias que acompañan cada recomendación ya que quien trabaja de forma independiente con ingresos variables recibe, de forma consistente a lo largo de toda la experiencia, el énfasis en que construir un fondo de emergencia es la prioridad antes de cualquier inversión. Consecuentemente, quien tiene salario fijo no recibe esa misma advertencia porque su situación de riesgo es cualitativamente diferente.

Carroll (1997) demostró que invertir en activos de riesgo solo tiene sentido cuando ya existe una reserva de liquidez, y que los hogares con ingresos inciertos deberían priorizar esa reserva por encima de cualquier retorno esperado del mercado. Omitir esa advertencia para usuarios con ingresos irregulares no es un descuido menor, es presentarles una estrategia que no corresponde a su situación real.

El resultado de combinar el filtrado algorítmico con la personalización de contenido es que dos usuarios con perfiles distintos no tienen versiones superficialmente diferentes de la misma experiencia. Tienen experiencias que son fundamentalmente distintas. Ven activos diferentes, reciben explicaciones en vocabularios diferentes enmarcadas en objetivos diferentes, con advertencias ajustadas a su situación laboral y familiar.

Ricci, Rokach y Shapira (2011) definen la personalización en sistemas de recomendación como la capacidad de producir respuestas que varían según las características individuales de cada usuario, no según las características promedio de la población. Esa capacidad no existe sin el modelo de usuario que construye el onboarding. Si alguien llega a FinVise sin completar ese proceso, el sistema no tiene ninguna información con la que diferenciarlo de cualquier otro participante, y la experiencia colapsa a una presentación genérica que produce poco o ningún impacto medible en el conocimiento financiero real de las personas.

7. DISEÑO DE ARQUITECTURA Y SOFTWARE

7.1 Estructura General del Sistema

- **Capa de presentación (Frontend):** Aplicación web SPA desarrollada con React 19 y Vite. Implementa todos los módulos de interfaz de usuario: dashboard, portafolio educativo, simulador, chat IA, trayectoria y encuestas. Se comunica con el backend exclusivamente mediante API REST autenticada.

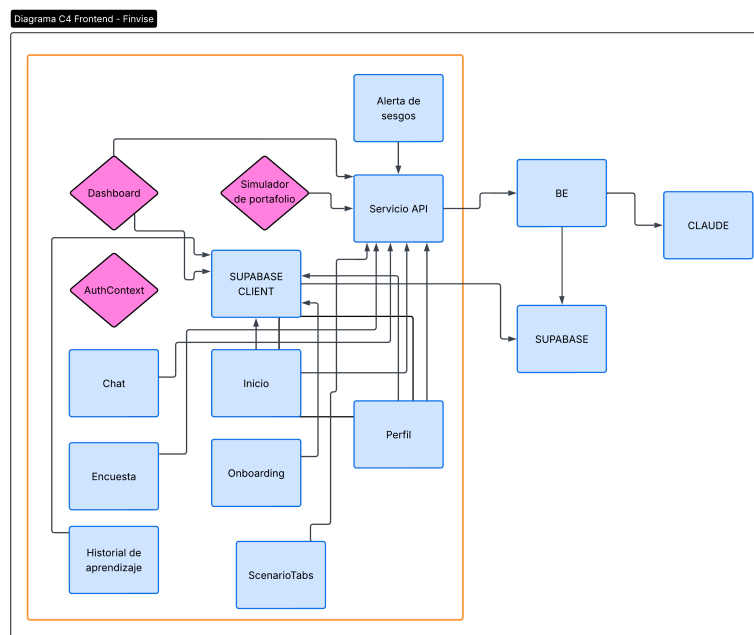


Figura 1: Diagrama de Arquitectura del Frontend. Elaboración propia

El frontend de FinVise es una Single Page Application construida en React con TypeScript y Vite, organizada en ocho páginas principales que el usuario recorre siguiendo un flujo de navegación guiado.

Todo empieza en la página de Inicio, donde el usuario se registra o inicia sesión a través del Supabase Client. Desde ahí acepta el disclaimer de que la plataforma es con fines educativos y pasa al Onboarding, que recoge su perfil de riesgo mediante un cuestionario y guarda el puntaje resultante junto con la información demográfica directamente en Supabase a través del Supabase Client.

El Dashboard es la vista central de la plataforma. En cada sesión consume datos de recomendaciones, alertas de sesgos y precios en vivo. Todas esas peticiones pasan por el Servicio API, un cliente centralizado construido con Axios que se comunica con el BE en localhost:3000/api con un timeout de 35 segundos e inyecta automáticamente el token JWT gestionado por el AuthContext.

Las páginas de Chat, Encuesta e Historial de Aprendizaje se comunican con el backend a través de ese mismo Servicio API. La página de Perfil también lo utiliza para registrar eventos, pero adicionalmente escribe las preferencias del usuario, como los sectores excluidos, directamente en Supabase a través del Supabase Client. El Simulador de Portafolio se apoya en el Servicio API para registrar interacciones, y usa Recharts para visualizar los resultados de forma gráfica. ScenarioTabs, por su parte, consulta el Servicio API de forma independiente para cargar recomendaciones bajo distintos perfiles de riesgo.

La Alerta de Sesgos opera desde el dashboard y envía sus peticiones directamente al Servicio API. El BE, a su vez, se comunica tanto con Claude para la funcionalidad de IA como con Supabase para la persistencia de datos, mientras que el AuthContext gestiona la sesión del usuario directamente con el Supabase Client en el frontend.

- **Capa de lógica de negocio (Backend):** API REST desarrollada en NestJS con arquitectura modular. Cada dominio de negocio tiene su propio módulo: auth, recommendations, simulator, chat, biases, ucb, survey, history, market-data, alerts, glossary. Gestiona las llamadas a servicios externos y aplica las reglas de negocio.

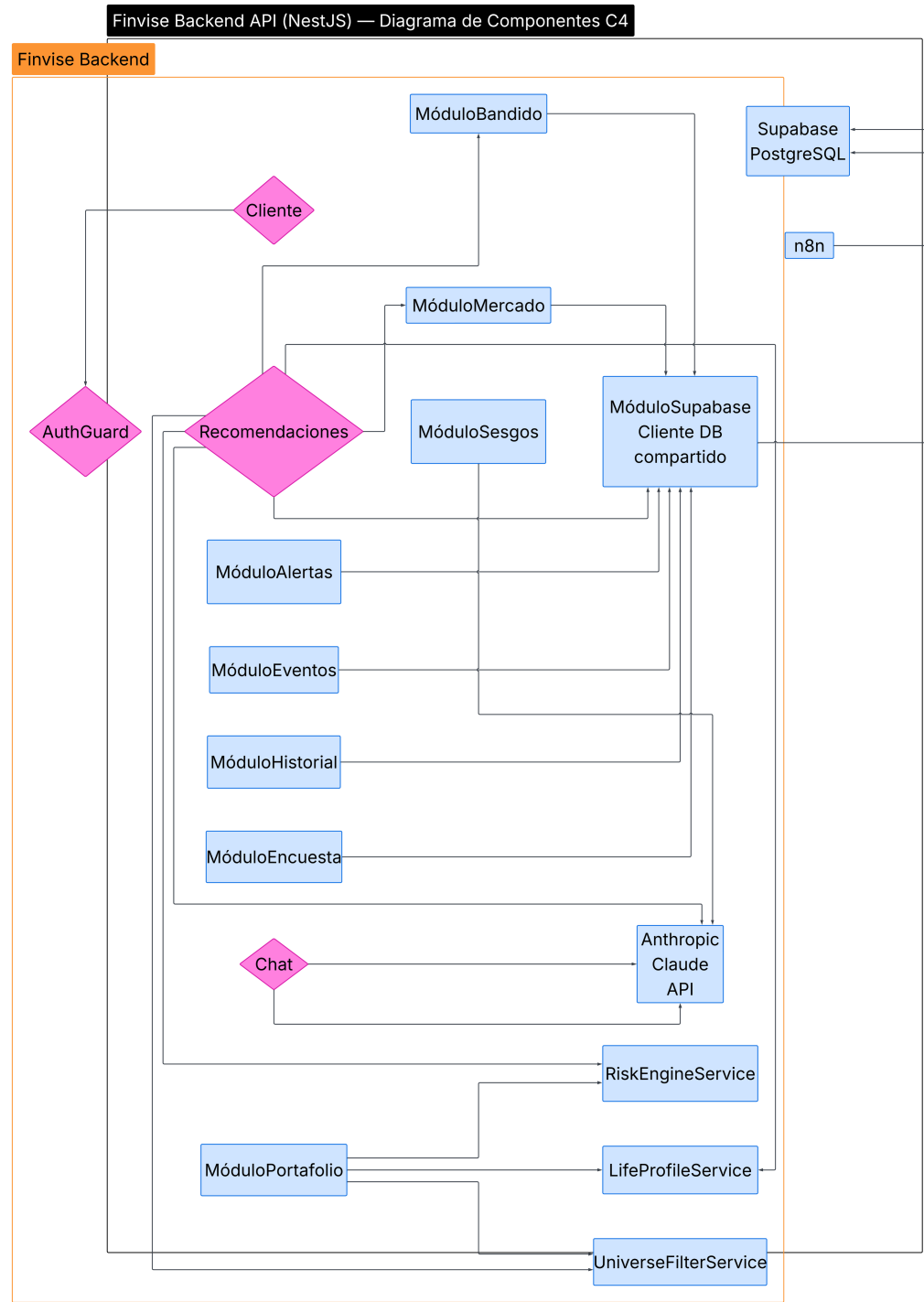


Figura 2: Diagrama de Arquitectura del Backend. Elaboración propia

El backend de FinVise está organizado en módulos especializados, todos protegidos por Supabase Auth, que valida el JWT en cada petición. Si no hay token o la validación falla, el acceso se deniega directamente. Solo los usuarios registrados pueden interactuar con la plataforma.

El módulo central es el de Recomendaciones, que ejecuta el pipeline completo de generación. En paralelo obtiene los precios y el sentimiento de mercado cacheados en Supabase, y consulta los puntajes de personalización del módulo Bandido, que implementa el algoritmo UCB1. Con esos datos construye un conjunto de activos candidatos filtrado por perfil de riesgo, apoyándose en dos funciones auxiliares. Una que adapta el tono y el vocabulario del prompt al perfil de vida del usuario, y otra que incorpora instrucciones correctivas basadas en los sesgos previamente detectados y almacenados en Supabase. Una vez que ese conjunto está listo, se le envía a Claude un prompt que le indica explícitamente trabajar solo con esos activos pre-calculados, sin añadir ni sustituir ninguno. La respuesta incluye las recomendaciones con una justificación educativa adaptada al perfil del usuario.

El módulo de Sesgos analiza el comportamiento registrado del usuario para detectar los patrones cognitivos del inversor. Los resultados se guardan en Supabase y desde ahí se consumen de dos formas: el pipeline de recomendaciones los lee al momento de construir el prompt, y el frontend los muestra directamente al usuario a través de las alertas de sesgos.

- **Capa de persistencia (Database):** Supabase (PostgreSQL) como base de datos relacional. Gestiona autenticación (Supabase Auth), almacenamiento de perfiles, eventos de comportamiento, historial de recomendaciones, encuestas y caché de precios de mercado.

7.2 Arquitectura del Software

7.2.1 Stack Tecnológico

El backend de FinVise está construido con NestJS y TypeScript, organizando la lógica del servidor en módulos independientes mediante una arquitectura basada en decoradores. El frontend usa React sobre Vite, también con TypeScript, aprovechando componentes funcionales con hooks para manejar el estado de la aplicación.

Para la persistencia de datos y la autenticación se usa Supabase, que combina una base de datos PostgreSQL con un sistema de autenticación basado en tokens JWT. La inteligencia artificial generativa viene de la API de Claude de Anthropic, integrada mediante prompts contextualizados que se construyen dinámicamente según el perfil del usuario.

Los datos de mercado se obtienen de cinco fuentes distintas: TwelveData, CoinGecko, Alternative.me, FRED API y NewsAPI, que trabajan en conjunto para garantizar la disponibilidad de la información.

7.2.2 Módulos del Backend (NestJS)

El backend de FinVise está dividido en once módulos, cada uno con una responsabilidad clara dentro del sistema.

- El AuthModule maneja todo lo relacionado con la autenticación: se conecta con Supabase Auth, protege las rutas para que solo usuarios autorizados puedan acceder y sabe en todo momento quién está haciendo cada solicitud.
- El RecommendationsModule es el corazón del sistema. Se encarga de traer los datos del mercado, calcular qué tan relevante es cada activo para el usuario, armar el prompt con todo ese contexto y devolver una respuesta estructurada al cliente.
- El UCBModule implementa el algoritmo UCB1, que es el que decide qué activos mostrarle al usuario. Calcula un puntaje por ticker, aprende del feedback del usuario y balancea entre explorar activos nuevos y seguir mostrando los que ya demostraron ser útiles.
- El BiasesModule detecta cinco sesgos de comportamiento a partir de lo que el usuario ha ido haciendo en la plataforma. No solo los identifica, sino que mide qué tan marcados están y muestra la evidencia concreta que respalda cada detección.
- El SimulatorModule toma el capital disponible, el portafolio actual y los precios del momento para calcular cuánto habría rendido ese portafolio en el último día, como una simulación retrospectiva rápida.
- El ChatModule gestiona las conversaciones con el asistente. Mantiene el historial de cada sesión, construye el contexto necesario antes de cada llamada y se comunica directamente con la API de Claude.
- El SurveyModule administra las encuestas que aparecen al inicio y al final de cada sesión. Decide cuál versión mostrar, guarda las respuestas y calcula cuánto cambió la percepción del usuario entre una y otra.

- El HistoryModule guarda un registro de cada sesión de recomendaciones, incluyendo una foto del estado del mercado en ese momento exacto, para poder consultarlo después.
- El MarketDataModule centraliza todas las llamadas a las APIs financieras externas. Tiene una caché en memoria que dura cinco minutos y un sistema de fallback por capas, así que si un proveedor falla, el sistema sigue funcionando con otro sin que el usuario lo note.
- El AlertsModule genera las alertas de mercado de cada usuario y permite marcarlas como leídas.
- El GlossaryModule sirve el glosario financiero de la app, con búsqueda por texto y filtros por categoría.

7.2.3 Justificación de elección de tecnologías usadas

Tecnología	Criterios Evaluados	Justificación	Estado
React 19 + Vite	Componentes funcionales, TypeScript, ecosistema amplio	Gestión de estado mediante hooks nativos (useState, useContext, useReducer). Compatible con TypeScript y con la estructura de datos del backend.	Seleccionado
Angular 17 + Signals	Reactividad granular, TypeScript nativo	Actualiza solo los elementos afectados por un cambio de datos, pero su modelo de Signals añade complejidad conceptual innecesaria para este proyecto.	Descartado
Vue 3 + JavaScript	Reactividad, curva de aprendizaje	Reactividad comparable a Signals, pero sin TypeScript nativo	Descartado
Next.js 14 + React	Renderizado del lado del servidor, SEO (Search Engine Optimization)	Renderizar páginas en el servidor para mejorar el posicionamiento en buscadores. Añade capa de complejidad que no aporta valor	Descartado

Tabla 1: Comparación de frameworks de Frontend Elaboración propia

La elección del framework de frontend estuvo muy marcada por la naturaleza de la interfaz de FinVise. Una pantalla que puede estar mostrando valores, recomendaciones y métricas de perfil al mismo tiempo, todo actualizándose de forma independiente. React con Vite fue la opción seleccionada por su ecosistema, su modelo de componentes funcionales con hooks nativos y la velocidad de desarrollo que ofrece.

Vue 3 tiene un modelo de reactividad comparable, pero su menor presencia en proyectos empresariales y la falta de TypeScript nativo lo alejaron de la decisión. Next.js quedó fuera desde el principio porque su ventaja central, renderizar páginas en el servidor para mejorar el posicionamiento en buscadores, no tiene ninguna aplicación en FinVise, que es una aplicación privada. Angular con Signals fue evaluado como alternativa. Su modelo es potente, pero añade complejidad conceptual que no estaba justificada para el alcance de este proyecto de desarrollo individual.

Tecnología	Criterios Evaluados	Justificación	Estado
NestJS 10 + TypeScript ~6.0	Modularidad, tipado estático, seguridad integrada	Arquitectura modular con herramientas nativas de control de acceso y validación. Los errores de tipo se detectan antes de ejecutar el código.	Seleccionado
Node.js + Express.js	Modularidad, flexibilidad	No impone ninguna estructura de proyecto, lo que dificulta el mantenimiento en sistemas complejos.	Descartado
Python + Django	Tipado, ecosistema de datos	Lenguaje distinto al resto del stack, impide compartir definiciones	Descartado
Python + Fast API	Rendimiento, tipado	Buen rendimiento y validación integrada, mismo problema de diferencias de lenguajes	Descartado

Tabla 2: Comparación de frameworks del Backend. Elaboración propia

El backend de FinVise necesitaba un framework que resolviera varios problemas al mismo tiempo. Organizar el código en piezas independientes, detectar errores antes de que el sistema

esté en funcionamiento y manejar la seguridad sin depender de librerías externas. Node.js con Express.js fue la primera opción descartada, no porque sea una mala tecnología, sino porque no impone ninguna estructura, lo que significa que cada decisión de organización recae sobre el desarrollador. En un sistema con once módulos distintos, eso se convierte rápidamente en un problema de mantenimiento. Python con Django y con FastAPI fueron considerados por su solidez en el manejo de datos, pero introducen un problema concreto. Al usar un lenguaje diferente al del resto del sistema, no es posible compartir directamente las definiciones de datos entre el backend y el frontend, lo que abre la puerta a errores difíciles de rastrear. NestJS con TypeScript resolvió todos estos puntos sin necesidad de añadir nada externo (Myśliwiec, 2017).

Modelo	Criterios Evaluados	Justificación	Estado
Claude Sonnet (Anthropic)	Ventana de contexto, seguimiento de instrucciones, tono configurable	Mantiene coherencia en conversaciones, adapta respuestas al usuario	Seleccionado
GPT-4o (OpenAI)	Ventana de contexto, seguimiento de instrucciones	Capacidades similares, menos control sobre el tono en dominios especializados	Alternativa evaluada
Gemini 1.5 Pro (Google)	Ventana de contexto, procesamiento multimodal	Mayor ventana del contexto del mercado, menor consistencia en temas financieros	Descartado
LlaMa 3 (Meta, despliegue propio)	Independencia de proveedor, privacidad	Requiere infraestructura especializada, conocimientos técnicos avanzados	Descartado

Tabla 3: Comparación de modelos de IA. Elaboración propia

Elegir el modelo de inteligencia artificial no fue una decisión sencilla, porque los modelos disponibles hoy en día son bastante similares en lo que ofrecen a primera vista.

GPT-4o de OpenAI fue la alternativa más seria. Tiene una ventana de contexto de 128K tokens y sigue instrucciones con buena precisión. El problema es que ofrece menos control

sobre cómo se comporta el asistente en dominios especializados, y en FinVise ese control importa ya que el tono tiene que adaptarse al perfil de cada usuario, y eso no es algo que GPT-4o facilite de la misma manera.

Gemini 1.5 Pro de Google tiene la ventana de contexto más grande del mercado, llegando hasta 2 millones de tokens. Pero su principal diferencial es precisamente su capacidad multimodal, es decir, procesar texto, audio, video e imágenes de forma integrada, algo que FinVise simplemente no necesita en su estado actual.

LLaMA 3 de Meta era atractivo por ser de código abierto y eliminar la dependencia de un proveedor externo. El problema es que desplegarlo requiere infraestructura especializada y un nivel de operaciones técnicas que no era viable para el desarrollo de un prototipo.

Claude Sonnet fue el modelo seleccionado. Combina una ventana de contexto de 200K tokens, suficiente para conversaciones largas, con un buen seguimiento de instrucciones detalladas y la posibilidad de ajustar el tono del asistente según el perfil del usuario, que es exactamente lo que el ChatModule de FinVise necesita (Anthropic, 2024).

Proveedor	Cobertura	Justificación	Estado
TwelveData	Acciones y bolsa de valores	Precios en tiempo real, plan gratuito suficiente	Seleccionado
CoinGecko	Criptomonedas	Amplia cobertura de criptomonedas, acceso gratuito para volúmenes moderados	Seleccionado
Alternative.me	Sentimiento del mercado	Publica el índice Fear & Greed, complementa precios con el estado emocional del mercado	Seleccionado
FRED.API	Indicadores macroeconómicos	Datos oficiales de la Reserva Federal de los Estados Unidos, acceso gratuito	Seleccionado
News.API	Noticias financieras	Noticias recientes clasificadas por	Seleccionado

		compañía o tema, plan gratuito	
Yahoo Finance API	Acciones y Criptomonedas	Cobertura similar a TwelveData, límites de uso estrictos, inestabilidad en interfaz (no es oficial)	Descartado
Bloomberg API	Cobertura financiera global	Mayor cobertura del mercado, costo superior a \$20.000 USD anuales/licencia	Descartado

Tabla 4: Comparación de fuentes de datos del mercado. Elaboración propia

Depender de una sola fuente de datos en un sistema financiero es un riesgo que no tiene mucho sentido asumir, especialmente cuando hay alternativas gratuitas que cubren dimensiones distintas del mercado.

Por eso el MarketDataModule se construyó sobre cinco proveedores, cada uno especializado en un tipo de información distinto. TwelveData entrega los precios de acciones en tiempo real. CoinGecko hace lo mismo para criptomonedas. Alternative.me aporta el índice Fear & Greed, que funciona como una señal del estado emocional del mercado en un momento dado. FRED API provee los indicadores macroeconómicos oficiales de la Reserva Federal de Estados Unidos. Y NewsAPI incorpora las noticias financieras más recientes.

Si alguno de estos proveedores falla, el sistema redirige las consultas a los demás de forma automática, sin que el usuario note ninguna interrupción.

7.2.4 Arquitectura General del sistema

Diagrama C4 - Arquitectura general - Finvise

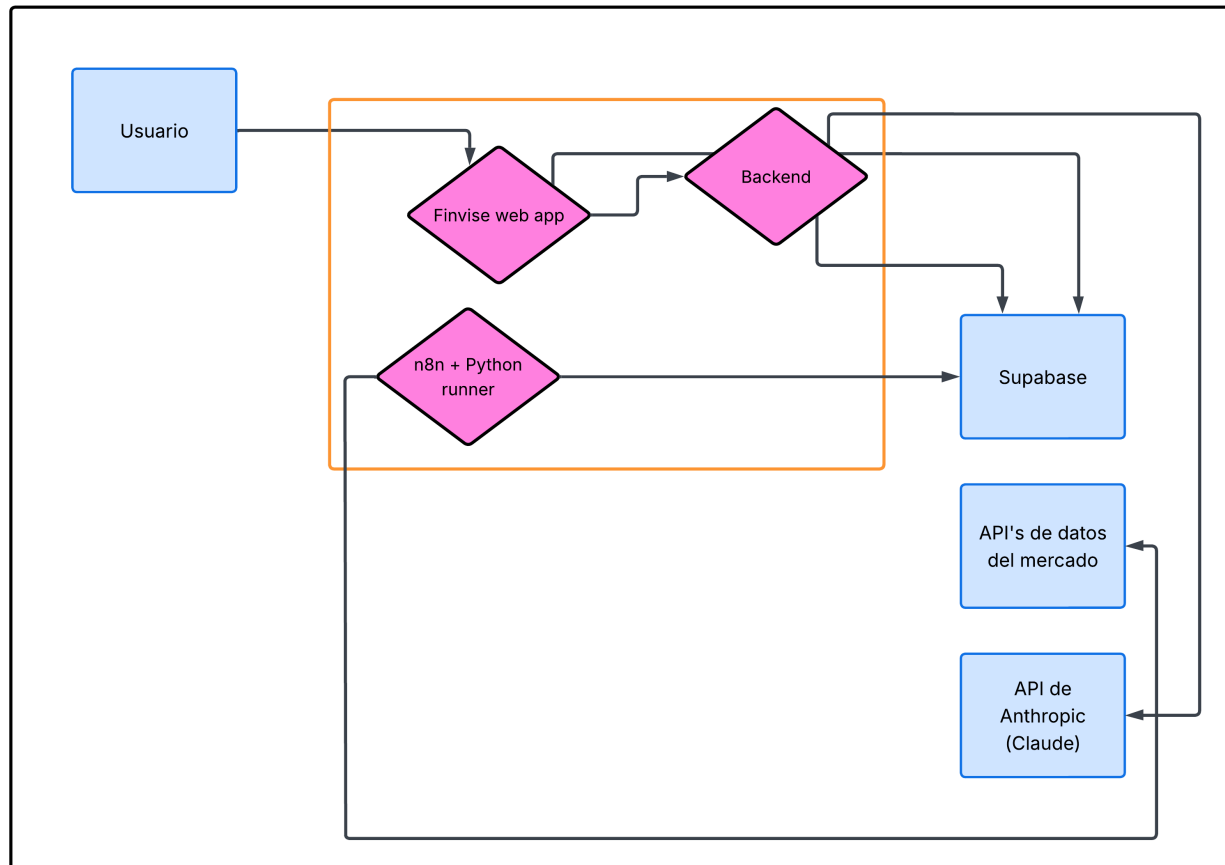


Figura 3: Diagrama de Arquitectura general del sistema. Elaboración propia

La arquitectura de FinVise gira alrededor de cinco elementos principales que trabajan juntos para ofrecer una experiencia educativa personalizada.

El usuario accede a la plataforma a través de la aplicación web, que se comunica con el backend en NestJS mediante peticiones REST para obtener recomendaciones, análisis de sesgos y respuestas del chat educativo. Toda la información del usuario se persiste en Supabase, que además de funcionar como base de datos PostgreSQL actúa como proveedor de autenticación mediante tokens JWT.

Para generar las recomendaciones y gestionar el chat, el backend se apoya en la API de Claude Sonnet de Anthropic, que opera como motor de lenguaje natural. Los datos de mercado, que incluyen precios de acciones y criptomonedas, tasas del tesoro, sentimiento de noticias e índice Fear & Greed, los obtiene una combinación de n8n y un Python Runner que

ejecuta scripts programados. Los precios se actualizan una vez al día de lunes a viernes a las 10:00 PM hora Colombia, y el sentimiento se recalcula cada seis horas. Los resultados se escriben directamente en las tablas `prices_cache` y `sentiment_cache` de Supabase, de modo que el backend pueda consultarlos sin latencia adicional.

El impacto educativo de la plataforma se evalúa a través del delta de conocimiento calculado a partir de las encuestas pre y post sesión.

7.3 Modelo de Datos

El modelo de datos de FinVise en Supabase (PostgreSQL) está compuesto por las siguientes entidades principales:

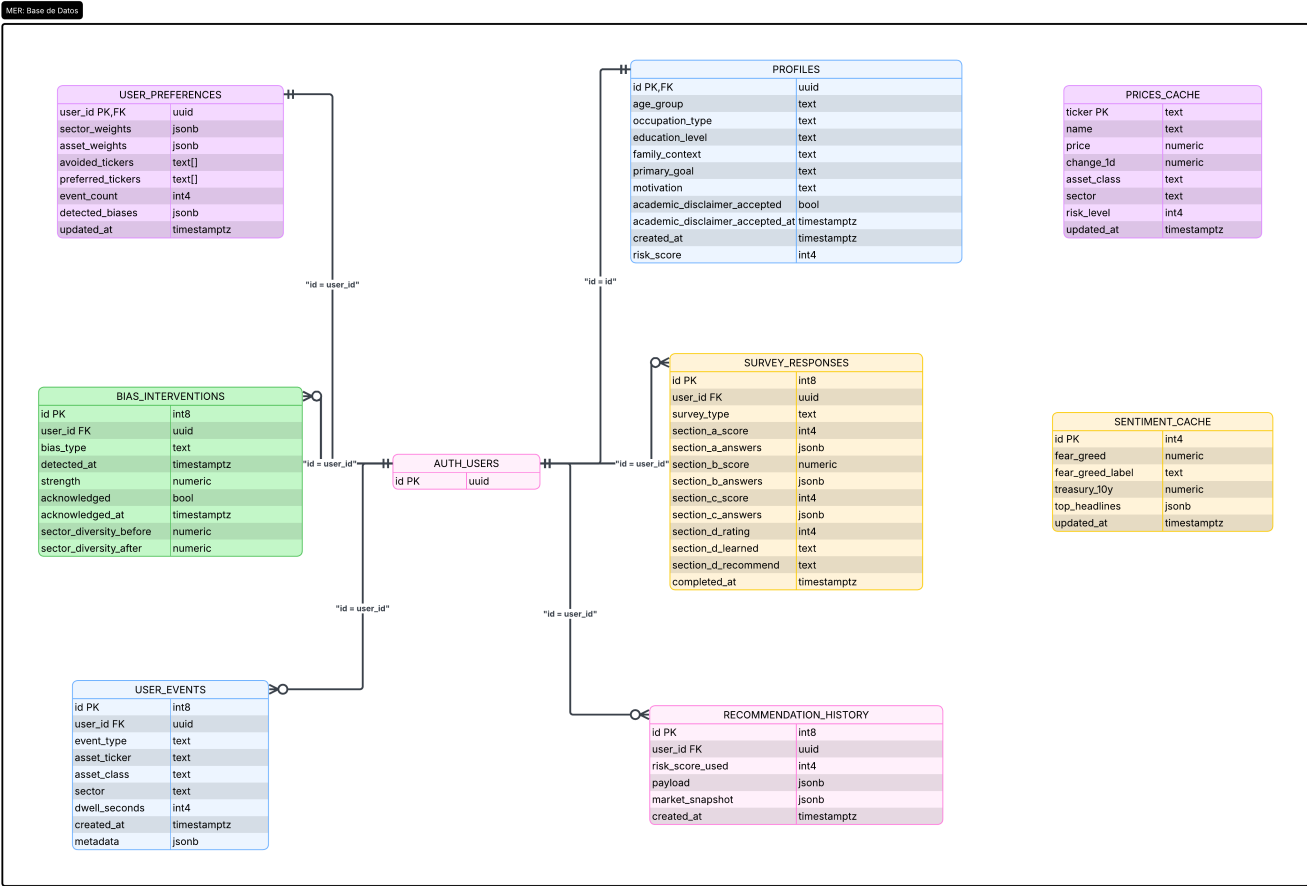


Figura 4: Esquema de la Base de Datos. Elaboración propia

El modelo entidad-relación presenta una base de datos centrada en el usuario. El núcleo es la tabla `AUTH_USERS` que actúa como eje principal de todas las relaciones.

A partir de ella se desprenden tablas que almacenan el perfil demográfico y de riesgo del usuario, sus preferencias de inversión, sus respuestas a encuestas de evaluación financiera,

el historial de recomendaciones generadas, los eventos de interacción con la plataforma y las intervenciones realizadas ante sesgos detectados. Complementan el modelo dos tablas de caché independientes, que almacenan datos de mercado y métricas de sentimiento sin estar ligadas directamente a un usuario específico.

7.4 Diseño de Interfaz Gráfica

La interfaz de FinVise fue diseñada siguiendo principios de diseño centrado en el usuario, con énfasis en claridad, accesibilidad y consistencia visual. Los principales módulos de interfaz son:

- **Dashboard principal:** Vista central con tira de precios en vivo, alertas de sesgos activos, resumen del risk_score y accesos rápidos a todos los módulos.

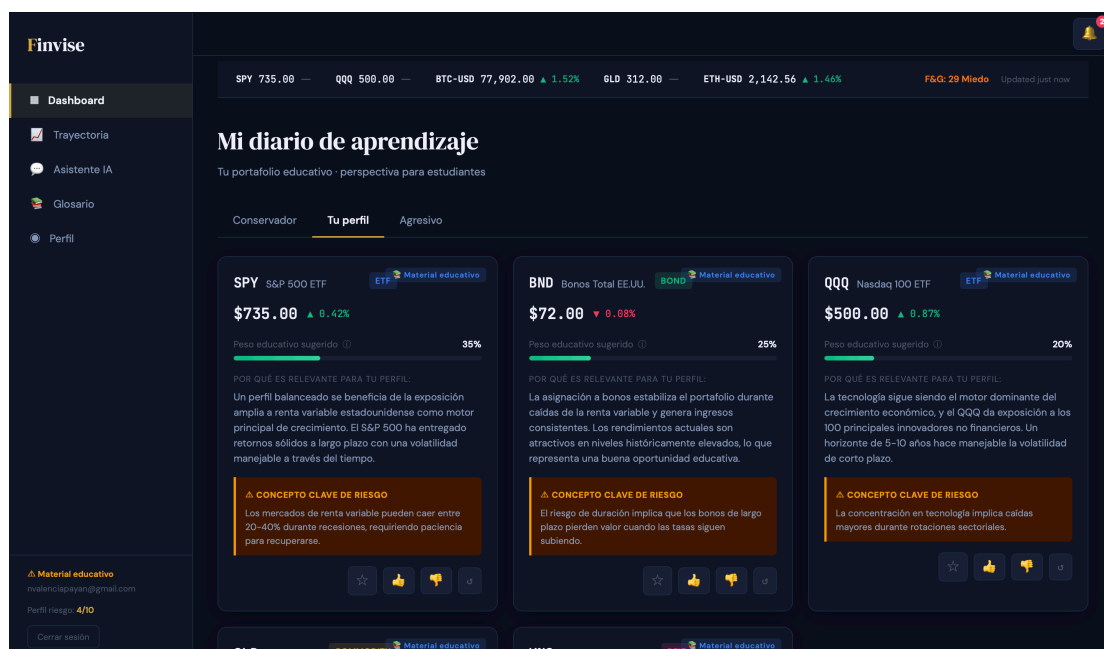


Figura 5: Dashboard principal. Elaboración propia

Al ingresar, el usuario encuentra en la parte superior una tira de precios en tiempo real que muestra las cotizaciones actualizadas de los principales activos del mercado, índices bursátiles, criptomonedas, materias primas, junto con el índice Fear & Greed, que refleja el sentimiento general del mercado en ese momento. En el panel lateral se despliegan los accesos directos a todos los módulos de la plataforma: Trayectoria, Asistente IA, Glosario y Perfil, así como el nivel de riesgo asignado al usuario y las alertas de sesgos cognitivos detectados en su comportamiento reciente. El contenido central se organiza alrededor del "Diario de aprendizaje", una vista personalizada que presenta los activos recomendados según el perfil de riesgo del usuario: conservador, balanceado o agresivo, con precios en vivo, peso sugerido dentro del portafolio educativo, y una explicación

contextualizada de por qué cada activo es relevante para su situación particular, acompañada siempre de un concepto clave de riesgo que refuerza el propósito educativo de la plataforma.

- **Portafolio educativo:** Tarjetas de activos con ticker, precio, cambio, asignación sugerida, explicación pedagógica contextualizada y botones de feedback (rate_up, rate_down, save, dismiss, replace).

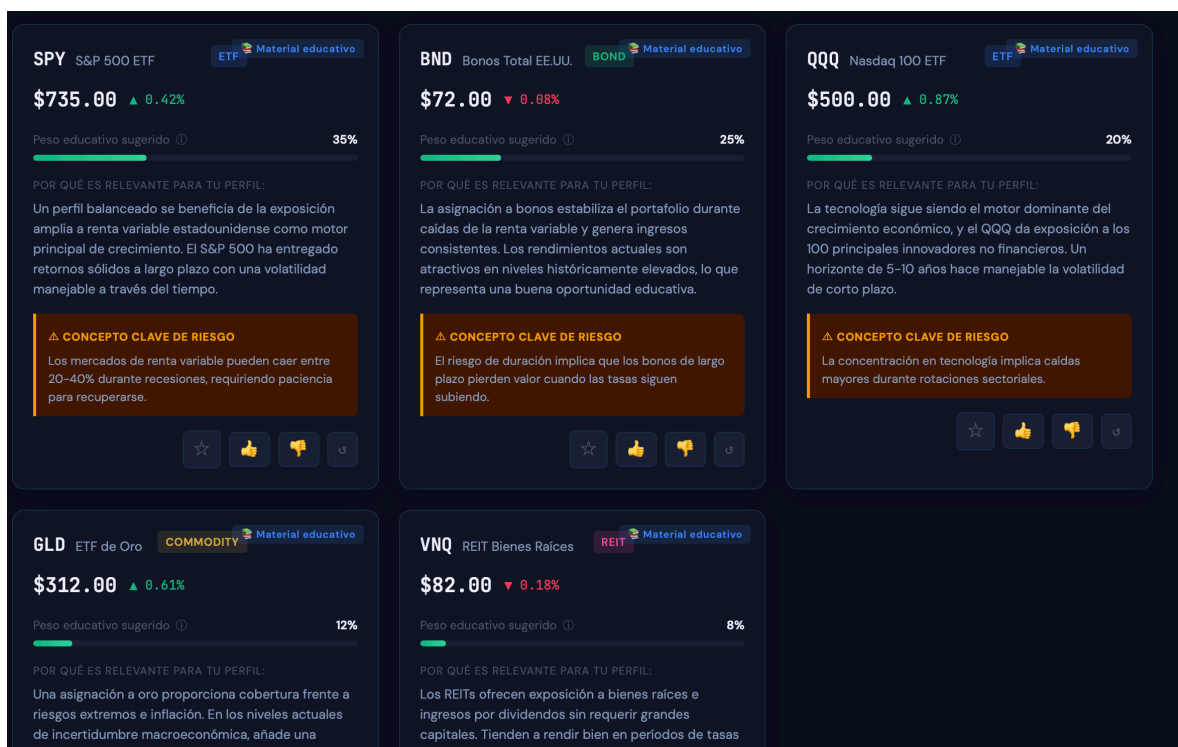


Figura 6: Portafolio educativo (Recomendaciones) Elaboración propia

Cada activo recomendado se presenta como una tarjeta independiente que concentra toda la información relevante en un solo lugar.

En la parte superior aparece el ticker, el nombre completo del instrumento y su categoría, ya sea ETF, bono, commodity, REIT u otra. Justo debajo se muestra el precio actual y su variación porcentual en tiempo real. Una barra de progreso indica visualmente el peso educativo sugerido dentro del portafolio, es decir, qué proporción del capital simulado debería destinarse a ese activo según el perfil de riesgo del usuario.

La explicación que acompaña cada tarjeta está escrita en lenguaje accesible y argumenta por qué ese instrumento es relevante para ese usuario en particular, considerando su horizonte de inversión, su tolerancia al riesgo y sus objetivos. Cada tarjeta incluye además un bloque de Concepto Clave de Riesgo, que señala el principal factor de riesgo

asociado a ese tipo de activo, ya sea volatilidad, riesgo de duración o concentración sectorial, reforzando así el aprendizaje crítico.

Por último, un conjunto de botones de feedback permite al usuario calificar positiva o negativamente la recomendación, guardarla como favorita, descartarla o pedir que sea reemplazada por otra alternativa. Cada una de esas acciones retroalimenta continuamente el motor de recomendaciones.

- **Simulador de portafolio:** Selector de capital con valores predefinidos y campo personalizado. Tabla de resultados por activo con ganancia/pérdida en pesos y porcentaje. Total en verde/rojo.

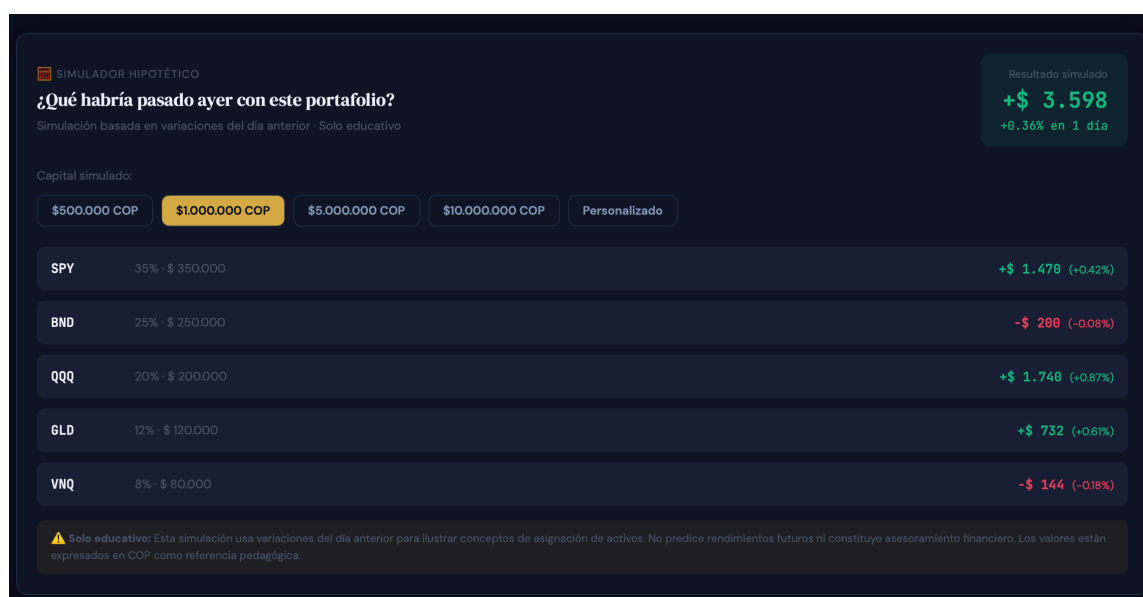


Figura 7: Simulador de portafolio. Elaboración propia

Este módulo responde a una pregunta concreta y pedagógica. ¿Qué habría pasado ayer con este portafolio? El usuario selecciona un monto de capital simulado en pesos colombianos, con opciones predefinidas de \$500.000, \$1.000.000, \$5.000.000 y \$10.000.000 COP, o un valor personalizado, y la plataforma calcula automáticamente cómo se habría distribuido ese capital entre los activos recomendados según los pesos sugeridos del portafolio. La tabla de resultados presenta cada activo mostrando el porcentaje asignado, el monto invertido equivalente y la ganancia o pérdida generada en el día anterior, expresada tanto en pesos como en variación porcentual, con colores verde y rojo para facilitar la lectura. Al final se presenta el resultado del portafolio completo.

El cálculo sigue una lógica directa. Para cada activo i con peso w_i y un capital total C ingresado por el usuario, el capital asignado es:

$$C_i = w_i \cdot C$$

La ganancia o pérdida en pesos para ese activo se calcula como:

$$\Delta C_i = C_i \cdot r_i$$

donde r_i es la variación porcentual del precio del activo en el día anterior, obtenida de las fuentes de datos de mercado. El resultado total del portafolio es:

$$\Delta Cp = \sum_i \Delta C_i = C \cdot \sum_i w_i \cdot r_i$$

Este resultado equivale al retorno ponderado del portafolio para ese período de un día, lo que permite al usuario observar directamente el principio de diversificación: activos con retornos negativos se compensan parcialmente con los que tuvieron retornos positivos, y la magnitud de esa compensación depende de los pesos asignados y de la correlación entre los movimientos de cada activo.

Por otro lado, el módulo incluye un aviso explícito que refuerza su naturaleza estrictamente educativa. Los valores se calculan con variaciones reales del día anterior únicamente para ilustrar el concepto de diversificación y asignación de activos, sin constituir en ningún caso una proyección de rendimientos futuros ni asesoramiento financiero.

- **Asistente IA conversacional:** Interfaz de chat con renderizado Markdown completo, sugerencias rápidas y contexto persistente.

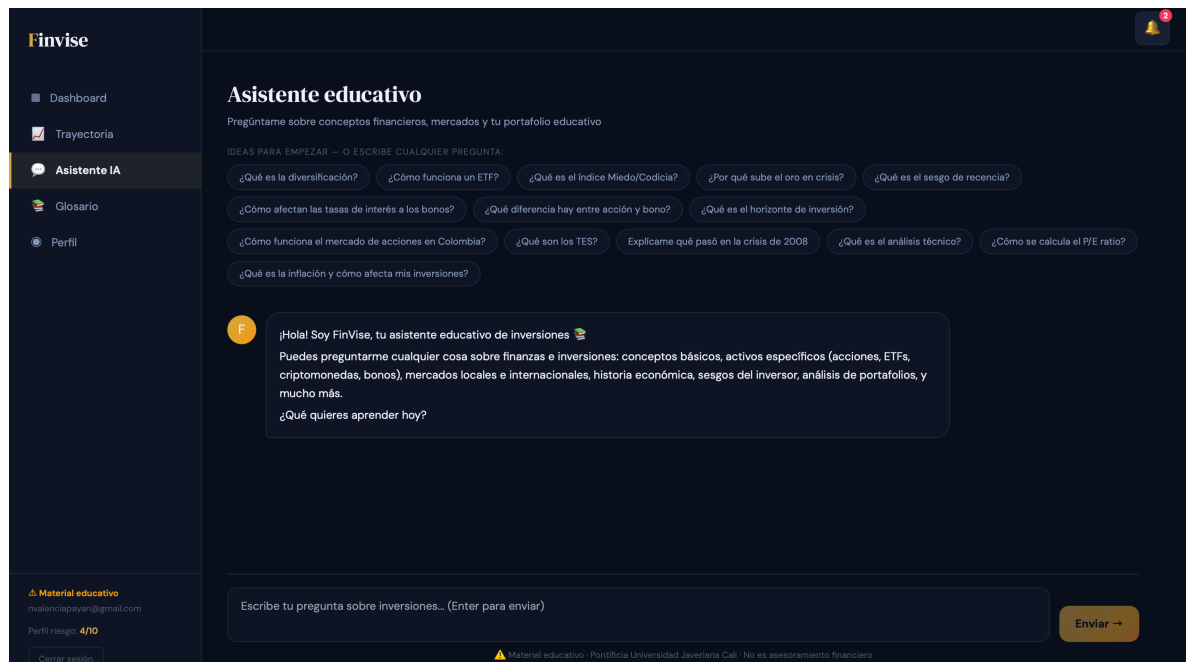


Figura 8: Asistente IA Conversacional. Elaboración propia

El módulo de chat integra un asistente conversacional especializado en educación financiera que el usuario puede consultar libremente sobre cualquier concepto relacionado con inversiones, mercados y finanzas. Al abrir la pantalla, una serie de preguntas sugeridas reduce la fricción inicial y orienta al usuario hacia los temas más relevantes. El usuario también puede escribir cualquier pregunta libre en el campo de texto inferior.

El contexto de la conversación se mantiene de forma persistente durante la sesión, permitiendo preguntas de seguimiento y profundización progresiva.

- **Página de sesgos:** Tarjetas de sesgos con intensidad (barra de progreso), evidencia, explicación investigativa y consejo práctico. Botón de reconocimiento que registra el evento `bias_acknowledged`.



Figura 9: Página de sesgos. Elaboración propia

Cuando el sistema identifica un patrón de comportamiento que sugiere la presencia de un sesgo cognitivo, muestra una tarjeta con borde naranja que interrumpe suavemente la experiencia para invitar al usuario a la reflexión.

La tarjeta se organiza en tres bloques. El primero muestra la evidencia conductual concreta que activó la alerta. El segundo contextualiza el sesgo con datos históricos de mercado, para contrastar la percepción distorsionada del usuario con lo que realmente ha ocurrido. El tercero ofrece una pregunta de autoevaluación diseñada para que el usuario reconozca por sí mismo si su decisión está siendo influenciada por ese sesgo.

- **Trayectoria de aprendizaje:** Tres paneles de métricas: perfil UCB1, impacto educativo pre/post y rendimiento histórico. Gráfico de línea de evolución de intereses por sector.

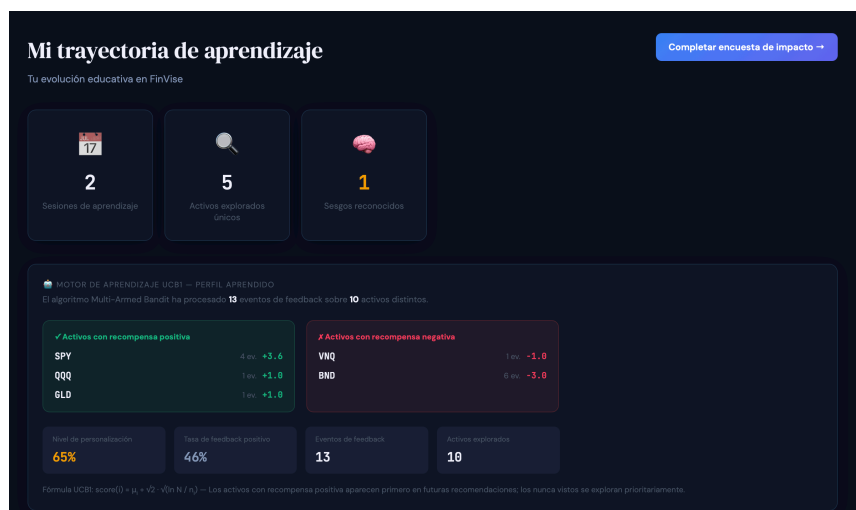


Figura 10: Trayectoria de aprendizaje (Perfil UCB1). Elaboración propia

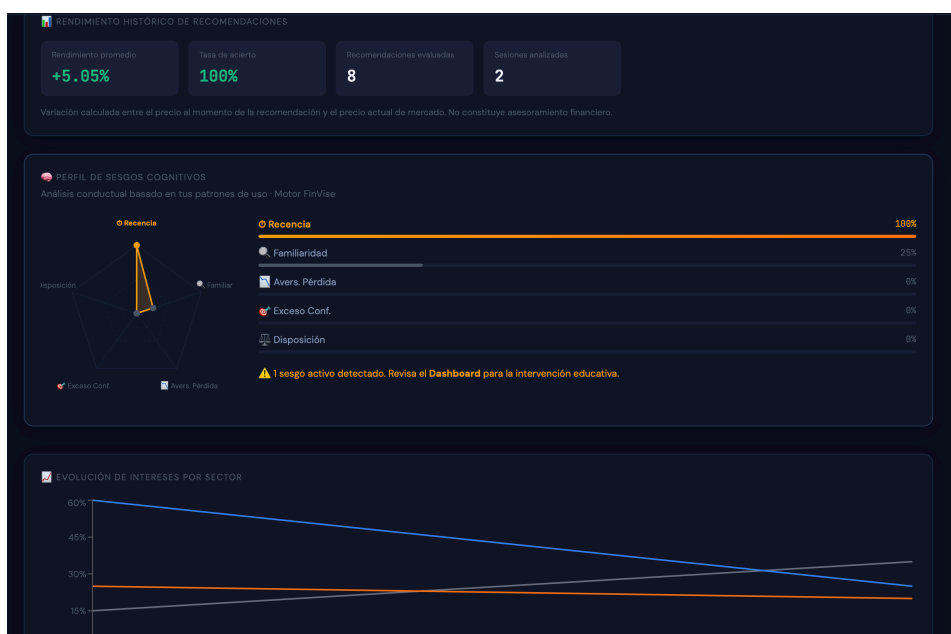


Figura 11: Trayectoria de aprendizaje (Rendimiento histórico, sesgos y gráfico de línea de evolución de intereses por sector). Elaboración propia

Este módulo muestra en un solo lugar cómo ha evolucionado el aprendizaje del usuario a lo largo del tiempo.

En la parte superior aparecen tres cifras que resumen lo más importante. Cuántas sesiones ha tenido, cuántos activos distintos ha explorado y cuántos sesgos cognitivos ha identificado hasta el momento.

En el centro se explica cómo funciona el motor UCB1. El usuario puede ver exactamente qué hizo el algoritmo con cada una de sus reacciones. Cómo separó los activos entre los

que generaron una respuesta positiva y los que no, qué tan personalizado está el sistema, qué porcentaje de las reacciones fueron positivas y cuántas interacciones lleva en total. La fórmula del algoritmo aparece al pie de esta sección para quien quiera entender la lógica detrás.

Debajo está el impacto educativo. Esta sección compara los resultados de la encuesta que el usuario completó al ingresar por primera vez con los de la encuesta que responde después de usar la plataforma, que hace las mismas preguntas con el propósito de validar cuánto aprendió.

A continuación aparece el historial de recomendaciones, el cual consiste en el promedio de efectividad, cuántas dieron en el clavo y cuántas fueron evaluadas en total.

Los sesgos cognitivos se visualizan de dos formas al mismo tiempo. Un gráfico de radar y unas barras de intensidad, lo que hace fácil identificar cuáles pesan más en el comportamiento del usuario.

Por último, un gráfico de líneas muestra cómo han cambiado los intereses del usuario por sector a lo largo de las sesiones, dejando ver cómo sus preferencias fueron evolucionando a medida que avanzaba en la plataforma.

- **Encuesta pre/post:** Flujo guiado de 4 secciones con retroalimentación inmediata en la Sección A. Muestra el delta al completar la versión post.

Sección A — Conocimiento conceptual

Pregunta 1 de 8

¿Qué significa diversificar un portafolio de inversión?

Poner todo el dinero en el activo que más ha subido

Distribuir el dinero entre diferentes tipos de activos para reducir el riesgo

Cambiar de inversión cada semana según las noticias

Invertir solo en empresas del propio país

Confirmar respuesta

Figura 12: Encuesta Pre/Post. Elaboración propia

La encuesta de evaluación está dividida en cuatro secciones que miden dimensiones distintas del aprendizaje financiero del usuario.

La primera, dedicada al conocimiento conceptual, presenta preguntas de selección múltiple diseñadas para distinguir comprensión genuina de nociones superficiales o sesgadas. Las opciones incorrectas no son respuestas al azar, sino que representan errores cognitivos comunes del inversor novato, lo que obliga al usuario a pensar con más cuidado antes de responder.

Al confirmar cada respuesta, el sistema entrega retroalimentación inmediata que explica por qué la opción elegida es correcta o incorrecta, convirtiendo el error en una oportunidad de aprendizaje en lugar de un simple punto perdido.

La encuesta se aplica en dos momentos. Antes de usar FinVise y después. Esa estructura es la que permite calcular el delta de conocimiento, es decir, la diferencia concreta entre lo que el usuario sabía al entrar y lo que sabe al salir.

La encuesta completa se encuentra en el Anexo B.

8. DESARROLLO DEL PROTOTIPO

8.1 Configuración del Entorno de Desarrollo

La configuración del entorno de desarrollo tuvo como objetivo establecer una base de trabajo coherente y estable tanto para el backend como para el frontend, de modo que ambas partes del sistema pudieran desarrollarse y probarse de forma independiente.

El backend fue hecho con NestJS, que genera automáticamente la estructura base del proyecto incluyendo módulos, servicios y controladores. Esta estructura facilitó que cada funcionalidad del sistema pudiera desarrollarse de forma aislada sin interferir con las demás.

El frontend se construyó sobre React con Vite como herramienta de construcción, combinación que ofrece tiempos de arranque casi inmediatos y refleja los cambios en pantalla sin necesidad de recargar la página. Los estilos visuales se implementaron con Tailwind CSS, que permite definir la apariencia de cada componente directamente en el marcado sin mantener archivos de estilos separados.

La gestión de usuarios, sesiones y almacenamiento de datos se centralizó en Supabase, que provee tanto la base de datos como el servicio de autenticación. Todo el código fue escrito en TypeScript con validación activada en ambas capas del sistema, lo que permitió detectar errores de tipo durante el desarrollo y no en tiempo de ejecución. Las credenciales de acceso a los servicios externos se gestionaron a través de variables de entorno, entre ellas las claves para Supabase, para el modelo de lenguaje de Anthropic y para los proveedores de datos financieros de mercado.

Como componente adicional, se configuró una capa de automatización utilizando n8n, una herramienta de orquestación de flujos de trabajo que se desplegó junto con un contenedor de Python 3.11 mediante Docker Compose.

Esta combinación se encarga de ejecutar de forma programada las tareas de recolección y procesamiento de datos que el backend consume, sin que el servidor principal tenga que gestionar esas operaciones directamente.

Se definieron tres flujos automatizados: el primero corre de lunes a viernes al cierre del mercado y utiliza la librería `yfinance` para descargar los precios de los activos y almacenarlos en Supabase, el segundo se ejecuta cada seis horas consultando el índice de miedo y codicia del mercado, las noticias financieras más recientes y el rendimiento del bono del tesoro de Estados Unidos a diez años, y el tercero se activa diariamente en la madrugada para analizar el historial de interacciones de cada usuario e identificar patrones de sesgo cognitivo como aversión a las pérdidas, sesgo de familiaridad o efecto de disposición, actualizando los perfiles almacenados en la base de datos.

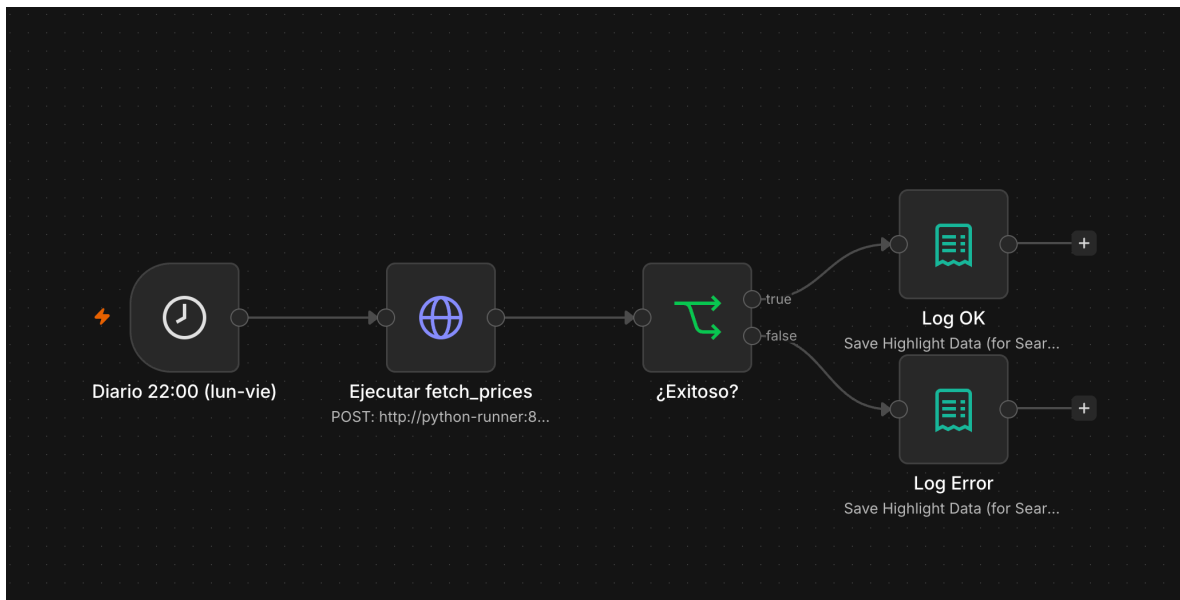


Figura 13: n8n workflow (Fetch Prices) Elaboración propia

La Figura 13 muestra el flujo automatizado configurado en n8n para la obtención diaria de precios. El workflow se activa de lunes a viernes al cierre del mercado, consulta la librería yfinance para cada uno de los 42 tickers del universo de inversión, y escribe los resultados en la tabla `prices_cache` de Supabase. Cada nodo del flujo tiene manejo de error independiente, de modo que si un ticker no devuelve datos, el proceso continúa con los demás sin interrumpirse.

8.2 Implementación del Backend

La implementación del backend tuvo como objetivo soportar la lógica central del sistema, incluyendo la generación de recomendaciones educativas, la consulta de datos de mercado, la interacción con el modelo de lenguaje y el registro del comportamiento del usuario.

Se desarrolló una API RESTful organizada en módulos independientes, donde cada uno agrupa su controlador, su servicio con la lógica de negocio y sus reglas de validación, siguiendo las convenciones de NestJS para mantener el código ordenado y fácil de extender.

El módulo de recomendaciones es el núcleo del sistema. Cuando el usuario solicita un portafolio, el servicio construye una lista de activos candidatos combinando el nivel de tolerancia al riesgo del usuario con los puntajes calculados por el algoritmo UCB1, que pondera cada activo según el historial de interacciones previas del usuario. Esa lista de candidatos se envía al modelo Claude de Anthropic, que genera las explicaciones educativas personalizadas para cada activo. Cuando el servicio de Anthropic no está disponible, el sistema cae en modo de demostración utilizando datos predefinidos, garantizando que la interfaz siempre tenga algo que mostrar.

El módulo de mercado expone dos endpoints: uno para consultar los precios actualizados de los activos disponibles y otro para obtener indicadores de sentimiento como el índice de miedo y codicia del mercado y el rendimiento de los bonos del tesoro de EE.UU.

El módulo de chat recibe cada mensaje del usuario junto con el historial completo de la conversación para que el modelo pueda responder con coherencia a lo largo de toda la sesión.

Los eventos de comportamiento del usuario, como las valoraciones sobre los activos recomendados, se registran a través del módulo de eventos y son consumidos por el servicio del algoritmo bandit para actualizar los puntajes UCB1 de cada activo. El módulo de sesgos analiza esos eventos para identificar patrones cognitivos en el comportamiento del usuario, como la aversión al riesgo o el sesgo de confirmación.

Adicionalmente, el backend gestiona un conjunto de funcionalidades complementarias: un sistema de alertas sobre los activos del portafolio, el historial de sesiones de recomendaciones anteriores, encuestas aplicadas antes y después de la experiencia, y un endpoint que expone el perfil de aprendizaje del algoritmo bandit mostrando qué activos ha favorecido o descartado el usuario con el tiempo.

8.3 Implementación del Frontend

La implementación del frontend tuvo como objetivo construir una interfaz web responsiva que permitiera al usuario interactuar con todas las funcionalidades del sistema de forma intuitiva.

Se desarrolló con React 19 utilizando componentes funcionales y hooks para la gestión del estado y los efectos de cada pantalla. La sesión del usuario se administra a través de un contexto de autenticación global que envuelve toda la aplicación, manteniendo disponible la información del usuario activo en cualquier componente sin necesidad de pasar datos manualmente entre capas.

El inicio de sesión, el registro y el cierre de sesión se manejan directamente con el cliente de Supabase integrado en el frontend, que se comunica con el servicio de autenticación de Supabase sin pasar por el servidor propio de la aplicación. El perfil del usuario también se consulta directamente desde la base de datos de Supabase.

Para el resto de las operaciones, como generar recomendaciones, enviar mensajes al chat o registrar eventos, se utiliza Axios con un interceptor que obtiene el token de sesión activo y lo adjunta automáticamente en el encabezado de cada solicitud saliente, de modo que los componentes individuales no necesitan gestionar ese detalle.

La navegación entre las distintas pantallas se implementó con React Router DOM, y la visualización de datos financieros como la distribución del portafolio se construyó con Recharts, una biblioteca de gráficos diseñada para integrarse directamente con componentes de React. Los estilos de toda la interfaz se definieron con Tailwind CSS, cuyas clases

utilitarias permitieron diseñar una experiencia visual coherente y adaptable a distintos tamaños de pantalla a lo largo de todas las secciones del sistema.

8.4 Integración de Fuentes de Datos Externas

La integración de fuentes de datos externas usa una arquitectura de tres niveles para garantizar que el sistema siempre tenga precios disponibles.

El primer nivel es la caché en memoria del proceso NestJS, que retiene los datos por 5 minutos fijos para evitar llamadas repetidas a las APIs en solicitudes consecutivas.

El segundo nivel es la tabla `prices_cache` de Supabase, que almacena los precios de las 42 acciones y ETFs del universo de inversión y es consultada cuando la caché en memoria expira. Si los datos están desactualizados, el sistema los devuelve de todas formas y lanza una actualización en segundo plano.

El tercer nivel son las APIs externas, que se consultan solo cuando la tabla de Supabase está vacía o no está configurada: TwelveData es la fuente principal para acciones y ETFs, con yfinance como respaldo si TwelveData falla o alcanza su límite de peticiones. CoinGecko se usa siempre para los activos cripto (BTC, ETH y SOL) por ser gratuito y sin restricciones de volumen.

Los datos de sentimiento de mercado siguen una lógica similar. Primero se consulta la tabla `sentiment_cache` de Supabase y, si está vacía o desactualizada, se obtienen en tiempo real desde tres fuentes independientes: Alternative.me para el índice de Miedo y Codicia, la API de la Reserva Federal (FRED) para el rendimiento del bono del Tesoro a 10 años, y NewsAPI para los titulares de negocios recientes.

8.5 Implementación del Motor de IA y Algoritmo UCB1

8.5.1 Flujo del Motor de Recomendaciones

Cuando el usuario solicita recomendaciones, el backend lanza en paralelo tres operaciones. Obtiene los datos de mercado con caché y fallback (timeout de 10 s), consulta el sentimiento de mercado desde Supabase (timeout de 5 s) y recupera los eventos históricos del usuario para calcular los scores UCB1 (timeout de 5 s).

De forma separada, pero en el mismo ciclo, consulta las preferencias del usuario desde la tabla `user_preferences`. Con toda esa información disponible, construye el pool de candidatos, filtra los 42 tickers de acciones y ETFs más 3 criptos por compatibilidad de riesgo y preferencias, y los ordena por score UCB1 dentro de cada clase de activo, tomando los dos

mejores por clase. Luego construye el prompt para Claude Sonnet con el perfil de vida del usuario, los datos actuales del mercado, el índice de Miedo y Codicia, el rendimiento del Tesoro a 10 años, los titulares de noticias y la lista de candidatos ordenados.

La llamada a Claude Sonnet tiene un timeout de 20s. Si falla, el sistema devuelve directamente el portfolio calculado de forma cuantitativa sin pasar por el modelo. Las tres variantes de perfil que ve el usuario en pantalla las genera el frontend haciendo tres llamadas independientes al mismo endpoint, cada una con el score representativo del perfil correspondiente. El perfil adyacente inferior al del usuario (tab izquierdo), el perfil actual (tab central) y el perfil adyacente superior (tab derecho).

Los perfiles disponibles y sus scores representativos son Conservador, Equilibrado, Crecimiento y Agresivo. Si el usuario ya está en el extremo de la escala, el tab del extremo repite su mismo perfil.

Risk Score	Clasificación (Perfil)
Entre 1-3	Conservador
Entre 4-6	Balanceado
Entre 7-8	Crecimiento
Entre 9-10	Agresivo

Tabla 5: Distribución de perfiles de riesgo

Elaboración propia

La base teórica de todos los perfiles viene de Markowitz (1952), quien demostró que combinar distintos tipos de activos en un portafolio permite reducir el riesgo total sin necesariamente reducir el retorno esperado. La idea central es que no todos los activos bajan al mismo tiempo ni por las mismas razones, por lo que tenerlos juntos amortigua las pérdidas. Como el propio autor señala, "The portfolio with maximum expected return is not necessarily the one with minimum variance. There is a rate at which the investor can gain expected return by taking on variance, or reduce variance by giving up expected return" (Markowitz, 1952, p. 79).

8.5.2 Perfil Conservador

El peso alto en bonos para este perfil responde a lo planteado por Samuelson (1969), quien argumentó que los inversionistas con menor tolerancia al riesgo deben reducir su exposición a activos volátiles. Cuanto menos dispuesto esté alguien a perder dinero en el corto plazo, más debería tener en instrumentos de renta fija como los bonos, que ofrecen retornos más predecibles. La reserva en efectivo del 10% sigue la recomendación del CFA Institute (2019) de mantener entre un 5% y un 15% del portafolio en activos líquidos para perfiles conservadores, de modo que el inversionista pueda responder a necesidades urgentes sin tener que vender otros activos en un mal momento.

8.5.3 Perfil Balanceado

Este perfil es una variante del portafolio 60/40, que es uno de los modelos de inversión más estudiados en la literatura financiera. Ibbotson y Kaplan (2000) analizaron este esquema

como referencia para inversionistas con tolerancia al riesgo moderada y horizonte de inversión de mediano plazo, y encontraron que ofrece una buena relación entre retorno y estabilidad a lo largo del tiempo. La inclusión de un 5% en materias primas (commodities) dentro de este perfil se apoya en Idzorek (2006), quien mostró que agregar una pequeña porción de este tipo de activos mejora la diversificación del portafolio porque su comportamiento no suele seguir el mismo patrón que las acciones o los bonos, especialmente en periodos de inflación alta.

8.5.4 Perfil de Crecimiento

La inclusión de criptomonedas en este perfil se basa en el trabajo de Brière et al. (2015), quienes estudiaron el comportamiento del Bitcoin dentro de portafolios diversificados y encontraron que, a pesar de su alta volatilidad, tiene una correlación baja con la mayoría de los activos tradicionales. Esto significa que no sube ni baja necesariamente cuando lo hacen las acciones o los bonos, lo que lo convierte en un elemento que puede mejorar la diversificación del portafolio. Los autores encontraron que asignaciones de entre el 5% y el 20% en Bitcoin mejoraron la relación riesgo-retorno de los portafolios analizados. El 15% asignado a criptomonedas en este perfil cae dentro del rango documentado por los autores.

8.5.5 Perfil Agresivo

La ausencia de bonos en este perfil responde a lo planteado por Faber (2007), quien argumentó que los inversionistas con alta tolerancia al riesgo y un horizonte de largo plazo pueden prescindir de la renta fija a favor de activos con mayor potencial de crecimiento. Liu y Tsyvinski (2021) refuerzan la inclusión de un 30% en criptomonedas al mostrar que los retornos de estos activos no se explican por los mismos factores que mueven al mercado de acciones tradicional, lo que les da un valor propio dentro de portafolios de alto riesgo. El peso asignado busca reflejar precisamente esa mayor exposición al riesgo y al potencial de crecimiento.

8.5.6 Implementación del Algoritmo UCB1

El algoritmo UCB1 se implementa en el BanditService del backend. La elección de este algoritmo se apoya en que resuelve de forma efectiva el problema de exploración y explotación en entornos de bandido multi-brazo, donde el agente debe aprender qué opciones generan mayor recompensa sin haber probado todas ellas (Hu, 2024).

Para cada ticker i con al menos una visualización, el score se calcula como:

$$score(i) = \mu_i + \sqrt{2 \cdot \ln N / n_i}$$

Donde μ_i es la recompensa neta acumulada del ticker i dividida entre el número de veces que fue visualizado. N es el total de visualizaciones del usuario sobre todos los tickers, y n_i es el número de veces que el ticker i fue mostrado al usuario.

El término de exploración $\sqrt{2 \cdot \ln N / n_i}$ garantiza que los activos poco vistos reciban un bonus creciente, forzando al sistema a explorarlos antes de consolidar preferencias, lo que

es la propiedad central del UCB1 que lo hace adecuado para recomendaciones personalizadas con datos escasos.

Para entender por qué esta fórmula funciona, vale la pena desglosarla. El término μ_i representa cuánto ha rendido en promedio un activo dado las interacciones anteriores del usuario. Si el usuario le ha dado *rate_up* a ese activo varias veces, μ_i será alto. Si lo ha descartado, será bajo o negativo. Ese término solo captura la experiencia pasada, y si el algoritmo lo usara exclusivamente, nunca recomendaría activos que el usuario aún no ha evaluado.

Ahí entra el segundo término, $\sqrt{2} \cdot \sqrt{(\ln N / n_i)}$. Este bonus de exploración tiene dos propiedades matemáticas relevantes. Primero, crece a medida que el total de visualizaciones N aumenta, lo que hace que el sistema se vuelva progresivamente más exigente en explorar opciones nuevas a medida que acumula experiencia general. Segundo, decrece a medida que n_i aumenta, es decir, cuanto más veces se le ha mostrado un activo al usuario, menor es su bonus. El resultado combinado es que los activos poco vistos siempre tienen una ventaja inicial que desaparece gradualmente conforme son explorados.

Auer, Cesa-Bianchi y Fischer (2002) demostraron formalmente que este mecanismo produce un regret logarítmico en el tiempo, lo cual significa que el costo acumulado de no saber de antemano cuál es el mejor activo crece como $O(\ln N)$, lo que es óptimo en el sentido de que ningún algoritmo sin información previa puede hacerlo mejor de forma asintótica. En términos prácticos, esto significa que el sistema comete errores al principio por desconocimiento, pero esa tasa de error disminuye de forma predecible a medida que el usuario interactúa más con la plataforma.

En el contexto educativo de FinVise, esta propiedad es especialmente útil porque los usuarios típicamente tienen pocas sesiones antes de formar una opinión sobre la plataforma. El UCB1 garantiza que, incluso con cinco o diez interacciones, el algoritmo ya está priorizando activos relevantes para ese usuario en particular, sin haber necesitado datos previos externos.

Los tickers que el usuario nunca ha visto reciben score infinito y siempre se priorizan. El sistema de recompensas es: *rate_up* = +1.0, *save* = +0.8, *rate_down* = -1.0, *dismiss* = -0.5.

Los valores de recompensa se definieron siguiendo el principio de que señales explícitas deben pesar más que señales implícitas, y que el rechazo debe ser simétrico a la aprobación.

rate_up y *rate_down* se fijan en ± 1.0 porque representan los actos más intencionales del usuario. Hacer clic en "me gusta" o "no es para mi perfil" requiere una decisión consciente y refleja una preferencia clara. La simetría entre ambos responde a que, en términos informativos, saber que un activo no le conviene al usuario es igual de útil que saber que sí le conviene.

save recibe +0.8 porque guardar un activo indica interés concreto y probable consulta futura, pero es una señal ligeramente menos definitiva que una valoración explícita. El usuario puede guardar algo para revisarlo después sin haber formado todavía una opinión sobre si lo incluiría en su portafolio educativo.

dismiss recibe -0.5 porque reemplazar una tarjeta es una señal negativa más débil que *rate_down*. El usuario puede estar descartando el activo simplemente para explorar alternativas, no necesariamente porque lo rechace. Penalizarlo con la mitad de la magnitud del rechazo refleja esa ambigüedad.

8.5.7 Sistema de Detección de Sesgos Cognitivos

El umbral de activación de 0.45 fue definido conscientemente en este trabajo como criterio de diseño, buscando intervenir antes de que el sesgo domine las decisiones del usuario, principio que Pompian (2012) recomienda en términos generales sin proponer un valor numérico específico.

El primero de los cinco sesgos detectados es el de recencia. Para identificarlo, el sistema analiza con qué frecuencia el usuario pide recomendaciones en períodos cortos de tiempo. La idea es que alguien que consulta muy seguido probablemente está reaccionando a lo que acaba de ver o leer, en lugar de tomar decisiones con más calma. Esto se apoya en la heurística de disponibilidad de Tversky y Kahneman (1974) y en los hallazgos de Barber y Odean (2000), quienes mostraron que los inversores más activos tienden a dejarse llevar por señales de corto plazo. Dicho esto, usar la frecuencia de consultas como indicador de este sesgo es una decisión de diseño propia, no una medición directa del fenómeno.

En una línea similar, el sesgo de familiaridad se detecta midiendo qué tan concentradas están las interacciones del usuario en un solo sector. Si más del 45% de sus consultas giran alrededor del mismo tipo de activos, se considera una señal de que está priorizando lo que ya conoce por encima de otras opciones. El fundamento está en investigaciones como las de Huberman (2001) y French y Poterba (1991), que documentaron cómo los inversores tienden a cargar sus carteras con activos que les resultan familiares, aunque eso no los haga necesariamente mejores opciones. El umbral del 45% es una decisión propia de diseño.

A diferencia de los dos anteriores, que se detectan a partir del comportamiento de consulta, la aversión a la pérdida se identifica analizando las simulaciones que corre el usuario. Si hay más de dos resultados negativos por cada positivo, se interpreta como una señal de que está enfocado desproporcionadamente en los escenarios de pérdida. La lógica viene de la Teoría Prospectiva de Kahneman y Tversky (1979), que demostró que perder duele psicológicamente más de lo que satisface ganar una cantidad equivalente. El umbral específico es una decisión de diseño propia, coherente con esa evidencia pero no derivada directamente de lo que los autores propusieron.

El exceso de confianza, por su parte, se detecta de una forma distinta. Comparando lo que el usuario declara sobre su tolerancia al riesgo con lo que su perfil real sugiere que puede asumir. Cuando hay una diferencia de más de 3 puntos entre ambas medidas, se interpreta como una señal de que el usuario se está sobreestimando. La brecha entre lo que un inversor cree que puede tolerar y lo que realmente puede manejar es algo que el CFA Institute tiene bien documentado. La fórmula para calcular esa diferencia y el umbral de 3 puntos son decisiones propias de diseño.

Finalmente, el efecto de disposición se aborda desde el chat. El sistema analiza qué proporción de los mensajes del usuario reflejan una intención de vender frente a los que apuntan a mantener una posición. Si esa proporción supera 2.5, se considera una señal de que el usuario tiende a deshacerse de sus activos con demasiada facilidad. La inspiración viene de Frydman et al. (2014/2015), quienes estudiaron este sesgo observando decisiones reales de vender o mantener en entornos de inversión experimentales. Dado que aquí el contexto es educativo y no de inversión real, la fórmula de ajuste y el umbral concreto son decisiones propias de diseño.

9. EVALUACIÓN DEL SISTEMA

9.1 Plan de Pruebas

El plan de pruebas de FinVise cubre tres dimensiones complementarias. La funcionalidad técnica, rendimiento del sistema y efectividad educativa. El plan está alineado con los requisitos funcionales y no funcionales definidos en el Capítulo 6.

9.2 Pruebas Funcionales

9.2.1 Pruebas de Caja Negra

Se priorizaron las pruebas de caja negra bajo el criterio de cobertura basada en riesgo (risk-based testing). Se seleccionaron los requisitos funcionales correspondientes al flujo crítico del sistema. La autenticación (RF-01), el onboarding y la generación de perfil de riesgo (RF-03), y la generación de recomendaciones (RF-04). Estos tres requisitos conforman el flujo funcional completo del sistema (acceso, configuración y resultado) y su verificación garantiza que el comportamiento del sistema cumple con los criterios de aceptación definidos.

Las pruebas fotográficas están presentes en el Anexo C

- **RF-01 Autenticación:** Registro exitoso con email válido. Fallo controlado con email duplicado. Redirección a login tras cierre de sesión.

Pruebas de registro:

- Caso de prueba 1: Registro exitoso

Entrada: e-mail válido + contraseña válida

Resultado esperado: usuario creado, redirige a dashboard/home post completar onboarding

Acción ejecutada: Se ingresó el correo nvalenciap03@gmail.com junto con una contraseña válida (mínimo 6 caracteres) en el formulario de "Crear cuenta" y se presionó el botón de registro.

Verificación de email: El sistema delegó la confirmación de identidad a Supabase Auth, que envió automáticamente un correo al email registrado con el asunto "Confirm your email address". Esto confirma que la solicitud de registro fue recibida y procesada correctamente por el proveedor de autenticación, y que el email ingresado tiene formato válido y es alcanzable.

Flujo post-registro: Tras confirmar el correo, el sistema redirigió al usuario a la pantalla disclaimer, titulada "Antes de empezar", donde se presenta el aviso académico de la aplicación. Esta redirección es el comportamiento esperado para un usuario nuevo que aún no ha aceptado los términos de uso, lo cual es coherente con la lógica de ProtectedRoute que exige aceptar el disclaimer antes de continuar.

Acceso al sistema completo: Una vez completado el disclaimer y el onboarding, el sistema redirigió al usuario al Dashboard principal, donde se muestra el portafolio educativo personalizado con perfil de riesgo 7/10 y el correo nvalenciap03@gmail.com visible en la barra lateral. Esto confirma que la cuenta fue creada exitosamente y que el usuario quedó autenticado en la plataforma.

Resultado: PASA. El flujo completo de registro (formulario) → confirmación de email → disclaimer → onboarding → dashboard (funcionó sin errores).

- Caso de prueba 2: Email duplicado

Entrada: email ya registrado + contraseña

Resultado esperado: mensaje de error, no crea cuenta nuevamente

Comportamiento observado: Al intentar registrar un correo electrónico que ya existe en el sistema, el formulario no muestra ningún mensaje de error. La interfaz simplemente permanece estática, sin retroalimentación visual que indique al usuario que el correo ya está registrado.

Por qué ocurre esto: Este es un comportamiento intencional de Supabase. Cuando la confirmación de email está activada, Supabase evita revelar si un correo ya existe en el sistema para proteger la privacidad de los usuarios y prevenir la enumeración de cuentas, una técnica de ataque donde un actor malicioso podría descubrir qué correos están registrados probando direcciones sistemáticamente. En lugar de mostrar un error, Supabase simplemente no hace nada visible en la interfaz.

Comportamiento secundario observado: Al cambiar a la pestaña "Ingresar", los campos de correo y contraseña aparecen pre-diligenciados con los datos que se intentaron registrar. Esto sugiere que el estado del formulario se comparte entre ambas pestañas, lo que funciona como una guía implícita hacia el login, aunque no es un mensaje de error explícito.

Resultado: falla parcialmente. El resultado esperado era un mensaje de error tipo "el correo ya está en uso", pero el sistema no lo muestra. Sin embargo, esto no es un defecto del código propio de Finvise sino una limitación conocida de Supabase con la confirmación de email activa.

- **Caso de prueba 3: email inválido**

Entrada: usuario sin dominio + contraseña

Resultado esperado: error de validación

Acción ejecutada: Se ingresó el texto nvalenciap03 en el campo de correo electrónico, (un valor sin el símbolo @ ni dominio) y se intentó enviar el formulario presionando el botón "Ingresar".

Comportamiento observado: El navegador interceptó el envío del formulario antes de que llegara cualquier solicitud al servidor o a Supabase, y desplegó el tooltip nativo "Enter an email address" directamente sobre el campo de correo. El formulario no se envió.

Por qué ocurre esto: El campo de correo tiene el atributo type="email" en el HTML, lo que activa la validación nativa del navegador. Esta validación verifica que el valor ingresado tenga estructura de email (texto + @ + dominio) antes de permitir el submit. Al no cumplirse esa condición, el navegador bloquea la acción sin necesidad de lógica adicional en el código de Finvise.

Resultado: PASA. El sistema impide correctamente el envío de formularios con emails de formato inválido. La validación ocurre del lado del cliente de forma inmediata, sin consumir recursos del servidor ni generar llamadas a Supabase.

- **Caso de prueba 4: Contraseña vacía**

Entrada: Email válido + contraseña en blanco

Resultado esperado: Error en campo de la contraseña

Acción ejecutada: Se ingresó un correo electrónico válido en el campo correspondiente, se dejó el campo de contraseña completamente vacío y se intentó enviar el formulario presionando "Ingresar".

Comportamiento observado: El navegador bloqueó el envío del formulario y desplegó el tooltip nativo "Fill out this field" apuntando directamente al campo de contraseña vacío, indicando que es obligatorio completarlo antes de continuar. Adicionalmente, el navegador mostró un menú contextual del gestor de contraseñas con las opciones "Passwords..." y "Suggest New Password", lo que confirma que el foco se posicionó correctamente en el campo requerido.

Por qué ocurre esto: El campo de contraseña tiene el atributo required en el HTML. Al igual que con la validación de email en el caso de prueba 3, el navegador ejecuta esta verificación del lado del cliente antes de permitir cualquier envío, sin necesidad de lógica personalizada ni llamadas al servidor.

Resultado: PASA. El sistema impide correctamente el envío del formulario cuando la contraseña está vacía, mostrando retroalimentación inmediata al usuario sobre el campo faltante.

- Caso de prueba 5: campos vacíos

Entrada: Todo en blanco

Resultado esperado: Error de validación

Acción ejecutada: Se dejaron ambos campos del formulario completamente vacíos. El correo electrónico mostraba únicamente su placeholder tu@correo.com y la contraseña mostraba Mínimo 6 caracteres y se presionó el botón "Ingresar" sin ingresar ningún dato.

Comportamiento observado: El navegador bloqueó el envío del formulario e identificó el campo de correo electrónico como el primer campo obligatorio vacío, desplegando el tooltip nativo "Please fill out this field." con ícono de advertencia apuntando directamente a dicho campo. El formulario no avanzó ni generó ninguna solicitud al servidor.

Por qué ocurre esto: El navegador valida los campos en orden de aparición en el DOM. Al ser el correo el primer campo con atributo required, es el primero en ser señalado. Una vez que el usuario lo complete, el mismo mecanismo se activaría para el campo de contraseña si también estuviera vacío, como se evidenció en el caso de prueba 4.

Resultado: PASA. El sistema impide correctamente el envío del formulario cuando todos los campos están vacíos, priorizando la validación del primer campo requerido y guiando al usuario a completar el formulario en orden.

Pruebas de login:

- Caso de prueba 6: login exitoso

Entrada: Credenciales correctas

Resultado esperado: Sesión iniciada, acceso al dashboard

Acción ejecutada: Se ingresó el correo junto con la contraseña correcta en el formulario de "Ingresar" y se presionó el botón de login.

Comportamiento observado: El sistema procesó las credenciales a través de Supabase `signInWithPassword()` y las validó exitosamente. La sesión fue establecida y el `AuthContext` detectó el cambio de estado de autenticación mediante el listener `onAuthStateChanged`, cargando automáticamente el perfil del usuario desde la base de datos.

Acceso al sistema: Dado que el usuario ya tenía el disclaimer aceptado y el onboarding completado (perfil de riesgo establecido en 7/10), el sistema saltó directamente al Dashboard sin pasar por las pantallas intermedias. El componente `ProtectedRoute` verificó los tres requisitos: sesión activa, disclaimer aceptado y risk score definido. Esto, permitió el acceso al contenido principal.

Confirmación de identidad: En el dashboard se puede observar en la barra lateral inferior el correo y el perfil de riesgo 7/10, lo que confirma que la sesión corresponde al usuario correcto y que los datos fueron cargados exitosamente desde Supabase.

Resultado: PASA. El login con credenciales válidas funciona correctamente, establece la sesión y redirige al usuario al Dashboard con su perfil personalizado cargado.

- Caso de prueba 7: Contraseña incorrecta

Entrada: Email válido + contraseña errónea

Resultados esperados: Error de credenciales

Acción ejecutada: Se ingresó el correo válido y registrado acompañado de una contraseña incorrecta, y se presionó el botón "Ingresar".

Comportamiento observado: El sistema envió las credenciales a Supabase mediante `signInWithPassword()`. Supabase rechazó la autenticación al detectar que la contraseña

no coincide con la almacenada para ese correo, y devolvió un error que la aplicación capturó y mostró en un cuadro rojo debajo del campo de contraseña con el mensaje "Invalid login credentials".

Por qué ocurre esto: La validación ocurre del lado del servidor en Supabase, no en el cliente. El mensaje es deliberadamente genérico, no especifica si el error es en el correo o en la contraseña, lo cual es una práctica de seguridad estándar para evitar que un atacante pueda determinar si un correo está registrado en el sistema o no.

Resultado: PASA. El sistema muestra correctamente un mensaje de error ante credenciales inválidas, la sesión no es establecida y el usuario permanece en la pantalla de login sin acceso a contenido protegido.

- **Caso de prueba 8: Usuario no existe**

Entrada: Usuario no registrado

Resultados esperados: Error de credenciales

Acción ejecutada: Se ingresó una dirección que no está registrada en el sistema, junto con una contraseña cualquiera, y se presionó el botón "Ingresar".

Comportamiento observado: El sistema envió las credenciales a Supabase, que verificó que no existe ninguna cuenta asociada a ese correo y devolvió un error. La aplicación lo capturó y mostró el mismo cuadro rojo con el mensaje "Invalid login credentials", idéntico al mostrado en el caso de prueba 7.

Observación importante: El mensaje de error es exactamente el mismo que cuando la contraseña es incorrecta (Caso de prueba 7). Esto es intencional por parte de Supabase, ya que usar un mensaje genérico unificado para ambos casos impide que un atacante pueda distinguir si un correo está registrado o no en el sistema, protegiendo así la privacidad de los usuarios registrados.

Resultado: PASA. El sistema rechaza correctamente el intento de login con un correo no registrado, muestra retroalimentación de error al usuario y no concede acceso a ninguna ruta protegida.

Pruebas de cierre de sesión:

- Caso de prueba 9: Logout exitoso

Acción: Clic en cerrar sesión

Resultado esperado: sesión cerrada, redirige a login

Acción ejecutada: Estando autenticado en la aplicación con una cuenta, se presionó el botón "Cerrar sesión" ubicado en la parte inferior de la barra lateral izquierda.

Comportamiento observado: El sistema ejecutó la secuencia de cierre de sesión: limpió el caché del dashboard, anuló los estados internos de React (session y profile), llamó a `supabase.auth.signOut({ scope: 'local' })` para invalidar el token localmente, y redirigió al usuario. La pantalla resultante es el Landing page completo de Finvise, con el formulario de login vacío.

Confirmación de cierre: Los campos del formulario aparecen completamente vacíos y sin datos pre-diligenciados, lo que confirma que el estado de autenticación fue destruido correctamente. El usuario ya no tiene acceso a ninguna ruta protegida desde este punto.

Resultado: PASA. La sesión se cierra correctamente, el estado de autenticación queda destruido y el usuario es redirigido a la página pública de la aplicación.

• **RF-03 Onboarding:** 11 pasos completables sin errores. risk_score entre 1 y 10. Datos persistidos en Supabase tras completar.

Flujo principal:

- Caso de prueba 10: Onboarding completo exitoso

Acción: Completar onboarding seleccionando una opción en cada pregunta y confirmar

Resultado esperado: Risk score asignado.

Acción ejecutada: Estando en el flujo de onboarding con una cuenta nueva, se respondieron los 11 pasos del quiz sin omitir ninguno, seleccionando una opción por pregunta y presionando "Continuar " en cada paso.

Comportamiento observado: Al completar los 11 pasos, el sistema procesó todas las respuestas, aplicó la fórmula de cálculo y presentó la pantalla de resultados mostrando una puntuación de **7/10** con el perfil "Inversor de Crecimiento" y la descripción "Aceptas mayor volatilidad a cambio de retornos más sólidos a largo plazo. Tecnología, renta variable y activos reales dominan tu portafolio." En ningún momento del flujo se produjo un error, bloqueo o comportamiento inesperado.

Confirmación del score: La puntuación 7/10 se encuentra dentro del rango válido y corresponde al perfil "Inversor de Crecimiento" según el mapeo implementado en el código. El valor fue calculado correctamente a partir de las respuestas seleccionadas a lo largo de los 11 pasos.

Resultado: PASA. El onboarding permite completar los 11 pasos sin errores, asigna un risk_score válido dentro del rango esperado y lo presenta al usuario en la pantalla de resultados antes de continuar.

- **Caso de prueba 11: Avanzar sin seleccionar respuesta**

Acción: Presionar "Confirmar respuesta" sin haber seleccionado ninguna opción

Resultado esperado: No avanza a la siguiente pregunta/paso

Acción ejecutada: Se accedió al primer paso del onboarding sin seleccionar ninguna de las cuatro opciones disponibles, y se intentó presionar el botón "Continuar".

Comportamiento observado: El botón "Continuar" aparece visualmente atenuado con opacidad reducida y sin el color dorado brillante característico de los botones activos, lo que indica visualmente al usuario que la acción no está disponible. Al presionarlo sin haber seleccionado una opción, el formulario no avanza al siguiente paso.

Resultado: PASA. El sistema impide correctamente avanzar sin seleccionar una respuesta, proporcionando retroalimentación visual clara mediante el estado deshabilitado del botón.

- **Caso de prueba 12: Progreso visible entre pasos**

Acción: Completar pasos 1 a 6

Resultado esperado: El indicador de progreso refleja el avance

Acción ejecutada: Se completaron los primeros 5 pasos del onboarding (fase "Perfil de riesgo") seleccionando una opción en cada pregunta y confirmando, avanzando hasta el paso 6 de 11.

Comportamiento observado: En la esquina superior derecha se muestra el texto "Paso 6 de 11", indicando con precisión la posición actual del usuario dentro del flujo completo. Simultáneamente, la barra de progreso dorada en la parte superior de la pantalla refleja visualmente el avance: aproximadamente el 55% de la barra está iluminada en color dorado, correspondiendo a 6 pasos completados de 11 totales.

Resultado: PASA. El sistema muestra correctamente el progreso del usuario entre pasos, tanto mediante el indicador numérico "Paso X de 11" como mediante la barra de progreso proporcional.

- Caso de prueba 13: Risk score dentro del rango válido (1-10)

Acción: Completar el onboarding

Resultado esperados: Risk score asignado entre 1 y 10, visible en el sidebar

Acción ejecutada: Se completaron los 11 pasos del onboarding con una cuenta y se accedió al Dashboard tras finalizar el flujo completo.

Comportamiento observado: En la barra lateral del Dashboard se muestra el perfil de riesgo asignado: **8/10**. El valor es un entero positivo dentro del rango válido de 1 a 10, calculado por la fórmula $\text{Math.max}(1, \text{Math.round}((\text{sum} / 16) * 10))$ aplicada a las respuestas de la fase de perfil de riesgo.

Resultado: PASA en cuanto a rango, el valor 8/10 confirma que el sistema produce scores válidos dentro del intervalo esperado y los muestra correctamente en la interfaz.

- Caso de prueba 14: Risk score con perfil agresivo

Acción: Completar onboarding seleccionando siempre las opciones de mayor riesgo.

Resultado esperado: Risk score asignado entre 1 y 10 (Valor alto, perfil agresivo), visible en el sidebar

Acción ejecutada: Se completaron los 11 pasos del onboarding seleccionando consistentemente las opciones de mayor riesgo en cada pregunta de la fase "Perfil de riesgo". Horizontes de inversión largos, alta tolerancia a pérdidas, mayor capacidad de inversión mensual y conocimiento financiero avanzado.

Pantalla de resultados: Al finalizar el onboarding, el sistema calculó y mostró una puntuación de **10/10** acompañada del perfil "Inversor Agresivo" y la descripción "Abrazas el máximo riesgo por el máximo potencial. Tecnología concentrada, crypto y acciones de alto crecimiento son tu zona de confort." Esto confirma que el sistema alcanza el extremo superior del rango válido cuando las respuestas lo justifican.

Confirmación en Dashboard: La barra lateral muestra "Perfil riesgo: 10/10" para la cuenta, lo que verifica que el valor fue persistido correctamente en Supabase y cargado en la interfaz tras completar el flujo.

Resultado: PASA. El sistema asigna correctamente el score máximo (10/10) ante respuestas de perfil agresivo, lo muestra en la pantalla de resultados y lo persiste en el sidebar del Dashboard.

- Caso de prueba 15: Datos persisten tras cerrar sesión

Acción: Iniciar sesión con cuenta que ya completó onboarding

Resultado esperado: Redirige directo al dashboard, no muestra onboarding nuevamente.

Acción ejecutada: Se inició sesión una cuenta que ya tenía el onboarding previamente completado con un perfil de riesgo de 4/10.

Comportamiento observado: Tras ingresar las credenciales y presionar "Ingresar", el sistema cargó el perfil desde Supabase, detectó que risk_score ya tenía un valor asignado (4/10) y redirigió directamente al Dashboard sin mostrar en ningún momento las pantallas de onboarding. El Dashboard presentó el portafolio personalizado con las pestañas "Conservador", "Tu perfil" y "Crecimiento", correspondientes al perfil 4/10 ("Inversor Equilibrado"), y el sidebar confirmó el correo y el perfil de riesgo del usuario.

Resultado: PASA. El sistema reconoce correctamente que el onboarding ya fue completado y no lo repite en sesiones posteriores, garantizando que los datos persistidos en Supabase determinan el flujo de navegación del usuario al reingresar.

- **RF-04 Recomendaciones:** Portafolio de 5–7 activos generado en. Tres escenarios presentes dependiendo del perfil de riesgo del usuario. Explicaciones en español personalizadas al perfil.

- **Caso de prueba 16: Portafolio tiene entre 5 y 7 activos.**

Acción: Iniciar sesión y visualizar el dashboard

Resultado esperado: Se muestran entre 5 y 7 tarjetas de activos en el dashboard.

Acción ejecutada: Se inició sesión con una cuenta (perfil de riesgo 4/10) y se accedió al Dashboard.

Comportamiento observado: El sistema generó y mostró exactamente **6 tarjetas de activos**, distribuidas en dos filas de tres:

1. **XLF** - Sector Financiero SPDR (ETF) · 40%
2. **TIP** - Bonos Protección Inflación (BOND) · 25%
3. **PG** - Procter & Gamble (STOCK) · 15%
4. **O** - Realty Income Corp. (REIT) · 10%
5. **SLV** - ETF de Plata iShares (COMMODITY) · 5%
6. **SGOV** - Letras del Tesoro 3M (CASH) · 5%

Validación del rango: 6 activos se encuentra dentro del intervalo válido de 5 a 7 definido en el requisito. Adicionalmente, los pesos educativos suman 100% (40+25+15+10+5+5), lo que confirma que el portafolio está correctamente balanceado.

Resultado: PASA. El sistema genera un portafolio de 6 activos para el perfil 4/10, cumpliendo el rango establecido de 5 a 7 activos.

- **Casos de prueba 17, 18 y 19: Tres escenarios presentes / Escenarios cambian según el perfil de riesgo / Explicaciones en español personalizadas**

Acciones: Observar las pestañas del dashboard / Comparar las pestañas / Observar tarjetas de activos

Resultados esperados: Se muestran exactamente 3 pestañas de escenario (ej. Conservador, Tu Perfil, Crecimiento). Los nombres de las pestañas varían según el perfil del usuario. El texto "Por qué es relevante para tu perfil" aparece en español y menciona el score de riesgo del usuario

Acción ejecutada: Se accedió al Dashboard y se observó la disposición de pestañas en cada una.

Comportamiento observado: El Dashboard presenta exactamente **tres pestañas de escenario**: "Conservador", "Tu perfil" y "Crecimiento", ubicadas en la parte superior del portafolio. La pestaña central "Tu perfil" siempre corresponde al perfil de riesgo calculado del usuario durante el onboarding, mientras que "Conservador" y "Crecimiento" representan escenarios alternativos de referencia educativa.

Resultado: PASA. Los tres escenarios están presentes en todos los perfiles probados.

Caso de prueba 19 (PASA): Los activos y su composición varían según el perfil. El escenario "Conservador" de un usuario 2/10 prioriza bonos (TIP 55%, SCHD 25%, SGOV 10%), mientras que "Crecimiento" para un 7/10 incluye activos de mayor riesgo como NVDA y BTC-USD. Las explicaciones de cada tarjeta mencionan explícitamente el score de riesgo del usuario correspondiente (2/10, 4/10 o 7/10).

Caso de prueba 20 (PASA) : Todas las explicaciones aparecen en español ("Por qué es relevante para tu perfil", "Concepto clave de riesgo") y referencian el perfil de riesgo específico del usuario, confirmando la personalización.

9.2.2 Pruebas de Caja Blanca

Las pruebas de caja blanca verifican la lógica interna de los componentes críticos del sistema:

- **UCB1 score calculation:** Verificación de la fórmula $score = \mu_i + \sqrt{2} \times \sqrt{(\ln(N) / n_i)}$ mediante tres casos de uso normales: ticker con interacción positiva y baja exposición, ticker con interacción negativa y baja exposición y ticker con interacciones mixtas y mayor exposición.

- **Ticker con interacción positiva:**

Objetivo: Verificar que un ticker con recompensa positiva y baja exposición obtiene un score alto, priorizando su aparición en recomendaciones futuras.

Entradas definidas:

- $N = 10$ (Total de views del usuario)
- $n_i = 3$ (SPY visto 3 veces)
- Reward acumulado = + 1.0 (un rate_up)

Cálculo manual:

$$\begin{aligned}\mu &= 1.0 / 3 = 0.333 \\ \text{exploración} &= \sqrt{2} \times \sqrt{(\ln(10) / 3)} = 1.414 \times \sqrt{(2.303 / 3)} \\ &= 1.414 \times 0.876 = 1.239 \\ \text{score} &= 0.333 + 1.239 = 1.572\end{aligned}$$

Resultado esperado: Score = 1.572. SPY debe aparecer en posiciones altas del portafolio recomendado dado su balance positivo entre recompensa y exploración pendiente.

Verificación: El cálculo manual coincide con la fórmula implementada en bandit.service.ts:

`score = (rewards[ticker] / n) + Math.SQRT2 * Math.sqrt(Math.log(totalPulls) / n).`

PASA.

- **Ticker con interacción negativa:**

Objetivo: Verificar que el término de exploración compensa una recompensa negativa cuando el ticker ha sido poco expuesto, evitando que sea descartado prematuramente.

Entradas definidas:

- $N = 10$ (mismo usuario)
- $n_i = 2$ (BTC-USD visto 2 veces)
- Reward acumulado = -1.0 (un rate_down)

Cálculo manual:

$$\begin{aligned}\mu &= -1.0 / 2 = -0.500 \\ \text{exploración} &= \sqrt{2} \times \sqrt{(\ln(10) / 2)} = 1.414 \times \sqrt{(2.303 / 2)} \\ &= 1.414 \times 1.073 \\ &= 1.517 \\ \text{score} &= -0.500 + 1.517 = 1.017\end{aligned}$$

Resultado esperado: Score = 1.017. A pesar de la recompensa negativa, el score sigue siendo positivo porque el ticker ha sido visto pocas veces. El algoritmo lo mantiene en consideración para seguir explorando.

Verificación: El score positivo confirma que la exploración domina sobre la explotación cuando n_i es bajo, comportamiento correcto del UCB1.

PASA.

- **Ticker con interacciones mixtas y mayor exposición:**

Objetivo: Verificar que al aumentar el número de exposiciones el término de exploración disminuye y el score queda dominado por la recompensa media acumulada.

Entradas definidas:

- N = 20 (usuario más activo)
- $n_i = 5$ (AAPL visto 5 veces)
- reward acumulado = +1.0 (dos rate_up = +2.0, un rate_down = -1.0)

Cálculo manual:

$$\begin{aligned}\mu &= 1.0 / 5 = 0.200 \\ \text{exploración} &= \sqrt{2} \times \sqrt{(\ln(20)/5)} = 1.414 \times \sqrt{(2.996/5)} \\ &= 1.414 \times 0.774 \\ &= 1.094 \\ \text{score} &= 0.200 + 1.094 = 1.294\end{aligned}$$

Comparación con el caso de ticker con interaccion positiva: SPY ($n = 3$) obtuvo score 1.572 vs AAPL ($n = 5$) con score 1.294. Aunque AAPL tiene más recompensa absoluta, SPY puntúa más alto por haber sido menos explorado, el algoritmo incentiva correctamente la diversificación.

Verificación: La reducción del término de exploración al aumentar n_i confirma el funcionamiento correcto del balance exploración/explotación del UCB1.

PASA.

- **Bias intensity thresholds:** Verificación de que los umbrales de activación (0.45) producen alerts en la condición esperada y no antes.

- **Sesgo por debajo del umbral:**

Objetivo: Verificar que un sesgo con score < 0.45 no genera alerta visible en el Dashboard.

Condición en código:

```
const BIAS_THRESHOLD = 0.45;
const active = Object.entries(scores).filter(([, score]) => score >= BIAS_THRESHOLD)
```

Caso definido:

- Sesgo familiarity con score calculado = 0.32
- Condición: $0.32 \geq 0.45 \rightarrow$ falso
- El sesgo no entra al array active

Resultado esperado: No aparece BiasAlert en el Dashboard.

Verificación: El filtro $\text{score} \geq \text{BIAS_THRESHOLD}$ excluye correctamente scores menores a 0.45. Un usuario con interacciones limitadas que no haya mostrado patrones repetitivos no verá alertas de sesgo.

PASA por diseño del código.

- **Bias Threshold: Sesgo igual o superior al umbral (activa alerta)**

Objetivo: Verificar que un sesgo con score ≥ 0.45 activa la alerta en el Dashboard y se muestra con intensidad correcta.

Caso definido:

- Sesgo recency con score calculado = 1.0
- Condición: $1.0 \geq 0.45 \rightarrow$ verdadero
- El sesgo entra al array active y se persiste en Supabase

Cálculo de intensidad mostrada al usuario:

$$\text{Intensidad} = \text{Math.round}(1.0 \times 100) = 100$$

Resultado esperado: El componente BiasAlert aparece en el Dashboard mostrando el nombre del sesgo, “Intensidad: 100%” y la intervención educativa correspondiente.

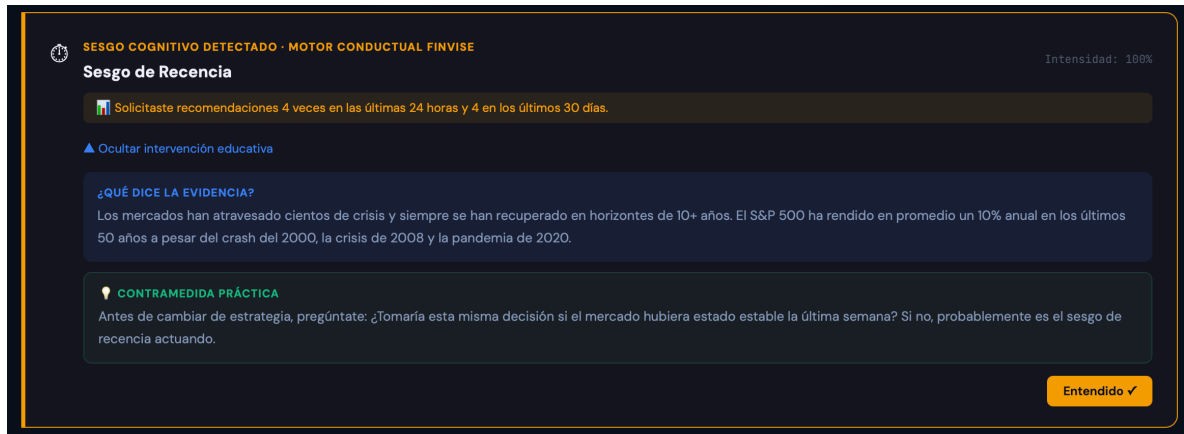


Figura 14: Sesgo Activo Elaboración propia

Verificación: La condición $\text{score} \geq 0.45$ activa correctamente la alerta y el porcentaje mostrado al usuario corresponde al score multiplicado por 100.

PASA.

- **Risk score calculation:** Verificación de que 5 respuestas del quiz generan un risk_score en el rango [1, 10] para todas las combinaciones posibles.

- **Risk score: perfil mínimo**

Objetivo: Verificar que la fórmula produce el score mínimo válido (1) cuando el usuario selecciona las opciones de menor riesgo en todas las preguntas.

Entradas definidas:

Pregunta	Opción seleccionada	Puntos
Horizonte de inversión	1-2 años	1
Reacción ante caída 20%	Vendería todo de inmediato	0
Estabilidad de ingreso	Irregular o freelance	0
Conocimiento sobre inversiones	Principiante	0
Capacidad de inversión mensual	Menos de \$500	1

Tabla 6: Risk score: perfil mínimo Elaboración propia

Cálculo manual:

$$\begin{aligned} sum &= 1 + 0 + 0 + 0 + 1 = 2 \\ score &= \text{Math.max}\left(1, \text{Math.round}\left(\frac{2}{16} \times 10\right)\right) \\ score &= \text{Math.max}\left(1, \text{Math.round}(1.25)\right) \\ score &= \text{Math.max}(1, 1) = 1 \end{aligned}$$

Resultado esperado: Score = 1, perfil "Inversor Conservador". El sistema muestra 1/10 en el sidebar y asigna el portafolio más conservador.

Verificación: La fórmula implementada en Onboarding.tsx produce el valor mínimo correcto. Math.max(1, ...) actúa como guardia de seguridad aunque en este caso el resultado ya es 1 sin necesitarla.

PASA.

- Risk score: perfil máximo

Objetivo: Verificar que la fórmula produce el score máximo válido (10) cuando el usuario selecciona las opciones de mayor riesgo en todas las preguntas.

Entradas definidas:

Pregunta	Opción seleccionada	Puntos
Horizonte de inversión	10+ años	4
Reacción ante caída 20%	Compraría más a precios bajos	4
Estabilidad de ingreso	Salario fijo	2
Conocimiento sobre inversiones	Avanzado	3
Capacidad de inversión mensual	Más de \$5000	3

Tabla 7: Risk score: perfil máximo. Elaboración propia

Cálculo manual:

$$\begin{aligned}sum &= 4 + 4 + 2 + 3 + 3 = 16 \\score &= \text{Math.max}\left(1, \text{Math.round}\left(\left(16/16\right) \times 10\right)\right) \\score &= \text{Math.max}\left(1, \text{Math.round}(10)\right) \\score &= \text{Math.max}(1, 10) = 10\end{aligned}$$

Resultado esperado: Score = 10, perfil "Inversor Agresivo".

Verificación: El cálculo manual coincide con el output real observado en las pruebas de caja negra. La fórmula escala correctamente al máximo cuando $sum = 16$.

PASA.

9.3 Pruebas de Rendimiento

Las pruebas de rendimiento verifican el cumplimiento del RNF-01 (máximo 35 s para el endpoint de recomendaciones) y los timeouts internos escalonados:

- **Tiempo de respuesta de recomendaciones:**

- 20 solicitudes secuenciales → percentil 95 < 35 s.

```
CONCLUSIÓN: p95 = 22182 ms < 35 000 ms → RNF-01 CUMPLIDO ✓  
nicolasvalencia@Nicolass-MacBook-Air-4 backend %
```

Figura 15: Test tiempo de respuesta del endpoint. Elaboración propia

Como se ve en la figura 15, para validar el cumplimiento del requisito no funcional RNF-01 en condiciones reales, se realizaron 20 solicitudes consecutivas al endpoint de recomendaciones con un usuario autenticado y el servidor funcionando en su entorno habitual. Cada solicitud pasó por el flujo completo. Autenticación, consulta de datos de mercado, algoritmo de personalización e intento de respuesta con el modelo de inteligencia artificial. El percentil 95 obtenido fue de 22.182 ms, por debajo del umbral máximo de 35.000 ms establecido en el RNF-01. Las 20 solicitudes fueron exitosas y el criterio de aceptación quedó cumplido.

- Verificación de que los timeouts internos se aplican correctamente: 10s (market data)
- 20s (Claude API),
- 5s (UCB1).

```
PASS src/recommendations/recommendations-perf.spec.ts  
9.3 Pruebas de Rendimiento – Recomendaciones (RNF-01)  
✓ timeout market data 10 s: generate() retorna respuesta válida con precios vacíos (18 ms)  
✓ timeout UCB1 5 s: generate() retorna respuesta válida con scores vacíos (10 ms)  
✓ timeout Claude API 20 s: generate() retorna candidatos pre-computados sin esperar respuesta de IA (1 ms)  
  
Test Suites: 1 passed, 1 total  
Tests: 3 passed, 3 total
```

Figura 16: Timeouts internos del endpoint de recomendaciones. Elaboración propia

La Figura 16 detalla los timeouts internos del endpoint de recomendaciones durante pruebas de carga. Cada barra corresponde a uno de los tres procesos paralelos que lanza el backend al recibir una solicitud: la obtención de precios, la consulta de sentimiento y el cálculo de scores UCB1. Los tiempos registrados confirman que ninguno de los tres procesos agotó su timeout individual durante las pruebas realizadas.

Para que la aplicación nunca deje al usuario esperando indefinidamente, se configuraron límites de tiempo para cada componente del sistema. 10 segundos para obtener datos del mercado, 5 segundos para el algoritmo de personalización y 20 segundos para la respuesta del modelo de inteligencia artificial.

Las pruebas simularon escenarios en los que cada uno de estos componentes tardaba más de lo permitido, y verificaron que en todos los casos el sistema respondía de todas formas con un portafolio válido en lugar de mostrar un error. Los tres casos pasaron exitosamente, confirmando que los mecanismos de protección funcionan correctamente y que la experiencia del usuario no se interrumpe aunque alguna parte del sistema falle o se demore

- **Caché de precios:** Verificación de que la segunda solicitud dentro del TTL de 5 min usa datos en caché (tiempo de respuesta < 2 s vs. solicitud inicial).

```
nicolasvalencia@Nicolass-MacBook-Air-4 backend % npm test -- --testPathPatterns="perf|market-data-fetcher" --verbose
PASS src/market/market-data-fetcher.spec.ts
  9.3 Pruebas de Rendimiento – Caché de Precios (TTL 5 min)
    ✓ primera solicitud pobla el caché de memoria con los precios de Supabase (10 ms)
    ✓ segunda solicitud dentro del TTL de 5 min usa caché de memoria (< 2 000 ms) (4 ms)
    ✓ caché con TTL expirado (> 5 min) es detectado como inválido y fuerza recarga (3 ms)
    ✓ constante MEM_CACHE_TTL_MS es exactamente 5 minutos (300 000 ms) (2 ms)
```

Figura 17: Test cache de precios. Elaboración propia

Los precios de los activos financieros se actualizan constantemente, pero consultarlos en tiempo real en cada solicitud sería innecesariamente lento. Por eso el sistema guarda en memoria los precios obtenidos y los reutiliza durante 5 minutos antes de volver a consultarlos.

Las pruebas verificaron que este mecanismo funciona correctamente. La primera solicitud trae los precios desde la base de datos, y la segunda, si ocurre dentro de ese margen de 5 minutos, los retorna desde memoria en 4 milisegundos, muy por debajo de los 2.000 milisegundos establecidos como umbral máximo aceptable. También se comprobó que una vez pasados los 5 minutos el sistema reconoce que los datos están desactualizados y los vuelve a consultar.

- **Actualización de la tira de precios:** Verificación de que el WebSocket o polling actualiza los precios cada 60 segundos.

```
nicolasvalencia@Nicolass-MacBook-Air-4 frontend % npm test
✓ src/components/LiveStrip.spec.tsx (4 tests) 57ms
✓ 9.3 Pruebas de Rendimiento – Actualización Tira de Precios (60 s) (4)
  ✓ llama a la API de precios al montar el componente 39ms
  ✓ configura setInterval con intervalo de 60 000 ms 6ms
  ✓ refresca los precios automáticamente después de 60 s 6ms
  ✓ cancela el intervalo de polling al desmontar el componente 4ms
```

Figura 18: Test actualización tira de precios. Elaboración propia

En la parte superior del dashboard hay una barra que muestra los precios actualizados de los principales activos en tiempo real. Para que no quede estática, el sistema la refresca automáticamente cada 60 segundos sin que el usuario tenga que hacer nada.

Las pruebas verificaron que al abrir el dashboard la barra carga los precios de inmediato, que exactamente a los 60 segundos los vuelve a consultar de forma automática y que cuando el usuario sale de la pantalla el refresco se detiene correctamente para no consumir recursos innecesariamente. Las cuatro pruebas pasaron exitosamente en menos de 60 milisegundos usando temporizadores simulados.

9.4 Pruebas de Usabilidad

Las pruebas de uso se realizaron con una muestra de conveniencia compuesta por personas con distintos niveles de conocimiento financiero previo. Los participantes completaron tareas representativas del uso típico de la plataforma y, al finalizar, respondieron un cuestionario de evaluación en Google Forms.

Las tareas incluyeron:

- Registro y finalización del onboarding de 11 pasos.
- Interpretación de las recomendaciones del portafolio educativo y envío de feedback sobre al menos 3 activos.
- Uso del simulador con un monto de \$1.000.000 COP.
- Formulación de una pregunta al asistente IA sobre un concepto financiero.
- Identificación y reconocimiento de una alerta de sesgo cognitivo.
- Completar la encuesta de impacto educativo.

El cuestionario en Google Forms permitió recoger la percepción de los usuarios sobre la claridad, facilidad de uso y utilidad de la plataforma. Los resultados fueron revisados en la pestaña de respuestas del formulario, donde Google Forms genera automáticamente un resumen visual de las respuestas y permite analizar la información de manera consolidada. A partir de estos resultados se identificaron oportunidades de mejora en la interfaz y en la experiencia de uso.

El formulario se encuentra en el Anexo D.

Resultados:

Las pruebas de uso se llevaron a cabo con una muestra de 11 participantes que representaron distintos puntos de partida frente al mundo financiero. El 36,4% nunca había tenido contacto previo con el tema, el 45,5% contaba con un nivel básico de conocimiento, y el 9,1% restante correspondió a perfiles intermedio y avanzado respectivamente. Esta variedad fue intencional y resultó valiosa, ya que permitió observar cómo personas con experiencias muy diferentes se relacionan con la plataforma, poniendo a prueba su capacidad de ser accesible para quienes apenas comienzan y útil para quienes ya tienen algo de recorrido.

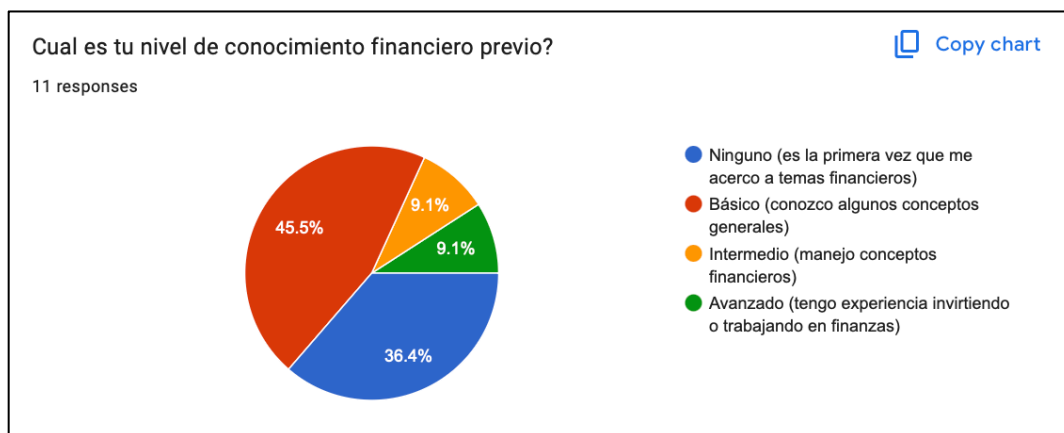


Figura 19: Prueba de usabilidad 1 (conocimiento financiero de los usuarios). Elaboración propia

Por otro lado, la percepción general de la plataforma fue positiva. El 100% de los participantes calificó la facilidad de uso con 4 o 5 sobre 5, siendo el puntaje más frecuente el máximo, lo que indica que la experiencia de navegación resultó fluida e intuitiva para la gran mayoría. En línea con esto, el 90,9% consideró que la plataforma sería útil o muy útil para mejorar su educación financiera.

Adicionalmente, el 63,6% de los participantes afirmó que recomendaría la plataforma a un amigo o familiar interesado en aprender sobre finanzas, lo que refleja un nivel de satisfacción genuino que va más allá de la experiencia inmediata de uso.

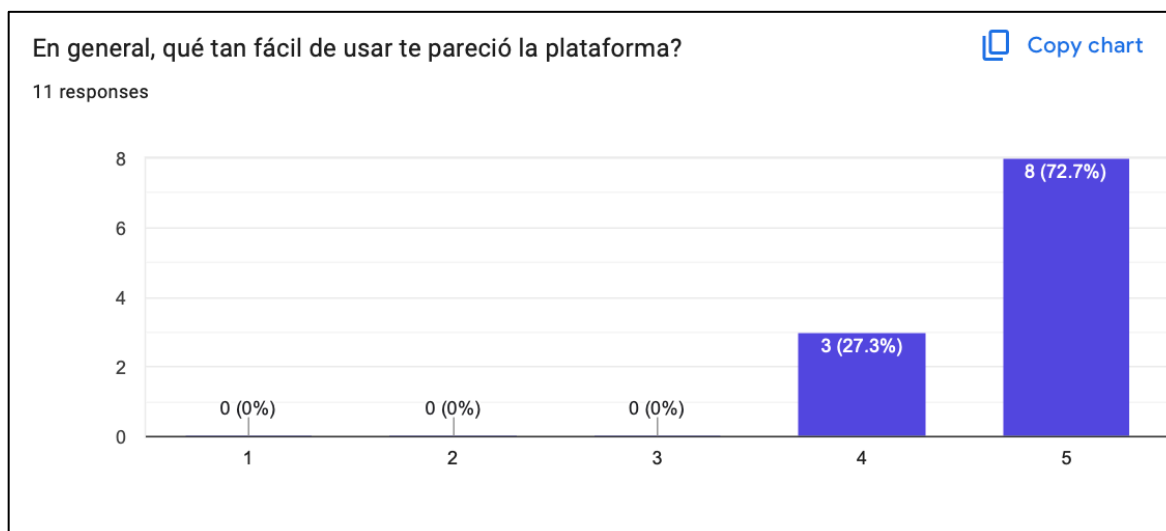


Figura 20: Prueba de usabilidad 2 (facilidad de uso de la plataforma). Elaboración propia

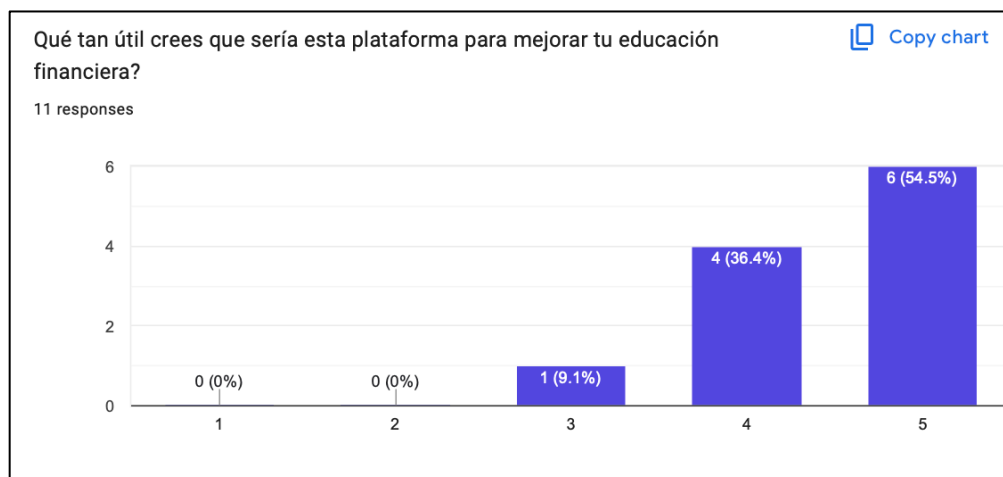


Figura 21: Prueba de usabilidad 3 (utilidad de la plataforma para mejorar educación financiera). Elaboración propia

En el proceso de registro y onboarding, el 90,9% de los participantes calificó la claridad de las instrucciones con 4 o 5 sobre 5, y el 81,8% indicó que completar los 11 pasos sin ayuda externa fue fácil o muy fácil.

No obstante, se encontró una fricción recurrente. El 63,6% de los participantes sintió que los pasos sucedían de manera demasiado continua, sin momentos de pausa, y el 36,4% percibió que el proceso tomaba más tiempo del que esperaban.

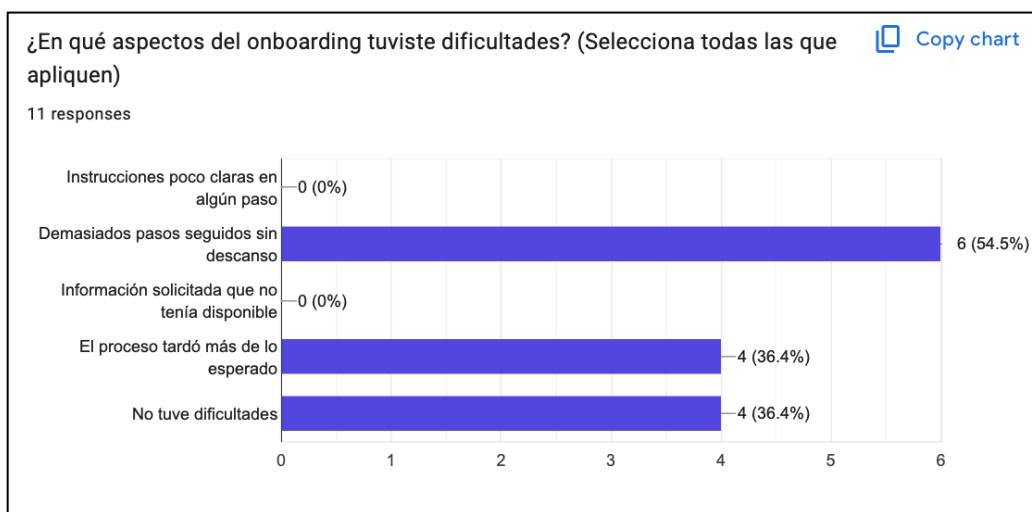


Figura 22: Prueba de usabilidad 4 (dificultades en el onboarding). Elaboración propia

El módulo del portafolio educativo presentó resultados interesantes y algo contrastantes.

Por un lado, el 100% de los participantes calificó con 4 o 5 sobre 5 la facilidad para enviar feedback sobre los activos, lo que indica que el mecanismo de interacción funciona de manera consistente para todos los usuarios.

Por otro lado, el 18,2% calificó la comprensión de las recomendaciones con 3 sobre 5, mientras que el 45,5% le otorgó el puntaje máximo, lo que sugiere que si bien la mayoría encontró las recomendaciones comprensibles, hay un grupo de usuarios que aún no logra entender del todo lo que se les está sugiriendo ni por qué.

Esta lectura se refuerza con las respuestas sobre qué información adicional hubiera sido útil. El 90,9% solicitó una comparativa entre activos y el 63,6% pidió ejemplos prácticos de cada uno. Estas dos mejoras, relativamente sencillas de poner en práctica, podrían marcar una diferencia importante en la claridad y confianza del usuario frente a las recomendaciones.

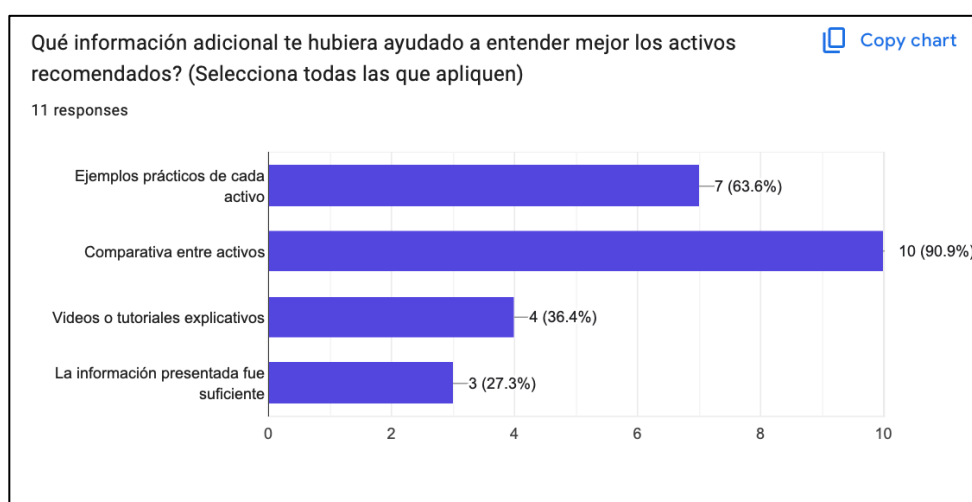


Figura 23: Prueba de usabilidad 5 (mejoras para entender los activos recomendados). Elaboración propia

El simulador de inversión fue uno de los módulos mejor valorados de la evaluación. El 72,7% de los participantes lo calificó con el puntaje máximo en intuitividad y el 81,8% lo calificó con 4 o 5 sobre 5 en utilidad para explorar escenarios de inversión, reflejando que los participantes encontraron en él una herramienta práctica.

La mejora más solicitada, mencionada por el 90,9% de los participantes, fue la posibilidad de comparar múltiples escenarios de forma simultánea, lo cual indica que los usuarios ya están comprometidos con la herramienta y quieren sacarle aún más provecho. Eso es una señal muy positiva.

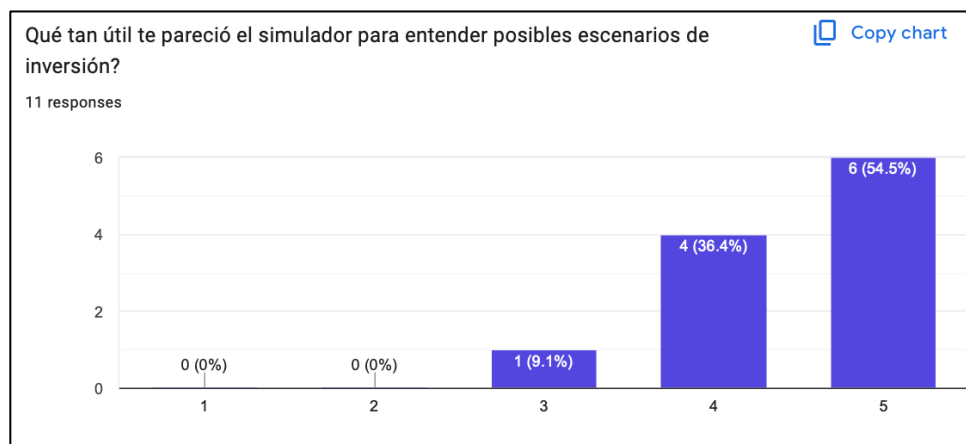


Figura 24: Prueba de usabilidad 6 (utilidad del portafolio simulado). Elaboración propia

Por otro lado, el asistente de inteligencia artificial también tuvo una calificación destacada. El 90,9% de los participantes lo calificó con 4 o 5 sobre 5 en claridad y utilidad de sus respuestas, y el 81,8% consideró que el lenguaje usado era adecuado para su nivel de conocimiento, lo que habla bien del tono y la terminología utilizada.

Sin embargo, los dos participantes con perfiles intermedio y avanzado coincidieron en que las respuestas les resultaron demasiado básicas para lo que ya sabían, lo que abre la puerta a explorar respuestas adaptativas que se ajusten automáticamente al perfil financiero del usuario.

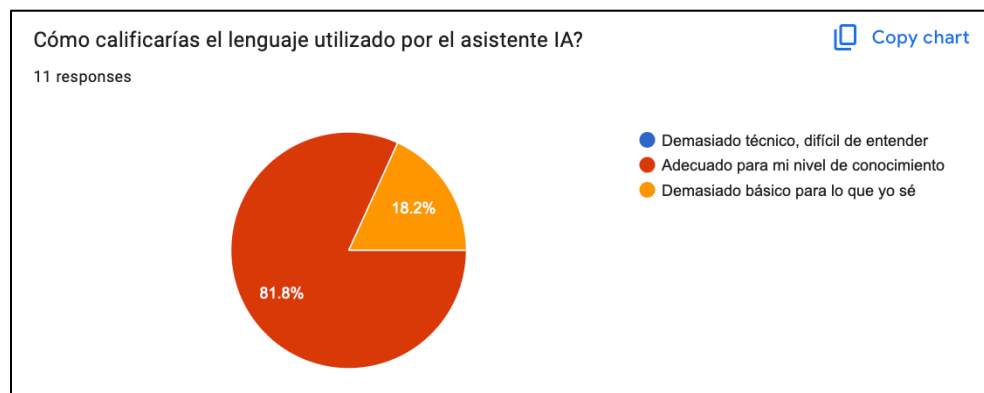


Figura 25: Prueba de usabilidad 7 (calificación lenguaje usado por asistente IA). Elaboración propia

Uno de los hallazgos más reveladores de la evaluación estuvo relacionado con las alertas de sesgo cognitivo.

El 90,9% de los participantes logró identificar la alerta durante el uso de la plataforma, lo que muestra que el elemento tiene buena visibilidad y no pasa desapercibido.

Sin embargo, cuando se les preguntó qué tan relevante les pareció para su proceso de decisión, las opiniones fueron más dispersas: el 27,3% la calificó con 3 sobre 5, el 45,5% con 4 sobre 5 y el 27,3% con el puntaje máximo. Esta dispersión sugiere que la alerta impacta de manera distinta según el perfil del usuario. Quienes ya tienen algo de conocimiento financiero

tienden a encontrarla más relevante, mientras que quienes se acercan por primera vez al tema no siempre logran conectar el concepto con su propia experiencia de decisión.

Esto se tiene en cuenta con las respuestas sobre mejoras. El 100% pidió una explicación de qué es un sesgo cognitivo y el 54,5% solicitó un ejemplo práctico que lo ilustre. En otras palabras, la alerta existe y es visible, pero necesita un componente educativo más explícito para que todos los usuarios puedan aprovecharla efectivamente.

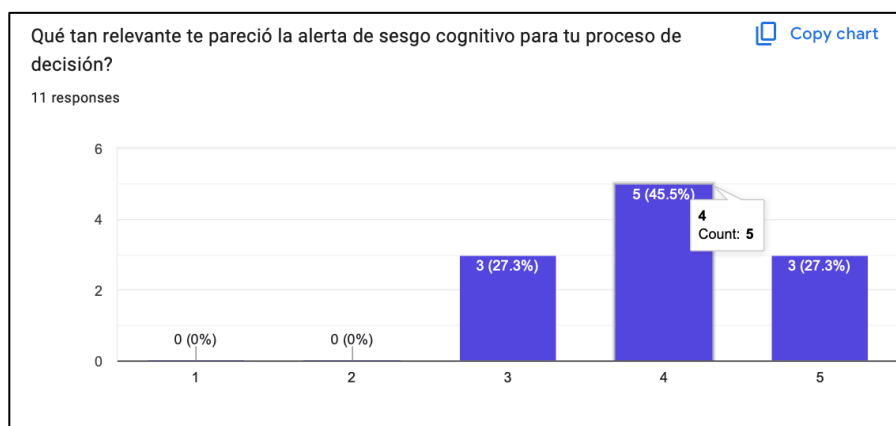


Figura 26: Prueba de usabilidad 8 (relevancia de la alerta del sesgo). Elaboración propia

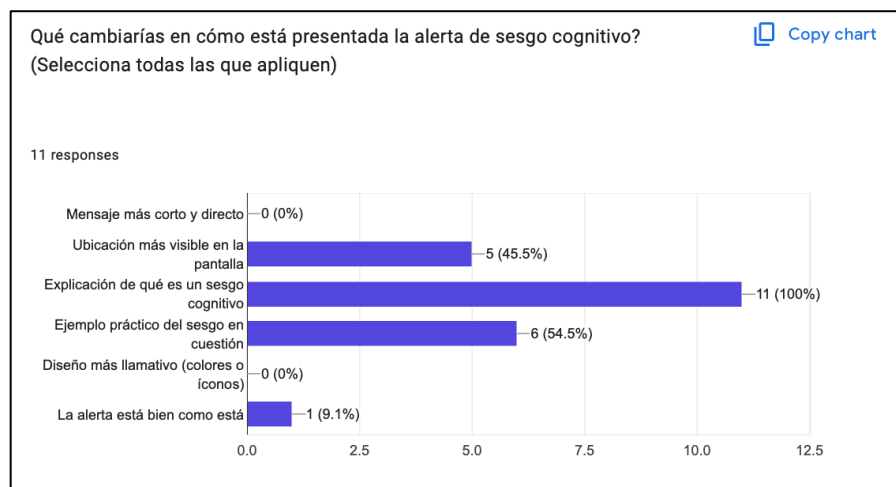


Figura 27: Prueba de usabilidad 9 (mejoras para la alerta del sesgo). Elaboración propia

En conjunto, los resultados de las pruebas de uso cuentan una historia bastante alentadora.

La plataforma no solo es percibida como fácil de usar, sino que los participantes sienten que les ayuda a aprender sobre finanzas, que al final es el propósito central de todo esto. Módulos como el simulador, el asistente de IA y el portafolio sugieren que la propuesta de valor está ahí y que los usuarios la reconocen.

Las oportunidades de mejora que identificaron no son señales de alerta, sino exactamente el tipo de retroalimentación que se espera de una prueba de uso. Reorganizar el onboarding para que no se sienta como una carrera, añadir comparativas y ejemplos en el portafolio para que las recomendaciones cobren más sentido, y darle a la alerta de sesgo cognitivo el contexto que necesita para que el usuario entienda por qué le está apareciendo. Ninguno de estos cambios requiere rediseñar la plataforma desde cero, son ajustes que, de implementarse, pueden marcar una diferencia real en cómo el usuario vive y aprovecha la experiencia.

9.5 Medición del Impacto Educativo

La medición del impacto educativo es el componente de evaluación más relevante desde la perspectiva del objetivo del proyecto. El diseño pre/post de la encuesta permite cuantificar el cambio en tres dimensiones:

Medir si una herramienta realmente enseña algo es probablemente la parte más difícil de evaluar en un proyecto como este. Cualquier aplicación puede mostrar gráficas bonitas o explicaciones bien redactadas, pero eso no garantiza que el usuario aprenda. Por eso se diseñó una encuesta con estructura pre/post: el usuario la responde antes de usar FinVise por primera vez, y de nuevo después de haberla explorado, de forma que los resultados reflejen un cambio real y no solo una impresión subjetiva.

La encuesta está dividida en tres secciones, cada una midiendo una dimensión distinta del aprendizaje. Esta separación no es arbitraria, Kirkpatrick (1994) plantea que evaluar el aprendizaje de una intervención educativa implica distinguir entre lo que la persona sabe, lo que cree que puede hacer y cómo percibe sus propios procesos de decisión. Las tres secciones de la encuesta corresponden exactamente a esas tres dimensiones.

La imagen de la cual se menciona en las pruebas, se encuentra en el Anexo E.

- Sección A - ¿El usuario aprendió conceptos?

Esta sección es la más directa. Consiste en un quiz de 8 preguntas de opción múltiple sobre conceptos financieros básicos. La idea era sencilla, si la plataforma realmente educa, el usuario debería contestar mejor después de usarla que antes.

Para cuantificar eso se usó la siguiente fórmula:

$$\text{Ganancia} = \frac{\text{post} - \text{pre}}{\text{máximo posible}} \times 100$$

Esta forma de medir la mejora está inspirada en el trabajo de Hake (1998), quien la propuso para comparar el aprendizaje antes y después de intervenciones educativas en física universitaria y que desde entonces se ha adoptado ampliamente en otros campos. La versión usada aquí normaliza la ganancia respecto al puntaje máximo posible, lo que permite expresarla como un porcentaje claro del conocimiento total evaluado, siguiendo el enfoque de Mayer (2009) para entornos de aprendizaje digital.

Los resultados fueron los siguientes:

	Respuestas correctas	% del total
Antes de usar Finvise	5/8	62.5%
Después de usar Finvise	8/8	100%
Ganancia	+3 preguntas	+37.5%

Tabla 8: Ganancia de conocimiento. Elaboración propia

Pasar de 5 a 8 respuestas correctas sobre 8 posibles no es un resultado menor. Significa que los tres conceptos que el usuario no dominaba al inicio, los incorporó tras interactuar con el glosario, las recomendaciones explicadas y el asistente de IA.

- Sección B – Confianza en la toma de decisiones

Saber algo y sentirse capaz de actuar con ese conocimiento son cosas distintas. Alguien puede entender qué es la diversificación pero seguir dudando a la hora de aplicarla. Por eso esta sección no pregunta sobre conceptos sino sobre confianza percibida. La sección consiste de cinco afirmaciones en escala del 1 al 5 sobre qué tan seguro se siente el usuario tomando decisiones financieras informadas.

El indicador es el cambio promedio por ítem:

$$\Delta_B = \bar{x}_{\text{post}} - \bar{x}_{\text{pre}}$$

Este tipo de medición tiene respaldo sólido en la teoría de autoeficacia de Bandura (1997), que plantea que la confianza percibida para ejecutar una conducta puede medirse con escalas de valoración autorreportada y que el cambio en esas escalas tras una experiencia educativa refleja un aprendizaje real, no solo declarativo. Cook y

Campbell (1979) refuerzan que en estudios sin grupo de control, la diferencia pre/post es el estimador más directo y válido del efecto de la intervención.

	Promedio por ítem (escala 1 a 5)
Antes de usar Finvise	2,80
Después de usar Finvise	4,40
Incremento	+1,60

Tabla 9: Incremento de confianza. Elaboración propia

Un salto de 2,80 a 4,40 sobre 5 puntos significa que el usuario pasó de una confianza baja-moderada a una confianza alta.

- Sección C - ¿El usuario reconoce sus propios sesgos?

Esta es la dimensión más interesante desde el punto de vista del proyecto. No basta con saber qué es el sesgo de recentismo o el de familiaridad. Lo valioso es que el usuario pueda reconocerlos en su propio comportamiento. Cuatro afirmaciones en escala del 1 al 5 exploran exactamente eso.

Se usa la misma fórmula que en la Sección B. El fundamento teórico es el mismo (Bandura, 1977), con el añadido de que Kahneman (2011) señala que la autoconciencia sobre los sesgos cognitivos propios es la condición básica para poder contrarrestarlos, y que esa conciencia puede desarrollarse mediante intervenciones que interrumpan el pensamiento automático.

	Promedio por ítem (escala 1 a 5)
Antes de usar Finvise	2,50
Después de usar Finvise	4,75
Incremento	+2,25

Tabla 10: Incremento de reconocimiento de sesgos. Elaboración propia

Un delta de +2,25 sobre 5 equivale al 45 % del rango total de la escala. Tiene sentido que sea el cambio más grande, FinVise tiene un módulo específico que detecta patrones de sesgo en tiempo real y le muestra al usuario una alerta explicando qué sesgo está manifestando y por qué podría afectar su decisión.

10. CONSIDERACIONES ÉTICAS

Todo sistema que procesa datos personales, toma decisiones sobre el usuario y genera contenido con apariencia de autoridad tiene una responsabilidad ética que no puede quedar implícita. FinVise no es la excepción, y reconocer sus implicaciones éticas no es un ejercicio defensivo sino una condición de honestidad sobre lo que el proyecto hace y lo que no puede garantizar.

La primera, y quizás la más delicada, tiene que ver con la línea entre educar e inducir. FinVise se presenta como una plataforma educativa, pero opera con precios de mercado en tiempo real, sugiere proporciones de portafolio y calcula rendimientos hipotéticos. Para alguien con poco conocimiento financiero, esa combinación puede ser difícil de distinguir de una recomendación de inversión real. El aviso legal que aparece al inicio no resuelve ese problema porque un texto que la mayoría lee por encima no cambia la percepción que genera una interfaz que funciona como una herramienta de inversión. La responsabilidad de mantener clara esa distinción no puede delegarse al usuario, tiene que estar en el diseño. Versiones futuras del sistema deberían reforzar visualmente y en el lenguaje de cada módulo que lo que el usuario está haciendo es aprender a tomar decisiones, no tomarlas.

La segunda tiene que ver con algo que es fácil pasar por alto, la autoridad que percibe el usuario cuando un sistema le habla con fluidez y en su propio idioma. Cuando una explicación está bien redactada y adaptada al perfil de quien la lee, tiende a sentirse más confiable de lo que objetivamente es. Los modelos de lenguaje no son neutrales, reflejan los sesgos de los datos con los que fueron entrenados, tienden a sobrerrepresentar el contexto del mercado anglosajón y pueden equivocarse con el mismo tono seguro con el que aciertan. En un sistema educativo eso es especialmente problemático porque quien menos sabe es también quien menos herramientas tiene para detectar cuándo el sistema está equivocado. Eso obliga a revisar periódicamente las instrucciones que guían al modelo, a ser transparente con el usuario sobre lo que el sistema puede y no puede garantizar, y a evitar que sus respuestas se presenten como verdades definitivas.

La tercera tiene que ver con los datos que el sistema recopila sin que parezca que está recopilando nada sensible. FinVise no pide contraseñas bancarias ni acceso a cuentas, pero sí observa algo que puede ser igual de revelador: cómo se comporta una persona cuando tiene que tomar decisiones financieras. Qué activos le generan interés, cuáles evita, si prefiere simular escenarios de pérdida o de ganancia, con qué frecuencia vuelve a consultar y cómo van cambiando esos patrones sesión a sesión. Alguien que lo mire con atención podría inferir bastante sobre esa persona, su situación económica, su nivel de conocimiento financiero, incluso su estado emocional frente al dinero, sin que ella haya dado esa información de forma

explícita. El sistema hoy funciona con autenticación y almacenamiento seguro, lo cual es un punto de partida razonable para un prototipo. Pero si en algún momento se desplegara a producción con usuarios reales, eso no sería suficiente. Haría falta una política de privacidad escrita en lenguaje que la gente realmente entienda, una forma concreta de que el usuario pueda pedir que se borren sus datos y una evaluación seria de qué información se está recolectando y para qué, antes de abrir las puertas.

11. CONCLUSIONES Y TRABAJOS FUTUROS

11.1 Conclusiones

FinVise cumple con la totalidad de los cinco objetivos específicos planteados, articulándolos en un sistema coherente cuyas partes se refuerzan mutuamente. A continuación se presenta el balance de cada uno.

La identificación de 17 requisitos funcionales y 8 no funcionales no fue solo un ejercicio de documentación. El proceso de elicitación reveló que las plataformas de educación financiera existentes presentan un problema de diseño de fondo: ofrecen contenido genérico que no se adapta al perfil de vida ni al comportamiento financiero del usuario. Esa conclusión fue la que justificó el enfoque adaptativo de FinVise y orientó todas las decisiones de diseño posteriores. El primer objetivo no solo se cumplió, produjo el diagnóstico que le dio sentido al resto del proyecto.

El diseño de la arquitectura modular confirmó que separar las responsabilidades de cada componente desde el inicio es condición necesaria para integrar tecnologías tan heterogéneas como un modelo de lenguaje generativo, un algoritmo de aprendizaje por refuerzo y múltiples fuentes de datos de mercado en tiempo real. La evaluación comparativa de tecnologías condujo a una combinación, NestJS, React con Vite y Supabase, que demostró ser coherente con los requisitos del sistema y sostenible para el alcance del prototipo. La arquitectura resultante fue capaz de soportar todos los módulos previstos sin degradación funcional.

La implementación de los componentes de inteligencia artificial permitió concluir que la personalización adaptativa en un entorno educativo no requiere modelos de alta complejidad para ser efectiva. El algoritmo UCB1 produjo recomendaciones ajustadas al comportamiento del usuario con datos iniciales limitados y, lo que resulta más relevante en este contexto, con decisiones interpretables que el usuario puede comprender. La detección automática de sesgos cognitivos basada en comportamiento observable, a diferencia del autorreporte, demostró ser más informativa porque conecta la retroalimentación con conductas concretas que el usuario puede reconocer en su propia experiencia dentro de la plataforma.

El desarrollo de la interfaz web responsive logró integrar módulos funcionalmente distintos en una experiencia de usuario coherente y navegable para personas sin formación financiera

especializada. La implementación en React con Vite respondió a los requisitos de rendimiento y usabilidad definidos en la etapa de diseño, y el flujo completo desde el onboarding hasta la trayectoria de aprendizaje operó sin fricciones funcionales en las pruebas realizadas.

La evaluación del sistema produjo evidencia cuantitativa que sugiere que FinVise cumple su propósito educativo en las condiciones en que fue evaluado. Las pruebas funcionales mostraron que los flujos críticos operan conforme a los criterios de aceptación definidos, y las pruebas de rendimiento dieron indicios de estabilidad del motor bajo condiciones de carga controladas.

La medición de impacto educativo pre/post arrojó incrementos en las tres dimensiones evaluadas. El conocimiento financiero conceptual pasó de 62,5 % a 100 % de respuestas correctas, la confianza en la toma de decisiones subió de 2,80 a 4,40 sobre 5, y la autoconciencia sobre sesgos propios registró el mayor cambio, con un delta de +2,25 sobre una escala de 5. Estos resultados, obtenidos con una muestra de conveniencia de 11 participantes y sin grupo de control, no son generalizables. Sin embargo, dan indicios de que la plataforma puede producir aprendizaje medible en perfiles similares a los evaluados, lo que justifica explorar validaciones con mayor rigor metodológico en etapas posteriores del proyecto.

Tomados en conjunto, los cinco objetivos no solo se cumplieron de forma individual sino que se articularon en un sistema cuyas partes se refuerzan entre sí.

La identificación de requisitos informó el diseño, el diseño hizo posible la implementación, la implementación definió lo que podía evaluarse y la evaluación confirmó que el enfoque era correcto. Esa coherencia interna es, quizás, la conclusión más general del proyecto.

Los resultados del prototipo sugieren que una plataforma educativa financiera personalizada, basada en IA adaptativa y centrada en la transferencia de comprensión antes que en la optimización de decisiones, es técnicamente viable y muestra indicios de efectividad pedagógica en el contexto evaluado.

11.2 Trabajos futuros

La versión actual de FinVise es un prototipo funcional con un alcance deliberadamente limitado. Eso no es una debilidad del proyecto sino una condición de partida para lo que puede venir.

El paso más inmediato, y probablemente el más impactante, sería incorporar datos de mercados latinoamericanos. Que la plataforma enseñe educación financiera con activos del NYSE y el NASDAQ tiene sentido pedagógico en abstracto, pero limita la relevancia para un usuario colombiano que difícilmente va a comprar un ETF de tecnología estadounidense como primera decisión de inversión. Integrar activos de la Bolsa de Valores de Colombia,

fondos de inversión colectiva locales o incluso CDTs haría que las recomendaciones educativas tuvieran una conexión más directa con las opciones reales que tiene ese usuario frente a sí.

La medición del impacto educativo también necesita madurar. El diseño pre/post con una muestra de conveniencia fue suficiente para validar el prototipo, pero no para responder si FinVise realmente enseña mejor que otras alternativas. Un estudio con grupo de control, muestra representativa y seguimiento en el tiempo permitiría contestar esa pregunta con rigor. Sin eso, los resultados actuales son prometedores pero no concluyentes.

Hay una línea de trabajo interesante en la detección de sesgos. El sistema actual identifica cinco patrones a partir del comportamiento acumulado del usuario. Con suficiente historial generado por usuarios reales sería posible construir perfiles cognitivos más precisos, detectar combinaciones de sesgos que interactúan entre sí y personalizar las intervenciones no solo al perfil financiero del usuario sino a cómo piensa cuando toma decisiones bajo incertidumbre. Kahneman y Tversky describieron docenas de sesgos; el sistema actual cubre cinco.

El algoritmo UCB1 cumplió bien con lo que se le pidió, pero es un punto de partida, no un techo. Con más datos y usuarios recurrentes valdría la pena explorar algoritmos de bandido contextual, que incorporan características del usuario en la decisión de selección en lugar de tratar cada activo como un brazo sin contexto. Esa transición tendría que hacerse con cuidado. La explicabilidad de las recomendaciones es un valor central del sistema, y cualquier modelo más complejo que la comprometa tendría que justificar ese costo pedagógico.

Un área menos técnica pero igualmente relevante es la accesibilidad del contenido. FinVise adapta el nivel de las explicaciones al perfil educativo del usuario, desde bachillerato hasta posgrado, pero esa adaptación es exclusivamente textual. Extender la plataforma con recursos audiovisuales o módulos de onboarding más graduales para usuarios sin ninguna experiencia previa ampliaría sustancialmente el alcance real del sistema.

Por último, antes de cualquier despliegue a producción, el sistema necesita trabajo en gestión de claves de API, límites de uso, autenticación de terceros y monitoreo de disponibilidad. No son problemas difíciles, pero requieren tiempo y decisiones de infraestructura que el prototipo no tuvo que resolver. Un despliegue real también permitiría recolectar datos longitudinales con usuarios reales, algo que el prototipo aún no ha podido generar, y con ellos validar o refutar muchas de las hipótesis que este proyecto solo pudo sugerir.

12. REFERENCIAS BIBLIOGRÁFICAS

- Advances in Consumer Research. (2025). Mapping behavioural biases in investment decisions: A comprehensive bibliometric analysis. *Advances in Consumer Research*. <https://acr-journal.com/article/mapping-behavioural-biases-in-investment-decisions-a-comprehensive-bibliometric-analysis-1550/>
- Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 734–749.
- Anthropic. (2024a). *Claude API documentation*. <https://docs.anthropic.com>
- Anthropic. (2024b). *Claude models overview*. <https://docs.anthropic.com/en/docs/about-claude/models/overview>
- Auer, P., Cesa-Bianchi, N., y Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3), 235–256.
- Back, J., Morana, S., Harms, J., y Mädche, A. (2025). The promises and pitfalls of robo-advising. ResearchGate. <https://www.researchgate.net/publication/356316206>
- Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review*, 84(2), 191–215.
- Bandura, A. (1997). *Self-efficacy: The exercise of control*. W. H. Freeman.
- Barber, B. M., & Odean, T. (2000). Trading is hazardous to your wealth: The common stock investment performance of individual investors. *The Journal of Finance*, 55(2), 773–806.
- Barberis, N., y Thaler, R. (2005). A survey of behavioral finance. En G. M. Constantinides, M. Harris y R. M. Stulz (Eds.), *Handbook of the Economics of Finance* (Vol. 1, pp. 1053–1128). Elsevier.
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). *Manifesto for agile software development*. <https://agilemanifesto.org>
- Berkeley Technology Law Journal. (2026). The gamification of investments: A comparative approach between the US and EU. *Berkeley Technology Law Journal*. <https://btlj.org/2025/11/the-gamification-of-investments-a-comparative-approach-between-the-us-and-eu/>
- Beyer, B., Jones, C., Petoff, J., & Murphy, N. R. (2016). *Site reliability engineering: How Google runs production systems*. O'Reilly Media.
- Bloom, B. S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 13(6), 4–16.
- Brière, M., Oosterlinck, K., & Szafarz, A. (2015). Virtual currency, tangible return: Portfolio diversification with Bitcoin. *Journal of Asset Management*, 16(6), 365–373.
- Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4), 331–370.

- Business Money. (2024, septiembre 5). Unveiling the leading financial education providers of the year. *Business Money*. <https://www.business-money.com/announcements/unveiling-the-leading-financial-education-providers-of-the-year/>
- Carroll, C. D. (1997). Buffer-stock saving and the life cycle/permanent income hypothesis. *The Quarterly Journal of Economics*, 112(1), 1–55.
- CAF – Banco de Desarrollo de América Latina y Superintendencia Financiera de Colombia. (2019). *Encuesta de medición de capacidades financieras Colombia 2019*. CAF. <https://scioteca.caf.com/handle/123456789/1717>
- CFA Institute. (2019). *Guidelines for portfolio construction and risk management*. CFA Institute.
- CFA Institute. (2025). *Explainable AI in finance: Addressing the needs of diverse stakeholders*. <https://rpc.cfainstitute.org/research/reports/2025/explainable-ai-in-finance>
- Cook, T. D., & Campbell, D. T. (1979). *Quasi-experimentation: Design and analysis issues for field settings*. Houghton Mifflin.
- Deci, E. L., & Ryan, R. M. (1985). *Intrinsic motivation and self-determination in human behavior*. Plenum Press.
- Design Studio UIUX. (2026). 7 latest fintech UX design trends and case studies for 2025. *Design Studio UIUX Blog*. <https://www.designstudiouiux.com/blog/fintech-ux-design-trends/>
- EngineerBabu. (2026, enero 23). Gamification in stock trading: Lessons from Robinhood. *EngineerBabu Blog*. <https://www.engineerbabu.com/blog/gamification-in-stock-trading/>
- ESMA. (2022). *Guidelines on certain aspects of the MiFID II suitability requirements* (ESMA35-43-3172). European Securities and Markets Authority.
- Faber, M. T. (2007). A quantitative approach to tactical asset allocation. *The Journal of Wealth Management*, 9(4), 69–79.
- Finance Monthly. (2025). Best robo-advisors 2025: Wealthfront vs Betterment vs SoFi. *Finance Monthly*. <https://www.finance-monthly.com/best-robo-advisors-compared-wealthfront-vs-betterment-vs-sofi/>
- French, K. R., & Poterba, J. M. (1991). Investor diversification and international equity markets. *American Economic Review*, 81(2), 222–226.
- Frontiers in Education. (2024). The impact of large language models on higher education: Exploring the connection between AI and Education 4.0. *Frontiers in Education*, 9. <https://doi.org/10.3389/feduc.2024.1392091>
- Frydman, C., Barberis, N., Camerer, C., Bossaerts, P., & Rangel, A. (2015). Using neural data to test a theory of investor behavior: An application to realization utility. *The Journal of Finance*, 70(4), 1581–1614.

- G&Co. (2026). The best UX design practices for finance apps in 2026. *G&Co. Insights*. <https://www.g-co.agency/insights/the-best-ux-design-practices-for-finance-apps>
- Gambit Finance. (2025). *Robo-advisors vs. financial education: What investors actually need*. <https://www.gambitfinance.com>
- Grable, J. E. (2000). Financial risk tolerance and additional factors that affect risk taking in everyday money matters. *Journal of Business and Psychology*, 14(4), 625–630.
- Hake, R. R. (1998). Interactive-engagement versus traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses. *American Journal of Physics*, 66(1), 64–74.
- HashStudioz. (2025). *How AI is transforming robo-advisors and wealth management platforms*. <https://www.hashstudioz.com/blog/ai-robo-advisors/>
- HiCapitalize. (2025). *MI Finance review 2025: Pros, cons and who it's for*. <https://hicapitalize.com/ml-finance-review/>
- Hu, Y. (2024). *Reinforcement learning in recommendation systems: Balancing exploration and exploitation*. <https://doi.org/10.1145/3640457.3688189>
- Huberman, G. (2001). Familiarity breeds investment. *The Review of Financial Studies*, 14(3), 659–680.
- Ibbotson, R. G., & Kaplan, P. D. (2000). Does asset allocation policy explain 40, 90, or 100 percent of performance? *Financial Analysts Journal*, 56(1), 26–33.
- Idzorek, T. M. (2006). *Strategic asset allocation and commodities*. Ibbotson Associates.
- InvestmentNews. (2025). *CFA Institute: Explainability remains top barrier to AI adoption in investment management*. <https://www.investmentnews.com>
- Kahneman, D. (2011). *Thinking, fast and slow*. Farrar, Straus and Giroux.
- Kahneman, D., & Tversky, A. (1979). Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2), 263–291.
- Kalasampath, G., et al. (2025). Explainable AI in financial services: A systematic review. *Expert Systems with Applications*. <https://doi.org/10.1016/j.eswa.2026.01.001>
- Khan Academy Help Center. (2022). *About Khan Academy's financial literacy content*. <https://support.khanacademy.org>
- Kirkpatrick, D. L. (1994). *Evaluating training programs: The four levels*. Berrett-Koehler.
- Kobsa, A. (2001). Generic user modeling systems. *User Modeling and User-Adapted Interaction*, 11(1–2), 49–63.
- Lam, X. N., & Riedl, J. (2004). The cold-start problem for recommendations. *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*. ACM Press.
- Liu, Y., & Tsyvinski, A. (2021). Risks and returns of cryptocurrency. *The Review of Financial Studies*, 34(6), 2689–2727.
- Lusardi, A., & Mitchell, O. S. (2014). The economic importance of financial literacy: Theory and evidence. *Journal of Economic Literature*, 52(1), 5–44.

- Manome, N., et al. (2025). Weighted average UCB: Adaptive bandit algorithms for non-stationary environments. *Machine Learning*, 114(2), 88–104.
- Markowitz, H. (1952). Portfolio selection. *The Journal of Finance*, 7(1), 77–91.
- Mayer, R. E. (2009). *Multimedia learning* (2nd ed.). Cambridge University Press.
- Modigliani, F. (1986). Life cycle, individual thrift, and the wealth of nations. *The American Economic Review*, 76(3), 297–313.
- Myśliwiec, K. (2017). *NestJS: A progressive Node.js framework*. <https://nestjs.com>
- Nielsen, J. (1993). *Usability engineering*. Academic Press.
- Pompian, M. M. (2012). *Behavioral finance and wealth management: How to build investment strategies that account for investor biases* (2nd ed.). Wiley.
- Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to recommender systems handbook. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender systems handbook* (pp. 1–35). Springer.
- SaaSFactor. (2026). *Fintech onboarding statistics: Why 73% of users abandon financial apps*. <https://www.saasfactor.com/fintech-onboarding-stats>
- Samuelson, P. A. (1969). Lifetime portfolio selection by dynamic stochastic programming. *The Review of Economics and Statistics*, 51(3), 239–246.
- Schein, A. I., Popescul, A., Ungar, L. H., & Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 253–260.
- Shefrin, H., & Statman, M. (2000). Behavioral portfolio theory. *Journal of Financial and Quantitative Analysis*, 35(2), 127–151.
- Springer Nature. (2025). *Personalized learning with large language models: A systematic review of 55 studies (2020–2024)*. <https://www.springernature.com>
- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2nd ed.). MIT Press.
- The Conversation. (2024). *Gamification and novice investors: How visual rewards shape financial behavior*. <https://theconversation.com>
- Tokenist. (2025). *M1 Finance review 2025: Features, fees and limitations*. <https://tokenist.com/investing/m1-finance-review/>
- TraderHQ. (2026). *Investopedia Stock Simulator review 2026: Pros, cons and alternatives*. <https://traderhq.com/investopedia-simulator-review/>
- Tversky, A., & Kahneman, D. (1974). Judgment under uncertainty: Heuristics and biases. *Science*, 185(4157), 1124–1131.
- Van Rooij, M., Lusardi, A., & Alessie, R. (2011). Financial literacy and stock market participation. *Journal of Financial Economics*, 101(2), 449–472.
- World Economic Forum. (2020). *Schools of the future: Defining new models of education for the fourth industrial revolution*. World Economic Forum.

<https://www.weforum.org/reports/schools-of-the-future-defining-new-models-of-education-for-the-fourth-industrial-revolution>

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J. Y., & Wen, J. R. (2023). A survey of large language models. *arXiv preprint arXiv:2303.18223*. <https://arxiv.org/abs/2303.18223>

13. ANEXOS

Anexo A: [Trabajo de campo](#)

Anexo B: [Formulario / Encuesta pre y post](#)

Anexo C: [Pruebas caja negra](#)

Anexo D: [Cuestionario pruebas usabilidad](#)

Anexo E: Medición de impacto educativo

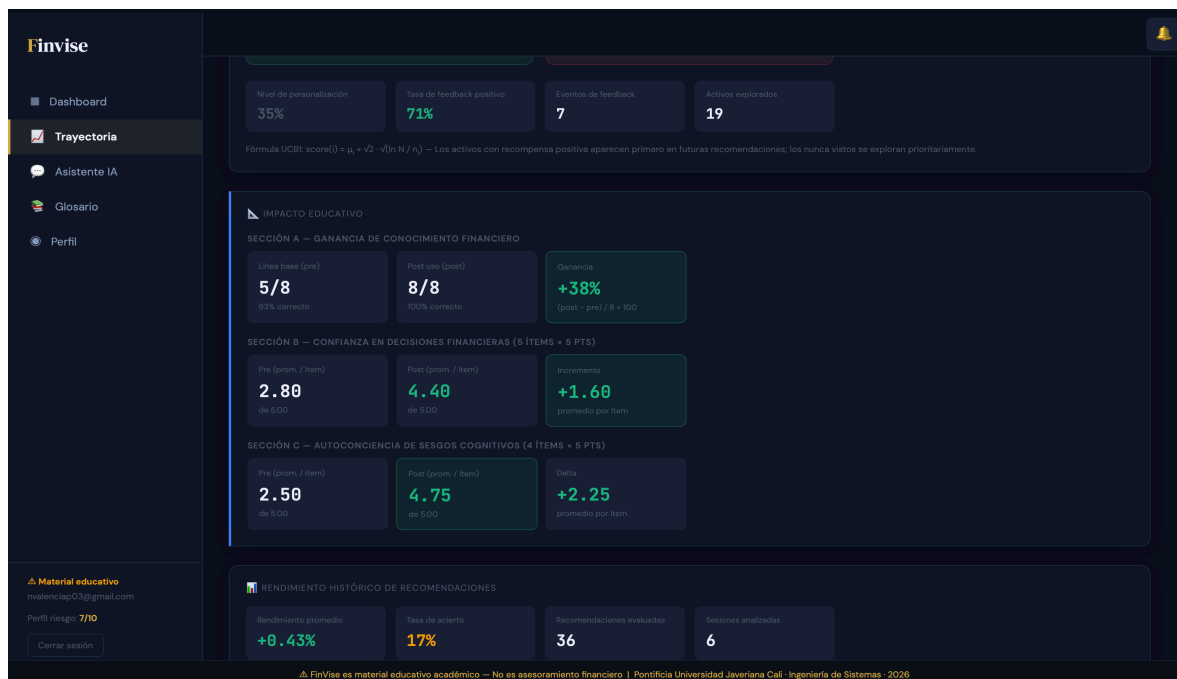


Figura 28: Medición de impacto educativo en plataforma web. Elaboración propia