

# FinQuest – WebApp de Educação Financeira Gamificada

**Título do Projeto:** FinQuest – WebApp de Educação Financeira Gamificada e Facilitada

**Nome do Estudante:** Vinícius Toni Rocha

**Curso:** Engenharia de Software

**Data de Entrega:** 2 de dezembro de 2025

## Resumo

O projeto FinQuest propõe a criação de um WebApp (PWA) de educação financeira que transforma teoria em prática. O objetivo é resolver a dificuldade que muitos iniciantes e investidores intermediários têm em aplicar conceitos financeiros no cotidiano. Para isso, o aplicativo combinará microlições (5–7 min), missões gamificadas com XP e pontuações, uma carteira de investimentos simulada (com dinheiro fictício e calculadoras de juros compostos, preço médio e renda fixa) e um plano de 30 dias composto por missões curtas e explicáveis (usando critérios como HHI para medir concentração da carteira). Ao final de cada ciclo o usuário recebe scores de literacia (acertos nos quizzes), saúde da carteira (diversificação, reserva) e comportamento (aderência às missões). O app opera em modo simulado por padrão, respeitando a LGPD ao solicitar apenas dados mínimos necessários e explicitar a política de privacidade[1]. O projeto visa permitir que qualquer pessoa aprenda, treine e mensure sua evolução financeira sem arriscar dinheiro real.

## 1. Introdução

### Contexto

A educação financeira tem se expandido no Brasil, mas a maioria das iniciativas concentra-se em conteúdos teóricos ou em aplicativos de monitoramento de investimentos. Estudos mostram que intervenções de arquitetura de escolha (nudges) promovem mudança de comportamento com efeito positivo, enquanto programas de educação financeira aumentam conhecimento e comportamento financeiro de forma causal. Ainda assim, muita gente não converte esse conhecimento em hábitos como formação de reserva, diversificação e rebalanceamento de carteira. A lacuna entre saber e fazer motiva este projeto.

### Justificativa

- **Necessidade real:** A alfabetização financeira ainda é baixa e muitos usuários não aplicam conceitos como reserva de emergência, diversificação e rebalanceamento. Falta uma solução que prescreva ações curtas, explique o “porquê” com critérios numéricos e permita medir a evolução dos usuários.
- **Lacuna de mercado:** Plataformas como Trademap, Kinvo, NuInvest ou academias de educação financeira oferecem monitoramento ou conteúdo, mas não integram

prescrição curta, explicabilidade e medição de resultados em um único ciclo educacional.

- **Evidência científica:** Meta-análises indicam que *nudges* e intervenções educacionais produzem mudanças comportamentais significativas, reforçando a importância de combinar conteúdo, prática e feedback.
- **Viabilidade acadêmica:** O escopo é compatível com o semestre, utiliza tecnologias dominadas (Angular, .NET 8, SQL), roda em hospedagens gratuitas (Vercel/Render), opera com dados simulados por padrão e possui métricas objetivas para avaliar o impacto.

## Objetivos

**Geral:** Promover inteligência financeira prática e transparente, transformando conteúdo em hábito por meio de gamificação, missões e simulação.

### Específicos:

1. Desenvolver onboarding que levante perfil de risco e objetivos, gerando um plano personalizado com várias missões adaptadas.
2. Criar trilhas de microlições com quizzes de reforço e feedback imediato.
3. Implementar uma carteira simulada permitindo comprar, vender e aportar ativos fictícios (ações, FIIs, renda fixa) baseados no mercado financeiro.
4. Fornecer **calculadoras** de juros compostos (com aportes), preço médio e renda fixa.
5. Estabelecer **missões** gamificadas com XP, badges e nudges explicáveis (limites de HHI, bandas de rebalanceamento, meses de reserva).
6. Exibir **scores e relatórios** (literacia, saúde/HHI, comportamento) e prover exportação de dados.
7. Garantir **segurança, privacidade e acessibilidade**, aderindo à LGPD[1], OWASP Top 10 e WCAG.

## 2. Descrição do Projeto

### Linha de Projeto

**Web Apps.** O produto será entregue como **PWA** (Progressive Web App) responsivo, disponível via navegadores modernos e instalável em smartphones sem necessidade de lojas de aplicativos. A arquitetura seguirá o modelo **cliente-servidor** com separação de camadas.

### Tema do Projeto

**FinQuest – WebApp de Educação Financeira Gamificada e Facilitada.** Trata-se de um hub de educação financeira que une ensino, simulação e gamificação para auxiliar iniciantes e investidores intermediários a desenvolverem hábitos financeiros saudáveis e medirem sua evolução.

## Propósito e Uso Prático

O FinQuest resolve o problema de **conversão do conhecimento financeiro em prática**. Ele guiará o usuário por meio de um **Plano de 30 dias** com missões curtas e mensuráveis, utilizando **dinheiro fictício** para treinar decisões sem risco. No futuro, a plataforma poderá incluir minicompetições de carteiras (utilizando valores fictícios) para engajar ainda mais os participantes. Além disso, o sistema oferecerá calculadoras e relatórios que auxiliam usuários mais experientes a validar estratégias.

## Público-Alvo

- **Iniciantes** em finanças pessoais, que desejam aprender os fundamentos e praticar sem risco.
- **Investidores intermediários**, que buscam uma ferramenta simples para testar estratégias, acompanhar sua evolução e comparar desempenho em ambientes simulados.

## Problemas a Resolver

1. Dificuldade de aplicar conteúdos aprendidos em livros/cursos.
2. Falta de feedback e acompanhamento mensurável de desempenho.
3. Risco de iniciar investimentos reais sem conhecimento prático.
4. Ausência de critérios transparentes para diversificação e rebalanceamento.

## Diferenciação/Ineditismo

- **Plano personalizado** com missões aplicáveis e explicáveis (por critérios como HHI, bandas de rebalanceamento, meses de reserva).
- **Carteira simulada** com dinheiro fictício e possibilidade de minicompetições sem risco.
- **Scores integrados** (Literacy, Health/HHI, Behavior) que medem aprendizado, saúde da carteira e formação de hábito.
- **Gamificação significativa**: XP, níveis e badges associados a comportamentos desejados (não apenas tarefas aleatórias).
- **Privacidade por design**: dados simulados por padrão e importação opcional via CSV, seguindo o princípio da **minimização de dados e consentimento explícito** da LGPD.
- **Foco local (BR)**: contempla ativos brasileiros, inclui conceitos de dividendos, JCP, FIIs e produtos de renda fixa.

## Limitações

- O MVP não integrará Open Finance nem corretoras; operações financeiras serão simuladas.
- A cobertura tributária será simplificada (tratamento de JCP/dividendos e eventos corporativos básicos).

- O **Copiloto de finanças (GPT das Finanças)** via IA avançada fica fora do escopo inicial; será considerado em fases posteriores.
- O suporte será exclusivo em **PT-BR** no MVP; a internacionalização é roadmap.

## Normas e Legislações Aplicáveis

- **LGPD (Lei 13.709/2018)**: coleta apenas dados necessários; consentimento explícito; possibilidade de exclusão; transparência sobre uso.
- **OWASP Top 10**: práticas de desenvolvimento seguro para prevenir injeção, XSS, CSRF, falhas de autenticação e autorização.
- **WCAG 2.1**: diretrizes de acessibilidade web, garantindo contraste adequado, navegação por teclado e suporte a leitores de tela.
- **ESRB/PEGI**: não se aplica diretamente, pois não é jogo de entretenimento, mas a gamificação seguirá princípios de ética e não promoverá comportamento de jogo de azar.
- **Normas de ética em IA (UNESCO/OECD)**: considerar vieses e transparência em eventuais módulos de IA, embora IA avançada não faça parte do MVP.

## Métricas de Sucesso (MVP)

- **Literacy Score**: aumento  $\geq 20$  p.p. nos pós-quiz em relação ao pré-quiz.
- **Portfolio Health (HHI)**: redução  $\geq 25\%$  no índice de concentração da carteira simulada após o plano de 30 dias.
- **Behavior Score**:  $\geq 70\%$  das missões concluídas no ciclo de 30 dias.
- **Engajamento**:  $\geq 1$  pergunta ou simulação por semana durante o plano.
- **Retenção**:  $\geq 50\%$  dos usuários completando todas as missões no ciclo piloto.

## 3. Especificação Técnica

### 3.1 Requisitos de Software

#### *Requisitos Funcionais (RF)*

Código	Descrição	Aderência à linha WebApp
RF01	Realizar cadastro e autenticação básica do usuário (JWT).	Requer autenticidade e sessão, conforme “Hospedagem pública e funcionalidades completas”.
RF02	Executar onboarding levantando perfil de risco, metas e gerando o Plano de 30 dias de missões.	Atende necessidade de plano personalizado.
RF03	Consumir trilhas de microlições com quizzes, registrando acertos e erros.	Garante interface navegável e conteúdo didático.
RF04	Gerenciar carteira simulada: comprar/vender ativos fictícios, registrar aportes, calcular preço médio.	Cumpre requisito de funcionalidade real; persistência via BD.

<b>RF05</b>	Disponibilizar calculadoras de juros compostos (com aportes), preço médio e renda fixa (pré e pós-fixada).	<b>Contribui para desempenho analítico.</b>
<b>RF06</b>	Gerar e controlar missões com XP, níveis e badges; emitir nudges baseados em critérios (HHI, bandas de rebalanceamento, reserva).	<b>Gamificação significativa.</b>
<b>RF07</b>	Exibir scores e relatórios (literacy, health/HHI, behavior) e permitir exportação de CSV/JSON.	<b>Métricas de sucesso definidas.</b>
<b>RF08</b>	(Roadmap) Permitir importação de extratos via CSV/OFX opcional, higienizando e categorizando dados.	<b>Persistência e integração externa desejável.</b>

#### ***Requisitos Não-Funcionais (RNF)***

- **RNF01 – Desempenho:** O tempo de resposta das APIs deve ser < 500 ms em consultas locais; uso de cache para dados estáticos.
- **RNF02 – Usabilidade:** Interface responsiva (mobile-first), com acessibilidade nível AA (WCAG 2.1).
- **RNF03 – Segurança:** Uso de HTTPS, proteção contra injeção e XSS (OWASP Top 10), autenticação JWT e renovação de tokens.
- **RNF04 – Escalabilidade:** Separação de front-end e back-end; possibilidade de escalar horizontalmente via serviços de nuvem; API stateless.
- **RNF05 – Disponibilidade:** Uptime ≥99% usando hospedagens públicas (Vercel/Render) e backups automáticos do BD.
- **RNF06 – Auditoria:** Logs de ações (missões, operações na carteira, quizzes), sem armazenar dados sensíveis.

#### ***Representação dos Requisitos***

Será utilizado um **Diagrama de Casos de Uso** para mapear os atores (Usuário, Sistema) e funcionalidades principais (Cadastro, Onboarding, Trilhas, Carteira, Calculadoras, Missões, Relatórios). Esse diagrama será apresentado em anexo. A especificação seguirá a **metodologia UML** para facilitar o entendimento.

#### ***Aderência aos Requisitos da Linha de Projeto***

- **Hospedagem pública:** deploy do front em **Vercel** e API em **Render/Fly**, com repositório público no GitHub (histórico de commits e CI/CD básico).
- **Interface navegável e usabilidade:** Páginas responsivas e interativas, com navegação clara entre trilhas, carteira, missões e relatórios.
- **Arquitetura em camadas:** client (Angular), API (.NET), domínio, infraestrutura (SQL).
- **Código-fonte versionado:** uso de Git com branches, pull requests e issues.

- **Documentação mínima:** README, diagramas, instruções de deploy e políticas de privacidade.

## 3.2 Considerações de Design

### *Visão Inicial da Arquitetura*

A arquitetura adotará o padrão **Cliente-Servidor em Camadas**, subdividida em: -

**Apresentação (front-end):** aplicação Angular PWA com componentes modulares (Home, Trilhas, Carteira, Calculadoras, Missões, Relatórios, Auth).

- **Serviços de Aplicação (API):** expostos via **.NET 8 Web API**, responsáveis por lógica de negócios, regras de gamificação e geração de missões.

- **Domínio:** classes de entidades (Usuário, Missão, Operação, Score), serviços de negócio (calculadoras, agregador de proventos) e validações.

- **Infraestrutura:** camada de persistência com **SQL Server** ou **PostgreSQL** (ORM via EF Core), armazenamento de arquivos CSV de proventos.

- **Camada de IA (futuro):** microserviço Python ou .NET para processamento de perguntas complexas (Cite→Compute→Explain), em roadmap.

### *Padrões de Arquitetura*

- **MVC/MVVM** no front-end (Angular).
- **Arquitetura em Camadas** no back-end (.NET) com separação de **Controllers, Services, Repositories** e **Models**.
- **DDD (Domain-Driven Design)** para estruturar o domínio de investimentos e gamificação.
- **C4 Model:** será utilizado para documentar os níveis de **Contexto, Contêiner, Componentes** e (quando aplicável) **Código**.

### *Modelos C4*

- **Contexto:** mostra o usuário interagindo com o WebApp (via navegador) e com a API hospedada na nuvem; realça a separação de responsabilidades.
- **Contêiner:** detalha a aplicação Angular, API .NET, banco de dados SQL e serviço de armazenamento de CSV.
- **Componentes:** identifica componentes internos da API (AuthenticationService, MissionGenerator, PortfolioService, ScoreCalculator) e do front (LessonComponent, MissionComponent, SimulatorComponent).
- **Código (opcional):** diagrama de classes simplificado das entidades principais será incluído nos apêndices.

### *Mockups das Telas Principais*

As principais telas (Dashboard, Trilhas, Missões, Carteira Simulada, Calculadoras, Relatórios) serão prototipadas em **Figma**, demonstrando navegação, layout e identidade visual. Serão anexadas imagens no apêndice do RFC.

### *Decisões e Alternativas Consideradas*

- **WebApp vs. Mobile nativo:** escolheu-se PWA por permitir acesso multiplataforma e facilitar deploy; mobile nativo e IoT foram adiados para o roadmap.
- **Simulação vs. Operação real:** optou-se pelo modo **simulado** para reduzir risco ao usuário e contornar exigências regulatórias.
- **Plano de 30 dias:** escolhido para equilibrar engajamento e evidência de impacto; planos mais longos poderiam causar desistência.

### *Critérios de Escalabilidade, Resiliência e Segurança*

- **Escalabilidade:** API stateless, podendo ser replicada e balanceada horizontalmente; BD escalável via serviço gerenciado.
- **Resiliência:** uso de retries, timeouts e circuit breakers nos acessos a dados; logs centralizados.
- **Segurança:** criptografia TLS, autenticação JWT, hashing de senhas, validação de entrada, proteção contra XSS e CSRF.
- **Backup e Recuperação:** backups automáticos do BD, replicação em zona distinta e testes de restauração.

## 3.3 Stack Tecnológica

Camada	Ferramenta / Linguagem	Justificativa
Front-end	Angular 16 (TypeScript), PWA, RxJS, NgRx	Framework moderno, tipado, com ecossistema maduro e suporte a PWAs; adequado para aplicações modulares.
Back-end	C# / .NET 8 Web API, EF Core	Framework robusto, suporte a APIs RESTful, fácil integração com SQL Server e Postgres; performance estável.
Banco de dados	SQL Server (ou PostgreSQL)	Garantia de ACID, suporte a transações e consultas complexas; possível troca para Postgres em hospedagens gratuitas.
Armazenamento de arquivos	Azure Blob Storage ou bucket S3 compatível (minio/Wasabi)	Para armazenar CSVs de proventos e anexos.
Ferramentas de desenvolvimento	Visual Studio Code, Git/GitHub, Docker, GitHub Actions para CI/CD	Suportam integração contínua, versionamento e pipelines automatizados.
Design e prototipagem	Figma, Lucidchart/draw.io	Criação de mockups e diagramas C4/UML.
Gerenciamento de	GitHub Projects/Issues, Trello ou	Controle de sprints, user stories

<b>Projeto</b>	Azure Boards	e tarefas.
<b>Licenciamento</b>	O projeto será distribuído sob licença MIT; bibliotecas (Angular, .NET) utilizam licenças permissivas (MIT/Apache).	

### 3.4 Considerações de Segurança

#### *Riscos Identificados*

- **Injeção de SQL/XSS:** falhas na validação de entrada podem comprometer o banco ou o front-end.
- **Quebra de autenticação:** vazamento de tokens ou senhas frágeis pode permitir acesso não autorizado.
- **Exposição de dados sensíveis:** caso haja importação de extratos, dados pessoais e financeiros podem ser expostos.
- **Ataques de força bruta e DDoS:** podem tornar o serviço indisponível.

#### *Medidas de Mitigação*

- Utilizar **ORM (EF Core)** com parâmetros para evitar injeção SQL; sanitizar entradas e saídas.
- Implementar **autenticação JWT** com expiração curta e refresh tokens; armazenar senhas com **BCrypt/Scrypt**.
- Aplicar **CORS** restrito; validar/filtrar uploads CSV; criptografar dados sensíveis em repouso.
- Configurar limites de requisições (rate limiting) e uso de CDN/firewall para mitigar DDoS.

#### *Normas e Boas Práticas Seguidas*

- **OWASP Top 10:** referência para proteger contra as principais vulnerabilidades web.
- **ISO/IEC 27001:** seguir diretrizes para gestão de segurança da informação.
- **LGPD:** coleta mínima de dados, consentimento informado, controle e exclusão.

#### *Responsabilidade Ética*

- O FinQuest será rotulado como **ferramenta educacional**, não oferecendo recomendações reguladas.
- Quaisquer módulos de IA futuros seguirão princípios de transparência, não discriminação e explicabilidade.

### 3.5 Conformidade e Normas Aplicáveis

- **LGPD – Lei 13.709/2018:** será coletado apenas o mínimo necessário (nome, e-mail, perfil de risco), com consentimento explícito. Usuários poderão acessar, corrigir e excluir seus dados.



- **WCAG 2.1:** garantir acessibilidade, proporcionando contraste adequado, navegação via teclado e descrição alternativa de imagens.
- **OWASP Top 10:** seguir práticas contra injeções, quebras de autenticação, exposição de dados sensíveis, XSS e CSRF.
- **ABNT NBR ISO 25010** (qualidade de software): cumprir requisitos de confiabilidade, usabilidade, eficiência, segurança, manutenibilidade e portabilidade.
- **Normas educacionais brasileiras:** o app será classificado como curso livre e seguir diretrizes de educação informal.

## 4. Próximos Passos

### Cronograma (Portfólio I e II)

Fase	Entrega	Data (sugerida)
Definição & RFC (N1)	Conclusão do RFC, levantamento de requisitos, protótipos iniciais.	até 15/03/2026
Protótipo Interativo (N2)	Apresentação dos mockups, diagrama de arquitetura, tabela de trabalhos correlatos, validação de requisitos.	até 30/04/2026
Sprint 1	Setup do repositório, autenticação, onboarding e plano de 30 dias.	15/05/2026
Sprint 2	Implementação das calculadoras e da carteira simulada.	25/05/2026
Sprint 3	Implementação das missões, XP e nudges.	01/06/2026
Sprint 4	Scores e relatórios; testes unitários e de usabilidade.	15/06/2026
Entrega Sprint	MVP funcional hospedado; documentação e relatório de avaliação.	20/06/2026
Entrega Final	Integrações extras (CSV/OFX), competições, evolução do Copiloto, avaliação com usuários reais e estudo de impacto.	junho/2026

### Marcos e Checkpoints

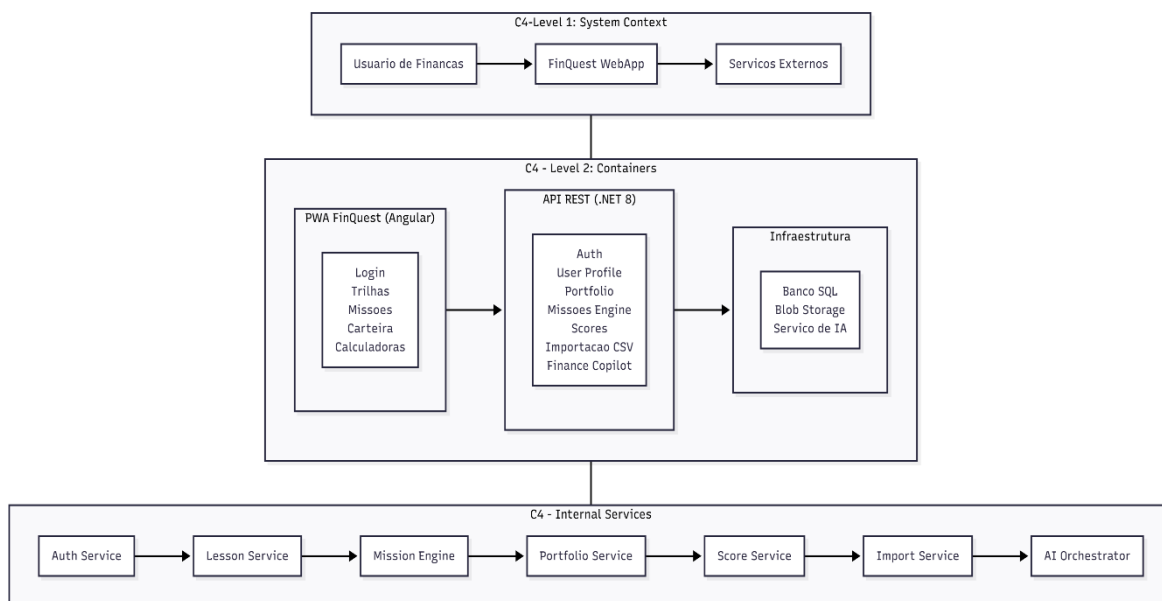
- Revisão semanal de requisitos e ajustes de backlog.
- Checkpoint com orientador ao final de cada sprint.
- Testes de usabilidade com 5–8 usuários ao final do Sprint 4.

## 5. Referências

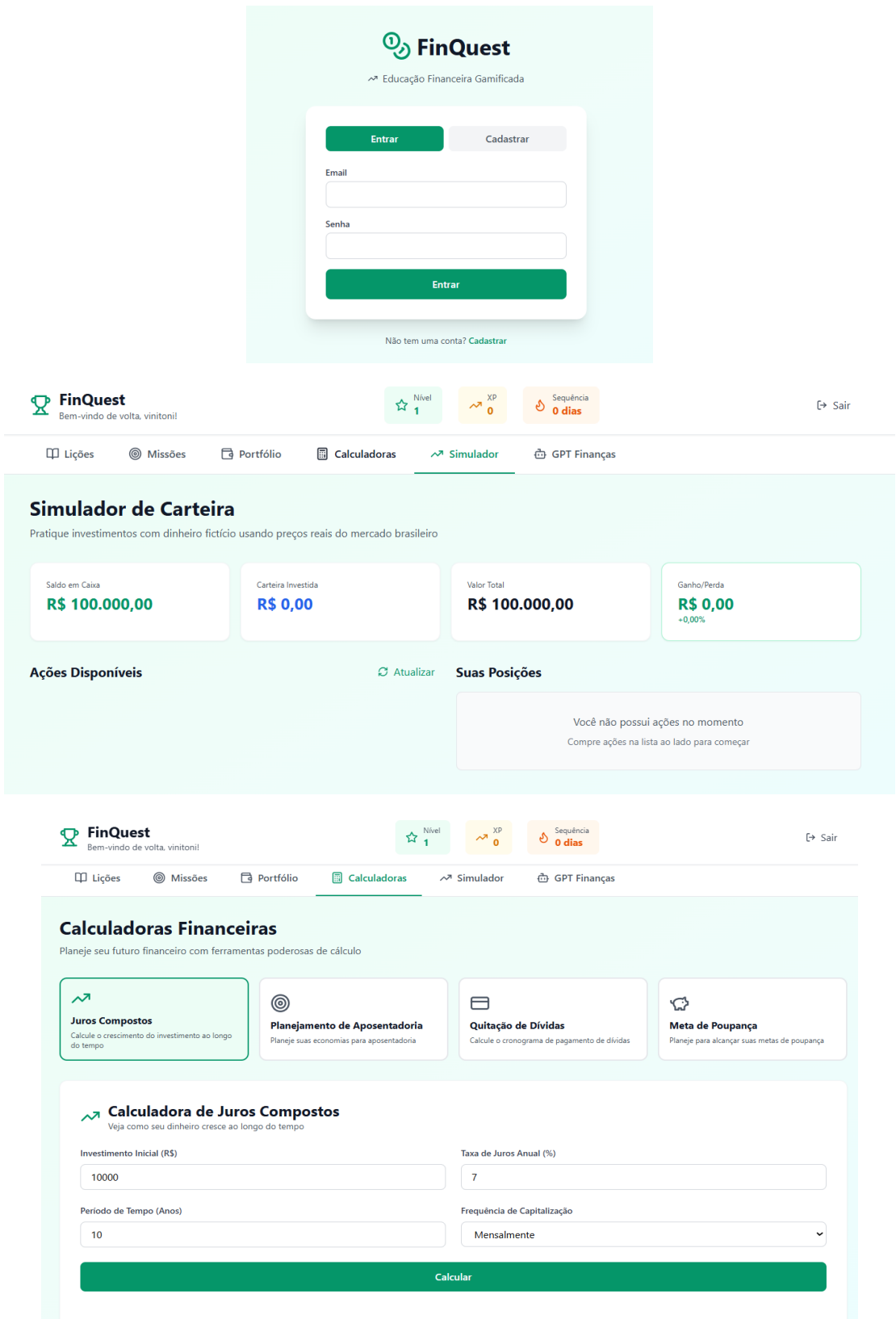
1. **Mertens et al. (2022)** – *The effectiveness of nudging: A meta-analysis of choice architecture interventions across behavioral domains*. PNAS. A meta-análise mostra que intervenções de arquitetura de escolha promovem mudanças comportamentais com efeito médio (Cohen's  $d \approx 0,45$ ).
2. **Kaiser et al. (2020)** – *Financial education affects financial knowledge and downstream behaviors*. NBER Working Paper 27057. A meta-análise de 76 RCTs demonstra que programas de educação financeira têm efeitos positivos e economicamente significativos no conhecimento e comportamento financeiro.
3. **Serpro (2021)** – *Princípios da LGPD*. Explica que o consentimento deve ser explícito e que a minimização de dados é premissa básica.
4. **OWASP Foundation** – *OWASP Top 10 (2021)*. Lista das principais vulnerabilidades de aplicações web e práticas de mitigação.
5. **WCAG 2.1** – *Web Content Accessibility Guidelines*. Diretrizes internacionais de acessibilidade na web.
6. **ISO/IEC 27001** – *Technology – Security techniques – Information security management systems*. Padrão para sistemas de gestão da segurança da informação.
7. **ISO/IEC 25010** – *Systems and software engineering – Systems and software quality requirements and evaluation (SQuaRE) – System and software quality models*.

## 6. Apêndices

### A. Diagrama de Casos de Uso (UML)



C. Mockups de Telas



## 7. Avaliações de Professores

**Considerações Professor/a:**

**Assinatura:** \_\_\_\_\_

**Considerações Professor/a:**

**Assinatura:** \_\_\_\_\_

**Considerações Professor/a:**

**Assinatura:** \_\_\_\_\_