

Finança: Uma Aplicação Web de Gestão Financeira Pessoal Com Análise Inteligente Baseada em Inteligência Artificial Generativa

Bruno Pereira Martins Silva¹; Arthur Silveira Oliveira²; Pedro Henrique Gomes Carneiro³

^{1, 2, 3} *Universidade de Uberaba*

brunopms123456789@gmail.com

Resumo: O controle das finanças pessoais constitui um desafio recorrente para a população brasileira, marcado por altos índices de endividamento e pela dificuldade de visualizar, de forma consolidada, o destino dos recursos financeiros individuais. Este trabalho apresenta o FinancelA, uma aplicação web desenvolvida para registrar receitas e despesas, importar extratos bancários em formato CSV, classificar transações de modo automático por meio de palavras-chave e gerar análises financeiras mensais em linguagem natural com o apoio de inteligência artificial generativa. O objetivo do estudo consiste em demonstrar como a integração de uma camada de interpretação textual fornecida por um modelo de linguagem pode complementar painéis financeiros tradicionais, tornando os dados mais compreensíveis para o usuário final. A metodologia adotada caracteriza-se como pesquisa aplicada de natureza qualitativa, com abordagem de pesquisa-desenvolvimento e prototipação evolutiva, organizada em seis etapas: modelagem do banco, implementação do backend, construção do frontend, integração com a inteligência artificial, validação e implantação. A solução utiliza React no frontend, Node.js com Express no backend, PostgreSQL acessado via Prisma e a API Gemini do Google para a análise mensal, hospedada em camadas gratuitas (Netlify, Render e Neon). Os resultados evidenciam o cumprimento integral do escopo definido, com nove módulos funcionais, categorização local consistente e geração de análises textuais coerentes a partir de dados agregados, preservando o sigilo dos dados pessoais. Conclui-se que o sistema cumpre seu propósito acadêmico e demonstra a viabilidade técnica de uma aplicação financeira pessoal apoiada por inteligência artificial.

Palavras-chave: gestão financeira pessoal; inteligência artificial generativa; aplicação web; arquitetura em camadas; categorização automática.

1 Introdução

A relação do brasileiro com o próprio dinheiro tem sido objeto de atenção crescente nos últimos anos. Pesquisas conduzidas anualmente pela Confederação Nacional do Comércio de Bens, Serviços e Turismo (CNC), por meio da Pesquisa de Endividamento e Inadimplência do Consumidor (PEIC), apontam que parcela significativa das famílias brasileiras encontra-se endividada, com dificuldades crescentes para honrar compromissos financeiros assumidos (CNC, 2024). Em paralelo, indicadores de educação financeira da Organização para a Cooperação e Desenvolvimento Econômico (OCDE) mostram que o brasileiro apresenta níveis abaixo da média internacional em conhecimento, comportamento e atitudes relacionadas ao dinheiro (OCDE, 2020). Esse cenário evidencia uma lacuna persistente entre a disponibilidade de ferramentas tecnológicas e a efetiva organização das finanças domésticas.

Na prática cotidiana, ainda é comum que o controle pessoal seja realizado por meio de cadernos, planilhas eletrônicas ou pequenas anotações esparsas. Tais abordagens exigem disciplina contínua, oferecem pouca visualização agregada e, principalmente, não traduzem números em interpretações compreensíveis sobre o comportamento financeiro do usuário. Aplicativos comerciais ofertados pelo mercado, por sua vez, frequentemente

concentram-se em integrações bancárias por meio do Open Finance ou em modelos de assinatura, deixando descobertas necessidades simples, porém recorrentes: registrar manualmente uma receita ou despesa, importar um extrato em arquivo CSV e receber uma leitura compreensível sobre o próprio mês.

Diante desse contexto, este trabalho apresenta o FinancelA, uma aplicação web de gestão financeira pessoal cuja proposta central reside na integração entre um painel analítico tradicional, baseado em indicadores e gráficos, e uma camada de interpretação textual produzida por um modelo de inteligência artificial generativa. O sistema permite o cadastro manual de receitas e despesas, a importação de extratos em CSV com prévia editável, a categorização automática local por meio de palavras-chave e a geração de uma análise mensal, em linguagem natural, baseada em dados agregados e anonimizados. Optou-se deliberadamente por preservar a privacidade do usuário, restringindo o uso da inteligência artificial à interpretação de totais agregados e impedindo qualquer envio de dados pessoais identificáveis ao serviço externo.

O problema central que motiva a pesquisa pode ser formulado da seguinte maneira: como oferecer ao usuário comum uma ferramenta web simples para acompanhar suas finanças e, ao mesmo tempo, traduzir os dados acumulados em uma interpretação compreensível, sem comprometer a segurança das informações pessoais? A justificativa do trabalho ampara-se tanto na relevância social do tema — fortalecimento da educação financeira por meio de tecnologia acessível — quanto na relevância acadêmica e tecnológica, dado que o estudo articula práticas contemporâneas de engenharia de software, arquitetura em camadas, segurança da informação e aplicação prática de inteligência artificial generativa em contexto analítico.

O objetivo geral deste trabalho consiste em desenvolver uma aplicação web de gestão financeira pessoal capaz de registrar, organizar e interpretar transações financeiras com o apoio de inteligência artificial generativa. Como objetivos específicos, destacam-se: (i) modelar e implementar um backend seguro, com autenticação por JSON Web Token e armazenamento adequado de senhas; (ii) construir uma interface web responsiva, organizada em torno de um painel central de indicadores e gráficos; (iii) implementar mecanismo local de categorização automática baseado em palavras-chave; (iv) integrar a aplicação a um modelo de linguagem generativo capaz de produzir análises mensais em texto; e (v) implantar o sistema em ambiente público gratuito, validando a viabilidade técnica da arquitetura proposta.

O presente artigo organiza-se em cinco seções, além desta introdução. A Seção 2 apresenta o referencial teórico que sustenta o trabalho, abordando educação financeira, arquiteturas para aplicações web modernas, inteligência artificial generativa e segurança da informação. A Seção 3 descreve a metodologia adotada e detalha a implementação técnica do sistema. A Seção 4 expõe os resultados obtidos e discute suas implicações. A Seção 5, por fim, retoma o problema, avalia o cumprimento do objetivo e indica trabalhos futuros, encerrando o estudo com as referências utilizadas.

2 Referencial Teórico

Esta seção apresenta os fundamentos teóricos que sustentam o desenvolvimento do FinancelA. Discutem-se, em ordem, a educação financeira pessoal como motivação social, as arquiteturas para aplicações web modernas, a aplicação de inteligência artificial generativa em contextos analíticos e os princípios de segurança da informação adotados.

2.1 Educação financeira e organização das finanças pessoais

A educação financeira pode ser compreendida como o processo pelo qual indivíduos aprimoram sua compreensão sobre conceitos e produtos financeiros, desenvolvendo competências para tomar decisões fundamentadas em informação (OCDE, 2020). No

Brasil, dados consolidados pela Confederação Nacional do Comércio de Bens, Serviços e Turismo demonstram que a parcela de famílias endividadadas se manteve em patamar elevado nos últimos anos (CNC, 2024), reforçando a necessidade de instrumentos que apoiem o planejamento financeiro doméstico.

Sob o ponto de vista prático, autores como Cerbasi (2022) defendem que o passo inicial para a organização financeira consiste em conhecer com precisão o destino do próprio dinheiro. Ferramentas que automatizam o registro, categorizam transações e oferecem visualizações claras tendem a reduzir o atrito desse processo e a aumentar a aderência do usuário ao planejamento. O FinancelA insere-se nesse contexto ao buscar uma interface acessível, com indicadores visuais simples e textos interpretativos curtos, evitando exigir do usuário conhecimento financeiro avançado.

2.2 Aplicações web modernas e arquitetura em camadas

Sistemas web contemporâneos são tipicamente estruturados em torno de uma Single Page Application (SPA) no lado do cliente e uma Application Programming Interface (API) no lado do servidor, comunicando-se por meio do protocolo HTTP em formato JSON (Richards; Ford, 2020). Essa separação favorece a evolução independente das camadas, a reutilização do backend por múltiplos clientes e a hospedagem distribuída entre provedores especializados.

A organização interna do servidor segue, neste trabalho, a abordagem de arquitetura em camadas descrita por Fowler (2002), na qual rotas e controllers atuam como uma camada fina de interface, delegando toda a lógica de negócio a serviços dedicados, que por sua vez acessam o banco de dados por meio de uma camada de persistência. No ecossistema Node.js, frameworks como Express oferecem o mínimo de infraestrutura necessária para implementar esse padrão, ao passo que mapeadores objeto-relacional reduzem a complexidade do acesso a dados e protegem a aplicação contra injeções de SQL ao gerar consultas parametrizadas automaticamente.

Do lado do cliente, a biblioteca React tornou-se uma das principais opções para a construção de interfaces declarativas, baseadas em componentes reutilizáveis e em um modelo unidirecional de fluxo de dados (Richards; Ford, 2020). A combinação com o empacotador Vite e com o utilitário de estilização Tailwind CSS permite produzir interfaces responsivas com agilidade, sem recorrer a folhas de estilo extensas e isoladas por componente.

2.3 Inteligência Artificial Generativa aplicada à análise textual

Modelos de linguagem de grande porte, conhecidos pela sigla LLM (Large Language Models), são redes neurais profundas treinadas em grandes corpora textuais com o objetivo de modelar a distribuição estatística da linguagem natural. A arquitetura predominante desses modelos é o Transformer, originalmente proposto por Vaswani et al. (2017), que substituiu mecanismos recorrentes pela operação de autoatenção e tornou viável o treinamento em larga escala. Sobre essa base, surgiram famílias de modelos generativos como a série Gemini, desenvolvida pelo Google DeepMind e disponibilizada por meio de uma API pública (Google, 2025).

Em aplicações de negócio, a integração com LLMs tipicamente segue o padrão de envio de um prompt estruturado, contendo instruções e dados de contexto, e o recebimento de uma resposta textual. No contexto financeiro, esse padrão permite traduzir conjuntos de números em narrativas compreensíveis para o usuário, sem a necessidade de treinar modelos próprios. Adota-se, neste trabalho, a recomendação de que apenas dados agregados sejam encaminhados ao serviço externo, em consonância com princípios de minimização de dados previstos na Lei Geral de Proteção de Dados Pessoais (Brasil, 2018).

2.4 Segurança da informação em aplicações web

A proteção das informações é requisito incontornável em qualquer aplicação que manipule dados financeiros. Stallings (2015) destaca três pilares fundamentais — confidencialidade, integridade e disponibilidade — que orientam as decisões de segurança ao longo do ciclo de vida do software. No nível da autenticação, o padrão JSON Web Token, descrito por Jones, Bradley e Sakimura (2015) na RFC 7519, fornece um mecanismo compacto e auto-contido para a representação de credenciais entre cliente e servidor, dispensando a manutenção de sessões em memória do servidor.

Para o armazenamento de senhas, recomenda-se o uso de funções de hash adaptativas, capazes de aumentar o custo computacional à medida que o hardware evolui. O algoritmo bcrypt, proposto por Provos e Mazières (1999), atende a esse requisito e tornou-se padrão de facto em aplicações web modernas. Complementarmente, recomendações da Open Web Application Security Project (OWASP, 2021) orientam práticas como restrição de origem por meio de Cross-Origin Resource Sharing (CORS), limitação de taxa em rotas sensíveis e uso obrigatório do protocolo HTTPS em produção. Todas essas práticas foram incorporadas ao projeto do FinancelA.

3 Metodologia e Descrição do Sistema

3.1 Natureza e abordagem da pesquisa

A pesquisa realizada classifica-se, quanto à sua natureza, como pesquisa aplicada, uma vez que visa gerar conhecimento de aplicação prática dirigido à solução de um problema específico, conforme tipologia descrita por Gil (2019). Quanto aos objetivos, caracteriza-se como descritiva e exploratória; quanto à abordagem, como qualitativa; e quanto aos procedimentos técnicos, como pesquisa-desenvolvimento, baseada em prototipação evolutiva e validação iterativa, prática recorrente na engenharia de software contemporânea (Sommerville, 2019).

O desenvolvimento foi organizado em seis etapas sequenciais, com critérios de conclusão explícitos para cada uma: (i) modelagem e configuração do banco de dados; (ii) implementação completa do backend, excetuando-se o serviço de inteligência artificial; (iii) construção do frontend e integração com a API; (iv) implementação da integração com o modelo generativo; (v) validação manual de ponta a ponta; e (vi) implantação pública em ambiente de produção. A passagem para uma nova etapa só se efetivou após o atendimento dos critérios definidos para a etapa anterior.

3.2 Stack tecnológica adotada

A escolha das tecnologias seguiu três princípios orientadores: viabilidade de execução em camadas gratuitas, maturidade comprovada no mercado e aderência ao perfil acadêmico do projeto. O Quadro 1 sintetiza as opções definitivas.

Quadro 1 – Stack tecnológica oficial do projeto FinancelA

Camada	Tecnologia	Justificativa
Frontend	React + Vite + Tailwind CSS	Componentes reutilizáveis e build ágil
Backend	Node.js + Express	Framework minimalista, aderente ao escopo
Banco de Dados	PostgreSQL (Neon)	Relacional maduro em camada gratuita
ORM	Prisma	Consultas parametrizadas e migrations automatizadas
Inteligência Artificial	Google Gemini (gemini-2.0-flash)	Camada gratuita permanente, sem cartão
Hospedagem do Frontend	Netlify	Implantação contínua a partir do repositório

Camada	Tecnologia	Justificativa
Hospedagem do Backend	Render	Web Service gratuito, integrado ao GitHub

Fonte: Elaborado pelos autores (2026).

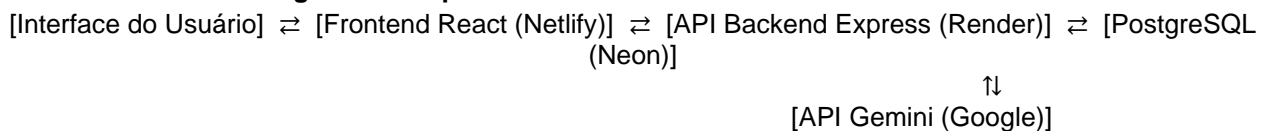
O modelo Gemini foi selecionado em detrimento de alternativas como GPT (OpenAI) e Claude (Anthropic) por oferecer camada gratuita permanente — limite de mil e quinhentas requisições diárias — sem exigência de cadastro de cartão de crédito, característica decisiva para um projeto acadêmico sem orçamento dedicado.

3.3 Arquitetura do sistema

A arquitetura adotada organiza-se em camadas, com responsabilidades estritamente delimitadas. O frontend constitui uma SPA hospedada no Netlify, responsável exclusivamente pela renderização da interface e por chamadas autenticadas à API. O backend, hospedado no Render, concentra toda a lógica de negócio, dividindo-se internamente em rotas, controllers e serviços. O banco de dados, hospedado no Neon, é acessado unicamente pelo backend. A API Gemini, por sua vez, é consultada apenas pelo serviço de análise mensal, jamais sendo acessada diretamente pelo cliente.

O Diagrama 1 ilustra, de forma esquemática, o fluxo de comunicação entre os componentes.

Figura 1 – Arquitetura em camadas do sistema FinanceLA



Fonte: Elaborada pelos autores (2026).

As requisições entre frontend e backend trafegam por HTTPS, em formato JSON, e carregam um token JWT no cabeçalho de autorização. O backend valida o token em todas as rotas privadas por meio de um middleware central e extrai o identificador do usuário diretamente do token validado, jamais do corpo da requisição. Esse procedimento garante o isolamento estrito de dados entre usuários, uma vez que toda consulta ao banco inclui obrigatoriamente o identificador do proprietário do recurso.

3.4 Modelagem do banco de dados

O modelo de dados é composto por cinco entidades: usuários (users), categorias (categories), transações (transactions), registros de importação (import_logs) e análises geradas pela IA (ai_analyses). A entidade transactions concentra tanto receitas quanto despesas, diferenciadas por um campo de tipo (income ou expense), evitando a duplicação de estrutura. As categorias formam uma tabela de referência fixa, populada por meio de um script de seed que associa, a cada categoria, um conjunto de palavras-chave utilizado pelo serviço de categorização automática.

Os relacionamentos seguem o padrão um-para-muitos: um usuário possui múltiplas transações, importações e análises; uma categoria classifica múltiplas transações; e um registro de importação pode originar múltiplas transações. Toda transação carrega obrigatoriamente o identificador do usuário e da categoria, e opcionalmente o identificador do registro de importação.

3.5 Categorização automática por palavras-chave

A categorização automática das transações foi implementada inteiramente em código local, sem recurso à inteligência artificial. Optou-se por essa abordagem por três razões: (i) garantir resposta em tempo determinístico, sem dependência de serviço externo;

(ii) eliminar custos por requisição em uma operação executada para cada transação cadastrada ou importada; e (iii) preservar a privacidade do usuário, evitando o envio de descrições individuais — que podem conter informações sensíveis — para um modelo externo.

O mecanismo consiste em um dicionário com quarenta entradas mínimas, no qual termos comuns em extratos bancários brasileiros — como "uber", "ifood", "netflix", "spotify", "salário" e "posto" — são associados a categorias específicas. Para cada nova transação, a descrição é normalizada e comparada ao dicionário; havendo correspondência, a categoria é sugerida automaticamente e o usuário pode confirmá-la ou alterá-la. Caso nenhuma palavra-chave seja identificada, a transação é classificada como "Outros".

3.6 Integração com a inteligência artificial generativa

A integração com o modelo Gemini é restrita ao serviço de análise mensal. Quando o usuário solicita a geração de uma análise, o backend consulta o banco para verificar a existência de uma análise recente para o mesmo período: havendo registro com menos de vinte e quatro horas de existência, o conteúdo é retornado diretamente do cache, sem nova chamada externa. Caso contrário, o serviço agrega os totais de receita, de despesa e o breakdown por categoria, monta um prompt estruturado em português e envia a requisição à API, com tempo máximo de espera de quinze segundos.

A política de envio é deliberadamente restritiva: nenhum dado pessoal identificável (nome, e-mail, descrição individual de transação) é enviado ao serviço externo. O modelo recebe apenas valores agregados, suficientes para produzir uma interpretação textual genérica. O resultado é persistido na entidade `ai_analyses`, juntamente com os totais utilizados no momento da geração, e fica disponível para consulta posterior sem nova requisição à API. Aplica-se ainda um limite de cinco requisições por hora por usuário, mitigando custos e potenciais usos indevidos.

3.7 Estratégia de implantação

A implantação foi planejada para utilizar exclusivamente camadas gratuitas. O frontend é compilado pelo Netlify a partir do repositório no GitHub e servido por meio de uma URL pública com certificado HTTPS automático. O backend é publicado como Web Service no Render, também conectado ao GitHub, com variáveis de ambiente configuradas diretamente no painel da plataforma — incluindo a URL do banco, o segredo do JWT e a chave de API do Gemini. O banco PostgreSQL é provisionado no Neon, com camada gratuita permanente e conexão SSL obrigatória. Essa combinação atende plenamente ao perfil de uso projetado para o sistema, dimensionado para um ou dois usuários simultâneos.

4 Resultados e Discussão

Esta seção descreve os resultados obtidos com o desenvolvimento e a implantação do FinancelA. São apresentadas, em sequência, as funcionalidades efetivamente entregues, a avaliação do mecanismo de categorização local, as características das análises produzidas pelo modelo generativo, as medidas de segurança adotadas e, ao final, uma discussão crítica sobre o alcance e as limitações da solução proposta.

4.1 Funcionalidades entregues

O escopo definido para o projeto foi integralmente cumprido. Foram implementadas as nove telas previstas — Login, Cadastro, Dashboard, Receitas, Despesas, Importação CSV, Relatórios, Análise IA e Perfil —, todas responsivas, navegáveis e integradas à API do backend. O Quadro 2 resume as funcionalidades por módulo e o status final de implementação.

Quadro 2 – Síntese das funcionalidades implementadas no FinanceIA

Módulo	Funcionalidade principal	Status
Autenticação	Cadastro, login e validação de JWT (24h)	Implementado
Usuários	Visualização e edição de perfil; troca de senha	Implementado
Receitas	CRUD com filtros, busca e paginação	Implementado
Despesas	CRUD com filtros, busca e paginação	Implementado
Categorias	Listagem fixa e sugestão por palavras-chave	Implementado
Importação CSV	Upload, prévia categorizada e confirmação	Implementado
Dashboard	Indicadores e gráficos de pizza e barras	Implementado
Relatórios	Intervalo livre + exportação em PDF	Implementado
IA	Análise mensal por Gemini com cache de 24h	Implementado

Fonte: Elaborado pelos autores (2026).

O painel principal sintetiza, para o período selecionado, quatro indicadores numéricos — receita total, despesa total, saldo e quantidade de transações — acompanhados de dois gráficos: um de pizza, exibindo a distribuição percentual dos gastos por categoria, com agrupamento automático das categorias menos relevantes em "Outros"; e um de barras agrupadas, comparando receitas e despesas dos seis meses mais recentes.

4.2 Avaliação da categorização automática local

O dicionário de palavras-chave foi populado com mais de quarenta entradas, cobrindo termos recorrentes em extratos bancários brasileiros e organizados por categoria de origem. Em testes realizados com arquivos CSV exportados de instituições financeiras de uso cotidiano, observou-se taxa elevada de acerto na sugestão automática de categorias para transações cujas descrições continham termos reconhecidos. Para descrições genéricas ou pouco padronizadas, a transação é classificada como "Outros", cabendo ao usuário a correção manual antes da confirmação da importação.

A opção pela categorização local, em vez do recurso à inteligência artificial para essa tarefa, representou uma decisão arquitetural relevante: a abordagem adotada elimina latência adicional, custos por requisição e exposição de descrições individuais a serviços externos, fatores especialmente importantes em uma aplicação financeira. A inteligência artificial generativa permanece, portanto, restrita à camada interpretativa do sistema, na qual oferece valor analítico sem implicar riscos à privacidade do usuário.

4.3 Análise gerada por inteligência artificial generativa

O serviço de análise mensal apresentou comportamento estável durante a validação. Os textos gerados pelo modelo Gemini, em português, descrevem a situação financeira do período em poucas frases, identificam categorias com participação proporcionalmente elevada nos gastos e oferecem sugestões genéricas de equilíbrio orçamentário. As análises são produzidas a partir exclusivamente de valores agregados, conforme política de minimização de dados, e armazenadas em banco para consulta posterior.

O mecanismo de cache de vinte e quatro horas operou conforme previsto: solicitações consecutivas para o mesmo período retornaram diretamente do registro persistido, sem nova requisição à API externa. Esse comportamento foi confirmado pela ausência de variação no carimbo de criação da análise e pela inspeção dos registros do backend. O limite de cinco requisições por hora também foi validado, retornando o código HTTP 429 (Too Many Requests) quando excedido em testes manuais.

4.4 Aspectos de segurança

As medidas de segurança implementadas seguem as recomendações apresentadas no referencial teórico. As senhas são armazenadas exclusivamente como hashes bcrypt,

em consonância com Provos e Mazières (1999). Os tokens JWT, com validade de vinte e quatro horas, transportam apenas o identificador do usuário, e o segredo de assinatura é mantido em variável de ambiente, fora do código-fonte versionado. O controle de origem cruzada (CORS) restringe o acesso ao backend a uma única origem em produção. As consultas ao banco são todas parametrizadas por meio do Prisma, eliminando a possibilidade de injeção de SQL. Por fim, a chave de API do Gemini nunca trafega pelo frontend, permanecendo confinada ao servidor.

Em testes manuais conduzidos com duas contas distintas, verificou-se que um usuário autenticado é incapaz de acessar, editar ou excluir recursos de outro usuário, ainda que conheça os identificadores correspondentes. Esse isolamento decorre da extração consistente do identificador do usuário a partir do token JWT validado, prática adotada de modo sistemático em todas as rotas privadas do sistema.

4.5 Discussão

Os resultados obtidos confirmam que a combinação proposta entre um painel financeiro tradicional e uma camada interpretativa baseada em inteligência artificial generativa é tecnicamente viável e funcional dentro de um escopo acadêmico, utilizando exclusivamente recursos de hospedagem gratuitos. O custo de operação aproxima-se de zero para o perfil de uso projetado, fator relevante para a aplicabilidade da solução em contextos de baixo orçamento.

Cabe ressaltar, contudo, que o sistema foi projetado para um a dois usuários simultâneos e que decisões deliberadas deixaram fora do escopo funcionalidades como integração com o Open Finance, recuperação automatizada de senha e suporte a múltiplos usuários com papéis diferenciados. Tais escolhas foram tomadas em favor da consolidação de uma base mínima funcional, prioridade compatível com o caráter acadêmico do projeto, e estão registradas como trabalhos futuros nas considerações finais.

5 Conclusão

Este trabalho retomou o problema da dificuldade enfrentada por pessoas físicas em organizar suas finanças pessoais de forma simples, segura e compreensível. Em resposta a esse desafio, propôs e desenvolveu o FinancelA, uma aplicação web que reúne o cadastro manual de receitas e despesas, a importação de extratos em formato CSV, a categorização automática local por palavras-chave, um painel analítico com indicadores e gráficos e, como diferencial, uma análise mensal em linguagem natural produzida por um modelo de inteligência artificial generativa.

O objetivo geral proposto foi alcançado. O sistema desenvolvido cumpre integralmente o escopo previsto, com nove telas funcionais, integração estável com a API Gemini do Google e implantação efetiva em ambiente público, utilizando exclusivamente camadas gratuitas de hospedagem (Netlify, Render e Neon). Os objetivos específicos relativos à segurança — autenticação por JWT, hash de senhas com bcrypt, restrição de origem por CORS, controle de taxa e isolamento estrito de dados entre usuários — foram igualmente atendidos.

Entre os principais resultados, destacam-se: a entrega de uma interface responsiva e organizada em torno de um painel central; a implementação de um mecanismo determinístico de categorização local, capaz de operar sem dependência de serviços externos; a geração de análises textuais coerentes a partir de dados exclusivamente agregados, preservando a privacidade do usuário; e a adoção sistemática de boas práticas de segurança em todas as camadas da aplicação. O sistema demonstrou ser capaz não apenas de armazenar transações, mas de traduzir números em interpretações compreensíveis, contribuindo para a finalidade educativa que motivou a pesquisa.

Algumas limitações devem ser reconhecidas. A solução foi dimensionada para um pequeno número de usuários simultâneos e não contempla, em sua versão atual, integração com o Open Finance, aplicativo móvel nativo, recuperação automatizada de senha ou perfis administrativos diferenciados. Adicionalmente, a hibernação automática do plano gratuito do Render introduz latência inicial em períodos de baixa utilização, característica conhecida e aceita no contexto acadêmico do projeto.

Como trabalhos futuros, sugerem-se: a implementação da recuperação de senha por correio eletrônico; a integração com o ecossistema Open Finance para sincronização automatizada de extratos; o desenvolvimento de uma versão mobile nativa, complementar à versão web responsiva já entregue; a introdução de testes automatizados; e a evolução do mecanismo de categorização para uma abordagem híbrida, capaz de aprender com correções realizadas pelo usuário.

Para além das contribuições técnicas, espera-se que o estudo apresente também relevância social, ao demonstrar a viabilidade de uma ferramenta financeira pessoal acessível e gratuita. Em um cenário de elevado endividamento das famílias brasileiras, instrumentos que reduzem o atrito do controle financeiro podem contribuir, ainda que de modo incremental, para o fortalecimento da educação financeira no país. Do ponto de vista acadêmico, o desenvolvimento do FinancelA permitiu ao grupo articular conhecimentos de engenharia de software, banco de dados, arquitetura de aplicações web, segurança da informação e inteligência artificial em torno de um produto único, coerente e plenamente funcional, satisfazendo os requisitos do componente curricular.

Referências

BRASIL. Lei nº 13.709, de 14 de agosto de 2018. Lei Geral de Proteção de Dados Pessoais (LGPD). Diário Oficial da União, Brasília, DF, 15 ago. 2018.

CERBASI, G. Como organizar sua vida financeira: inteligência financeira pessoal na prática. 2. ed. Rio de Janeiro: Sextante, 2022.

CONFEDERAÇÃO NACIONAL DO COMÉRCIO DE BENS, SERVIÇOS E TURISMO (CNC). Pesquisa de Endividamento e Inadimplência do Consumidor (PEIC) – Relatório Anual. Rio de Janeiro: CNC, 2024.

FOWLER, M. Patterns of Enterprise Application Architecture. Boston: Addison-Wesley, 2002.

GIL, A. C. Métodos e técnicas de pesquisa social. 7. ed. São Paulo: Atlas, 2019.

GOOGLE. Gemini API: Reference Documentation. Mountain View: Google, 2025. Disponível em: <https://ai.google.dev/gemini-api/docs>. Acesso em: 14 jun. 2026.

JONES, M.; BRADLEY, J.; SAKIMURA, N. JSON Web Token (JWT). RFC 7519. Internet Engineering Task Force (IETF), 2015.

ORGANIZAÇÃO PARA A COOPERAÇÃO E DESENVOLVIMENTO ECONÔMICO (OCDE). OECD/INFE 2020 International Survey of Adult Financial Literacy. Paris: OECD Publishing, 2020.

OPEN WEB APPLICATION SECURITY PROJECT (OWASP). OWASP Top 10: 2021 – The Ten Most Critical Web Application Security Risks. [S.l.]: OWASP Foundation, 2021.

PROVOS, N.; MAZIÈRES, D. A future-adaptable password scheme. In: USENIX ANNUAL

TECHNICAL CONFERENCE, 1999, Monterey. Proceedings [...]. Berkeley: USENIX Association, 1999. p. 81-92.

RICHARDS, M.; FORD, N. Fundamentals of Software Architecture: an engineering approach. Sebastopol: O'Reilly Media, 2020.

SOMMERVILLE, I. Engenharia de software. 10. ed. São Paulo: Pearson Education do Brasil, 2019.

STALLINGS, W. Criptografia e segurança de redes: princípios e práticas. 6. ed. São Paulo: Pearson, 2015.

VASWANI, A. *et al.* Attention is all you need. *In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS (NeurIPS)*, 31., 2017, Long Beach. Proceedings [...]. [S.l.]: NeurIPS, 2017. p. 5998-6008.