

G.COM: Sistema de Controle de Gastos Financeiros Pessoais

Gabriel C. O. Moura, Daiane M. Tomazet¹

¹Instituto Federal de Educação, Ciência e Tecnologia de São Paulo,
Câmpus Hortolândia (IFSP)
Caixa Postal 13183-250 – Hortolândia – SP – Brazil

g.cerdan@ifsp.edu.br, daianetomazeti@ifsp.edu.br

Abstract. *This study discusses the relevance of financial management tools to assist the Brazilian population in planning and controlling their personal finances, especially in a context where indebtedness affects a significant portion of families. The web system G.COM was developed to enable users to register, monitor, and manage their incomes and expenses, as well as to set financial goals and analyze credits and pending payments. The system development process faced challenges, particularly regarding responsive design and interface usability, which were overcome through incremental approaches and constant testing throughout the development process.*

Resumo. *O presente trabalho discute a relevância de ferramentas de gerenciamento financeiro para auxiliar a população brasileira no planejamento e controle de suas finanças pessoais, em um contexto no qual o endividamento afeta grande parte das famílias. O sistema web G.COM foi desenvolvido para possibilitar aos usuários o cadastro, monitoramento e gerenciamento de receitas e despesas, além do estabelecimento de metas e análise de créditos e pendências. O processo de construção do sistema enfrentou desafios, sobretudo no que diz respeito ao design responsivo e à usabilidade da interface, os quais foram superados por meio de abordagens incrementais e testes constantes ao longo do desenvolvimento.*

1. Introdução

Um estudo da Confederação Nacional do Comércio (CNC) [Abdala 2023], divulgado em maio de 2023, aponta que 29,1% das famílias brasileiras estão endividadas, e que, em comparação entre maio e abril deste mesmo ano, teve um aumento de aproximadamente 1% do total de lares endividados do país.

Segundo dados da *InfoMoney* [Godoy 2022], uma das causas que levam muitas pessoas ao endividamento é a incapacidade de administrar o dinheiro, seja gastando o dinheiro que ainda não recebeu ou não sabendo para onde seu dinheiro está sendo destinado.

Com base nos dados pesquisados sobre o endividamento da população brasileira, este trabalho tem como objetivo, desenvolver um *software* financeiro, que auxilie as pessoas a visualizar e controlar seus gastos contábeis.

1.1. Objetivos

Nesta seção, serão estabelecidos os objetivos, ou seja, os fins norteadores do projeto, articulando de maneira integrada tanto o Objetivo Geral quanto os Objetivos Específicos.

Nesse contexto, o Objetivo Geral define a finalidade central do sistema G.COM, enquanto os Objetivos Específicos delineiam as metas operacionais e funcionais necessárias para a implementação da solução.

1.1.1. Objetivo Geral

Desenvolver um sistema *web* de controle financeiro, denominado **G.COM**, capaz de auxiliar o usuário no planejamento, monitoramento e gerenciamento de suas finanças pessoais, proporcionando maior clareza sobre receitas, despesas, créditos e metas financeiras.

1.1.2. Objetivos Específicos

Os objetivos específicos do projeto G.COM são delineados de forma a garantir uma solução completa e eficaz para o gerenciamento das finanças pessoais, alinhando cada funcionalidade aos requisitos do usuário. Nesse contexto, o sistema é concebido para integrar diversas funcionalidades que, em conjunto, promovem um controle financeiro robusto e orientado para a tomada de decisões informadas.

Em primeiro lugar, o sistema visa permitir o cadastro e o monitoramento detalhado das receitas e despesas, possibilitando que o usuário insira e acompanhe cada movimentação financeira com uma classificação clara e organizada. Essa funcionalidade é fundamental para facilitar a análise dos fluxos de caixa, proporcionando uma visão precisa do comportamento financeiro. Paralelamente, o software incorpora mecanismos para o estabelecimento de metas e objetivos financeiros, permitindo que o usuário defina, visualize e acompanhe parâmetros de gastos e economia, contribuindo para um controle orçamentário mais rigoroso.

Ademais, o projeto inclui a implementação de um módulo específico para o gerenciamento de créditos e pendências, o qual se propõe a analisar de maneira detalhada os créditos, as parcelas e as dívidas, fornecendo subsídios essenciais para uma tomada de decisão financeira mais informada. Por fim, o sistema se compromete a oferecer relatórios e gráficos personalizados, que consolidem os dados das transações de forma visual e intuitiva, permitindo ao usuário interpretar e analisar com profundidade sua situação financeira. Dessa forma, cada um desses objetivos específicos contribui para a concretização do objetivo geral, reforçando a eficácia do G.COM no apoio ao planejamento e controle das finanças pessoais.

2. Referencial Teórico

Esta seção integrará, principalmente, os conhecimentos relacionados ao desenvolvimento *web*, à arquitetura cliente-servidor e à gestão de bancos de dados, permitindo uma compreensão aprofundada dos desafios técnicos e das soluções implementadas.

2.1. Desenvolvimento Web

Desenvolvimento *Web* é o processo de criar e manter sistemas *web* que funcionam em diferentes plataformas e dispositivos. Uma parte fundamental desse processo é a implementação da arquitetura cliente-servidor [Hartl 2017b].

A arquitetura cliente-servidor é um modelo de *design* de sistema onde o servidor hospeda, entrega e mantém a maioria dos recursos e serviços para serem usados pelo cliente. Neste contexto, o cliente é um aplicativo ou sistema que acessa os serviços fornecidos pelo servidor [Gallotti 2016].

Um sistema *web* de controle de gastos financeiros pessoais, construído com a arquitetura cliente-servidor, é uma ferramenta que ajuda o usuário a gerenciar suas receitas e despesas, planejar seu orçamento e alcançar seus objetivos financeiros. O servidor pode armazenar e processar dados, enquanto o cliente, geralmente um navegador *web*, solicita esses dados e apresenta-os ao usuário.

A importância do desenvolvimento *web* para esse tipo de sistema é garantir que ele seja seguro, confiável, fácil de usar e acessível para o público-alvo. A arquitetura cliente-servidor permite uma distribuição eficiente dos recursos, facilitando a manutenção e a escalabilidade do sistema [G. Coulouris and Blair 2011b].

Além disso, o desenvolvimento *web* permite que o sistema se adapte às mudanças nas necessidades e preferências dos usuários, bem como às novas tecnologias e tendências do mercado. A arquitetura cliente-servidor suporta essa adaptabilidade, pois permite que atualizações e melhorias sejam implementadas no servidor sem afetar diretamente o cliente [G. Coulouris and Blair 2011a].

2.2. Banco de Dados

O sistema permitirá ao usuário registrar, acompanhar e analisar suas despesas e receitas.

Para que esse sistema funcione, é necessário armazenar e manipular os dados dos usuários de forma organizada e estruturada. É aí que entra o conceito de Banco de Dados, que é um conjunto de informações relacionadas entre si, que podem ser acessadas e modificadas por meio de programas específicos [Carvalho 2015].

Um Banco de Dados permite ao sistema *web* consultar, inserir, atualizar e excluir os dados dos usuários conforme suas necessidades e preferências. Além disso, um Banco de Dados também oferece recursos para garantir a integridade, a consistência e a segurança dos dados, evitando perdas, erros ou invasões [Oracle 2022].

Portanto, um Banco de Dados é um elemento fundamental para o desenvolvimento e o funcionamento de um sistema *web* de controle de gastos financeiros pessoais.

2.2.1. Modelo Conceitual

No modelo conceitual, as principais entidades identificadas foram Usuário, Crédito, Meta, Carteira e uma entidade intermediária que registra as Movimentações (receitas e despesas). As relações definem como cada entidade se conecta, por exemplo, um usuário pode ter várias metas, assumir diferentes créditos e registrar diversas movimentações. Esse estágio busca representar de forma abstrata as informações necessárias para atender aos requisitos do sistema. A **Figura 1** apresenta o modelo conceitual proposto para o G.COM:

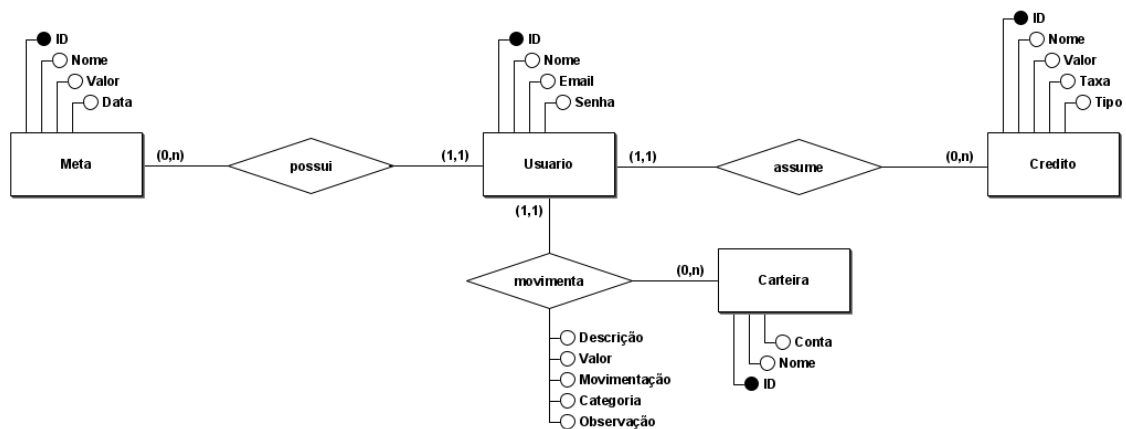


Figura 1. Diagrama apresentando os relacionamentos conceituais entre as entidades "Usuário", "Crédito" e "Meta".

2.2.2. Modelo Lógico

Na fase lógica, as entidades conceituais são traduzidas em tabelas com atributos e relacionamentos, considerando cardinalidades e chaves primárias e estrangeiras.

- **Usuário:** ID, nome, email, senha;
- **Crédito:** ID, nome, valor, taxa, tipo, data planejada, chave estrangeira para o usuário;
- **Meta:** ID, nome, valor, data planejada, chave estrangeira para o usuário;
- **Carteira:** ID, conta, nome, chave estrangeira para o usuário;
- **movimenta (ou Movimentações):** ID, descrição, valor, observação, categoria, data planejada, data realizada, chave estrangeira para usuário e carteira, entre outros.

As tabelas são normalizadas de modo a reduzir redundâncias e facilitar o gerenciamento das informações. Modelo Lógico utilizado:

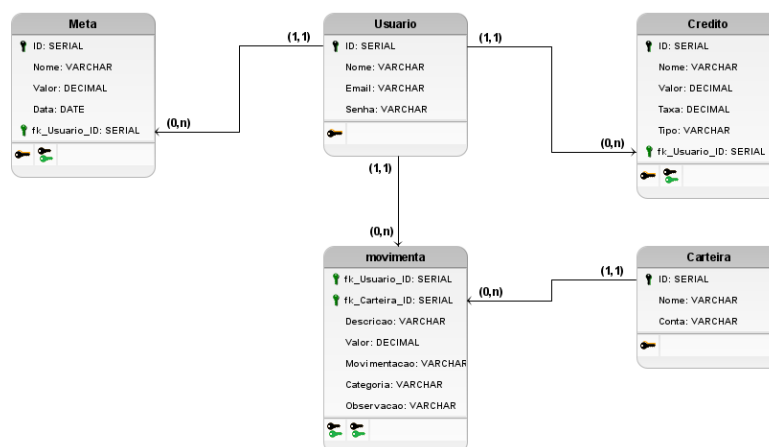


Figura 2. Diagrama apresentando os relacionamentos lógicos entre as entidades "Usuário", "Crédito" e "Meta".

2.2.3. Modelo Físico

No modelo físico, o banco de dados é implementado efetivamente no MySQL, com a criação de tabelas, definição de tipos de dados (por exemplo, *VARCHAR*, *DECIMAL*, *DATE*), índices e relacionamentos (chaves primárias e estrangeiras). Também são definidos aspectos de desempenho, como configuração de índices, e aspectos de segurança, como controle de acesso aos dados e permissões para diferentes usuários do sistema.

2.3. Padrão de Arquitetura MVC

O padrão de arquitetura MVC (*Model-View-Controller*) é uma forma de organizar o desenvolvimento de um sistema *web* que divide o código em três camadas: modelo, visão e controlador [Sommerville 2018b].

O modelo é responsável pela lógica de negócio e pela interação com os dados. A visão é a camada que apresenta a interface gráfica para o usuário. O controlador é o intermediário entre o modelo e a visão, recebendo as requisições do usuário e enviando as respostas adequadas.

Esse padrão pode trazer vários benefícios para o nosso sistema, como: facilitar a manutenção e a evolução do código, separando as responsabilidades de cada camada; aumentar a reusabilidade e a modularidade dos componentes, permitindo criar diferentes visões para o mesmo modelo; melhorar a testabilidade e a qualidade do código, isolando as dependências e os comportamentos de cada camada [E. Gamma and Vlissides 1994].

2.4. Tecnologias Utilizadas

O desenvolvimento de aplicações web modernas e robustas depende fortemente da integração harmoniosa de diversas tecnologias que possibilitam a construção de sistemas dinâmicos, seguros e escaláveis. Nesse sentido, as *Java Web Pages* (JSP) emergem como uma solução poderosa para a criação de conteúdos dinâmicos no lado do servidor. A utilização de JSP permite a incorporação de código Java diretamente em páginas HTML, facilitando a implementação do padrão arquitetural MVC.

2.4.1. JSP

Java Web Pages (JSP) é uma tecnologia amplamente utilizada no desenvolvimento de aplicações web dinâmicas. Ela permite a integração de conteúdo estático com componentes interativos, utilizando a plataforma Java [Oracle 2020]. A sua principal vantagem é a separação clara entre a lógica de negócios e a apresentação, o que facilita a manutenção e escalabilidade do sistema. Quando combinado com o padrão MVC, o JSP oferece uma estrutura bem definida em que a camada de controle gerencia as entradas do usuário e invoca a lógica de negócios, enquanto as páginas JSP rendem a interface visual com dados dinâmicos inseridos em componentes HTML. No contexto do G.COM, o uso de JSP permite uma entrega consistente e eficiente de dados entre o servidor e as páginas do usuário, garantindo uma navegação interativa e fluida.

2.4.2. MySQL

O MySQL é um dos sistemas de gerenciamento de banco de dados mais populares, baseado no modelo relacional [Erickson 2024]. Sua principal utilização em sistemas *web* está na organização estruturada e recuperação eficiente de dados, o que é essencial no gerenciamento de grandes volumes de informações, como as relacionadas às finanças pessoais, receitas, despesas, créditos e metas financeiras. Com a utilização do MySQL, é possível manter a integridade e consistência dos dados por meio do uso de transações, controle de concorrência e mecanismos de recuperação. Sua flexibilidade, juntamente com o phpMyAdmin, ferramenta de administração gráfica para MySQL, permite ao desenvolvedor realizar operações de forma rápida e eficaz, como o monitoramento de dados e execução de consultas complexas, além de facilitar o gerenciamento das bases de dados [Delisle 2012].

2.4.3. CSS

O uso de CSS (*Cascading Style Sheets*) é fundamental para atribuir estilo e formatação às páginas da aplicação [Ariane 2022]. A compatibilidade do CSS com as plataformas *web* permite que as páginas sejam desenvolvidas de maneira acessível e visualmente agradável, possibilitando a customização de todos os elementos da interface, como fontes, cores e layouts [Hartl 2017a]. A inclusão de *design* responsivo no desenvolvimento de páginas com CSS garante que o sistema G.COM seja acessível e funcional em diferentes dispositivos, como desktops, *tablets* e *smartphones*. O conceito de *design* responsivo trata-se de adaptar o *layout* da página conforme a resolução de tela do dispositivo do usuário, melhorando a experiência global. A utilização desse conceito é essencial em um sistema financeiro, onde o acesso contínuo e eficiente às funcionalidades da plataforma pode ocorrer a qualquer momento e lugar.

2.4.4. JavaScript

O JavaScript é uma das principais linguagens de programação do lado do cliente, sendo crucial no desenvolvimento de aplicações *web* ricas em interatividade e dinâmica [Flanagan 2012]. No sistema desenvolvido, a implementação de JavaScript permite realizar operações do lado do cliente sem a necessidade de interações constantes com o servidor, tornando o sistema mais rápido e responsivo. Funções como a validação de formulários, a atualização dinâmica de listas e gráficos financeiros, e a manipulação de elementos da interface em resposta a comandos do usuário são contempladas por JavaScript.

2.4.5. Bootstrap

O Bootstrap é um *framework open-source* que proporciona componentes reutilizáveis para a construção rápida de interfaces *web*. Desenvolvido pelo *Twitter*, o Bootstrap oferece uma coleção de componentes, modelos e estilos pré-definidos, a fim de facilitar a criação de layouts consistentes e atraentes [Bootstrap 2023]. No G.COM, ele foi utili-

zado para garantir que o sistema fosse acessível em diversos tipos de dispositivos e navegadores, independente de suas resoluções. O *design* modular e flexível desse *framework*, juntamente com as classes CSS predefinidas, elimina a necessidade de escrever código complexo, garantindo que o sistema tenha uma interface intuitiva e coerente.

2.4.6. NetBeans

É um Ambiente de Desenvolvimento Integrado (*IDE*) que facilita a criação de aplicações *web* e de sistemas em Java. Ele permite aos desenvolvedores escrever, compilar, depurar e testar seus códigos em um ambiente integrado e eficiente. Durante o desenvolvimento, esta ferramenta foi fundamental para acelerar o processo de codificação e testes em JSP [Netbeans 2019]. Juntamente com o *Java Development Kit* (JDK), ele possibilitou uma integração eficiente das tecnologias Java com os demais componentes do sistema.

2.4.7. Tomcat

Um servidor de aplicação de código aberto que implementa a especificação Java Servlet e *Java Server Pages* [Tomcat 1999]. Ele é amplamente utilizado para o desenvolvimento e testes de aplicações em Java, como no caso do G.COM. O Tomcat gerencia a execução da aplicação, controlando aspectos como a alocação de recursos e a segurança, além de permitir que os desenvolvedores realizem o *deploy* de novas versões da aplicação com facilidade.

2.5. Requisitos Funcionais

Os requisitos funcionais são descrições do comportamento que o *software* deve ter em termos das tarefas ou serviços do usuário. Eles definem as funcionalidades específicas que o *software* deve ser capaz de executar para atender às necessidades do usuário [Vazquez and Travassos 2016]. Em outras palavras, os requisitos funcionais descrevem o que o *software* deve fazer.

No sistema G.COM, foram aprofundados de maneira a evidenciar a robustez e a abrangência das funcionalidades implementadas. Nesse sentido, o sistema foi concebido para permitir o cadastro minucioso de todas as receitas e despesas, registrando não apenas os valores, mas também as origens, a natureza e as datas correspondentes a cada transação. Essa abordagem detalhada garante a organização e a rastreabilidade das informações, essenciais para a análise crítica do fluxo financeiro. Além disso, o controle do saldo de caixa é implementado de forma a oferecer uma visão integrada e atualizada das entradas e saídas de recursos, facilitando a identificação de eventuais desequilíbrios orçamentários. Outro aspecto central é a geração de relatórios personalizados, os quais possibilitam ao usuário obter informações detalhadas sobre suas movimentações financeiras, promovendo uma compreensão aprofundada do comportamento econômico e contribuindo para decisões mais embasadas. Complementarmente, o sistema contempla o gerenciamento das contas a pagar e a receber, proporcionando uma visão prospectiva das obrigações e dos recebimentos, o que é fundamental para o planejamento financeiro. Por fim, o estabelecimento de metas orçamentárias e objetivos financeiros integra uma funcionalidade estratégica, incentivando os usuários a definir parâmetros claros para a economia e o equilíbrio

dos gastos, alinhando o uso do sistema com práticas de gestão financeira sustentáveis e eficazes.

2.6. Requisitos Não-Funcionais

Os requisitos não funcionais são descrições dos atributos de qualidade que o *software* deve ter, como desempenho, segurança, usabilidade, confiabilidade, escalabilidade, entre outros. Eles definem as características que o *software* deve apresentar para atender às necessidades do usuário em termos de qualidade [Vazquez and Travassos 2016].

Diferentemente dos requisitos funcionais, que descrevem o que o *software* deve fazer, os requisitos não funcionais descrevem como o *software* deve fazer [Sommerville 2018a]. Eles são profundamente afetados pela arquitetura da solução e pelas decisões arquiteturais tomadas no início do ciclo de vida do *software*.

No sistema G.COM, foram descritos com ênfase nos atributos de qualidade que sustentam a eficiência e a experiência do usuário. Primeiramente, a usabilidade se destaca como um dos pilares do sistema, exigindo que a interface seja intuitiva e de fácil navegação, permitindo que usuários de diferentes perfis possam controlar suas finanças de maneira simples e eficiente, sem a necessidade de extensos treinamentos. Em complemento, o desempenho é rigorosamente considerado, de forma que o processamento de dados seja realizado de maneira ágil, garantindo respostas rápidas e evitando atrasos que possam comprometer a análise financeira em tempo real. Adicionalmente, a portabilidade é um requisito importante, pois o sistema deve operar de forma consistente em uma variedade de dispositivos e sistemas operacionais, incluindo computadores, tablets e smartphones, assegurando que o acesso às funcionalidades seja ininterrupto e adaptável às diversas condições de uso. Dessa forma, os requisitos não funcionais reforçam a robustez e a flexibilidade do G.COM, assegurando uma experiência de uso que alia eficiência, praticidade e acessibilidade.

2.7. Personas e Cenários

Personas são representações fictícias e generalizadas de um grupo de usuários idealizados que representam as necessidades de um grupo maior de usuários. Em *design* e desenvolvimento de produtos, as personas são usadas para ajudar a equipe a entender e se conectar com os usuários finais do produto. As personas geralmente incluem detalhes como comportamento, atitudes, dores e objetivos [Nielsen 2007].

Por exemplo, uma persona para um aplicativo de controle de gastos financeiros pessoais pode ser "João, um profissional de 30 anos que luta para economizar dinheiro e precisa de ajuda para gerenciar suas finanças pessoais".

Cenários, por outro lado, são histórias ou narrativas que descrevem como as personas interagem com produtos em um contexto específico [Nielsen 2007]. Os cenários ajudam a equipe de *design* a visualizar como um usuário usará o produto na vida real e podem ajudar a identificar problemas ou oportunidades de *design*.

Continuando com o exemplo acima, um cenário para a persona João pode ser: "João recebe seu salário e precisa dividir o dinheiro entre suas contas, economias e despesas diárias. Ele abre o aplicativo de controle de gastos financeiros pessoais e começa a alocar seu dinheiro".

A construção de personas e cenários de uso para o sistema G.COM possibilita uma compreensão aprofundada dos diferentes perfis de usuários e de suas respectivas necessidades financeiras. Ao estruturar essas representações, busca-se evidenciar como o sistema pode oferecer suporte eficiente e personalizado para desafios reais enfrentados por indivíduos de distintas faixas etárias, ocupações e contextos socioeconômicos. Dessa forma, a definição dessas personas fundamenta a concepção das funcionalidades do G.COM, garantindo que o sistema atenda de maneira prática e acessível às demandas específicas de cada usuário. A partir dessa abordagem, pode-se analisar em detalhes como o sistema auxilia no gerenciamento financeiro, iniciando pela experiência da empresária Ana, que enfrenta dificuldades na organização contábil de seu negócio e necessita de uma ferramenta intuitiva para facilitar a geração de relatórios financeiros.

A primeira persona é representada pela empresária Ana, uma profissional autônoma de 35 anos, proprietária de uma pequena empresa. Ana enfrenta desafios típicos da gestão financeira, sobretudo por não possuir uma vasta experiência em contabilidade, o que a leva a buscar uma ferramenta que ofereça simplicidade e clareza na apresentação dos dados. Em seu cenário, ela utiliza o G.COM para gerar um relatório detalhado das receitas e despesas referentes ao último trimestre, informação crucial para a prestação de contas ao seu contador. Essa funcionalidade não apenas facilita a análise dos fluxos de caixa, mas também contribui para a tomada de decisões estratégicas, garantindo a saúde financeira do negócio.

A segunda persona descreve o estudante João, um universitário de 20 anos que, além dos compromissos acadêmicos, trabalha meio período para se sustentar. Com um orçamento limitado, João necessita de um controle rigoroso de suas finanças para assegurar o cumprimento de suas obrigações sem comprometer os recursos destinados à sua formação. Nesse contexto, o G.COM torna-se uma ferramenta essencial, pois permite que ele verifique, de maneira rápida e eficiente, se dispõe de saldo suficiente para o pagamento de contas essenciais, como a conta de luz, evidenciando o valor de uma gestão financeira organizada para a manutenção do equilíbrio pessoal.

No caso dos recém-casados, o casal formado por André e Juliana, ambos com 30 anos e atuantes nas áreas de engenharia e medicina, respectivamente, exemplifica o perfil de usuários que necessitam de uma abordagem integrada para o gerenciamento de suas finanças. Ao unirem suas fontes de renda e despesas, eles enfrentam o desafio de equilibrar os gastos domésticos e planejar projetos conjuntos. Em seu cenário típico, o casal recorre ao G.COM para realizar um planejamento financeiro que lhes permita reservar recursos para uma viagem de férias, consultando os saldos individuais e, a partir dessa análise, definindo um orçamento que contemple tanto as obrigações cotidianas quanto os objetivos de lazer.

A quarta persona é representada por Maria, uma dona de casa de 35 anos, cuja principal responsabilidade é a administração das finanças do lar. Diante do desafio de gerir os recursos familiares, Maria busca uma solução que simplifique o acompanhamento dos gastos e a organização dos pagamentos essenciais, como as mensalidades escolares de seus filhos. Utilizando o G.COM, ela consegue verificar de forma imediata se o saldo de caixa disponível é suficiente para arcar com essas despesas, o que demonstra como o sistema pode ser um instrumento indispensável para manter a estabilidade financeira e assegurar o bem-estar da família.

Por fim, o empreendedor Pedro, com 27 anos e atuante no meio digital, representa um perfil de usuário que valoriza a eficiência e a agilidade na administração de suas finanças. Gerenciando um negócio online, Pedro dispõe de tempo limitado para a realização de análises financeiras detalhadas, necessitando de um sistema que ofereça informações rápidas e precisas sobre as entradas e saídas de recursos. Em seu cenário de uso, ele recorre ao G.COM para avaliar se possui recursos suficientes para a aquisição de equipamentos necessários ao seu empreendimento. Ao verificar o saldo bancário e comparar com as despesas previstas, Pedro pode decidir de forma embasada se a compra pode ser realizada imediatamente ou se será mais prudente aguardar uma melhoria na situação financeira.

2.8. Testes de Funcionalidade

Os testes de funcionalidade configuram uma abordagem metódica e sistemática para a validação do comportamento do sistema, concentrando-se na verificação de que cada fluxo e componente operacional atende rigorosamente aos requisitos previamente estabelecidos [Sommerville 2018c]. Essa modalidade de teste envolve a execução de casos de uso que simulam a interação do usuário com a aplicação, abrangendo processos que vão desde o cadastro e autenticação até a realização de transações e a geração de relatórios financeiros. Eles permitem a execução repetida e contínua destes processos críticos, proporcionando uma detecção precoce de inconsistências e falhas, o que, por sua vez, reforça a robustez e confiabilidade do *software*.

3. Trabalhos Correlatos

Existem diversos trabalhos correlatos na área de sistemas de controle de gastos financeiros, que podem ser classificados em três categorias principais: aplicativos móveis, planilhas eletrônicas e sistemas *web*.

Os aplicativos móveis são soluções que permitem aos usuários registrar e acompanhar seus gastos financeiros por meio de dispositivos móveis, como *smartphones* e *tablets*. Esses aplicativos geralmente oferecem recursos como categorização de despesas, gráficos e relatórios, alertas e lembretes, sincronização com contas bancárias e cartões de crédito, entre outros. Alguns exemplos de aplicativos móveis de controle de gastos financeiros são: Minhas Finanças, Mobills, Organize e Minhas Economias.

As planilhas eletrônicas são ferramentas que permitem aos usuários criar e editar tabelas e gráficos para organizar e visualizar seus dados financeiros. Essas ferramentas geralmente requerem que os usuários insiram manualmente seus dados e fórmulas, o que pode demandar tempo e conhecimento técnico. Além disso, as planilhas eletrônicas podem apresentar limitações de armazenamento, compartilhamento e segurança dos dados. Alguns exemplos de planilhas eletrônicas de controle de gastos financeiros são: Excel, Google Sheets e LibreOffice Calc.

Os sistemas *web* são soluções que permitem aos usuários acessar e gerenciar seus gastos financeiros por meio de navegadores *web*, como Chrome, Firefox e Safari. Esses sistemas geralmente oferecem recursos semelhantes aos dos aplicativos móveis, mas com a vantagem de não dependerem da instalação de *software* nos dispositivos dos usuários. Além disso, os sistemas *web* podem oferecer maior capacidade de armazenamento, compartilhamento e segurança dos dados. Alguns exemplos de sistemas *web* de controle de gastos financeiros são: Meu Dinheiro e Granatum.

Tabela 1. Tabela de comparação de trabalhos correlatos

Aplicativo	Minhas Finanças	Mobills	Organizze	Minhas Economias	Meu Dinheiro	Granatum Financeiro	ContaAzul	G.COM
Gerenciador financeiro	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Controle de cartão de crédito	Não	Sim	Sim	Sim	Sim	Sim	Sim	Não
Contas digitais	Não	Sim	Não	Não	Não	Não	Não	Não
Empréstimos	Não	Não	Não	Não	Não	Não	Não	Sim
Planejador financeiro	Não	Sim	Não	Sim	Sim	Sim	Não	Sim
Investimentos	Não	Não	Não	Não	Não	Não	Não	Sim

4. Ferramentas de Desenvolvimento

Inicialmente, será executada a pesquisa bibliográfica relacionada ao assunto do sistema, como Desenvolvimento *Web*, Desenvolvimento para Dispositivos Móveis, Persistência de Dados, etc.

Será conduzida uma busca por aplicativos que visam atender à mesma necessidade. Através dessa pesquisa, será possível identificar quais aspectos foram abordados, quais são os pontos que ainda não foram resolvidos e possíveis aprimoramentos.

A abordagem metodológica adotada será incremental, em que cada funcionalidade será planejada, modelada de acordo com os requisitos refinados, desenvolvida, testada e, por fim, implementada ao *software*.

5. Desenvolvimento do trabalho

Esta seção são descritas, de forma sistemática, as etapas e metodologias que nortearam a execução do projeto. A partir da definição dos requisitos funcionais e não funcionais, passando pela implementação e pelos testes de funcionalidade, esta seção ilustra como cada componente do sistema G.COM foi planejado e desenvolvido. Tal abordagem evidencia a integração entre os diversos processos e tecnologias, demonstrando a coerência e a eficácia das soluções adotadas para atender às demandas identificadas com as personas e cenários citados anteriormente.

5.1. Materiais e Métodos

Serão utilizadas, como material de desenvolvimento, as linguagens Java, afim da programação da lógica da aplicação, MySQL, como sistema gerenciador de banco de dados, HTML, para o corpo das páginas *Web*, CSS, aplicando estilo visual e PHP para construção de gráficos.

5.2. Arquitetura e Tecnologias Utilizadas

O G.COM foi desenvolvido com base na arquitetura de aplicações *web* em Java, seguindo o paradigma de separação de camadas (como o padrão MVC). Na camada de *Model*, concentram-se as regras de negócio e a comunicação com o banco de dados. A camada de *View* é composta pelas páginas JSP (*Java Server Pages*) estilizadas com CSS e Bootstrap, oferecendo interfaces responsivas e atraentes. Já a camada de *Controller* recebe as requisições, processa as regras necessárias e interage com o modelo para gerar as respostas adequadas ao usuário.

Para o desenvolvimento e testes, foi utilizado o NetBeans 18, em conjunto com o Tomcat 8.5.96 como servidor de aplicação. O banco de dados adotado foi o MySQL,

gerenciado pelo XAMPP 3.3.0 e acessado via phpMyAdmin 5.2.1. A linguagem JavaScript foi empregada na camada de *front-end* para aprimorar a interatividade das páginas e, quando necessário, manipular dinamicamente elementos da interface. Além disso, foi incorporado o Bootstrap 5.3.3 para garantir *design* responsivo e padronizado. O projeto foi compilado com o JDK 20, assegurando compatibilidade com as funcionalidades mais recentes da linguagem Java.

5.3. Protótipo

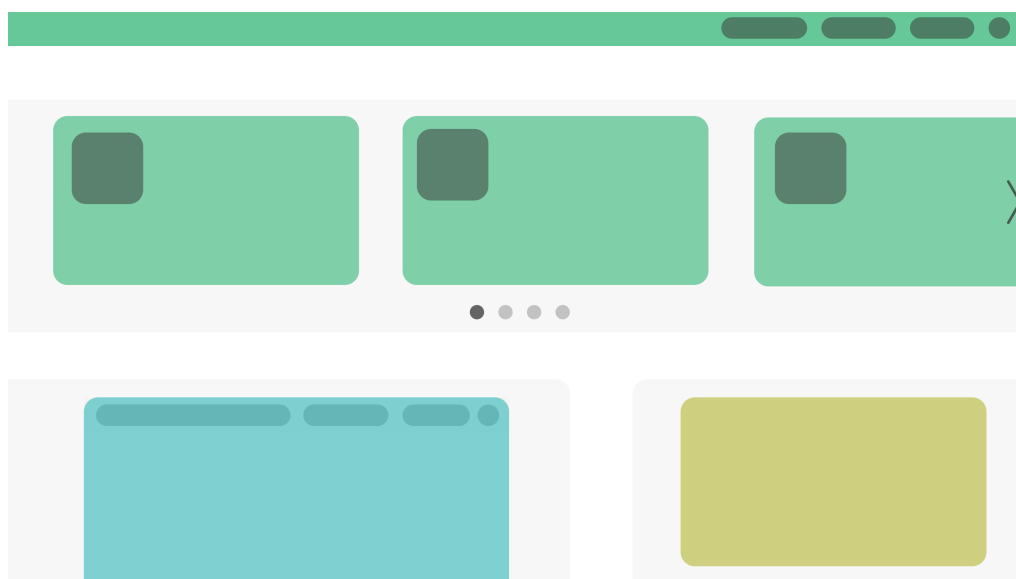


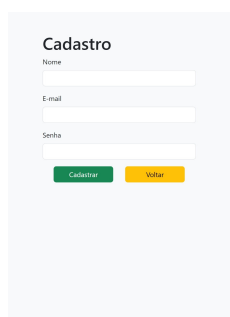
Figura 3. Protótipo da tela principal da aplicação.

A **Figura 3** apresenta o protótipo da tela principal do sistema G.COM, denominada "Resumo". Nesta interface, é possível visualizar de forma integrada os principais indicadores da situação financeira do usuário, tais como os saldos das carteiras, o progresso das metas estabelecidas e os detalhes das movimentações programadas para os próximos 30 dias. O design orientado por princípios de usabilidade e responsividade evidencia a clareza e a acessibilidade das informações, facilitando uma análise rápida e precisa do cenário financeiro, aspecto fundamental para a tomada de decisões estratégicas.

5.4. Telas

Esta seção apresenta uma visão abrangente das interfaces gráficas que compõem o sistema G.COM, detalhando a organização e o design de cada módulo que interage com o usuário. Ao descrever as funcionalidades e a estrutura visual das telas – desde o cadastro e login até o gerenciamento de movimentações e carteiras – esta parte do trabalho evidencia como os princípios de usabilidade, responsividade e consistência foram aplicados para garantir uma experiência de usuário intuitiva e eficiente. Dessa forma, a análise das telas reforça a importância do design na promoção de uma interface que facilita o controle financeiro e a interação com o sistema.

5.4.1. Cadastro

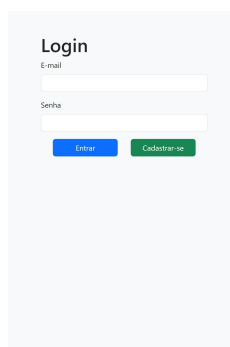


The image shows a mobile registration form titled 'Cadastro'. It contains three text input fields labeled 'Nome', 'Email', and 'Senha'. At the bottom of the form, there are two buttons: a green button labeled 'Cadastrar' and a yellow button labeled 'Validar'.

Figura 4. Tela de cadastro da aplicação em versão para dispositivos móveis.

A interface de cadastro foi concebida para viabilizar o registro inicial do usuário no sistema, permitindo a criação de uma conta personalizada. Nesta tela, o usuário fornece informações essenciais – como nome, e-mail e senha – que são submetidas a um processo de validação, garantindo a integridade dos dados e a segurança do acesso. Funcionalmente, esta etapa possibilita a confirmação do registro, bem como o armazenamento das credenciais, estabelecendo a base para uma experiência personalizada desde o primeiro acesso.

5.4.2. Login



The image shows a mobile login form titled 'Login'. It contains two text input fields labeled 'Email' and 'Senha'. At the bottom of the form, there are two buttons: a blue button labeled 'Entrar' and a green button labeled 'Cadastrar-se'.

Figura 5. Tela de login da aplicação em versão para dispositivos móveis.

A tela de login atua como o ponto de entrada para o ambiente do G.COM, onde o usuário insere suas credenciais previamente cadastradas (e-mail e senha) para autenticação. Este mecanismo de acesso não só valida os dados informados, mas também previne acessos não autorizados. Entre as funcionalidades desta interface, destaca-se a exibição de mensagens de alerta em caso de tentativas de acesso inválidas, reforçando a proteção dos dados pessoais.

5.4.3. Resumo

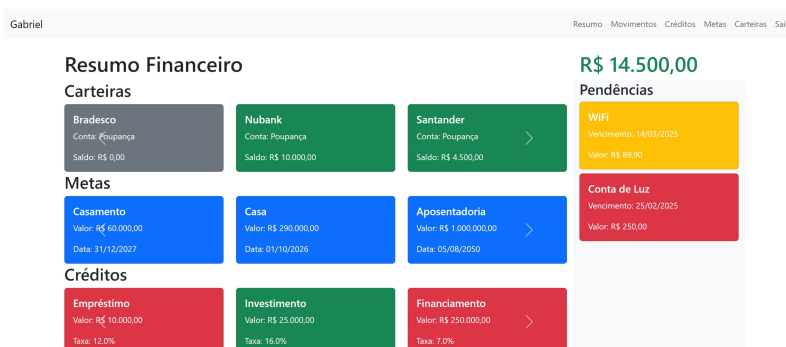


Figura 6. Tela de resumo da aplicação.

A tela de Resumo do sistema G.COM apresenta uma disposição intuitiva e informativa, organizada em dois quadrantes que facilitam a visualização dos dados financeiros. Nos três carrosséis localizados à esquerda, que oferecem um panorama resumido das Carteiras, Metas e Créditos, utiliza-se uma codificação de cores para otimizar a interpretação das informações: a cor vermelha sinaliza saldos negativos, a cor verde destaca os saldos positivos, a cor azul identifica os objetivos financeiros e a cor cinza indica a neutralidade de saldos zerados. Paralelamente, o quadrante da lateral direita exibe cards referentes às pendências, onde as obrigações com vencimento dentro de 30 dias são apresentadas em amarelo, evidenciando a proximidade do vencimento, e as pendências já atrasadas são ressaltadas em vermelho, reforçando a necessidade de ações imediatas.

5.4.4. Movimentos

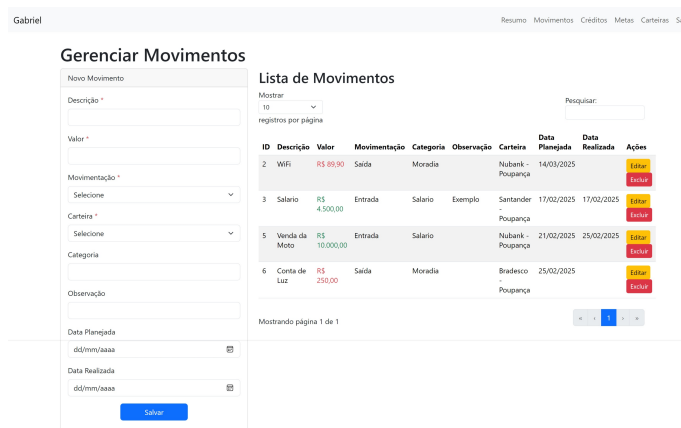


Figura 7. Tela de movimentos da aplicação.

Desenvolvida para o gerenciamento minucioso das transações financeiras, a tela de movimentos possibilita ao usuário a criação, edição e exclusão de registros. Cada movimento financeiro pode ser detalhadamente especificado – englobando dados como descrição, valor, tipo de movimentação, carteira associada, categoria, observações e datas

planejadas e realizadas. Dentre as funcionalidades desta tela, incluem-se a filtragem por categorias, a ordenação cronológica e a atualização imediata dos saldos, o que colabora para um controle rigoroso e transparente dos fluxos financeiros.

5.4.5. Créditos

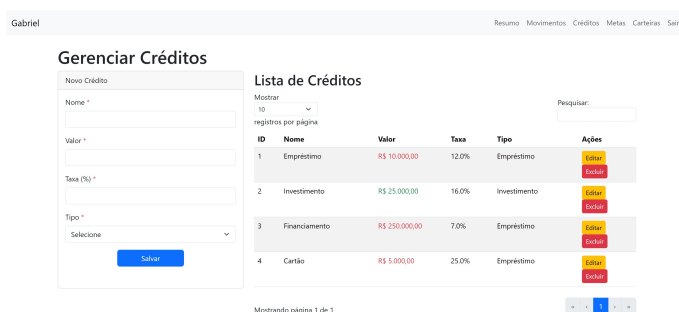


Figura 8. Tela de créditos da aplicação.

A interface de créditos foi especialmente desenhada para a administração de operações relativas a investimentos e empréstimos. Por meio desta tela, o usuário tem a possibilidade de inserir novos registros de crédito, bem como modificar ou excluir transações existentes. Cada operação é caracterizada por atributos específicos, tais como nome, valor, taxa de juros e tipo de crédito. Funcionalmente, esta tela incorpora mecanismos de validação e atualização dos saldos, além de permitir o acompanhamento do desempenho financeiro em relação a créditos e débitos.

5.4.6. Metas

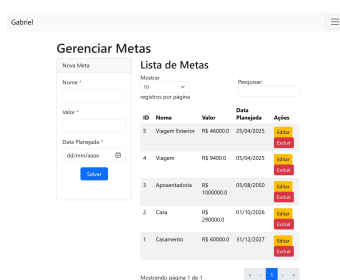


Figura 9. Tela de metas da aplicação em versão para tablets.

Destinada ao planejamento financeiro, a tela de metas permite ao usuário definir, monitorar e concluir objetivos econômicos. Nesta interface, o usuário pode estabelecer

novas metas de economia ou investimento, acompanhar o progresso através de indicadores gráficos e atualizar o status das metas à medida que os objetivos são alcançados. As funcionalidades aqui disponibilizadas incentivam a disciplina orçamentária, oferecendo também opções para a edição ou remoção de metas, o que contribui para a criação de um planejamento financeiro realista e motivador.

5.4.7. Carteiras

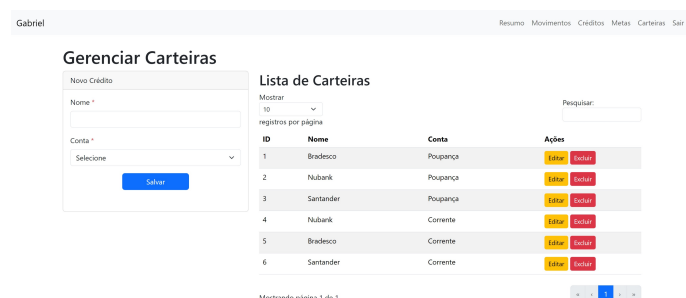


Figura 10. Tela de cadastro da aplicação.

A tela de carteiras é o módulo responsável pelo registro e gerenciamento das diversas contas ou carteiras financeiras do usuário. A interface permite que cada carteira seja cadastrada com atributos relevantes – como nome e identificação da conta – possibilitando a segmentação dos fluxos financeiros. Entre as funcionalidades oferecidas, destaca-se a integração com os demais módulos do sistema, de forma que os saldos e as movimentações sejam consolidados e apresentados de maneira segmentada, facilitando a visualização e o controle das diferentes fontes de recursos.

Cada uma dessas interfaces foi projetada considerando os princípios de usabilidade e responsividade, de modo a assegurar que o usuário possa navegar e interagir com o sistema de forma intuitiva e eficiente, independentemente do dispositivo utilizado. Dessa forma, o G.COM consolida-se como uma ferramenta robusta para o controle financeiro pessoal, atendendo às demandas de diversos perfis de usuários por meio de funcionalidades específicas e interligadas.

5.5. Testes de Funcionalidade

Durante a execução dos testes funcionais, foram replicados cenários do mundo real e trajetórias típicas de interação dos usuários, permitindo a validação prática de cada funcionalidade do sistema. Nesse processo, foram utilizados conjuntos específicos de entradas, correlacionadas com as saídas esperadas, para assegurar que o comportamento do software correspondia rigorosamente aos requisitos definidos. Essa abordagem sistemática possibilitou a verificação precisa das respostas geradas, facilitando a identificação e correção de eventuais inconsistências, e, assim, contribuindo para a robustez e confiabilidade da aplicação.

6. Conclusão

O desenvolvimento do G.COM evidencia a importância de ferramentas de controle financeiro para o contexto brasileiro, onde muitas famílias enfrentam dificuldades em organizar

suas contas e planejar gastos futuros. A implementação de um sistema *web*, acessível a partir de qualquer dispositivo conectado à internet, oferece maior comodidade e praticidade aos usuários, que passam a ter uma visão consolidada de suas receitas, despesas, metas e créditos pendentes.

Durante o processo de criação do sistema, foram utilizados os conhecimentos obtidos nas disciplinas Análise Orientada a Objetos, Banco de Dados, Desenvolvimento de Sistemas *Web*, Engenharia de *Software*, Programação Orientada a Objetos, entre outras. Os principais desafios envolveram o *design* responsivo e a usabilidade. Foi necessário estudar diferentes abordagens para *layout* adaptativo, assim como realizar testes de usabilidade para garantir que o fluxo de navegação fosse intuitivo, evitando sobrecarga de informações e passos desnecessários para o usuário. Esses obstáculos foram superados com a adoção de metodologias iterativas e utilização de *frameworks* como o Bootstrap, que simplificou a implementação de *layouts* responsivos.

Os benefícios do sistema desenvolvido incluem a possibilidade de melhor planejamento financeiro, uma vez que o usuário pode acompanhar seus gastos em tempo real e projetar cenários futuros, definindo metas de economia ou investimento. Além disso, a ferramenta permite ao usuário compreender melhor sua saúde financeira e tomar decisões mais embasadas, reduzindo riscos de endividamento e contribuindo para a construção de uma cultura de educação financeira.

O desenvolvimento do G.COM demonstrou ser uma contribuição efetiva no enfrentamento dos desafios de gestão financeira pessoal. Cada funcionalidade implementada foi pensada para responder às necessidades específicas das personas envolvidas no estudo. Assim, a empresária Ana, que demanda relatórios detalhados para a gestão de seu negócio, beneficia-se do módulo de geração de relatórios customizados, que consolida informações de receitas, despesas e créditos de forma clara e objetiva. Já o estudante João, com um orçamento restrito, utiliza as funcionalidades de monitoramento em tempo real e categorização de despesas, facilitando o controle rigoroso de seus gastos diários. No caso do casal formado por André e Juliana, a integração de múltiplas fontes de renda e a visualização conjunta dos saldos permitem um planejamento financeiro integrado, atendendo à necessidade de equilibrar gastos domésticos com objetivos comuns. Por fim, a dona de casa Maria e o empreendedor Pedro encontram na interface intuitiva e responsiva do sistema a ferramenta ideal para o acompanhamento simplificado das transações e para a tomada de decisões ágeis, demonstrando que a aplicação é capaz de adaptar-se a diferentes perfis e contextos. Dessa forma, o G.COM reafirma seu potencial de promover uma gestão financeira consciente e eficiente, contribuindo para a estabilidade econômica dos usuários por meio de soluções tecnológicas alinhadas às suas demandas específicas.

7. Projetos Futuros

Para aprimorar a usabilidade e escalabilidade do sistema, futuras pesquisas e desenvolvimentos podem focar na otimização da interface (*UI*) e na experiência do usuário (*UX*), tornando a navegação mais intuitiva e acessível a diferentes perfis. Pode-se explorar melhorias no *design* responsivo, garantindo compatibilidade aprimorada com dispositivos móveis e desktops, além da implementação de fluxos mais eficientes para o cadastro e gerenciamento de despesas.

Outro aspecto relevante a ser considerado é a personalização da interface, per-

mitindo que os usuários escolham temas visuais de acordo com suas preferências, como modos claro e escuro, além de opções avançadas de customização, incluindo organização de painéis, *dashboards* dinâmicos e *widgets* configuráveis.

Em relação à escalabilidade, pesquisadores podem estudar formas de aprimorar a arquitetura do sistema para suportar um volume crescente de usuários e transações financeiras sem comprometer o desempenho. Estratégias como a otimização de consultas ao banco de dados, adoção de técnicas de *caching* e implementação de serviços escaláveis em nuvem podem ser exploradas para esse fim.

Além disso, futuras implementações podem investigar a incorporação de funcionalidades baseadas em inteligência artificial, permitindo a geração de *insights* personalizados para os usuários. Algoritmos preditivos poderiam sugerir estratégias de economia e alertas financeiros inteligentes com base nos padrões de consumo dos usuários, agregando valor à experiência de gerenciamento financeiro.

Dessa forma, o G.COM pode continuar evoluindo por meio de novas contribuições acadêmicas e técnicas, garantindo um sistema cada vez mais eficiente, adaptável e alinhado às necessidades dos usuários.

Referências

- Abdala, V. (2023). Endividamento atinge 78,3% das famílias brasileiras, diz cnc. <https://pesquisascnc.com.br/pesquisa-peic/>. [Online; Acessado em 16 de maio de 2023].
- Ariane, G. (2022). O que é css? guia básico para iniciantes. <https://www.hostinger.com.br/tutoriais/o-que-e-css-guia-basico-de-css>. [Online; Acessado em 15 de junho de 2024].
- Bootstrap (2023). Get started with bootstrap. <https://getbootstrap.com/docs/5.3/getting-started/introduction/>. [Online; Acessado em 24 de julho de 2024].
- Carvalho, V. (2015). 1.2 banco de dados. In Pearson, editor, *MySQL: comece com o principal banco de dados open source do mercado*. São Paulo.
- Delisle, M. (2012). Server view. In Publishing, P., editor, *Mastering Phpmyadmin 3.4 for Effective MySQL Management*, pages 102–103. Birmingham.
- E. Gamma, R. Helm, R. J. and Vlissides, J. (1994). 1.6 how design patterns solve design problems. In Professional, A.-W., editor, *Design Patterns: Elements of Reusable Object-Oriented Software*, pages 18–19. Indiana.
- Erickson, J. (2024). Mysql: Entendendo o que é e como é usado. <https://www.oracle.com/br/mysql/what-is-mysql/>. [Online; Acessado em 09 de novembro de 2024].
- Flanagan, D. (2012). 13.7 estruturas do lado do cliente. In Bookman, editor, *JavaScript: O Guia Definitivo*, pages 330–331. Porto Alegre.
- G. Coulouris, J. Dollimore, T. K. and Blair, G. (2011a). 2.3.2: Architectural patterns. In Pearson, editor, *Distributed Systems: Concepts and Design*, pages 51–52. Indiana.

- G. Coulouris, J. Dollimore, T. K. and Blair, G. (2011b). 3.3: Network principles. In Pearson, editor, *Distributed Systems: Concepts and Design*. Indiana.
- Gallotti, A. (2016). Arquitetura cliente-servidor. In Pearson, editor, *Arquitetura de software*, pages 34–35. São Paulo.
- Godoy, T. (2022). Conheça as 5 alavancas do endividamento. <https://www.infomoney.com.br/colunistas/thiago-godoy/conheca-as-5-alavancas-do-endividamento/>. [Online; Acessado em 13 de setembro de 2022].
- Hartl, M. (2017a). 5.1.2 bootstrap and custom css. In Pearson, editor, *Ruby on Rails Tutorial: Learn Web Development with Rails*. Indiana.
- Hartl, M. (2017b). Chapter 1: From zero to deploy. In Pearson, editor, *Ruby on Rails Tutorial: Learn Web Development with Rails*, pages 1–3. Indiana.
- Netbeans (2019). Code assistance in the netbeans ide java editor: A reference guide. <https://netbeans.apache.org/tutorial/main/kb/docs/java/editor-codereference/>. [Online; Acessado em 27 de fevereiro de 2025].
- Nielsen, J. (2007). In LTC, G., editor, *Usabilidade na web: Projetando Websites com Qualidade*. São Paulo.
- Oracle (2020). Javaserer pages technology. <https://www.oracle.com/java/technologies/jspt.html>. [Online; Acessado em 05 de agosto de 2023].
- Oracle (2022). What is a database. <https://www.oracle.com/database/what-is-database/>. [Online; Acessado em 06 de setembro de 2022].
- Sommerville, I. (2018a). 4.1. requisitos funcionais e não funcionais. In Pearson, editor, *Engenharia de software*, pages 88–94. São Paulo.
- Sommerville, I. (2018b). 6.4. o padrão mvc (modelo-visão-controlador). In Pearson, editor, *Engenharia de software*, pages 155–156. São Paulo.
- Sommerville, I. (2018c). 8. testes de software. In Pearson, editor, *Engenharia de software*, pages 203–204. São Paulo.
- Tomcat (1999). Apache tomcat. <https://tomcat.apache.org/>. [Online; Acessado em 27 de fevereiro de 2025].
- Vazquez, C. and Travassos, G. (2016). 5.9. requisitos funcionais. In Brasport, editor, *Engenharia de Requisitos: software orientado ao negócio*, pages 155–156. Brasport.

Documento Digitalizado Restrito

Artigo de TCC do aluno Gabriel Cerdan Oste de Moura

Assunto: Artigo de TCC do aluno Gabriel Cerdan Oste de Moura

Assinado por: Daiane Tomazeti

Tipo do Documento: Comprovante

Situação: Finalizado

Nível de Acesso: Restrito

Hipótese Legal: Informação Pessoal - dados pessoais e dados pessoais sensíveis (Art. 31 da Lei nº 12.527/2011)

Tipo do Conferência: Cópia Simples

Documento assinado eletronicamente por:

- **Daiane Mastrangelo Tomazeti, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 11/03/2025 12:33:16.

Este documento foi armazenado no SUAP em 11/03/2025. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifsp.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1962030

Código de Autenticação: d7cce5ae58

