



# **CASHFLOW – CONTROLE FINANCEIRO PESSOAL**

## **CIÊNCIA DA COMPUTAÇÃO**

### **5º SEMESTRE**

*Daniel Bouças, Erick Diniz, João Dimitroulia, Matheus Gomes, Matheus Luigi*

## **1. Introdução**

### **1.1 Descrição Geral**

Este documento apresenta a especificação técnica e funcional do sistema CashFlow, uma aplicação desenvolvida com o objetivo de promover a organização financeira pessoal. Criado como projeto acadêmico no contexto universitário, o CashFlow representa uma iniciativa prática e moderna voltada à educação financeira, à experiência do usuário e ao uso de tecnologias.

Com isso, busca-se incentivar o hábito de organização financeira, fomentar o engajamento dos usuários com a ferramenta e modernizar a experiência de controle pessoal de finanças, alinhando o sistema CashFlow às práticas contemporâneas de educação financeira e transformação digital.

O sistema foi concebido para permitir o cadastro de usuários, autenticação por e-mail e senha, e a navegação entre diferentes etapas do processo de registro, finalizando com o acesso a uma tela principal, pronta para expansão com funcionalidades de controle financeiro.

### **1.2 Justificativa e Objetivos**

O crescente número de usuários que necessitam de ferramentas acessíveis e didáticas para acompanhar suas finanças pessoais motivou a criação de um sistema simples, modular e funcional, capaz de servir como base para implementações futuras.

#### **Objetivo Estratégico**

Desenvolver uma aplicação que sirva como modelo funcional para gerenciamento pessoal de dados, com foco em clareza de código, responsividade e usabilidade.

## Objetivos Específicos

- aplicar boas práticas de desenvolvimento front-end com HTML, CSS e Javascript;
- Implementar um fluxo de navegação com autenticação e cadastro de novos usuários;
- Promover a separação entre lógica, estrutura e estilo;
- Fornecer base sólida para futura expansão com backend e banco de dados.

### 1.3 Escopo

O escopo funcional e técnico da versão atual do sistema CashFlow abrange as seguintes entregas:

- **Tela de Login:** Estrutura de autenticação baseada em validação de campos, com redirecionamento condicional para a área principal mediante entrada de credenciais válidas.
- **Fluxo de Cadastro em Múltiplas Etapas:** Processo segmentado em três páginas sequenciais:
  - **Etapas 1 – Dados Pessoais:** Coleta de informações como nome, CPF, e-mail e telefone, com validação de preenchimento obrigatório e formatação;
  - **Etapas 2 – Endereço:** Captura de endereço completo com campos para CEP, logradouro, número, cidade e estado;
  - **Etapas 3 – Finalização:** Tela de confirmação do sucesso do processo de cadastro, com redirecionamento para o login.
- **Tela Principal (Home):** Estrutura inicial de dashboard (*home.html*) que serve como ponto de entrada para futuras funcionalidades, com layout visual básico já implementado.
- **Validações de Formulário:** Implementadas via JavaScript nativo, utilizando manipulação do DOM e lógica condicional para garantir integridade mínima dos dados antes do envio.

- **Navegação Controlada entre Telas:** Redirecionamento automatizado via `location.href` e condicionais de controle de fluxo em JavaScript, assegurando a sequência lógica entre etapas.
- **Estrutura Modularizada:** Separação clara entre arquivos HTML, CSS e JS específicos para cada página, promovendo manutenibilidade, reutilização de código e legibilidade.

**Ficam fora do escopo nesta fase:**

- Integração com banco de dados ou serviços de persistência (como Firebase ou APIs REST);
- Mecanismos reais de controle financeiro (lançamento, categorização e saldo), que serão abordados em versões futuras;
- Implementação completa de responsividade e usabilidade mobile (a aplicação apresenta responsividade limitada e parcial).

## **2. Visão Técnica da Solução**

### **2.1 Arquitetura da Solução**

A arquitetura do sistema CashFlow adota um modelo estático de front-end, estruturado segundo o paradigma de separação de responsabilidades entre camadas, visando clareza, manutenibilidade e expansibilidade futura.

- **Camada de Apresentação:** Responsável pela renderização das interfaces e interação visual com o usuário, construída utilizando HTML para estruturação semântica e CSS para estilização e responsividade básica.
- **Camada de Lógica de Interface:** Implementada em JavaScript, atua diretamente na validação dos dados de entrada, controle do fluxo de navegação entre páginas, gerenciamento de estados de formulário e manipulação dinâmica do DOM. Cada etapa possui um script dedicado, mantendo a lógica modularizada.

A comunicação entre as etapas é controlada localmente, via redirecionamentos e verificações condicionais de preenchimento, simulando um fluxo dinâmico e sequencial de interação.

O sistema está projetado de forma que possa, futuramente, integrar-se com serviços externos (como APIs REST, Firebase ou bancos relacionais), sem necessidade de reestruturação significativa da arquitetura base.

Essa abordagem garante:

- Baixa complexidade para prototipagem e testes;
- Facilidade de manutenção e leitura do código;
- Prontidão para expansão com backend real e autenticação;
- Isolamento de responsabilidades, seguindo boas práticas de desenvolvimento front-end.

## **2.2 Tecnologias Utilizadas**

- **HTML:** Estruturação semântica das interfaces;
- **CSS:** Estilização visual e responsividade básica;
- **JavaScript:** Validação, interações, transições de telas e controle da lógica de front-end.

## **3. Requisitos do Sistema**

### **3.1 Requisitos Funcionais**

- Tela de login com autenticação por e-mail e senha;
- Cadastro de novo usuário em três etapas;
- Redirecionamento automático para a tela principal após login;
- Feedback visual para campos obrigatórios;
- Estrutura modularizada de arquivos JS e CSS para cada tela.

### **3.2 Requisitos Não Funcionais**

- Navegação compatível com navegadores modernos (Chrome, Firefox, Edge);
- Estruturação clara dos arquivos e pastas;
- Carregamento rápido (< 2s);
- Separação entre lógica, visual e estrutura;
- Código limpo, comentado e padronizado.

## **4. Análise de Riscos e Estratégias de Mitigação**

### **4.1 Ausência de Persistência de Dados**

#### **Descrição do Risco**

Atualmente, o sistema CashFlow opera exclusivamente em ambiente front-end, sem qualquer mecanismo de persistência de dados (como banco de dados ou armazenamento local). Dessa forma, todas as informações inseridas pelo usuário como nome, CPF, endereço e demais dados cadastrais são temporários e perdidas ao recarregar ou fechar o navegador.

#### **Estratégia de Mitigação**

A médio prazo, será implementada a integração com serviços de backend (para gerenciamento de usuários) e banco de dados relacional.

Isso permitirá:

- Armazenamento persistente e seguro dos dados;
- Criação de sessões de login reais;
- Implementação de histórico, perfis personalizados e controle financeiro vinculado ao usuário.

## 4.2 Falhas de Navegação entre Etapas

### Descrição do Risco

Devido à ausência de backend e de controle robusto de sessões, usuários podem acessar diretamente URLs intermediárias sem ter concluído as etapas anteriores, comprometendo o fluxo lógico do sistema.

### Estratégia de Mitigação

O sistema conta atualmente com validações condicionais implementadas em JavaScript, que executam checagens básicas antes de permitir o avanço entre telas.

Isso inclui:

- Verificação de campos obrigatórios com bloqueio de navegação;
- Alertas visuais em campos vazios ou com formatação incorreta;
- Controle de fluxo baseado em eventos de clique e, futuramente, via localStorage.

## 5. Cronograma e Status Atual

| Módulo                  | Status       |
|-------------------------|--------------|
| Tela de login           | Concluído    |
| Cadastro - Etapa 1      | Concluído    |
| Cadastro - Etapa 2      | Concluído    |
| Cadastro - Etapa 3      | Concluído    |
| Tela inicial (home)     | Concluído    |
| Integração com backend  | Não iniciado |
| Responsividade completa | Parcial      |

## **6. Considerações Finais**

O projeto CashFlow representa um primeiro passo em direções relevantes, como alfabetização financeira digital e desenvolvimento web. Ao aplicar os conhecimentos adquiridos ao longo do curso, o sistema demonstra capacidade de organização, lógica e estruturação de interfaces interativas.

Apesar de suas limitações, o projeto constitui uma base sólida para futuras versões, que podem incluir integração com serviços de backend, dashboards financeiros, gráficos interativos e funcionalidades avançadas de controle de receitas e despesas. A clareza e modularidade da arquitetura garantem fácil evolução do sistema conforme a complexidade e o escopo se expandem.