SENAI ARMANDO DE ARRUDA PEREIRA -   
CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS

**PROJETO INTEGRADOR – 2MD**  
**ATHEOS – SISTEMA DE CONTROLE FINANCEIRO PESSOAL**

DAVI RODRIGUES PIRES

RAMON COELHO MELO

SAMUEL JUAN BARBOSA SANTOS

.

SÃO CAETANO DO SUL  
2025

**SUMÁRIO**

**1. INTRODUÇÃO AO PROJETO** ................................................................................................ [Página]  
1.1 NOME DO PROJETO ........................................................................................................ [Página]  
1.2 OBJETIVO ......................................................................................................................... [Página]  
1.3 RESUMO FUNCIONAL .................................................................................................... [Página]  
1.4 REQUISITOS FUNCIONAIS (RFS) ................................................................................. [Página]  
1.5 REQUISITOS NÃO FUNCIONAIS (RNFS) ...................................................................... [Página]  
1.6 REGRAS DE NEGÓCIO (RNS) ...................................................................................... [Página]  
**2. HISTÓRIA E JUSTIFICATIVA** ............................................................................................ [Página]  
**3. TECNOLOGIAS UTILIZADAS** ........................................................................................... [Página]  
3.1 FRONT-END .................................................................................................................... [Página]  
3.2 BACK-END ..................................................................................................................... [Página]  
3.3 BANCO DE DADOS ......................................................................................................... [Página]  
3.4 FERRAMENTAS AUXILIARES ........................................................................................ [Página]  
**4. MODELAGEM DE DADOS** ............................................................................................... [Página]  
**5. ORGANIZAÇÃO DAS ROTAS DA API** ............................................................................. [Página]  
5.1 LÓGICA DE ACESSO INICIAL ........................................................................................ [Página]  
5.2 ROTAS PÚBLICAS (NÃO AUTENTICADAS) ............................................................... [Página]  
5.3 ROTAS PRIVADAS (REQUEREM TOKEN JWT) ............................................................ [Página]  
**6. PRINTS DE TELAS PRINCIPAIS E DA EXECUÇÃO FUNCIONAL DO SISTEMA** ..... [Página]  
**7. CONCLUSÃO E POSSÍVEIS MELHORIAS FUTURAS** ................................................. [Página]  
7.1 CONCLUSÃO ................................................................................................................. [Página]  
7.2 MELHORIAS FUTURAS PLANEJADAS ........................................................................ [Página]  
**8. LINK PARA REPOSITÓRIO NO GITHUB** ...................................................................... [Página]

**1. INTRODUÇÃO AO PROJETO**

Este documento detalha o projeto Atheos, um sistema de controle financeiro pessoal desenvolvido no âmbito da disciplina de Projeto Integrador do SENAI São Caetano do Sul, sob a orientação dos professores Gleydson Monteiro, Rodrigo Alvarez e William Reis.

* 1. **Nome do autor**

Atheos – Sistema de Controle Financeiro Pessoal.

* 1. **Objetivo**

O objetivo principal é fornecer uma plataforma multiplataforma e intuitiva que permita aos usuários:

* Registrar transações.
* Analisar gastos.
* Gerenciar suas finanças de forma eficaz.
  1. **Resumo funcional**

O projeto Atheos nasceu da necessidade identificada entre jovens estudantes de gerenciar melhor suas finanças pessoais. Trata-se de uma plataforma multiplataforma e intuitiva para gerenciamento completo das finanças, incluindo cadastro e análise de transações, definição de metas financeiras e visualização de gráficos para um controle eficaz dos gastos. O sistema foi desenvolvido com foco em segurança, utilizando autenticação JWT, e priorizando uma experiência de usuário responsiva, além de prever extensibilidade futura para notificações, versão mobile e exportação de relatórios.

* 1. **Requisitos funcionais (RFS)**

A tabela a seguir apresenta os requisitos funcionais do sistema Atheos.

|  |  |
| --- | --- |
| Código | Descrição |
| RF01 | Permitir o cadastro de novos usuários (nome, e-mail, senha). |
| RF02 | Permitir o login de usuários (e-mail, senha). |
| RF03 | Permitir o cadastro de transações financeiras (tipo: entrada/saída, valor, data, descrição, categoria). |
| RF04 | Permitir a listagem de todas as transações do usuário logado. |
| RF05 | Calcular e exibir o saldo atual com base nas transações. |
| RF06 | Permitir a visualização de gráficos de entradas e saídas mensais. |
| RF07 | Permitir o cadastro de metas financeiras mensais por categoria. |
| RF08 | Notificar visualmente quando o usuário ultrapassar uma meta de gasto. |
| RF09 | Manter os dados separados por usuário (isolamento total de dados). |
| RF10 | Registrar a data de criação e atualização em todas as entidades principais. |

* 1. **Requisitos não funcionais (RNFS)**

A tabela a seguir apresenta os requisitos não funcionais do sistema Atheos.

|  |  |
| --- | --- |
| Código | Descrição |
| RNF01 | O sistema deve ser acessível via navegador web moderno. |
| RNF02 | A interface deve ser responsiva, adaptando-se a dispositivos móveis e desktops. |
| RNF03 | As senhas dos usuários devem ser armazenadas de forma segura (com hash). |
| RNF04 | A aplicação deve utilizar banco de dados relacional (MySQL). |
| RNF05 | O back-end deve seguir uma arquitetura RESTful. |

* 1. **Regras de negócio (RN)**

A tabela a seguir apresenta as regras de negócio do sistema Atheos.

|  |  |
| --- | --- |
| Código | Descrição |
| RN01 | Cada transação deve pertencer a exatamente um usuário. |
| RN02 | A transação do tipo “entrada” soma ao saldo; “saída” subtrai. |
| RN03 | O saldo exibido deve ser calculado em tempo real (ou o mais próximo disso). |
| RN04 | Cada meta está associada a um mês, ano e categoria específica. |
| RN05 | Ao deletar um usuário, todas as suas transações, metas e categorias devem ser removidas automaticamente. |
| RN06 | O sistema não deve permitir transações com valor menor que R$ 0,01. |

1. **HISTÓRIA E JUSTIFICATIVA**

Durante uma palestra sobre educação financeira no SENAI São Caetano do Sul, muitos alunos relataram dificuldades em controlar seus gastos e evitar endividamentos. Percebendo a importância do tema, os professores propuseram aos alunos o desenvolvimento de uma aplicação que auxiliasse jovens estudantes no gerenciamento de finanças pessoais, promovendo educação financeira e autonomia.  
Assim nasceu o projeto Atheos: uma plataforma multiplataforma e intuitiva para gerenciamento completo das finanças pessoais, incluindo cadastro e análise de transações, definição de metas financeiras e visualização de gráficos para um controle eficaz dos gastos. Desenvolvido com foco em segurança através de autenticação JWT, experiência de usuário responsiva e com extensibilidade futura para notificações, versão mobile e exportação de relatórios.

1. **TECNOLOGIAS UTILIZADAS**

As tecnologias empregadas no desenvolvimento do projeto Atheos foram selecionadas visando robustez, escalabilidade e uma boa experiência de desenvolvimento e de usuário.

**3.1 Front-end**

* **Core e Framework:** Next.js (React para SSR/SSG), React, React DOM.
* **Estilização e UI:** Tailwind CSS (CSS utilitário), Framer Motion (animações), clsx (classes condicionais).
* **Ícones:** React Icons, FontAwesome, Lucide React.
* **Gráficos:** Chart.js, React-Chartjs-2.
* **Componentes e Utilitários:** React Hot Toast e React Toastify (notificações), React Responsive Carousel (carrossel), Validator (validação de strings).

**3.2 Back-end**

* **Ambiente e Framework:** Node.js (ambiente de execução), Express (framework para API).
* **Segurança e Autenticação:** bcrypt/bcryptjs (hashing de senhas), jsonwebtoken (JWT), crypto (funcionalidades criptográficas).
* **Driver de Banco de Dados:** mysql2 (cliente MySQL).
* **Middleware:** cors (habilita CORS), multer (upload de arquivos).

**3.3 Banco de dados**

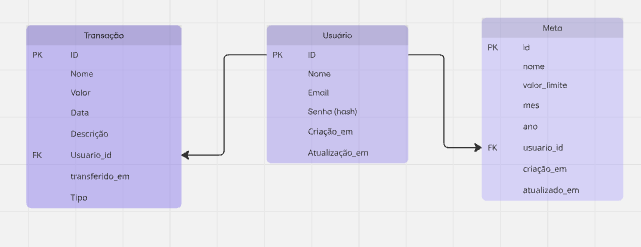
* MySQL (SGBD relacional).

**3.4 Ferramentas auxiliares**

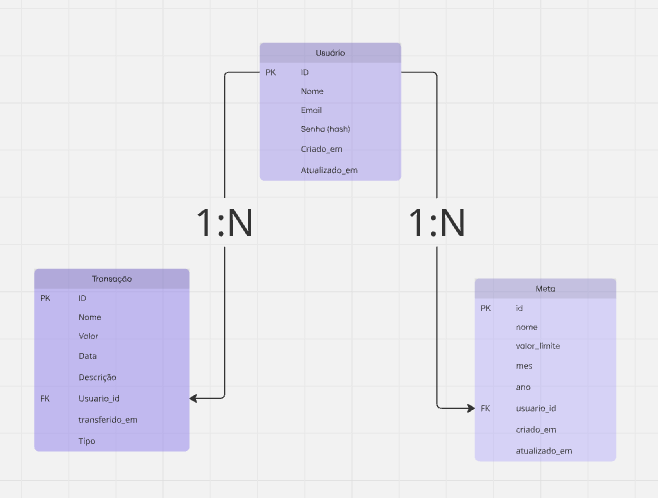
* Miro (prototipagem, diagramação, Kanban).
* Canva (design e elementos visuais).
* [Git e GitHub para versionamento de código e colaboração]
* [VS Code como ambiente de desenvolvimento integrado (IDE)]
* [Insomnia/Postman para testes de API]

**4. MODELAGEM DE DADOS**

A modelagem de dados do sistema Atheos é composta por três entidades principais: Usuários, Transações e Metas, organizadas conforme o diagrama lógico relacional abaixo, apresentando seus atributos, tipos e relacionamentos.

MER: 

DER:

[https://miro.com/welcomeonboard/c0Z0bnEyM1FpTFpFeDJVUjVnSkhjalpDNHVIdW9CUXE5ZEg4YjRubGZDd2NzUmNvTUtFdVZIbFMyTFlHY2lFU21hOWx3LzhCb3BFOVRyTm1PYWRPdGRDQXQ0TXRuME1sODdzcHZueFlWUHBrVythTUhzdVlqWENhcHhDSHBUb2t0R2lncW1vRmFBVnlLcVJzTmdFdlNRPT0hdjE=?share\_link\_id=777999103349](https://www.google.com/url?sa=E&q=https%3A%2F%2Fmiro.com%2Fwelcomeonboard%2Fc0Z0bnEyM1FpTFpFeDJVUjVnSkhjalpDNHVIdW9CUXE5ZEg4YjRubGZDd2NzUmNvTUtFdVZIbFMyTFlHY2lFU21hOWx3LzhCb3BFOVRyTm1PYWRPdGRDQXQ0TXRuME1sODdzcHZueFlWUHBrVythTUhzdVlqWENhcHhDSHBUb2t0R2lncW1vRmFBVnlLcVJzTmdFdlNRPT0hdjE%3D%3Fshare_link_id%3D777999103349)

**5. ORGANIZAÇÃO DAS ROTAS DA API**

A API do Atheos foi projetada com uma lógica de acesso inicial que prioriza a autenticação do usuário para garantir a segurança e privacidade dos dados financeiros.

**5.1 Lógica de acesso inicial**

* **Usuário Logado:** Acesso direto à homepage (/dashboard) com os dados do usuário.
* **Usuário Deslogado:** Redirecionamento para a rota / (ou uma rota específica como /verificacao, se implementada), que oferece opções de Login e Registro. Após o registro, o usuário deve efetuar login para ser redirecionado à homepage.

**5.2 Rotas públicas**

* **GET /**: Tela inicial com introdução e links para as opções de login e registro.
* **POST /registro**: Criação de uma nova conta de usuário.
  + Corpo da Requisição: { "nome": "string", "email": "string", "senha": "string" }
  + Resposta de Sucesso (201): { "message": "Usuário criado com sucesso!" }
* **POST /login**: Autenticação do usuário e geração de um token JWT.
  + Corpo da Requisição: { "email": "string", "senha": "string" }
  + Resposta de Sucesso (200): { "token": "jwt\_token", "usuario": { "id": "number", "nome": "string", "email": "string" } }

**5.3** R**otas privadas (requerem token jwt no header)**

5.3.1 **Dashboard principal e extratos**

* **GET /dashboard**: Homepage do usuário com dashboard (saldo, gráficos).
  + Resposta de Sucesso (200): { "saldoAtual": "number", "dadosGraficos": "object", ... }
* **GET /dashboard/extratos**: Lista todas as transações financeiras do usuário.
  + Resposta de Sucesso (200): [ { "id": "number", "tipo": "string", "valor": "number", ... }, ... ]
* **POST /dashboard/extratos**: Criação de uma nova transação.
  + Corpo da Requisição: { "tipo": "entrada|saida", "valor": "number", "data": "YYYY-MM-DD", "descricao": "string", "categoria": "string" }
  + Resposta de Sucesso (201): { "message": "Transação criada com sucesso!", "transacao": { ... } }
* **PUT /dashboard/extratos/:id**: Atualização de uma transação existente pelo ID.
  + Corpo da Requisição: { "tipo": "entrada|saida", "valor": "number", "data": "YYYY-MM-DD", "descricao": "string", "categoria": "string" } (campos opcionais para atualização)
  + Resposta de Sucesso (200): { "message": "Transação atualizada com sucesso!", "transacao": { ... } }
* **DELETE /dashboard/extratos/:id**: Exclusão de uma transação pelo ID.
  + Resposta de Sucesso (200): { "message": "Transação excluída com sucesso!" }

5.3.2 **Configurações**

* **GET /dashboard/config**: Tela/configurações do usuário. [Pode incluir rotas para PUT /usuario para atualizar dados do usuário, etc.]
  + [Definir o que será retornado/atualizado nesta rota, ex: dados do perfil do usuário]

5.3.3 **Rotas de Metas**

* **GET /dashboard/metas**: Listar metas financeiras do usuário.
  + Resposta de Sucesso (200): [ { "id": "number", "mes": "number", "ano": "number", "categoria": "string", "valor\_meta": "number" }, ... ]
* **POST /dashboard/metas**: Criação de uma nova meta.
  + Corpo da Requisição: { "mes": "number", "ano": "number", "categoria": "string", "valor\_meta": "number" }
  + Resposta de Sucesso (201): { "message": "Meta criada com sucesso!", "meta": { ... } }
* **PUT /dashboard/metas/:id**: Atualização de uma meta existente pelo ID.
  + Corpo da Requisição: { "mes": "number", "ano": "number", "categoria": "string", "valor\_meta": "number" } (campos opcionais para atualização)
  + Resposta de Sucesso (200): { "message": "Meta atualizada com sucesso!", "meta": { ... } }
* **DELETE /dashboard/metas/:id**: Exclusão de uma meta pelo ID.
  + Resposta de Sucesso (200): { "message": "Meta excluída com sucesso!" }

**6. PRINTS DE TELAS PRINCIPAIS E DA EXECUÇÃO FUNCIONAL DO SISTEMA**

📸 (As screenshots serão adicionadas após o desenvolvimento e testes finais das telas.)

**7. CONCLUSÃO E POSSÍVEIS MELHORIAS FUTURAS**

7.1 Conclusão

O projeto Atheos visa oferecer um sistema de controle financeiro pessoal prático e organizado, contribuindo para a educação financeira dos usuários, especialmente jovens estudantes que buscam maior autonomia e controle sobre seus gastos e receitas. A plataforma desenvolvida cumpre os requisitos funcionais e não funcionais estabelecidos, proporcionando uma base sólida para o gerenciamento financeiro.

7.2 MELHORIAS FUTURAS PLANEJADAS

Visando a contínua evolução e aprimoramento da plataforma Atheos, algumas melhorias futuras são planejadas:

* **Implementação de notificações proativas:** Envio de alertas (e-mail, push no app) para metas de gastos próximas de serem atingidas ou já excedidas.
* **Desenvolvimento de uma versão mobile dedicada:** Criação de aplicativos nativos (Android/iOS) ou PWA (Progressive Web App) para maior acessibilidade e experiência otimizada em dispositivos móveis.
* **Funcionalidade de exportação de relatórios:** Permitir que o usuário exporte seus dados financeiros e relatórios em formatos como PDF, CSV ou Excel.
* **Inclusão de modo escuro (Dark Mode):** Oferecer uma alternativa de visualização para preferência do usuário e conforto visual em ambientes com pouca luz.
* **Aprimoramentos de acessibilidade (A11Y):** Garantir que o sistema seja utilizável pelo maior número de pessoas, incluindo aquelas com deficiências, seguindo as diretrizes do WCAG.
* [Integração com Open Finance para importação automática de transações bancárias (requer análise de segurança e viabilidade).]
* [Criação de categorias personalizadas pelo usuário.]
* [Funcionalidade de planejamento orçamentário mais detalhada.]

**8. LINK PARA REPOSITÓRIO NO GITHUB**

O código-fonte completo do projeto Atheos, juntamente com um arquivo README.md explicativo sobre como configurar e executar a aplicação, está disponível no seguinte repositório do GitHub:

🔗 [https://github.com/seu-usuario/atheos-financas]  
*(Substituir pelo link real após a criação e publicização do repositório.)*

O README do repositório deve conter:

* Descrição do projeto.
* Tecnologias utilizadas.
* Pré-requisitos para execução.
* Instruções de instalação de dependências (front-end e back-end).
* Instruções para configurar o banco de dados.
* Como executar o projeto localmente (front-end e back-end).
* [Estrutura de pastas do projeto (opcional, mas útil).]
* [Como contribuir (se aplicável).]

Lembre-se de preencher os campos entre [] com as informações específicas do seu grupo e projeto. Boa sorte com a entrega!

9. REFERÊNCIAS

MICROSOFT: