

Cygni - AGIS

DOCUMENTO DE ARQUITETURA

Versão 0.1

Histórico de Revisão

Data	Versão	Descrição	Autor
15/05/2024	1.0	Documentação inicial	Todos do grupo
18/05/2024	1.1	Revisão do documento	Todos do grupo
20/05/2024	1.2	Revisão do documento	Ana e Weverton

Tabela 1: Tabela de Revisão

Autores:

Matrícula	Nome	Descrição do papel assumido na equipe	% de contribuição ao trabalho
180114964	Adriano Fonseca Carvalho Costa	Participou no desenvolvimento dos documentos de visão e arquitetura. BackEnd e Banco de Dados	12,5%
221007798	Ana Julia Mendes Santos	Participou no desenvolvimento dos documentos de visão e arquitetura. FrontEnd	12,5%
221022462	Arthur da Silveira Sousa	Participou no desenvolvimento dos documentos de visão e arquitetura. FrontEnd	12,5%
221022490	Cauã Araujo dos Santos	Participou no desenvolvimento dos documentos de visão e arquitetura. FrontEnd e Banco de Dados	12,5%
222014859	Ian Costa Guimarães	Participou no desenvolvimento dos documentos de visão e arquitetura. BackEnd e Banco de Dados	12,5%
200025571	Necivaldo Amaral Pinto Junior	Participou no desenvolvimento dos documentos de visão e arquitetura. BackEnd	12,5%
231036980	Pedro Luiz Fonseca da Silva	Participou no desenvolvimento dos documentos de visão e arquitetura. BackEnd e Banco de Dados	12,5%
221022767	Weverton Rodrigues da Costa Silva	Participou no desenvolvimento dos documentos de visão e arquitetura. FrontEnd	12,5%

Tabela 2: Tabela com os participantes

Sumário

1	<i>INTRODUÇÃO</i>	4
1.1	Propósito	4
1.2	Escopo	4
2	<i>REPRESENTAÇÃO ARQUITETURAL</i>	4
2.1	Definições	4
2.2	Justifique sua escolha	4
2.3	Detalhamento	4
2.4	Metas e restrições arquiteturais	5
2.5	Visão de Casos de uso (escopo do produto)	5
2.6	Visão lógica	5
2.7	Visão de Implementação	5
2.8	Visão de Implantação	6
2.9	Restrições adicionais	6
3	<i>BIBLIOGRAFIA</i>	7

1 INTRODUÇÃO

1.1 Propósito

Este documento descreve a arquitetura do sistema AGIS, sendo desenvolvido pelo grupo Cygni, na disciplina de MDS – Métodos de Desenvolvimento de Software – edição do primeiro semestre de 2024, a fim de fornecer uma visão abrangente do sistema para desenvolvedores, testadores e demais interessados.

1.2 Escopo

O software AGIS é uma aplicação que busca otimizar a organização do trabalho do cliente. Ele permite o cadastro de funcionários e suas respectivas especificidades e, a partir disso, fornece ferramentas para analisar esses cadastros por meio de persistência e gestão de dados de forma a possibilitar a geração de relatórios de forma eficiente. O detalhamento do escopo se encontra no documento de visão do produto <[Link do documento](#)>.

2 REPRESENTAÇÃO ARQUITETURAL

2.1 Definições

O sistema seguirá uma Arquitetura em Camadas.

2.2 Justifique sua escolha

A arquitetura em camadas possui o formato ideal para a implementação de CRUDs, conceito que será amplamente usado no decorrer do projeto. Dado o contexto de utilização do nosso software, será imprescindível focar no que será mostrado, além de ter uma camada para aplicação de regras de negócio e para a persistência dos dados.

Há também benefícios para o processo de desenvolvimento, tendo em vista que essa arquitetura possibilita delimitar bem o escopo de atuação de cada membro e também que cada componente nele inserido seja testado separadamente, possibilitando, assim, uma maior abrangência de testes.

2.3 Detalhamento

A ideia por trás da Arquitetura em Camadas é que os módulos com funcionalidades semelhantes são organizados em camadas horizontais. Esse estilo de arquitetura abstrai a visão do sistema como um todo e, ao mesmo tempo, fornece detalhes suficientes para entender as funções e responsabilidades de cada camada e o relacionamento entre elas.

A camada de apresentação contém todas as classes responsáveis por apresentar a UI/visualização para o usuário final e pelo tratamento da lógica de comunicação do navegador. É a camada com a qual os clientes interagem.

Já a camada de negócios/lógica contém toda a lógica exigida pelo aplicativo para atender seus requisitos funcionais. Geralmente, lida com a agregação de dados, computação e solicitações de consulta.

A camada de dados/física e a de persistência é onde as informações recuperáveis são armazenadas. Essa camada consiste em aspectos lógicos e físicos e implementa o modelo lógico na plataforma de banco de dados físico.

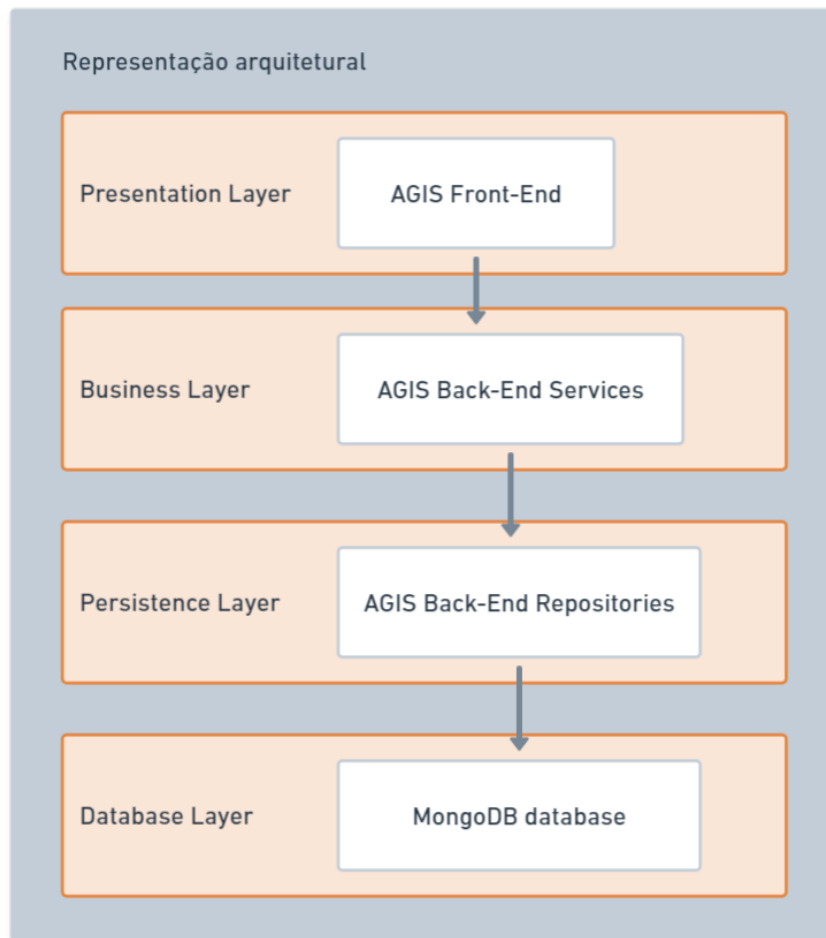


Figura 1: Desenho esquemático da representação arquitetural

2.4 Metas e restrições arquiteturais

- O sistema deverá funcionar, sem haver problemas, em 10 máquinas ao mesmo tempo pelo menos;
- Ter uma vida útil maior possível;
- O sistema deverá funcionar da forma mais auto sustentável possível;
- O sistema não deve precisar se adaptar a novas mudanças nas tecnologias utilizadas no mesmo;
- A única alteração esperada para o sistema é uma atualização da máquina que hospeda o sistema localmente;
- Sistema funcionará apenas na rede local;
- Hardware local, sistema leve para funcionar com hardware “simples”;
- Base de dados totalmente nova (nenhuma será reutilizada);
- Segurança reforçada dos dados;

2.5 Visão de Casos de uso (escopo do produto)

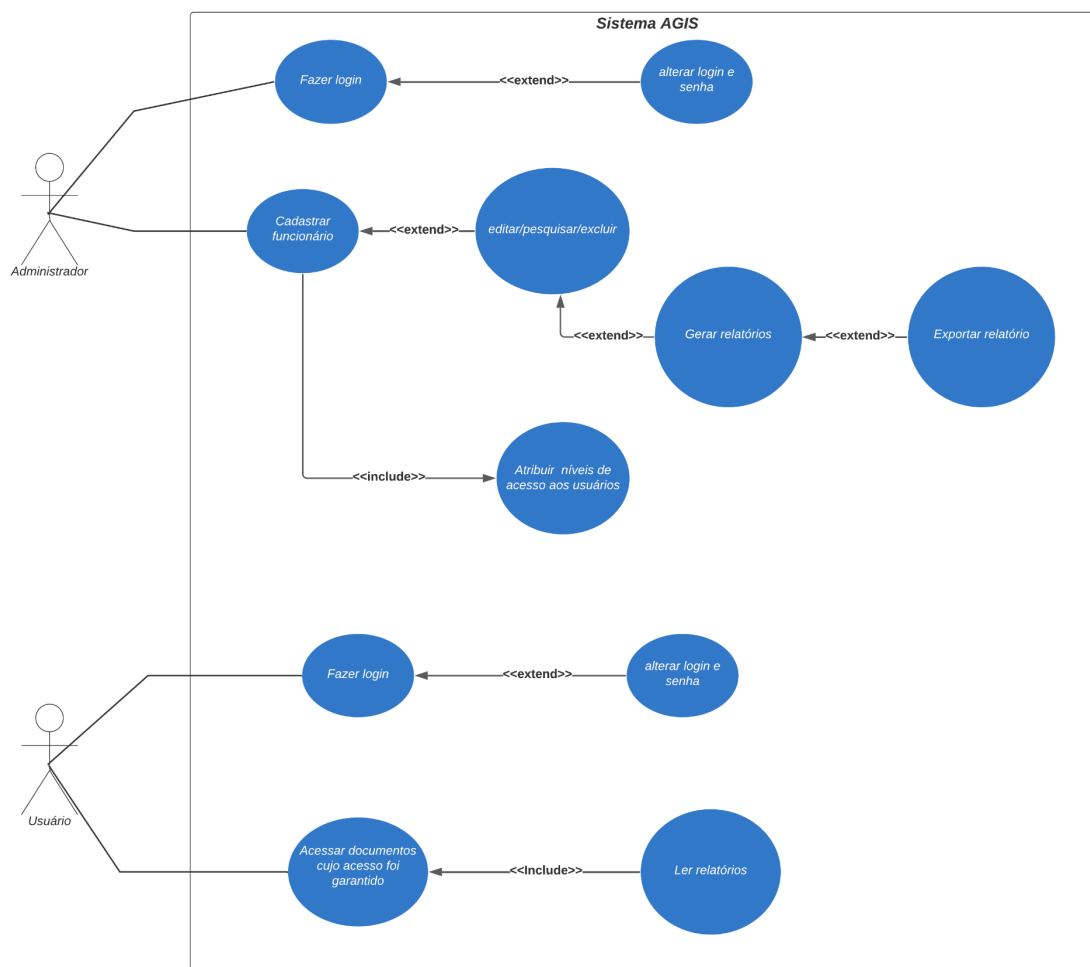


Figura 2: Desenho esquemático de casos de uso

Os casos de uso do Sistema AGIS são simples e diretos, focando nas principais funcionalidades administrativas e de acesso de usuários. O Administrador pode realizar ações como fazer login, cadastrar funcionários (incluindo atribuir níveis de acesso), editar, pesquisar e excluir dados, além de gerar e exportar relatórios (inclui relatório mensal, campanhas e escalas). O Usuário pode fazer login, acessar documentos cujo acesso foi autorizado e ler relatórios. Ambos os tipos de usuários têm a possibilidade de alterar login e senha. Este fluxo simples facilita a gestão de usuários e documentos, integrando todas essas funcionalidades de forma clara e eficiente.

2.6 Visão lógica

A Arquitetura em Camadas organiza módulos similares em camadas horizontais, abstraindo a visão do sistema como um todo. A camada de apresentação lida com a interface do usuário e a comunicação do navegador, a de negócios com a lógica do aplicativo e a de dados/persistência com armazenamento de informações.

O sistema é subdividido nos seguintes módulos: Camada de Apresentação, Camada de negócio, Camada de persistência e Camada do banco de dados.

- Camada de Apresentação: é responsável por lidar com toda a interface do(a) usuário(a) e lógicas de comunicação com navegadores.

- Camada de Negócio: é responsável por executar operações e fluxos de negócios específicos associados à uma requisição.
- Camada de Persistência: é responsável por persistir ou recuperar informações fisicamente salvas em armazenamento.
- Camada de Banco de Dados: mantém os dados salvos de forma física.

Todas as camadas são independentes e formam uma abstração em volta de sua necessidade e o trabalho a ser realizado dentro de uma requisição em particular. A comunicação entre as camadas é definida pelos membros da equipe, podendo elas serem abertas ou fechadas para as outras camadas.

2.6.1 Diagrama de Estados

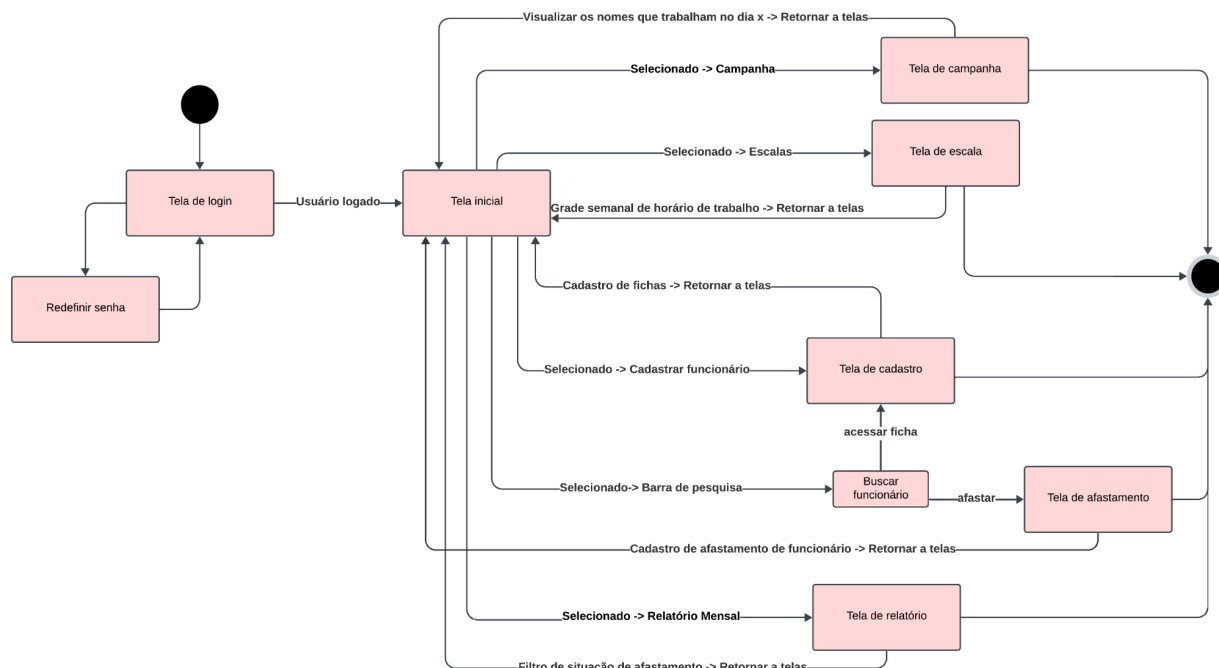


Figura 3: Desenho esquemático do diagrama de estados

O diagrama de estados é aplicado para demonstrar o comportamento geral do sistema, mostrando os estados e transições do funcionamento assim como quais eventos levam a transição de um estado para outro do AGIS. Assim, é possível compreender o funcionamento da aplicação desktop conforme é utilizado. A aplicação começa pela:

1. **Tela de login:** Para ter acesso ao sistema, os usuários terão que fazer login. Os usuários são divididos em Administrador e Usuário, sendo o primeiro capaz de cadastrar funcionários e gerir suas escalas enquanto o segundo apenas lê. Caso o usuário esqueça a senha, o mesmo será direcionado para uma tela de redefinição de senha.
2. **Tela Inicial:** Caso o usuário efetue o login com sucesso, ele terá acesso à tela inicial, que oferece acesso a todas as outras telas do sistema, sendo elas:
 - a. Tela de Escala
 - b. Tela de Campanha
 - c. Tela de Cadastro
 - d. Tela de Afastamento
 - e. Tela de Relatório

Além disso, na Tela Inicial, será possível realizar buscas de todos os funcionários cadastrados, permitindo o acesso às suas fichas e ao cadastro de afastamentos. Após acessar os serviços desejados, é possível voltar à Tela Inicial, acessar outras telas e, também, sair do programa.

2.6.2 Diagrama de Atividades

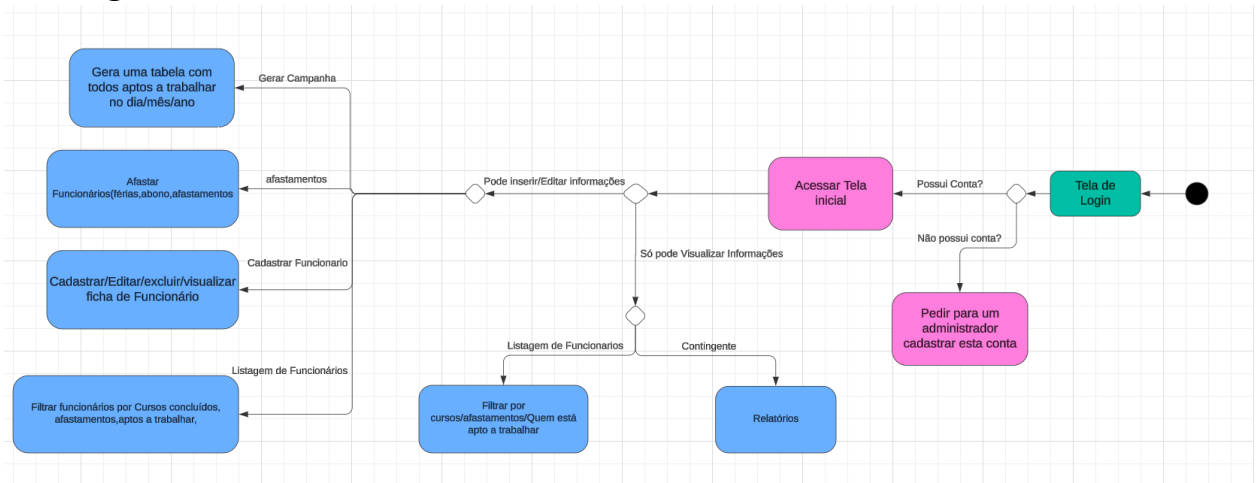


Figura 4: Desenho esquemático do diagrama de atividades

Esse diagrama de atividades representa um fluxo de trabalho para um sistema de gerenciamento de funcionários, detalhando as diferentes etapas e decisões envolvidas no processo. Aqui está a explicação passo a passo:

1. Tela de Login:

- O usuário inicia acessando a Tela de Login.
- É feita uma verificação se o usuário possui uma conta:
- Se sim, ele acessa a Tela Inicial.
- Se não, ele deve pedir para um administrador cadastrar esta conta.

2. Tela inicial:

- Dependendo das permissões do usuário, ele pode:
- Inserir/Editar informações, ou
- Apenas visualizar informações.

3. Ações Disponíveis:

- Gerar Campanha: Gera uma tabela com todos os funcionários aptos a trabalhar no dia/mês/ano.
- Afastar Funcionários: Afastar funcionários por férias, abono, ou outros motivos de afastamento.
- Cadastrar/Editar/Excluir/Visualizar ficha de Funcionário: Gerenciar os dados dos funcionários.
- Listagem de Funcionários: Exibir uma lista de todos os funcionários.

4. Filtragem e Relatórios:

- Na Listagem de Funcionários, o usuário pode filtrar por:
- Cursos concluídos,
- Afastamentos,
- Funcionários aptos a trabalhar.
- Também pode gerar Relatório com base nas informações filtradas.

5. Decisões:

- As permissões do usuário determinam se ele pode inserir/editar informações ou só visualizar informações:

- Se o usuário pode inserir/editar, ele pode realizar todas as ações descritas.
- Se o usuário só pode visualizar, ele tem acesso limitado às informações.

Este diagrama ilustra o fluxo lógico do sistema desde o login até a execução de várias operações de gerenciamento de funcionários, com base nas permissões do usuário.

2.6.3 Diagrama de Classes

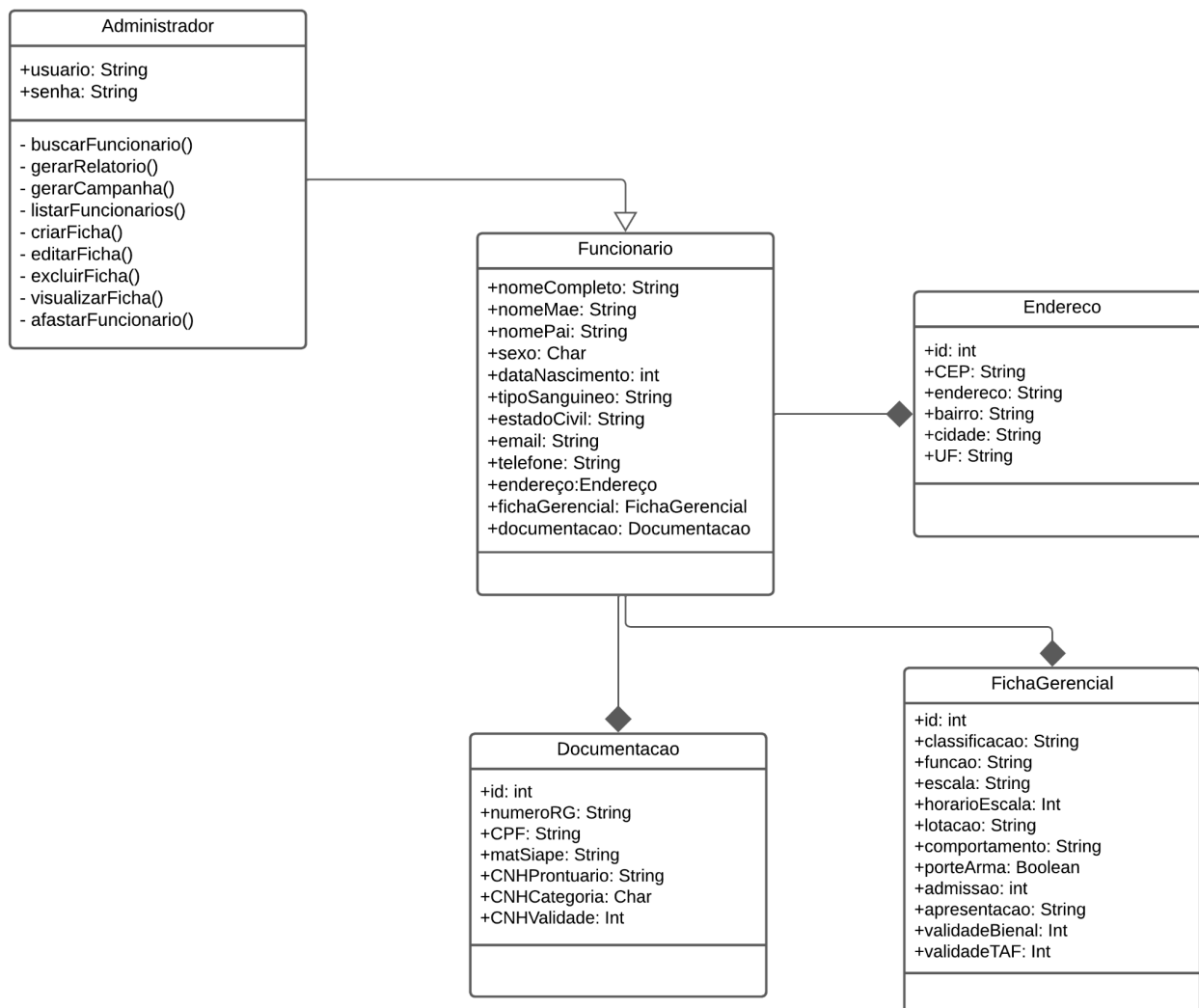


Figura 5: Desenho esquemático do diagrama de classes

Esse diagrama de classes representa um modelo de dados para um sistema de gerenciamento de funcionários, mostrando como as diferentes entidades do sistema estão relacionadas entre si e quais atributos e métodos elas possuem. Vamos detalhar cada uma das classes e suas relações:

A classe ADM representa um administrador do sistema, com atributos de nome de usuário e senha para login. Ela possui vários métodos para gerenciar os funcionários, incluindo busca, criação, edição, exclusão de fichas, geração de relatórios e campanhas, listagem de funcionários, visualização de fichas e afastamento de funcionários.

A classe Funcionário representa um funcionário com diversos atributos pessoais e profissionais. Ela se relaciona com as classes Endereço, FichaGerencial e Documentação.

A classe Endereço contém os detalhes de endereço do funcionário, incluindo CEP, endereço, bairro, cidade e unidade federativa (UF).

A classe FichaGerencial inclui informações gerenciais e operacionais sobre o funcionário, como classificação, função, escala, lotação, comportamento, porte de arma, datas de admissão e apresentação, e validade de certificações e exames.

A classe Documentação armazena informações sobre os documentos do funcionário, como RG, CPF, matrícula SIAPE, CNH (Carteira Nacional de Habilitação) com prontuário, categoria e validade.

Relações entre as classes

- A classe Funcionário tem uma associação com a classe Endereço, indicando que cada funcionário tem um endereço.

- A classe Funcionário tem uma associação com a classe FichaGerencial, indicando que cada funcionário tem uma ficha gerencial.

- A classe Funcionário tem uma associação com a classe Documentação, indicando que cada funcionário possui documentos associados.

Esse diagrama de classes fornece uma visão estruturada do sistema, especificando como os dados sobre funcionários são organizados e manipulados.

2.7 Visão de Implementação

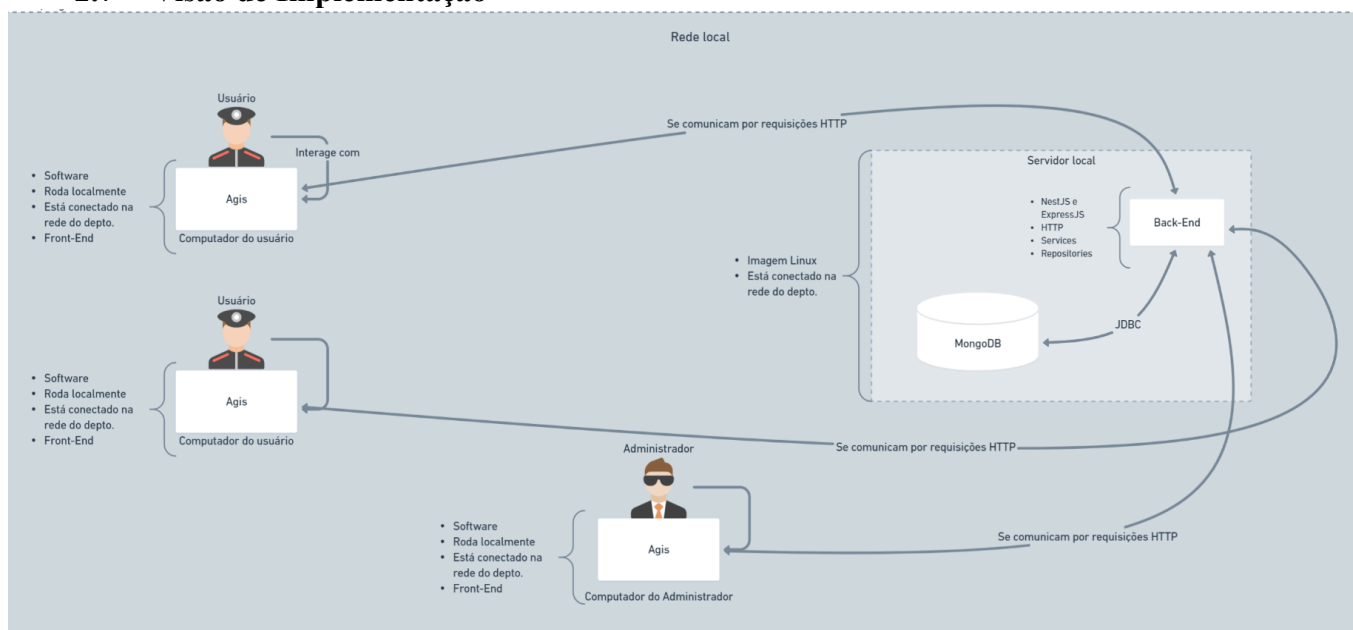


Figura 6: Desenho esquemático da visão de implementação

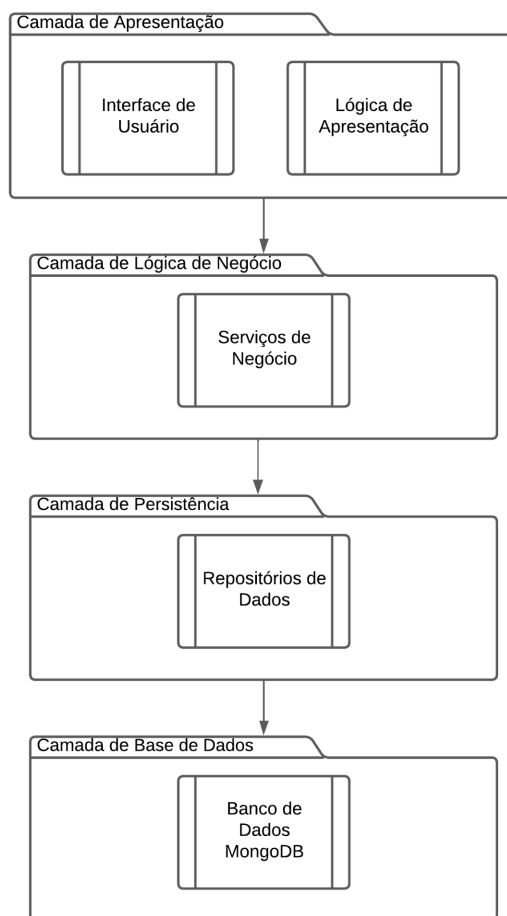


Figura 7: Desenho esquemático do diagrama de pacotes

Quanto ao diagrama de pacotes, apresenta os seguintes componentes principais: a Interface de Usuário, responsável pela interação direta com os usuários, e a Lógica de Apresentação, que manipula e transforma dados para a interface; os Serviços de Negócio, que implementam as regras de negócios e processos centrais do sistema; os Repositórios de Dados, que gerenciam operações de persistência e recuperação de dados armazenados; e o Banco de Dados MongoDB, que armazena os dados de maneira persistente utilizando um banco de dados NoSQL.

2.8 Visão de Implantação

O software será implantado em um servidor local para ser executado em um desktop. Para o desenvolvimento, utilizaremos HTML, CSS e JavaScript como marcadores e linguagens de programação, em conjunto com o framework React para FrontEnd, visando uma experiência de usuário mais dinâmica e responsiva.

Para o Backend, usaremos JavaScript, em conjunto com o NestJS e Express. Optamos pelo MongoDB, um banco de dados No-SQL que oferece praticidade e flexibilidade no controle de coleções de dados e, devido a maturidade do projeto, também oferece de forma nativa ferramentas para sua gestão, dispensando o uso de ORMs e ferramentas de visualização de dados à parte.

2.9 Restrições adicionais

O software deve ser usado a partir da rede local do cliente que, por sua vez, exige a identificação e login do usuário. O software é preparado para 10 usuários conectados simultaneamente em rede local.

Confiabilidade: Deve apresentar um ambiente suficientemente seguro, a fim de garantir que apenas usuários registrados no sistema tenham acesso às informações.

Interface: Deve conter estilização das páginas com cores, fontes, botões e ícones padronizados.

Suportabilidade: Deve ser compatível com as versões de até maio de 2024 do sistema operacional Windows.

Usabilidade: Deve ajustar dinamicamente o layout e o conteúdo para diferentes tamanhos de tela, permitindo uma visualização apropriada em telas pequenas, médias e grandes.

3 Bibliografia

- Guideline: Descrever a Arquitetura. **Centro de Informática da UFPE**. Disponível em: <https://www.cin.ufpe.br/~rls2/processo_tg/Metodologia%20S&B/guidances/guidelines/outline_the_architecture_6DDDD817.html>. Acesso em: 15 de maio de 2024
- Layered Architecture - Software Architecture Patterns. **O'Reilly Media**. Disponível em: <<https://www.oreilly.com/library/view/software-architecture-patterns/9781491971437/ch01.html>>. Acesso em: 15 de maio de 2024
- O que é um diagrama de máquina de estados? **Lucidchart**. Disponível em: <<https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-maquina-de-estados-uml>>. Acesso em: 16 de maio de 2024