



CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

LUIS AFONSO MINEO

RICARDO PETRI SILVA

PRONTUÁRIO DIGITAL ANNA MARIA PIETTA

Londrina
2025

AGRADECIMENTOS

Gostaria de expressar minha profunda gratidão a todos que, de alguma forma, contribuíram para a realização deste trabalho e para a minha formação.

Agradeço imensamente ao meu orientador, Professor Doutor Ricardo Petri, pela paciência, pelo conhecimento compartilhado e pela orientação ao longo de todo o desenvolvimento deste projeto. Seus conselhos foram fundamentais.

Agradeço também aos membros da banca examinadora, pela disponibilidade e por dedicarem seu tempo e conhecimento à avaliação deste trabalho.

Aos meus pais, agradeço pelo amor incondicional, pelo apoio constante e por serem a base de tudo. Sem o incentivo e a confiança de vocês, esta jornada não seria possível.

A minha irmã, por todos os conselhos, amizade e palavras de incentivo, não somente durante este projeto, mas também ao longo de todo o meu período acadêmico.

Aos meus amigos, pela companhia, pelas palavras de encorajamento e por todos os momentos de alegria que tornaram, e tornam, o caminho mais leve.

A todos, o meu muito obrigado!

LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama de Caso de Uso do Sistema ProAMP	8
Figura 2 – Diagrama de Sequência para Geração de Relatório Mensal	10
Figura 3 – Diagrama de Classe do Sistema ProAMP	12
Figura 4 – Diagrama Entidade-Relacionamento do Sistema ProAMP	13
Figura 5 – Diagrama de Estado para a entidade Estudante	14

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface (Interface de Programação de Aplicações)
CRUD	Create, Read, Update, Delete (Criar, Ler, Atualizar, Excluir)
DRF	Django REST Framework
ORM	Object-Relational Mapping (Mapeamento Objeto-Relacional)
ProAMP	Prontuário Anna Maria Pietta
SUS	Sistema Único de Saúde
UML	Unified Modeling Language (Linguagem de Modelagem Unificada)

SUMÁRIO

Agradecimentos	1
Lista de ilustrações	2
Lista de Abreviaturas e Siglas	3
1 INTRODUÇÃO	6
2 DESCRIÇÃO DO SISTEMA	7
2.1 Arquitetura e Recursos do Sistema	7
3 MODELAGEM DO SISTEMA	8
3.1 DIAGRAMA DE CASO DE USO	8
3.2 DIAGRAMA DE SEQUÊNCIA	9
3.3 DIAGRAMA DE CLASSE	10
3.4 DIAGRAMA DE ENTIDADE E RELACIONAMENTO	12
3.5 DIAGRAMA DE ESTADO	13
4 CONCLUSÕES E TRABALHOS FUTUROS	15
REFERÊNCIAS	16
GLOSSÁRIO	17
APÊNDICE A – REPOSITÓRIO DE DOCUMENTAÇÃO	18
A.1 Página do Projeto	18
A.2 Glossário Interativo Online	18
APÊNDICE B – ARTEFATOS DO PROJETO	19
B.1 Pedido do Investidor	19
B.2 Especificação Suplementar	19
B.3 Documento de Visão	19
B.4 Glossário	19
B.5 Especificação de Casos de Uso	19
B.5.1 Gerenciar Usuários	20
B.5.2 Gerenciar Estudantes	20
B.5.3 Ver Profissionais da Saúde	20
B.5.4 Gerenciar Perfil	20
B.5.5 Consultar Detalhes de Estudante	20

B.5.6	Gerar Relatório Mensal	20
B.5.7	Gerar Relatório Específico	20
	APÊNDICE C – DIAGRAMAS DO PROJETO	21
C.1	Diagrama de Caso de Uso	21
C.2	Diagrama de Classes	21
C.3	Diagrama de Entidade-Relacionamento (DER)	21
C.4	Diagrama de Implantação	21
C.5	Diagramas de Estado	22
C.5.1	Estado de Estudantes	22
C.5.2	Estado de Entradas de Prontuário	22
C.6	Workflow AS-IS	22
C.7	Diagramas de Sequência	22
C.7.1	Sequência: Gerenciar Usuários	22
C.7.2	Sequência: Gerenciar Estudantes	23
C.7.3	Sequência: Gerenciar Perfil	23
C.7.4	Sequência: Ver Profissionais da Saúde	23
C.7.5	Sequência: Consultar Detalhes de Estudante	23
C.7.6	Sequência: Gerar Relatório Mensal	23
C.7.7	Sequência: Gerar Relatório Específico A	23
C.7.8	Sequência: Gerar Relatório Específico B	23

1 INTRODUÇÃO

A informatização de processos clínicos é um elemento essencial para garantir a eficiência, segurança e a correta organização das informações de pacientes. No ambiente escolar, em instituições que promovem atendimento clínico especializado, a implementação de um sistema de prontuário digital pode gerar benefícios significativos tanto para os profissionais da saúde quanto para a administração.

Este projeto de estágio propôs o desenvolvimento do ProAMP, um sistema de prontuário digital destinado ao setor clínico da escola de educação especial Anna Maria Pietta. O sistema foi projetado para permitir o registro eletrônico dos atendimentos, organizar os dados dos pacientes e automatizar a geração de relatórios, o que promoveu maior agilidade e precisão na gestão das informações clínicas.

A justificativa para o sistema baseou-se na necessidade de otimizar o gerenciamento dos atendimentos clínicos. A escola, mantida pela APAE de Florestópolis, opera com um convênio com o Sistema Único de Saúde (SUS), que repassa recursos financeiros com base na quantidade de procedimentos clínicos realizados mensalmente. Atualmente, o processo de registro e compilação desses atendimentos é realizado de forma inteiramente manual. Os profissionais preenchem fichas físicas, que posteriormente são usadas para a contabilização do valor de repasse.

Esta metodologia manual apresentava deficiências que comprometiam a eficiência e a precisão das informações, resultando em relatórios entregues com atraso e com inconsistências, como nomes incompletos de alunos ou a omissão de atendimentos. A implementação do prontuário digital visou mitigar esses problemas, garantindo maior precisão na coleta de dados e agilidade na geração de relatórios, o que, por sua vez, contribuiu para a estabilidade financeira da instituição, conforme os benefícios de eficiência e exatidão dos registros de saúde apontados por Seymour (SEYMOUR; FRANTSVOG; GRAEBER, 2012, p. 207).

O objetivo geral deste projeto foi, portanto, desenvolver um sistema de prontuário digital para otimizar a gestão dos atendimentos e garantir maior eficiência nos processos administrativos do setor clínico da escola.

2 DESCRIÇÃO DO SISTEMA

Este capítulo descreve os aspectos técnicos do sistema ProAMP, detalhando os recursos, ferramentas e o ambiente de desenvolvimento utilizados. O projeto foi construído com base em uma arquitetura moderna e modular, visando escalabilidade, segurança e manutenibilidade.

2.1 ARQUITETURA E RECURSOS DO SISTEMA

A arquitetura do sistema ProAMP foi dividida em *backend* e *frontend*, com comunicação via *APIs RESTful*.

A linguagem de programação principal para o *backend* foi o *Python*¹ na versão 3.12.4. A escolha se justificou pelo seu vasto ecossistema de bibliotecas, que atenderam plenamente aos requisitos do projeto, incluindo *frameworks web* robustos e ferramentas para geração de relatórios.

Para o desenvolvimento web do *backend*, o *Django Framework*² versão 5.2 foi o escolhido. O *Django ORM (Object-Relational Mapping)*, por sua vez, facilitou a interação com o banco de dados de forma segura e intuitiva. Complementarmente, foi utilizado o *Django REST Framework (DRF)*³ versão 3.16.0 para a construção de *APIs RESTful*, simplificando a comunicação entre o *frontend* e o *backend*.

O sistema de gerenciamento de banco de dados selecionado foi o *PostgreSQL*⁴ versão 17. Ele é conhecido por seu suporte a grandes volumes de dados e consultas complexas, garantindo a escalabilidade futura do sistema, além de ser um software de código aberto, o que eliminou custos de licenciamento.

Para o *frontend*, optou-se pelo *framework Vue.js*⁵ versão 3.5.18, permitindo a criação de uma interface de usuário reativa e moderna. A estilização das telas foi desenvolvida utilizando o *framework CSS Tailwind*⁶ versão 4.1, que possibilitou a construção de layouts personalizados de forma mais rápida.

O ambiente de desenvolvimento foi composto pelo *Visual Studio Code*⁷ versão 1.94, com extensões para *Python*, *Django*, *Vue.js* e *TailwindCSS*. O controle de versão foi realizado com *Git*⁸, e o código-fonte foi hospedado em um repositório no *GitHub*⁹, garantindo o versionamento seguro do projeto.

¹ Disponível em: <<https://www.python.org/>>

² Disponível em: <<https://www.djangoproject.com/>>

³ Disponível em: <<https://www.django-rest-framework.org/>>

⁴ Disponível em: <<https://www.postgresql.org/>>

⁵ Disponível em: <<https://vuejs.org/>>

⁶ Disponível em: <<https://tailwindcss.com/>>

⁷ Disponível em: <<https://code.visualstudio.com/>>

⁸ Disponível em: <<https://git-scm.com/>>

⁹ Disponível em: <<https://github.com/>>

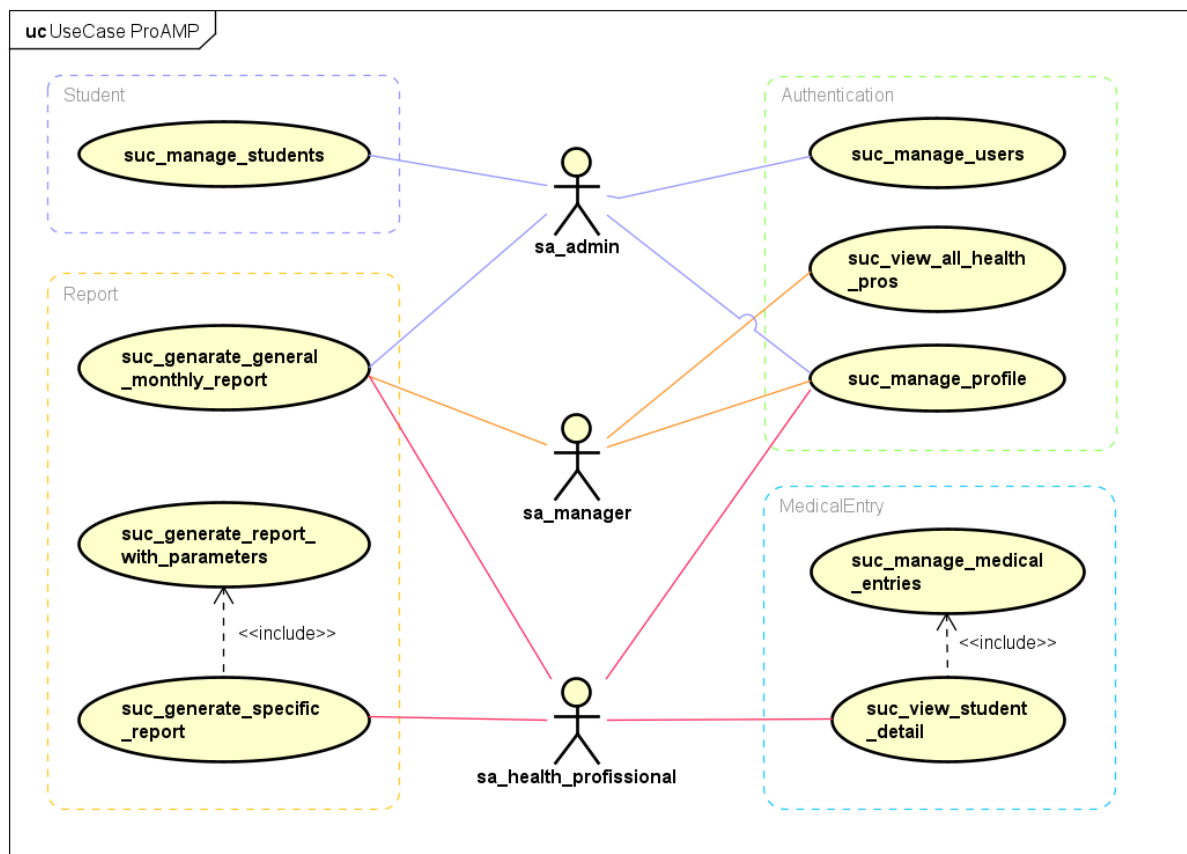
3 MODELAGEM DO SISTEMA

Neste capítulo são apresentados os diagramas que representam a estrutura e o funcionamento do sistema ProAMP. Os artefatos e diagramas criados durante o projeto, seguem o modelo UML, descritos por Fowler (FOWLER, 2003), e podem ser acessados através dos Apêndices B e C.

3.1 DIAGRAMA DE CASO DE USO

O Diagrama de Caso de Uso oferece uma visão geral das funcionalidades do sistema e de como os diferentes tipos de usuários, ou atores, interagem com ele. Para refletir a arquitetura modular do projeto, os casos de uso ilustrados no diagrama abaixo foram organizados dentro de contêineres pontilhados. Cada contêiner representa um dos "apps" do *Django*, o qual agrupa funcionalidades relacionadas. A seguir, na Figura 1, os principais casos de uso são apresentados visualmente e, na sequência, descritos em detalhe.

Figura 1 – Diagrama de Caso de Uso do Sistema ProAMP



- **suc_manage_users** (Gerenciar Usuários): Permite ao administrador executar operações CRUD (Criar, Ler, Atualizar, Excluir) em todos os usuários do sistema.
- **suc_manage_students** (Gerenciar Estudantes): Permite ao administrador executar operações de CRUD sobre os estudantes da escola.
- **suc_view_all_health_pros** (Visualizar Profissionais de Saúde): Permite ao gestor visualizar uma lista com as informações dos profissionais de saúde, com acesso apenas de leitura.
- **suc_manage_profile** (Gerenciar Próprio Perfil): Permite a qualquer ator (admin, gestor ou profissional de saúde) visualizar e editar suas próprias informações pessoais.
- **suc_generate_general_monthly_report** (Gerar Relatório Mensal Geral): Permite a administradores e gestores gerar relatórios de atendimento do último mês.
- **suc_view_student_detail** (Visualizar Detalhes do Estudante): Permite ao profissional de saúde acessar a ficha completa de um estudante, incluindo seus dados pessoais e o histórico de prontuários.
- **suc_manage_medical_entries** (Gerenciar Entradas Médicas): Subfluxo que permite ao profissional de saúde adicionar ou excluir entradas no prontuário de um estudante.
- **suc_generate_specific_report** (Gerar Relatório Específico): Permite ao profissional de saúde gerar relatórios detalhados com diferentes critérios de busca.
- **suc_generate_report_with_parameters** (Gerar Relatório com Parâmetros): Subfluxo que possibilita a geração de relatórios por estudante ou por um período de tempo específico.

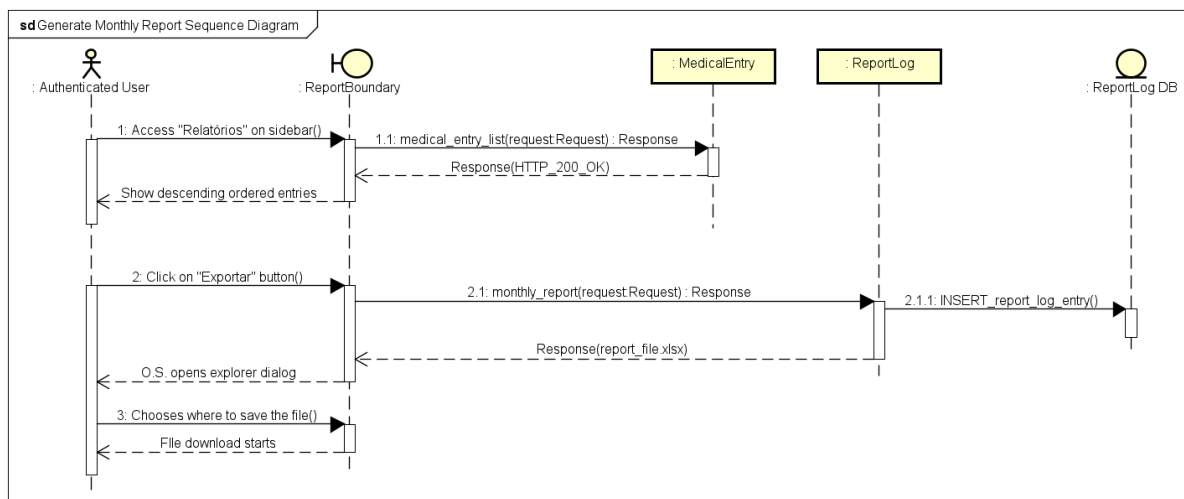
3.2 DIAGRAMA DE SEQUÊNCIA

O Diagrama de Sequência é uma representação visual do fluxo de ações possíveis por parte de um ator, ao interagir com o sistema. Para este projeto, foram elaborados oito diagramas de sequência que representam o fluxo dos sete casos de uso principais, onde dois diagramas representam cenários possíveis de um único caso de uso (*suc_generate_specific_report*), e outros dois (*suc_generate_specific_report* e *suc_view_student_detail*) possuem casos de uso com relação de inclusão. Relação à qual é descrita nas respectivas Especificações de Caso de Uso (consultar apêndice), e ilustrada nos subsequentes diagramas de sequência.

Para este documento, foi escolhido um dos oito diagramas de sequência para exemplificação. Os demais diagramas podem ser consultados no apêndice C. Abaixo,

na Figura 2, está ilustrado o Diagrama de Sequência de Geração de Relatório Mensal, descrito pelo caso de uso *suc_generate_general_monthly_report*.

Figura 2 – Diagrama de Sequência para Geração de Relatório Mensal



O fluxo demonstrado na Figura 2 pode ser descrito nos seguintes passos:

1. Um usuário autenticado, independente de seu perfil, acessa a seção de "Relatórios" do sistema.
2. A interface solicita ao *backend* e exibe a lista com todas as entradas de prontuário registradas no último mês.
3. O usuário então aciona a funcionalidade clicando no botão "Exportar".
4. A interface envia uma requisição ao sistema, que por sua vez busca os dados, utiliza uma lógica interna para compilar o arquivo no formato *xlsx* e, por fim, registra a operação na tabela de auditoria *ReportLog*.
5. Ao final do processo, o sistema retorna o arquivo gerado para o usuário, que é solicitado a fazer o download através de seu navegador.

3.3 DIAGRAMA DE CLASSE

O Diagrama de Classes representa a estrutura orientada a objetos do sistema, definindo suas principais entidades e relacionamentos. Este projeto adota o padrão arquitetural *Model-View-Template (MVT)*, uma variação do conhecido *Model-View-Controller (MVC)* utilizada pelo framework Django. Este padrão modulariza a aplicação em três componentes com responsabilidades distintas: o *Model*, responsável pela modelagem, regras de negócio e comunicação com o banco de dados; o

Template, a camada de apresentação que exibe os dados ao usuário (o *frontend*); e a *View*, componente que no Django assume o papel da *Controller* no MVC, sendo responsável por receber as requisições, interagir com o *Model* para manipular dados e renderizar o *Template* como resposta.

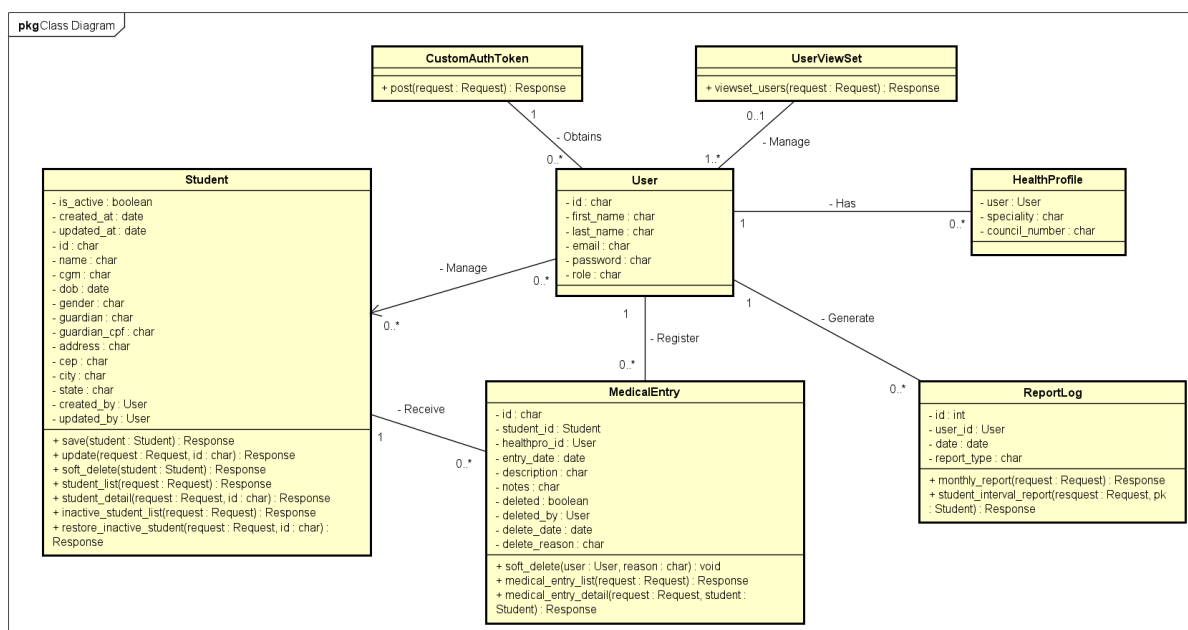
A estrita separação de responsabilidades no padrão MVT resulta em múltiplas classes. Mapear todas elas individualmente no diagrama de classes adicionaria uma complexidade desnecessária. Por isso, optou-se por uma abordagem que mescla a representação das *Models* e *Views*. Na classe *Student*, por exemplo (Figura 3), foram mapeados os atributos da *Model Student* e os métodos presentes na *View* que gerencia os estudantes.

Existem duas exceções a essa abordagem no diagrama: *CustomAuthToken* e *UserViewSet*. Para a gestão de usuários, foi utilizado um modelo customizado que estende o modelo abstrato do Django, complementando-o com atributos necessários ao projeto, como o campo **role**. O gerenciamento desse modelo é realizada por um *ViewSet*, aqui renomeado como *UserViewSet* para refletir qual modelo este se associa, uma classe do *Django REST Framework* que implementa automaticamente as operações básicas de CRUD (Create, Read, Update, Delete). Esta escolha agiliza o desenvolvimento e, por isso, foi representada separadamente para destacar seu papel arquitetural.

Já a classe *CustomAuthToken* é responsável exclusivamente pelo fluxo de autenticação via token (login), uma funcionalidade não presente nativamente pelo *ViewSet*. Ambas as classes auxiliam na separação de responsabilidades do *backend* e foram mantidas de forma distinta no diagrama para evidenciar sua importância.

O modelo final do projeto é composto por sete classes principais: *User*, *Student*, *HealthProfile*, *MedicalEntry*, *ReportLog*, *CustomAuthToken* e *UserViewSet*. A classe *User* gerencia os diferentes perfis (administradores, gestores, profissionais de saúde), enquanto a *Student* armazena as informações dos estudantes. A *HealthProfile* registra os dados específicos dos profissionais de saúde, a *MedicalEntry* centraliza os registros de atendimento dos estudantes e a *ReportLog* é responsável pela lógica de geração de relatórios e auditoria. As relações entre estas classes podem ser observadas na Figura 3.

Figura 3 – Diagrama de Classe do Sistema ProAMP



3.4 DIAGRAMA DE ENTIDADE E RELACIONAMENTO

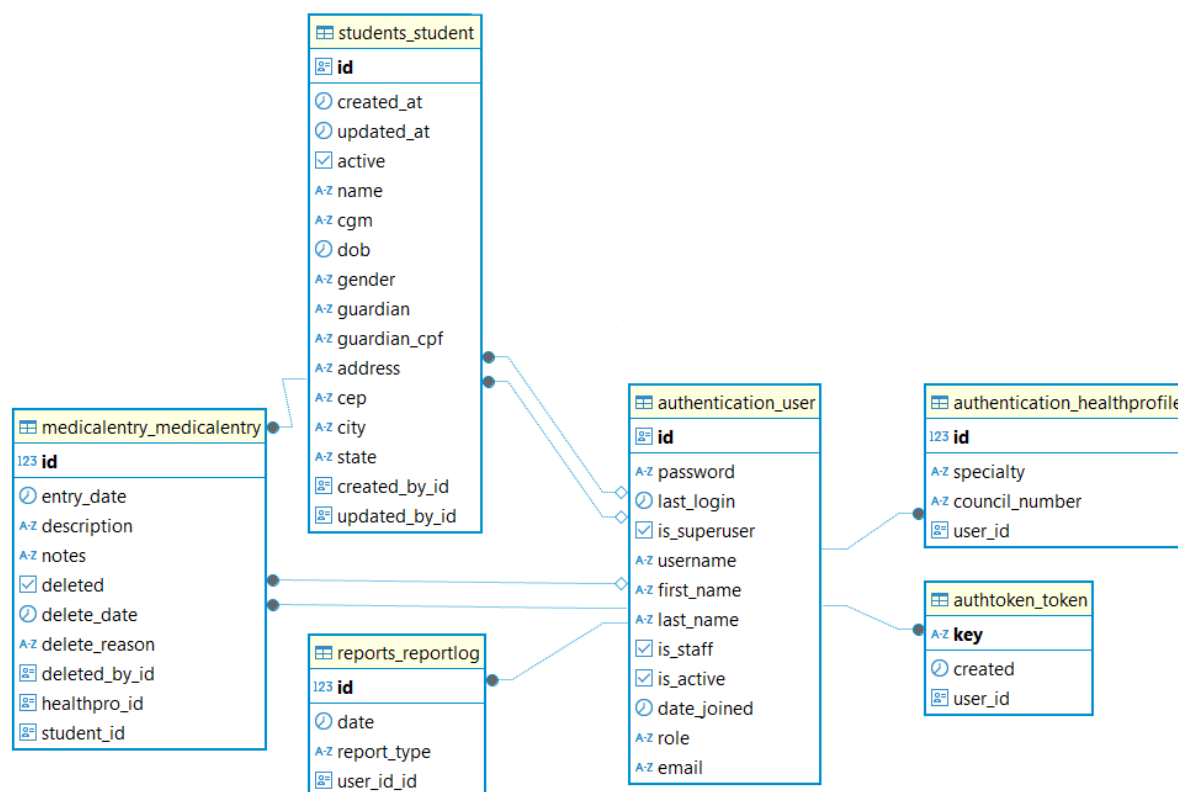
Este diagrama fornece uma visão conceitual da modelagem do banco de dados, onde os nomes das tabelas são prefixados pelo nome da aplicação Django à qual pertencem (e.g., **students_student**).

A entidade **authentication_user** atua como pilar para autenticação e controle de acesso, utilizando o atributo "role" para diferenciar os perfis de usuário. Informações específicas dos profissionais da saúde, como especialidade e número de conselho, são armazenadas na tabela **authentication_healthprofile**, que possui um relacionamento de um para um com **authentication_user**.

A entidade central do domínio é a **Student students_student**, que armazena os dados dos alunos e se conecta à **MedicalEntry (medicalentry_medicalentry)** em um relacionamento de um para muitos, representando os diversos registros de atendimento de cada aluno. A tabela **medicalentry_medicalentry** também se vincula diretamente ao profissional de saúde responsável pelo registro (**healthpro_id**).

Para fins de auditoria, a tabela **ReportLog (reports_reportlog)** documenta a geração de relatórios, associando cada log a um usuário. Por fim, a tabela **auth-token_token**, gerenciada automaticamente pelo *framework* Django, é responsável pela gestão dos *tokens* de autenticação para o acesso via API.

Figura 4 – Diagrama Entidade-Relacionamento do Sistema ProAMP

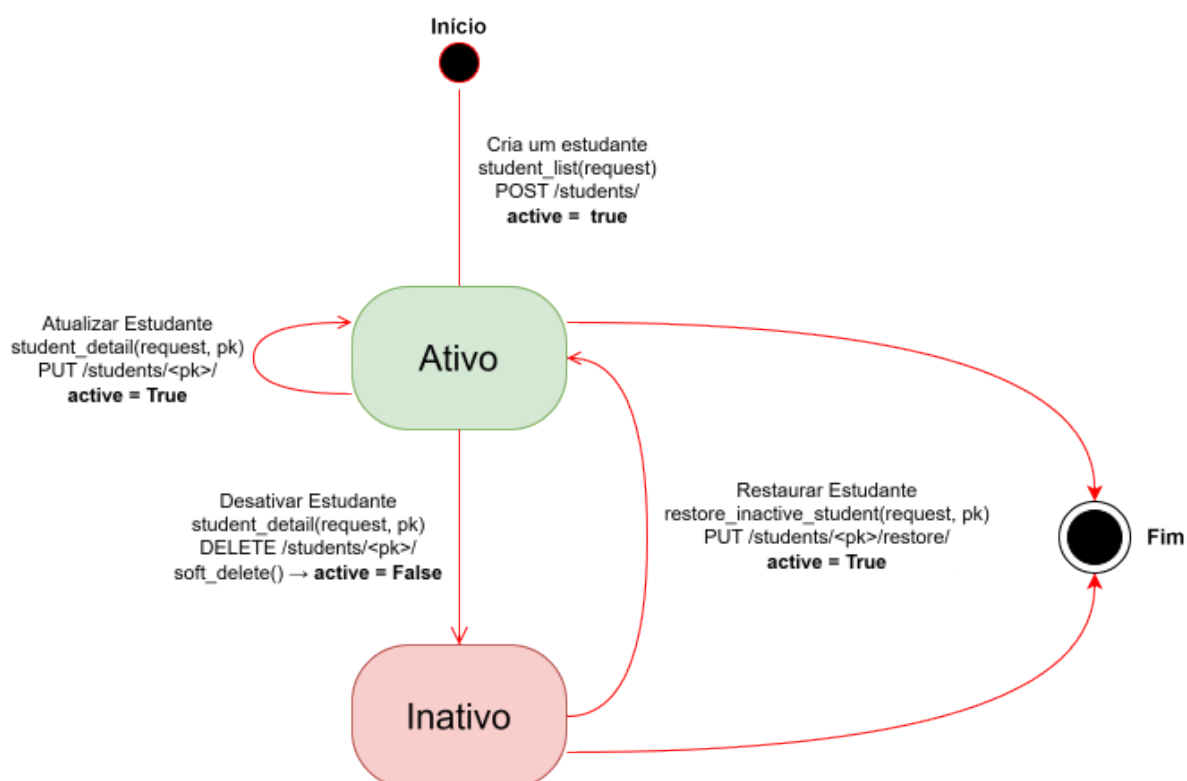


3.5 DIAGRAMA DE ESTADO

O Diagrama de Estado modela o comportamento dinâmico de um objeto ao longo de seu ciclo de vida, descrevendo seus possíveis estados e as transições entre eles. No sistema ProAMP, foram modelados os ciclos de vida de duas entidades principais: Estudantes e Entradas de Prontuário.

Para exemplificar, a Figura 5 apresenta o diagrama de estado para a entidade Estudante. O diagrama referente às Entradas de Prontuário pode ser consultado no Apêndice C.

Figura 5 – Diagrama de Estado para a entidade Estudante



O diagrama ilustra os dois estados principais de um estudante no sistema: *Ativo* e *Inativo*. Um novo estudante é criado e entra diretamente no estado *Ativo*. A transição para o estado *Inativo* ocorre por meio de uma operação de desativação, que implementa um mecanismo de *soft delete*, garantindo que o registro não seja permanentemente excluído. Um estudante inativo pode ser restaurado, retornando ao estado *Ativo*. Este modelo assegura a manutenção da integridade histórica dos dados, fundamental para a geração de relatórios.

4 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho teve como objetivo principal o desenvolvimento de um sistema de prontuário digital, o *ProAMP*, para o setor clínico da escola de educação especial Anna Maria Piettá. A conclusão deste projeto representa um possível avanço, caso devidamente implementado no processo administrativo da instituição, na modernização de seus processos, respondendo aos desafios propostos na introdução.

O sistema desenvolvido alcançou seus objetivos específicos com sucesso. Ele facilita o registro eletrônico de atendimentos, promovendo maior organização e acessibilidade às informações clínicas. A funcionalidade de geração automática de relatórios mensais foi implementada, otimizando um processo administrativo que antes era manual e suscetível a erros. A interface foi projetada para ser intuitiva, visando uma rápida adoção pelos usuários, e a estrutura do sistema garantiu a segurança e a proteção dos dados sensíveis dos alunos.

Com a implementação do *ProAMP*, espera-se uma redução nas inconsistências de dados e nos atrasos na entrega de relatórios, o que irá contribuir diretamente para a estabilidade financeira da instituição, garantindo o recebimento integral dos recursos do convênio com o SUS. O aprendizado obtido com este projeto reforçou a importância da tecnologia como ferramenta de otimização e suporte à gestão em instituições.

Como sugestão para trabalhos futuros, propõe-se a evolução da plataforma com funcionalidades que ampliem seu impacto. Dentre as possibilidades, destacam-se a implementação de um portal interativo para a comunicação com os responsáveis e a integração de ferramentas de análise de dados para subsidiar a tomada de decisões estratégicas na instituição. E para novas produções acadêmicas, um melhor planejamento semanal, e divisão entre documentação e desenvolvimento do projeto fariam grande diferença no tempo de produção.

REFERÊNCIAS

FOWLER, M. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. 3rd ed.. ed. [S.l.]: Addison-Wesley Professional, 2003. (The Addison-Wesley Object Technology Series). ISBN 0321193687. 8

SEYMOUR, T.; FRANTSVOG, D.; GRAEBER, T. Electronic health records (ehr). *American Journal of Health Sciences*, The Clute Institute, v. 3, n. 3, p. 201, 2012. Disponível em: <https://www.researchgate.net/profile/Dr-Tom-Seymour-2/publication/267226700_Electronic_Health_Records_EHR/links/57594e9508ae414b8e43a316/Electronic-Health-Records-EHR.pdf>. 6

GLOSSÁRIO

API RESTful: Um estilo de arquitetura de software para a criação de serviços web que facilitam a comunicação entre sistemas, como o *backend* e o *frontend* do *ProAMP*.

Ator: Em um Diagrama de Caso de Uso, representa um papel desempenhado por um usuário ou outro sistema que interage com o *ProAMP*. Exemplos no projeto são o administrador, o gestor e o profissional de saúde.

Backend: A parte do sistema que roda no servidor e é responsável pela lógica de negócio, processamento de dados, interação com o banco de dados e fornecimento de informações para o *frontend* via *APIs*.

Framework: Um conjunto de ferramentas, bibliotecas e boas práticas que serve como base para a construção de um software. No projeto, foram utilizados *Django*, *DRF* e *Vue.js*.

Frontend: A parte do sistema com a qual o usuário interage diretamente; a interface gráfica. No projeto, foi desenvolvida com o *framework* *Vue.js*.

ProAMP: Nome do sistema de prontuário digital desenvolvido, destinado ao setor clínico da escola de educação especial Anna Maria Piettá.

Versionamento: Processo de controle de diferentes versões de um código-fonte ou documento. No projeto, foi realizado com a ferramenta *Git* e hospedado na plataforma *GitHub*.

APÊNDICE A – REPOSITÓRIO DE DOCUMENTAÇÃO

A.1 PÁGINA DO PROJETO

Para facilitar o acesso e a compilação das documentações relacionadas ao projeto, foi organizado um repositório publicado através do *Github Pages*. Nele, todos os artefatos, diagramas e especificações estão disponíveis. A página pode ser acessada através do endereço abaixo:

Disponível em: <<https://placeholder.github.io/projeto-docs/>>

A.2 GLOSSÁRIO INTERATIVO ONLINE

Para facilitar a compreensão dos termos técnicos e específicos utilizados em todo o projeto, foi desenvolvido um glossário interativo como parte da documentação online. O glossário está sempre atualizado e pode ser acessado publicamente através do seguinte endereço:

Disponível em: <<https://placeholder.github.io/projeto-docs/pages/glossario.html>>

APÊNDICE B – ARTEFATOS DO PROJETO

B.1 PEDIDO DO INVESTIDOR

Este documento detalha as solicitações e os requisitos iniciais propostos pelo investidor, servindo como ponto de partida para a definição do escopo do projeto e o desenvolvimento do Documento de Visão.

Disponível em:

<<https://luismineo.github.io/proamp-docs/docs/artefatos/1-art-pedido-investidor.pdf>>

B.2 ESPECIFICAÇÃO SUPLEMENTAR

A Especificação Suplementar consolida todos os requisitos não funcionais do sistema, como usabilidade, confiabilidade, desempenho, segurança e manutenibilidade, que não são cobertos pelos casos de uso.

Disponível em:

<<https://luismineo.github.io/proamp-docs/docs/artefatos/2-art-espec-suplementar.pdf>>

B.3 DOCUMENTO DE VISÃO

O Documento de Visão oferece uma perspectiva geral do projeto, descrevendo o problema a ser solucionado, o público-alvo, as funcionalidades-chave e os objetivos de negócio que norteiam o desenvolvimento.

Disponível em:

<<https://luismineo.github.io/proamp-docs/docs/artefatos/3-art-visao.pdf>>

B.4 GLOSSÁRIO

O glossário compila todos os termos técnicos, gerais ou em inglês, além de siglas e acrônimos do projeto.

Disponível em:

<<https://luismineo.github.io/proamp-docs/docs/artefatos/0-art-glossario-projeto.pdf>>

B.5 ESPECIFICAÇÃO DE CASOS DE USO

As especificações de caso de uso descrevem os fluxos de eventos de cada funcionalidade do sistema. Abaixo estão listados os links de acesso para cada artefato detalhado.

B.5.1 Gerenciar Usuários

Disponível em:

<<https://luismineo.github.io/proamp-docs/docs/artefatos/especificacoes-caso-de-uso/specuc-01-pdf>>

B.5.2 Gerenciar Estudantes

Disponível em:

<<https://luismineo.github.io/proamp-docs/docs/artefatos/especificacoes-caso-de-uso/specuc-02-pdf>>

B.5.3 Ver Profissionais da Saúde

Disponível em:

<<https://luismineo.github.io/proamp-docs/docs/artefatos/especificacoes-caso-de-uso/specuc-03-pdf>>

B.5.4 Gerenciar Perfil

Disponível em:

<<https://luismineo.github.io/proamp-docs/docs/artefatos/especificacoes-caso-de-uso/specuc-04-pdf>>

B.5.5 Consultar Detalhes de Estudante

Disponível em:

<<https://luismineo.github.io/proamp-docs/docs/artefatos/especificacoes-caso-de-uso/specuc-05-pdf>>

B.5.6 Gerar Relatório Mensal

Disponível em:

<<https://luismineo.github.io/proamp-docs/docs/artefatos/especificacoes-caso-de-uso/specuc-06-pdf>>

B.5.7 Gerar Relatório Específico

Disponível em:

<<https://luismineo.github.io/proamp-docs/docs/artefatos/especificacoes-caso-de-uso/specuc-07-pdf>>

APÊNDICE C – DIAGRAMAS DO PROJETO

C.1 DIAGRAMA DE CASO DE USO

O Diagrama de Caso de Uso ilustra os atores do sistema e suas interações com as principais funcionalidades. Uma versão de maior qualidade pode ser acessada pelo endereço abaixo.

Disponível em:

<<https://luismineo.github.io/proamp-docs/docs/diagramas/svg/svg-diagrama-de-caso-de-uso.svg>>

C.2 DIAGRAMA DE CLASSES

O Diagrama de Classes detalha a estrutura estática do sistema, apresentando suas classes, atributos, métodos e os relacionamentos entre eles. Uma versão em alta resolução está disponível para consulta no link a seguir.

Disponível em:

<<https://luismineo.github.io/proamp-docs/docs/diagramas/svg/svg-diagrama-de-classe.svg>>

C.3 DIAGRAMA DE ENTIDADE-RELACIONAMENTO (DER)

Este diagrama modela a estrutura do banco de dados, representando as entidades, seus atributos e como elas se relacionam para garantir a integridade e a consistência dos dados.

Disponível em:

<<https://luismineo.github.io/proamp-docs/docs/diagramas/6-diagramas-der.pdf>>

C.4 DIAGRAMA DE IMPLANTAÇÃO

O Diagrama de Implantação descreve a arquitetura física do sistema, mostrando como os componentes de software são distribuídos nos componentes de hardware (nós) da infraestrutura.

Disponível em:

<<https://luismineo.github.io/proamp-docs/docs/diagramas/svg/svg-diagrama-de-implantacao.svg>>

C.5 DIAGRAMAS DE ESTADO

Os diagramas de estado descrevem as transições de status de objetos específicos do sistema. Abaixo, estão os links para os principais diagramas de estado do projeto.

C.5.1 Estado de Estudantes

Disponível em:

<<https://luismineo.github.io/proamp-docs/docs/diagramas/svg/svg-diagramas-estado/svg-diagrama-estudantes.svg>>

C.5.2 Estado de Entradas de Prontuário

Disponível em:

<<https://luismineo.github.io/proamp-docs/docs/diagramas/svg/svg-diagramas-estado/svg-diagrama-entradas-de-prontuario.svg>>

C.6 WORKFLOW AS-IS

Disponível em:

<<https://luismineo.github.io/proamp-docs/docs/diagramas/7-diagramas-workflow-as-is.pdf>>

C.7 DIAGRAMAS DE SEQUÊNCIA

Esta seção disponibiliza todos os diagramas de sequência, que detalham a interação entre objetos ao longo do tempo para cada caso de uso principal.

Disponível em:

<<https://luismineo.github.io/proamp-docs/docs/diagramas/5-diagramas-sequencia.pdf>>

Caso deseje acessar um diagrama de sequência específico, estes estão listados abaixo.

C.7.1 Sequência: Gerenciar Usuários

Disponível em:

<<https://luismineo.github.io/proamp-docs/docs/diagramas/svg/svg-diagramas-sequencia/01-Manage-Users-SD.svg>>

C.7.2 Sequência: Gerenciar Estudantes

Disponível em:

<<https://luismineo.github.io/proamp-docs/docs/diagramas/svg/svg-diagramas-sequencia/02-Manage-Students-SD.svg>>

C.7.3 Sequência: Gerenciar Perfil

Disponível em:

<<https://luismineo.github.io/proamp-docs/docs/diagramas/svg/svg-diagramas-sequencia/04-Manage-Profile-SD.svg>>

C.7.4 Sequência: Ver Profissionais da Saúde

Disponível em:

<<https://luismineo.github.io/proamp-docs/docs/diagramas/svg/svg-diagramas-sequencia/03-View-All-Health-Pros-SD.svg>>

C.7.5 Sequência: Consultar Detalhes de Estudante

Disponível em:

<<https://luismineo.github.io/proamp-docs/docs/diagramas/svg/svg-diagramas-sequencia/05-View-Student-Details-SD.svg>>

C.7.6 Sequência: Gerar Relatório Mensal

Disponível em:

<<https://luismineo.github.io/proamp-docs/docs/diagramas/svg/svg-diagramas-sequencia/06-Generate-General-Monthly-Report-SD.svg>>

C.7.7 Sequência: Gerar Relatório Específico A

Disponível em:

<<https://luismineo.github.io/proamp-docs/docs/diagramas/svg/svg-diagramas-sequencia/07A-Generate-Specific-Report-SD.svg>>

C.7.8 Sequência: Gerar Relatório Específico B

Disponível em:

<<https://luismineo.github.io/proamp-docs/docs/diagramas/svg/svg-diagramas-sequencia/07B-Generate-Specific-Report-SD.svg>>