

## **FACULDADE DE TECNOLOGIA DE FRANCA**

**Bianca Peixoto Lima  
Fabrício Zaneratto Ansanello  
Isaac de Sousa Resende  
May Uni II Kim  
Miguel Lima dos Reis**

**Sistema para a APADA: Melhoria do Processo de Cadastro e Organização Interna**

**UCE - Engenharia de Software II**

**FRANCA  
2025**

**Bianca Peixoto Lima  
Fabrício Zaneratto Ansanello  
Isaac de Sousa Resende  
May Uni II Kim  
Miguel Lima dos reis**

**Sistema para a APADA: Melhoria do Processo de Cadastro e Organização Interna**

**UCE - Engenharia de Software II**

Relatório Parcial de Unidade Curricular de Extensão – UCE de Engenharia de Software II, apresentado ao curso de Análise e Desenvolvimento de Sistemas da Faculdade de Tecnologia de Franca, para atender às disposições da Resolução nº 7 de 18 de dezembro de 2018, que estabelece as Diretrizes para a Extensão na Educação Superior.

Prof. Responsável: Me. Carlos Alberto Lucas

**FRANCA  
2025**

## SUMÁRIO

1 INTRODUÇÃO	4
2 PROJETO DE UNIDADE CURRICULAR DE EXTENSÃO - UCE	5
3 ESTUDOS TEÓRICOS	6
4 ATORES E ENTREVISTAS COM A COMUNIDADE	7
5. INTERVENÇÕES	8
6 PRINCIPAIS RESULTADOS	9
7 CONSIDERAÇÕES FINAIS	10
8 CONTRIBUIÇÕES DA UCE PARA A FORMAÇÃO DISCENTE	11

## 1 INTRODUÇÃO

O coordenador do curso de Análise e Desenvolvimento de Sistemas da Faculdade de Tecnologia de Franca no cumprimento de suas atribuições, apresentou às disposições da Resolução nº 7, e para atender tal demanda, os envolvidos optaram pela construção de uma Landing Page para Instituições Filantrópicas, Organização Não Governamental, Institutos e afins.

No início das aulas regulares, o projeto nos foi apresentado [alunos do 1º semestre do curso de Análise e Desenvolvimento de Sistemas = matutino e noturno]. Em seguida, nos organizamos em grupos com as respectivas responsabilidades, e na sequência buscamos a definição do ‘cliente’.

Após nos reunirmos com os responsáveis [cliente], foi elaborada e apresentada uma lista de ‘problemas’, que necessitavam de um recurso [software] sistêmico. Em seguida, estas questões problemas foram divididas em possíveis soluções sistêmicas.

Tivemos que analisar e identificar as regras do negócio da Instituição, e em seguida, iniciamos o desenvolvimento dos artefatos [a documentação completa de uma solução sistêmica] de Engenharia de Software.

É importante ressaltar que atenderemos com este projeto, as indicações da ONU através da articulação com os Objetivos do Desenvolvimento Sustentável [ODS]: a ODS 1 (erradicação da pobreza), a ODS 8 (trabalho decente e crescimento econômico), a ODS 9 (indústria, inovação e infraestrutura), a ODS 10 (redução das desigualdades) e a ODS 11 (cidades e comunidades sustentáveis).

## 2 PROJETO DE UNIDADE CURRICULAR DE EXTENSÃO – UCE

O presente projeto descreve o trabalho desenvolvido por um grupo de estudantes do curso de Análise e Desenvolvimento de Sistemas da Faculdade de Tecnologia de Franca, com o objetivo de criar um sistema para a APADA, com foco na melhoria do processo de cadastro e na organização interna da associação. A iniciativa busca aprimorar a gestão da instituição, alinhando-se aos valores de eficiência e desenvolvimento sustentável, que são fundamentais para a sociedade.

## OBJETIVO GERAL

O projeto tem como principal objetivo propor e desenvolver uma solução tecnológica abrangente para a APADA (Associação de Pais e Amigos dos Deficientes Auditivos), visando modernizar sua presença digital, otimizar sua gestão administrativa e fortalecer sua comunicação com o público. Para isso, o projeto contempla a criação de um novo site institucional — mais acessível, intuitivo e visualmente moderno —, a implementação de um sistema de cadastro de alunos, responsáveis, voluntários e colaboradores, e a estruturação de um banco de dados capaz de centralizar e organizar as informações essenciais da instituição.

A proposta nasce da observação das dificuldades enfrentadas pela APADA no tocante à gestão da informação, à visibilidade online e à integração de dados. Atualmente, a associação lida com processos manuais ou descentralizados, o que compromete a eficiência no atendimento e o alcance de suas ações sociais. A ausência de um sistema digital integrado gera gargalos operacionais, dificulta o acompanhamento de cadastros e limita a capacidade de planejamento e tomada de decisão da diretoria e dos demais responsáveis.

Com a implementação da solução proposta, espera-se não apenas otimizar os fluxos de trabalho da instituição, mas também ampliar sua capacidade de atuação e captação de apoio. O sistema permitirá uma gestão mais eficiente e transparente, ao mesmo tempo em que o novo site servirá como um canal de divulgação das atividades, projetos e campanhas, contribuindo para a ampliação do engajamento da comunidade e de possíveis parceiros.

Além de resolver problemas internos, o projeto busca promover a inclusão digital e o fortalecimento institucional da APADA, alinhando-se aos seus valores de acessibilidade, acolhimento e compromisso social. Esta iniciativa representa, portanto, uma importante ponte entre o conhecimento acadêmico e a realidade comunitária, reafirmando o papel transformador da tecnologia no apoio a organizações do terceiro setor.

## Objetivos específicos

A APADA enfrenta desafios estruturais e operacionais que comprometem sua eficiência organizacional, visibilidade institucional e capacidade de gestão de dados. A ausência de um sistema informatizado e a carência de ferramentas digitais dificultam a realização de atividades essenciais, como o cadastro e controle de alunos, voluntários e colaboradores, além de limitar a atuação da entidade no ambiente digital. Diante desse cenário, o presente projeto propõe os seguintes objetivos específicos:

- **Desenvolver um novo site institucional** para a APADA, com layout moderno, responsivo e acessível, com o intuito de ampliar a presença digital da instituição, facilitar a divulgação de projetos e eventos, e melhorar a comunicação com o público externo;
- **Implementar um sistema de cadastro digital**, capaz de registrar, armazenar e gerenciar informações de alunos, responsáveis, voluntários, colaboradores e demais envolvidos, proporcionando maior organização interna e acesso rápido a dados relevantes;
- **Projetar e estruturar um banco de dados relacional**, seguro e funcional, que permita a centralização e integridade das informações, além de possibilitar consultas e atualizações de forma eficiente;
- **Oferecer uma solução tecnológica de fácil uso e manutenção**, visando capacitar os membros da associação a operar o sistema de forma autônoma e contínua, sem depender de conhecimento técnico avançado;
- **Contribuir para a transformação digital da associação**, fortalecendo seus processos internos e sua imagem institucional diante da comunidade, apoiadores e órgãos públicos;
- **Estreitar a relação entre o ambiente acadêmico e o terceiro setor**, promovendo a aplicação prática dos conhecimentos adquiridos no curso de Análise e Desenvolvimento de Sistemas e estimulando o engajamento social dos estudantes.

## Metodologia

Para o desenvolvimento deste projeto, foram adotadas abordagens metodológicas combinando pesquisa bibliográfica, coleta de dados em campo e aplicação prática de técnicas da área de Análise e Desenvolvimento de Sistemas.

Inicialmente, realizou-se uma pesquisa exploratória e bibliográfica, com o objetivo de compreender as necessidades de organizações do terceiro setor em relação à transformação digital e gestão de dados. Foram analisados artigos acadêmicos, estudos de caso e referências técnicas sobre desenvolvimento de sites institucionais, sistemas de cadastro e banco de dados relacionais.

Posteriormente, foi realizada uma visita técnica à sede da APADA, com o intuito de levantar informações diretamente com os membros da associação. A coleta de dados envolveu entrevistas informais com representantes da instituição — como a presidente e a secretária —, observação do funcionamento atual da gestão interna, e análise dos principais desafios enfrentados pela organização. Esse contato direto com a realidade da instituição foi fundamental para a personalização da proposta às necessidades reais da entidade.

Com base nas informações coletadas, iniciou-se a fase de planejamento do sistema, definindo os requisitos essenciais para a construção do novo site, do sistema de cadastro e do banco de dados. Nessa etapa, foram utilizados diagramas (como casos de uso e BPMN) para mapear os fluxos de usuários e funcionalidades necessárias.

O desenvolvimento do projeto foi dividido em etapas, seguindo princípios de metodologias ágeis, permitindo revisões constantes, testes incrementais e adaptações conforme novas demandas surgiram durante a evolução do sistema. Além disso, está previsto o treinamento dos membros da associação que utilizarão o sistema, a fim de garantir sua correta utilização e manutenção futura. Também será realizada uma fase de testes e validação com dados reais, a fim de identificar possíveis ajustes e garantir o funcionamento pleno da solução antes da entrega final.

A metodologia adotada valoriza, portanto, a integração entre teoria e prática, priorizando a escuta ativa da instituição beneficiada, a aplicação dos

conhecimentos técnicos dos alunos envolvidos e o impacto social positivo da solução desenvolvida.

### **Termo de Abertura do Projeto (TAP)**

De acordo com o Guia PMBOK, o TAP é um dos documentos que faz parte de todo o ciclo de vida de um projeto. Nele, irá constar os objetivos e benefícios do projeto, estudos de viabilidade, as restrições de prazo e orçamento (ARTIA, 2024).

É a partir do TAP que é permitida a autorização de inicialização do projeto.

O quadro apresenta de forma parcial, algumas informações contempladas neste documento tão importante para o sucesso deste projeto

### **TAP**

**Situação atual:** A APADA, atua no apoio a pessoas com deficiência auditiva, a pesar de ter um impacto social, ainda é pouco conhecida, por grande parte da população, possuindo dificuldades em divulgação, captação de recursos, analfabetismo digital. **Justificativa do projeto:** Com o crescimento da comunicação digital, a presença online é essencial para sua visibilidade e captação de apoio. A criação da LP (Landing Page) tem como objetivo fortalecer sua presença na internet.

**Metas:**

- Desenvolvimento de uma Landing Page responsiva para a ONG.
- Apresentar a missão, visão, serviços e formas de contato. Criar um espaço que facilite a entrada de novos voluntários.

**Descrição do projeto:** Desenvolvimento de uma LP, com foco em acessibilidade, clareza de informações e identidade visual.

**Premissas:**

- A ONG forneceu todas as informações necessárias para a criação da LP
- A LP será acessada principalmente por meio de dispositivos moveis, desenvolvido de forma responsiva.

• O projeto realizado com os recursos da faculdade **Restrições:**

- Prazo limitado até o fim de semestre.
- A ONG possui limitações tecnológicas, como: site desatualizado, falta de equipe para manusear o site depois de concluído, falta de recursos tecnológicos.
- Nem todos os membros possuem disponibilidade para trabalhar fora do horário de aula.

**Usuário final:** Equipe gestora da ONG APADA, e visitantes como: (visitantes, voluntários, e até mesmo doadores) **Riscos:**

- Falta de resposta ou atraso na entrega de informações da ONG.
- Falta de engajamento do público.

**Cronograma:**

- Semanas 1-2 (Entrevista)
- Semanas 3-4 (Planejamento dos documentos)
- Semanas 5-6 (Elaboração dos documentos iniciais)
- Semanas 7-8 (Elaboração do BPMN)
- Semanas 9-10 (Finalização de documentos: Digitalização)

### **Matriz SWOT**

A análise SWOT é uma ferramenta de planejamento estratégico usada para identificar os pontos fortes (Strengths), pontos fracos (Weaknesses), oportunidades (Opportunities) e ameaças (Threats) de uma organização ou projeto. A sigla SWOT vem das iniciais desses quatro fatores em inglês. Essa análise é amplamente utilizada para avaliar a posição competitiva de uma empresa e desenvolver estratégias eficazes. A análise SWOT é importante por várias razões: Identificação de Potenciais Estratégicos: Ajuda a identificar as áreas onde a empresa pode maximizar suas forças e aproveitar as oportunidades, ao mesmo tempo em que mitiga as fraquezas e se prepara para as ameaças. Tomada de Decisões Informadas: Fornece uma visão clara das circunstâncias internas e externas, permitindo que os gestores tomem decisões mais informadas e estratégicas. Planejamento Estratégico: Auxilia no desenvolvimento de estratégias de curto e longo prazo, alinhando os objetivos da empresa com o ambiente em que ela opera. Comunicação e Alinhamento: Facilita a comunicação e o alinhamento dentro da organização, pois fornece uma linguagem comum para discutir os fatores que afetam o sucesso da empresa.

Abaixo está a Matriz SWOT esquematizada com base na análise geral da situação atual da APADA, considerando seus pontos fortes, fracos, oportunidades e ameaças:

Pontos fortes	Pontos fracos
Integração Social Muitos Eventos Apoio social	Sites desatualizados Grande rotatividade Analfabetismo digital
Representatividade	Pouca visibilidade
Oportunidades	Ameaças
Cursos EAD Parcerias com universidades	Desafios financeiros Concorrência

## 2W1H

A identificação das fraquezas na APADA é fundamental para a aplicação de medidas corretivas e para o aprimoramento contínuo de suas operações. Uma das principais fragilidades observadas é a ausência de um controle de estoque eficaz. Essa carência pode gerar problemas como o acúmulo desnecessário de materiais, a indisponibilidade de itens essenciais no momento certo e o aumento dos custos com armazenamento. Esses fatores podem resultar em perdas de recursos, dificuldades operacionais e comprometer a qualidade dos serviços prestados pela associação.

A metodologia 2W1H é uma ferramenta de apoio à gestão e ao planejamento estratégico, baseada na formulação de três questões norteadoras: What (O quê?), Why (Por quê?) e How (Como?). Sua aplicação tem como objetivo garantir maior clareza e assertividade na definição de metas, justificativas e formas de execução de uma ação ou projeto.

- What (O quê?): corresponde à identificação clara da atividade ou intervenção a ser realizada;
- Why (Por quê?): refere-se à justificativa da ação, considerando os benefícios esperados ou a necessidade diagnosticada;
- How (Como?): diz respeito à descrição dos procedimentos, recursos e estratégias que serão empregados para a realização da ação.

Abaixo está a 2W1H esquematizada diante do desenvolvimento da landing page e sistema para a Associação APADA

Questão Problema: Analfabetismo Digital

Oque?	Porque?	Como?
Realizar treinamento destinado aos responsáveis para cuidar do sistema	Afim de capacitar digitalmente os atores principais	Realizando treinamento em base nas funcionalidades presentes na solução sistêmicas
Criar um sistema de fácil uso e entendimento	Para tornar o uso simplificado e fácil aos atores	Desenvolvendo o sistema de fácil utilização

Questão Problema: Grande Rotatividade

Oque?	Porque?	Como?
Criar um tutorial com informações básicos do sistema	Para treinar os novos voluntários	Desenvolvendo um passo a passo de como utilizar o sistema
Realizar um teste para saber se o usuário está apto para usar o sistema	Para reconhecer o nível de conhecimento digital do usuário	Fazendo um teste com perguntas básicas sobre conhecimento digital

Questão Problema: Pouca visibilidade

Oque?	Porque?	Como?
Analizar o motivo da falta de acesso	Para analisar com eficiência	Contribuindo com boas divulgações
Analizar os meios de divulgações	Afim de entender como é feito as divulgações atualmente	Melhorando os meios de divulgações

Questão Problema: Site desatualizado

O'Que?	Porque?	Como?
Atualizar o site	Afim de tornar o site mais intuitivo e eficiente	Atualizando o site atual
Criar um novo site	Afim de contribuir para o mais acesso	Criando um novo site

O plano de ação desenvolvido para a resolução das fraquezas apontadas na Matriz SWOT, teve como objetivo a solução integral de cada uma das fraquezas a partir de duas soluções possíveis. Cada uma das soluções trouxe o entendimento necessário para a resolução dos problemas, além do detalhamento passo a passo para ser realizado.

Com base no diagnóstico realizado e validado com o docente responsável, foram propostas soluções para as principais fragilidades identificadas na APADA, como o analfabetismo digital, o site desatualizado, a baixa visibilidade na mídia e a alta rotatividade de membros. Entre as soluções levantadas, destacam-se o desenvolvimento de um novo site institucional, com foco em acessibilidade e divulgação, e a implementação de um sistema de cadastro que facilite a organização interna e a comunicação com os beneficiários.

A elicitação de requisitos foi uma etapa importante no desenvolvimento do sistema da APADA. Embora não tenha sido possível mapear todos os processos de forma completa, fizemos entrevistas com os gestores da Associação e realizamos uma visita à organização para observar diretamente as rotinas diárias. Durante essa visita, tivemos a oportunidade de interagir com alguns usuários, o que nos proporcionou uma visão mais clara das necessidades em relação ao sistema. Com as informações obtidas, conseguimos mapear os processos essenciais para o desenvolvimento das soluções propostas

### **A Estrutura Analítica do Projeto (EAP)**

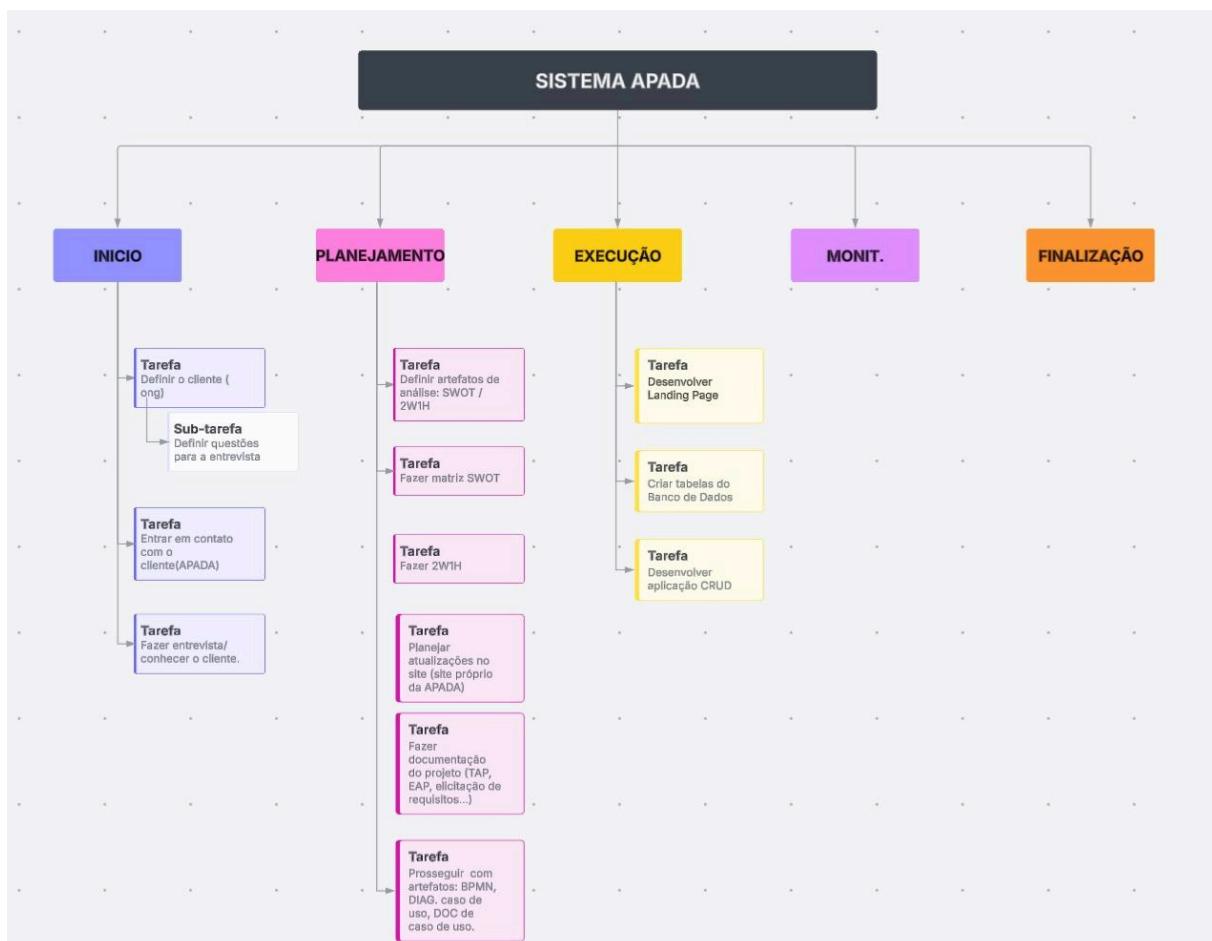
A Estrutura Analítica do Projeto (EAP) é uma ferramenta fundamental na gestão de projetos, utilizada para decompor o escopo do projeto em partes menores e mais gerenciáveis. A EAP organiza e estrutura as entregas e atividades em níveis hierárquicos, facilitando o planejamento, a execução, o controle e o monitoramento

do projeto.

Essa estrutura fornece uma visão clara de tudo o que precisa ser feito, desde as entregas principais até as tarefas mais detalhadas, garantindo que nenhuma parte importante seja negligenciada. Além disso, contribui para a definição precisa de responsabilidades, alocação de recursos e estimativas de tempo e custo.

No contexto do projeto desenvolvido para a APADA, a EAP foi elaborada com o objetivo de representar, de forma organizada, todas as etapas e entregas envolvidas no desenvolvimento da solução proposta, incluindo o novo site institucional e o sistema de cadastro de usuários.

A seguir, apresenta-se a Estrutura Analítica do Projeto (EAP) elaborada com base nas atividades realizadas no desenvolvimento do sistema para a APADA.

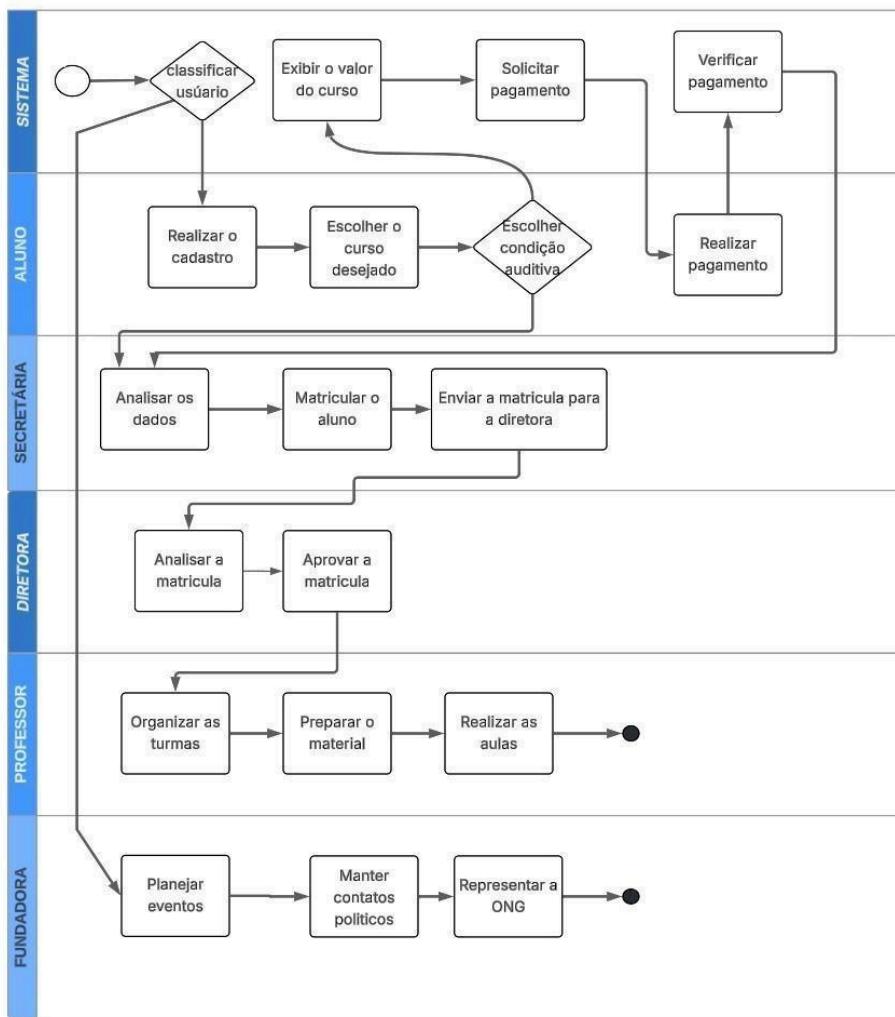


## BPMN

O BPMN (Business Process Model and Notation) é uma ferramenta padrão para a modelagem de processos de negócios. No contexto da APADA, o BPMN foi utilizado para mapear e representar os fluxos de trabalho essenciais no desenvolvimento do novo sistema proposto. Essa notação gráfica fornece uma visualização clara dos processos, identificando as etapas envolvidas, os responsáveis e os pontos críticos para otimização.

Ao utilizar o BPMN, buscamos criar um diagrama intuitivo e comprehensível, facilitando a comunicação entre as partes envolvidas no projeto e assegurando que todos os requisitos do sistema sejam bem documentados e implementados de forma eficiente. A padronização fornecida pelo BPMN também permite identificar possíveis falhas nos processos existentes e pontos de melhoria para a implementação do sistema de cadastro e do novo site.

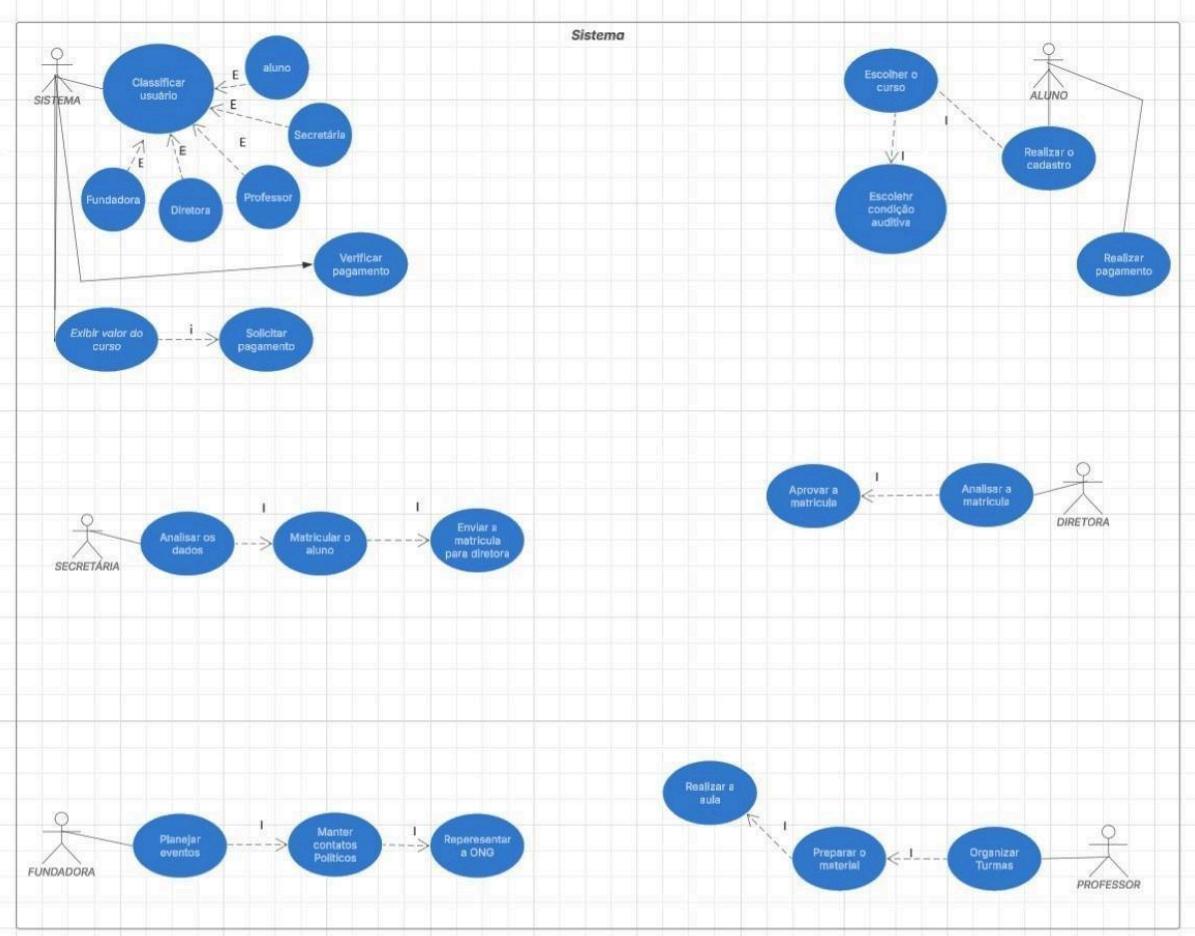
A baixo apresenta-se o diagrama BPMN referente aos processos mapeados no projeto desenvolvido para a APADA.



**Diagrama de Caso de Uso**

O Diagrama de Caso de Uso é uma ferramenta da UML (Unified Modeling Language) utilizada para representar, de forma clara e objetiva, as funcionalidades que o sistema deve oferecer e a interação dos usuários com essas funcionalidades. A elaboração deste diagrama foi baseada no mapeamento do processo realizado previamente por meio do BPMN (Business Process Model and Notation), o que permitiu identificar os principais atores e casos de uso envolvidos. Dessa forma, o Diagrama de Caso de Uso auxilia na definição dos requisitos funcionais do sistema, servindo como referência para as etapas seguintes do desenvolvimento.

A seguir, apresenta-se o Diagrama de Caso de Uso, elaborado com base no mapeamento de processos representado no BPMN, com o objetivo de identificar as principais interações entre os atores e o sistema desenvolvido para a APADA.



## DOCUMENTO DE CASO DE USO

O Documento de Caso de Uso descreve de forma detalhada cada funcionalidade identificada no Diagrama de Caso de Uso. Ele apresenta informações como os atores envolvidos, o fluxo de eventos, as pré-condições, as pós-condições e possíveis exceções. Esse documento é fundamental para garantir o entendimento claro e completo dos requisitos funcionais do sistema, servindo como base para as próximas etapas do desenvolvimento, como a modelagem de sequências e a implementação das funcionalidades.

A seguir, apresenta-se o Documento de Caso de Uso, contendo a descrição detalhada das funcionalidades identificadas, seus atores, fluxos de eventos e condições, fundamentando os requisitos funcionais do sistema desenvolvido para a APADA.

**ID:** UC001: Classificar o usuário

**Raia:** Sistema

**Descrição:** Mostrar a tela inicial para que o usuário selecione seu perfil.

**Ator Primário:** Usuário

**Pré-condição:** O sistema deve estar acessível.

**Cenário Principal:**

1.O sistema pergunta quem está acessando.

2.O usuário escolhe o perfil.

3.O sistema direciona o fluxo conforme a escolha.

**Pós-condição:** O sistema segue o fluxo adequado.

**Cenário Alternativo:** Nenhum perfil selecionado. O sistema aguarda.

**Inclusão:** Não possui.

**Extensão:** Aluno, secretária, diretora, professor e fundadora.

**ID:** UC002: Realizar o cadastro

**Raia:** Aluno

**Descrição:** Preencher informações pessoais para criar o cadastro no sistema.

**Ator Primário:** Aluno

**Pré-condição:** O aluno deve ter escolhido o perfil “Aluno”.

**Cenário Principal:**

1.O aluno preenche seus dados.

2.O sistema armazena as informações.

**Pós-condição:** Cadastro do aluno criado.

**Cenário Alternativo:** Dados incompletos. O sistema avisa o erro.

**Inclusão:** UC003: Escolher curso desejado.

**Extensão:** Não possui.

**ID:** UC003: Escolher o curso desejado

**Raia:** Aluno

**Descrição:** Escolher o curso que deseja se matricular.

**Ator Primário:** Aluno

**Pré-condição:** O aluno já deve estar com o cadastro feito.

**Cenário Principal:**

1.O sistema mostra os cursos disponíveis.

2.O aluno escolhe o curso.

**Pós-condição:** Curso escolhido.

**Cenário Alternativo:** Nenhum curso selecionado. O sistema continua aguardando.

**Inclusão:** UC004: Escolher condição auditiva.

**Extensão:** Não possui.

**ID:** UC004: Escolher condição auditiva

**Raia:** Aluno

**Descrição:** Informar se é surdo ou ouvinte.

**Ator Primário:** Aluno

**Pré-condição:** Ter escolhido o curso.

**Cenário Principal:**

1.O sistema pergunta sobre a condição auditiva.

2.O aluno informa se é surdo ou ouvinte.

**Pós-condição:** Sistema armazena a informação.

**Cenário Alternativo:** Informação não fornecida. O sistema alerta.

**Inclusão:** UC003: Escolher curso desejado. **UC009:** Analisar o cadastro.

**Extensão:** Não possui.

**ID:** UC005: Exibir valor do curso

**Raia:** Sistema

**Descrição:** Mostrar o valor do curso de acordo com o perfil do aluno.

**Ator Primário:** Sistema

**Pré-condição:** O aluno deve ter informado a condição auditiva.

**Cenário Principal:**

1.O sistema mostra o valor do curso, conforme o caso.

**Pós-condição:** Valor exibido.

**Cenário Alternativo:** Erro no sistema. Valor não exibido.

**Inclusão:** UC006: Solicitar pagamento.

**Extensão:** UC004: Escolher condição auditiva.

**ID:** UC006: Solicitar pagamento

**Raia:** Sistema

**Descrição:** Pedir que o aluno realize o pagamento.

**Ator Primário:** Sistema

**Pré-condição:** Valor do curso exibido.

**Cenário Principal:**

1.O sistema apresenta opções de pagamento.

2.O aluno escolhe como pagar.

**Pós-condição:** Pagamento iniciado.

**Cenário Alternativo:** Pagamento não iniciado. O sistema aguarda.

**Inclusão:** UC007: Realizar pagamento.

**Extensão:** Não possui.

**ID:** UC007: Realizar pagamento

**Raia:** Aluno

**Descrição:** Efetuar o pagamento do curso.

**Ator Primário:** Aluno

**Pré-condição:** Ter escolhido uma forma de pagamento.

**Cenário Principal:**

1. O aluno realiza o pagamento.

**Pós-condição:** Pagamento enviado ao sistema.

**Cenário Alternativo:** Falha no pagamento. Aluno tenta novamente.

**Inclusão:** UC008: Verificar pagamento.

**Extensão:** Não possui

**ID:** UC008: Verificar pagamento

**Raia:** Sistema

**Descrição:** Conferir se o pagamento foi aprovado.

**Ator Primário:** Sistema

**Pré-condição:** O pagamento foi enviado.

**Cenário Principal:**

1.O sistema verifica o pagamento.

2.Se aprovado, libera os dados do aluno.

**Pós-condição:** Pagamento confirmado.

**Cenário Alternativo:** Pagamento recusado. O sistema avisa o erro.

**Inclusão:** UC009: Analisar os dados.

**Extensão:** Não possui.

**ID:** UC009: Analisar os dados

**Raia:** Secretaria

**Descrição:** Verificar os dados do aluno após o pagamento.

**Ator Primário:** Secretaria

**Pré-condição:** Pagamento aprovado.

**Cenário Principal:**

1.A secretaria analisa os dados.

2.Organiza as informações do aluno.

**Pós-condição:** Dados prontos para matrícula.

**Cenário Alternativo:** Dado incompleto. A secretaria pede correção.

**Inclusão:** UC010: Matricular o aluno.

**Extensão:** Não possui.

**ID:** UC010: Matricular o aluno

**Raia:** Secretaria

**Descrição:** Fazer a matrícula do aluno no sistema.

**Ator Primário:** Secretaria

**Pré-condição:** Dados analisados.

**Cenário Principal:**

1. A secretaria registra a matrícula.

**Pós-condição:** Matrícula feita.

**Cenário Alternativo:** Erro no sistema. Matrícula não concluída.

**Inclusão:** UC011: Enviar a matrícula para a diretora.

**Extensão:** Não possui.

**ID:** UC011: Enviar a matrícula para a diretora

**Raia:** Secretaria

**Descrição:** Encaminhar os dados da matrícula para a diretora.

**Ator Primário:** Secretaria

**Pré-condição:** Matrícula feita.

**Cenário Principal:**

1. A secretaria envia os dados.

**Pós-condição:** Diretora recebe a matrícula.

**Cenário Alternativo:** Falha no envio. A secretaria tenta novamente.

**Inclusão:** UC012: Analisar a matrícula.

**Extensão:** Não possui.

**ID:** UC012: Analisar a matrícula

**Raia:** Diretora

**Descrição:** Verificar as informações da matrícula do aluno.

**Ator Primário:** Diretora **Pré-condição:** Matrícula recebida.

**Cenário Principal:**

1. A diretora analisa os dados.

**Pós-condição:** Matrícula pronta para aprovação.

**Cenário Alternativo:** Dúvida. A diretora encaminha para a fundadora.

**Inclusão:** UC013: Aprova matricula.

**Extensão:** Não possui.

**ID:** UC013: Aprovar a matrícula

**Raia:** Diretora

**Descrição:** Autorizar a matrícula do aluno.

**Ator Primário:** Diretora

**Pré-condição:** Análise feita.

**Cenário Principal:**

- 1.A diretora aprova a matrícula.
- 2.Encaminha ao professor.

**Pós-condição:** Matrícula finalizada.

**Cenário Alternativo:** Encaminhar à fundadora em caso de dúvida.

**ID:** UC014: Organizar as turmas

**Raia:** Professor

**Descrição:** Separar os alunos em turmas conforme perfil.

**Ator Primário:** Professor

**Pré-condição:** Matrícula aprovada.

**Cenário Principal:**

1. O professor agrupa os alunos em turmas.

**Pós-condição:** Turmas organizadas.

**Cenário Alternativo:** Nenhum.

**Inclusão:** UC015: Preparar matricula.

**Extensão:** Não possui.

**ID:** UC015: Preparar o material

**Raia:** Professor

**Descrição:** Criar o material para as aulas, adaptando se necessário.

**Ator Primário:** Professor

**Pré-condição:** Turmas formadas.

**Cenário Principal:**

- 1.O professor prepara o material.
- 2.Se houver aluno surdo, adapta o conteúdo.

**Pós-condição:** Material pronto.

**Cenário Alternativo:** Nenhum.

**Inclusão:** UC016: Realizar as aulas.

**Extensão:** Não possui.

**ID:** UC016: Realizar as aulas

**Raia:** Professor

**Descrição:** Ministrar as aulas para os alunos.

**Ator Primário:** Professor

**Pré-condição:** Material preparado.

**Cenário Principal:**

1. O professor dá as aulas.

**Pós-condição:** Alunos assistem às aulas.

**Cenário Alternativo:** Nenhum.

**Inclusão:** Não possui.

**Extensão:** Não possui.

**ID:** UC017: Planejar eventos

**Raia:** Fundadora

**Descrição:** Organizar eventos especiais da ONG.

**Autor Primário:** Fundadora

**Pré-condição:** Nenhuma.

**Cenário Principal:**

1. A fundadora planeja os eventos.

**Pós-condição:** Eventos planejado

**Cenário Alternativo:** Nenhum.

**Inclusão:** UC018: Manter contatos políticos.

**Extensão:** UC001 Classificar usuário

**ID:** UC018: Manter contatos políticos

**Raia:** Fundadora

**Descrição:** Manter relacionamentos com políticos e parceiros.

**Autor Primário:** Fundadora

**Pré-condição:** Nenhuma

**Cenário Principal:**

1. A fundadora entra em contato com representantes.

**Pós-condição:** Relações mantidas.

**Cenário Alternativo:** Nenhum.

**Inclusão:** Representar a ONG

**Extensão:** Não possui.

**ID:** UC019: Representar a ONG

**Raia:** Fundadora

**Descrição:** Falar em nome da ONG em eventos e reuniões

**Autor Primário:** Fundadora

**Pré-condição:** Ser convidada ou convocada

**Cenário Principal:**

1. A fundadora representa a ONG publicamente

**Pós-condição:** ONG representada

**Cenário Alternativo:** Nenhum.

**Inclusão:** Representar a ONG

**Extensão:** Não possui.

## DOCUMENTO DE REQUISITOS

A Documentação de Requisitos é um documento essencial no desenvolvimento de sistemas, pois reúne todos os requisitos que o sistema deve atender, incluindo requisitos funcionais, não funcionais e regras de negócio. Ela serve como base para orientar a equipe de desenvolvimento, garantindo que as funcionalidades e restrições estejam claras e que o sistema atenda às necessidades do cliente.

A seguir, apresenta-se a Documentação de Requisitos elaborada para o sistema desenvolvido para a APADA, contendo os requisitos funcionais, não funcionais e regras de negócio identificados durante o levantamento de necessidades.

### (RAIA SISTEMA)

- **ID:** RF01
  - **NOME:** Classificar Usuário.
  - **Descrição:** O sistema deverá identificar o tipo de usuário que o acessam, para direcionar o usuário para o campo correspondente (Aluno, Secretária, Fundadora, professor).
  - **CATEGORIA:** Evidente.
  - **PRIORIDADE:** Essencial.
  - **INFORMAÇÕES:** aluno, Secretária, Fundadora, diretora. • **REGRA DO NEGÓCIO:** (Não Existente).
  
- **ID:** RF02
  - **NOME:** Exibir valor do curso.
  - **Descrição:** O Sistema deverá, após o login, direcionamento e escolha do curso, exibir o valor do curso com base no plano e forma de pagamento escolhidos posteriormente pelo usuário.
  - **CATEGORIA:** Evidente.
  - **PROPRIEDADES:** Essencial
  - **INFORMAÇÕES:** Usuário, login, escolha de curso.
  - **REGRA DO NEGÓCIO:** Não possui.
  
- **ID:** RF03
  - **NOME:** Solicitar pagamento.
  - **Descrição:** O sistema deverá, após exibir a forma de pagamento, solicitar o mesmo.
  - **CATEGORIA:** Evidente.
  - **PROPRIEDADES:** Essencial.
  - **INFORMAÇÕES:** Usuário, condições de pagamento.
  - **REGRA DO NEGÓCIO:** Escolha do curso e forma de pagamento.
  
- **ID:** RF04

- **NOME:** Verificar pagamento.
- **Descrição:** O sistema deverá conferir se o pagamento foi realizado.
- **CATEGORIA:** Oculto
- **PROPRIEDADE:** Essencial.
- **INFORMAÇÕES:** Pagamento, revisão, aprovação.
- **REGRA DO NEGÓCIO:** O cliente deve realizar o pagamento.

**(RAIA ALUNO)**

- **ID:** RF05
  - **Nome:** Escolher curso desejado.
  - **Descrição:** O sistema deverá permitir que o usuário selecione o curso de interesse, a partir de uma lista de cursos disponíveis.
  - **CATEGORIA:** Evidente.
  - **PROPRIEDADE:** Essencial.
  - **INFORMAÇÕES:** Usuário, curso.
  - **REGRA DO NEGÓCIO:** O curso deve ter vaga, o usuário deve estar logado
- **ID:** RF06
  - **Nome:** Selecionar condição auditiva.
  - **Descrição:** O sistema deverá permitir que o usuário selecione sua condição auditiva a partir de uma lista pré-definida.
  - **CATEGORIA:** Evidente.
  - **PROPRIEDADE:** Essencial.
  - **INFORMAÇÕES:** Usuário, condição auditiva.
  - **REGRA DO NEGÓCIO:** O usuário deverá provar que possui alguma condição auditiva.
- **ID:** RF07
  - **NOME:** Realizar cadastro.
  - **Descrição:** O sistema deverá permitir que o usuário preencha e envie seus dados pessoais de contato.
  - **CATEGORIA:** Evidente.
  - **PROPRIEDADE:** Essencial.
  - **INFORMAÇÕES:** Cadastro.
  - **REGRA DO NEGÓCIO:** Passar informações corretas.
- **ID:** RF08
  - **NOME:** Realizar pagamento.
  - **Descrição:** O sistema deverá permitir que o usuário efetue o pagamento do curso selecionado por meio de diferentes formas de pagamentos disponíveis.
  - **CATEGORIA:** Evidente.
  - **PROPRIEDADE:** Essencial.
  - **INFORMAÇÕES:** Pagamento.
  - **REGRA DO NEGÓCIO:** O pagamento deve ser efetuado.

**(RAIA SECRETARIA)**

- **ID:** RF09
- **NOME:** Analisar dados.
- **Descrição:** A secretaria deverá analisar os dados da matrícula e pagamento do aluno que não possui deficiência auditiva.
- **CATEGORIA:** Oculta.
- **PROPRIEDADE:** Essencial.
- **INFORMAÇÕES:** Análise, dados, matrícula.
- **REGRA DO NEGÓCIO:** Não possui.
  
- **ID:** RF010
- **NOME:** Matrícula do aluno.
- **Descrição:** A secretaria deverá efetuar a matrícula do aluno com base nas informações dele.
- **CATEGORIA:** Oculta.
- **PROPRIEDADE:** Essencial.
- **INFORMAÇÕES:** Matrícula, aluno.
- **REGRA DO NEGÓCIO:** Não possui.
  
- **ID:** RF011
- **NOME:** Enviar a matrícula para a diretora.
- **Descrição:** A secretaria deverá enviar os dados dos usuários que se inscreveram para o curso para a diretora.
- **CATEGORIA:** Oculta.
- **PROPRIEDADE:** Essencial.
- **INFORMAÇÕES:** Envio, matrícula.
- **REGRA DO NEGÓCIO:** Não possui.

**(RAIA DIRETORA)**

- **ID:** RF012
- **NOME:** Analisar a matrícula.
- **Descrição:** A diretora deverá analisar a matrícula do aluno para garantir sua participação no curso.
- **CATEGORIA:** Oculta.
- **PROPRIEDADE:** Importante.
- **INFORMAÇÕES:** Análise, matrícula.
- **REGRA DO NEGÓCIO:** Não possui.
  
- **ID:** RF013
- **NOME:** Aprovar a matrícula.
- **Descrição:** A diretora deverá aprovar a matrícula caso as informações estejam todas corretas e o pagamento efetuado.
- **CATEGORIA:** Oculta.
- **PROPRIEDADE:** Essencial.
- **INFORMAÇÕES:** Aprovação, matrícula.

- **REGRA DO NEGÓCIO:** Não possui

**(RAIA PROFESSOR)**

- **ID:** RF014
- **NOME:** Organizar turmas.
- **DESCRIÇÃO:** O professor deverá com base nos alunos matriculados organizar as turmas.
- **CATEGORIA:** Oculto.
- **PROPRIEDADE:** Importante.
- **INFORMAÇÕES:** Matrículas, planejamento.
- **REGRA DO NEGÓCIO:** Informações de alunos e planejamento de eventos devem estar disponíveis e repassadas para o professor.
  
- **ID:** RF015
- **NOME:** Preparar o material.
- **DESCRIÇÃO:** O professor deverá fazer o planejamento das aulas junto do material de aula.
- **CATEGORIA:** Oculto.
- **PROPRIEDADE:** Importante.
- **INFORMAÇÕES:** Planejamento, turmas (já formadas).
- **REGRA DO NEGÓCIO:** O professor deve ter as turmas já formadas, ter acesso ao planejamento.
  
- **ID:** RF016
- **NOME:** Realizar as aulas.
- **DESCRIÇÃO:** O professor deve realizar as aulas de acordo com o cronograma estabelecido após realizar a preparação do material.
- **CATEGORIA:** Evidente.
- **PROPRIEDADE:** Essencial.
- **INFORMAÇÕES:** Planejamento, aulas.
- **REGRA DO NEGÓCIO:** O Professor deve ter acesso as turmas (já formadas), planejamento e ter material já pronto.

**(RAIA FUNDADORA)**

- **ID:** RF017
- **NOME:** Planejar eventos.
- **DESCRIÇÃO:** A fundadora deverá analisar o contexto social atual e desenvolver eventos que busquem ampliar o alcance de divulgação da instituição, tendo também como objetivo principal a arrecadação de fundos para manter a Associação.
- **CATEGORIA:** Evidente.
- **PROPRIEDADE:** Essencial.
- **INFORMAÇÕES:** Fundadora, analisar contexto, planejar evento.
- **REGRA DO NEGÓCIO:** Desenvolver evento que seja condizente com o momento/contexto social.
  
- **ID:** RF018

- **NOME:** Manter contatos “políticos”.
  - **Descrição:** A fundadora deverá cultivar relações com apoiadores e patrocinadores da instituição.
  - **CATEGORIA:** Evidente.
  - **PROPRIEDADE:** Essencial.
  - **INFORMAÇÕES:** Fundadora, apoiadores/patrocinadores.
  - **REGRA DO NEGÓCIO:** Não possui
- 
- **ID:** RF019
  - **NOME:** Representar a ONG.
  - **Descrição:** A fundadora deverá frequentar eventos sociais que possam trazer oportunidades para a associação e seus usuários, agregar apoiadores e melhorar a visibilidade a instituição.
  - **CATEGORIA:** Evidente.
  - **PROPRIEDADE:** Essencial.
  - **INFORMAÇÕES:** Eventos sociais, fundadora, expor a visão, missão e valores da ONG.
  - **REGRA DO NEGÓCIO:** Não possui.
  - **REGRA DO NEGÓCIO:** Não possui.

### 3 ESTUDOS TEÓRICOS

Na presente seção, devem ser apresentados os materiais de estudo para a criação do software, como exemplos abaixo:

1)COHN, Mike. Desenvolvimento de Software com Scrum. Editora Bookman. 1<sup>a</sup>. Edição. (2023). ISBN: 9788577808076

Concepções/estudos empregados: Leitura e prática = desenvolvimento dos artefatos de software

2)VALENTE, Marco Túlio. Engenharia de Software Moderna. Editora: Independente. 1<sup>a</sup> edição (2022). ISBN-10: 6500019504

Concepções/estudos empregados: Leitura e prática = desenvolvimento dos artefatos de software

3)SUTHERLAND, Jeff. Scrum: A arte de fazer o dobro do trabalho na metade do tempo. (2019); Editora Sextante. ISBN-10: 8543107164

Concepções/estudos empregados: Leitura e prática = desenvolvimento dos

artefatos de software

4)PRESSMAN, Roger. Engenharia de Software. 9<sup>a</sup> Edição. (2021). ISBN-10: 6558040107

Concepções/estudos empregados: Leitura e prática = desenvolvimento dos artefatos de software

## 4 ATORES E ENTREVISTAS COM A COMUNIDADE

### ATORES E ENTREVISTAS COM A COMUNIDADE

Na presente seção, será apresentada e caracterizada a comunidade social atendida pela APADA, com foco em seus atores principais e a coleta de dados relacionados à ação social realizada pela ONG. Além disso, serão descritos os locais de coleta de dados, destacando as interações entre os membros da associação.

Na APADA, os stakeholders desempenham papéis essenciais na condução e funcionamento da instituição. Os principais membros da organização se destacam nesse contexto: a fundadora, a coordenadora, a Diretora, o Professor e a Secretária.

**Fundadora:** A fundadora desempenha um papel fundamental na liderança da APADA. Ela é responsável pela organização e planejamento de eventos e pelo contato com políticos, além de representar a ONG em diversas ocasiões. Sua visão e habilidades de gestão permitem que ela guie a APADA com determinação, sempre buscando o melhor para a comunidade.

**Coordenadora:** A coordenadora supervisiona as atividades diárias da associação, garantindo que as operações fluam corretamente. Ela também auxilia na coordenação de eventos e no desenvolvimento de estratégias educacionais e sociais para os membros da APADA.

**Diretora:** A diretora atua de maneira fundamental na gestão interna e externa da associação, supervisionando o progresso das atividades e coordenando as equipes envolvidas. Ela também tem um papel importante na organização e acompanhamento dos projetos e na gestão de recursos.

**Professor:** O professor é responsável pelas atividades pedagógicas da APADA, contribuindo para o desenvolvimento educacional dos membros. Ele realiza treinamentos, aulas e workshops, garantindo que os alunos adquiram os conhecimentos necessários para superar o analfabetismo digital e outras dificuldades.

**Secretária:** A secretária tem a função de administrar as questões administrativas e burocráticas da associação. Ela é responsável por aprovar as matrículas dos alunos, organizar os documentos e auxiliar na gestão de registros internos da organização.

O gerenciamento de processos na APADA enfrenta desafios relacionados à integração das tecnologias acessíveis e à adaptação de sistemas que atendam às necessidades específicas dos membros. A implementação de soluções como sistemas de cadastro e atualizações do site é essencial para garantir que todos os

membros tenham acesso às informações necessárias de forma eficiente.

O Projeto Interdisciplinar (PI), voltado para a criação de uma solução digital para a APADA, teve início em fevereiro de 2025. Desde então, foram realizadas diversas etapas, com a divisão dos alunos em equipes responsáveis por áreas específicas, além da escolha de um gerente por grupo para coordenar as atividades.

No início do projeto, foram definidos os objetivos da equipe, seguido da elaboração de artefatos iniciais, como a matriz SWOT, o 2W1H e o levantamento preliminar de requisitos. Em seguida, desenvolveu-se a EAP (Estrutura Analítica do Projeto) e o BPMN, representando os processos da associação com base nas informações levantadas.

A seguir, apresenta-se um resumo das atividades desenvolvidas pela equipe durante a execução do projeto, destacando a contribuição conjunta de todos os integrantes nas diferentes etapas do trabalho.

- **Isaac de Sousa Resende:** Gerente do módulo da APADA, responsável pelo desenvolvimento da Missão Visão e Valor, SWOT, 2W1H, Perguntas da Entrevista, BPMN, EAP, TAP, Documentação de requisitos, Diagrama de Atividades, Diagrama de Máquina de Estado, Diagrama de sequência, Portabilidade e Digitalização dos documentos.
- **Bianca Peixoto Lima:** Vice gerente do módulo da APADA, responsável pelo desenvolvimento do SWOT, 2W1H, perguntas da entrevista, Digitalização dos documentos, 2W1H, Documentos de requisitos e Diagrama de sequência.
- **Miguel Lima dos Reis:** Responsável pelo desenvolvimento do SWOT, 2W1H, Perguntas da Entrevista, TAP, Digitalização dos documentos, Documentos de requisitos e Diagrama de sequência.
- **May Uni II Kim:** Responsável pelo desenvolvimento do 2W1H, perguntas da entrevista e Diagrama de Caso se Uso, EAP, Diagrama de sequência e Digitalização dos documentos.
- **Fabrício Zaneratto Ansanello:** Responsável pelo desenvolvimento das perguntas para a entrevista, 2W1H, Digitalização dos documentos, Documentos de requisitos, SWOT, Diagrama de Atividades e Diagrama de Máquina de Estado.

[OBJ-103]

## 5 INTERVENÇÕES

As intervenções realizadas ao longo do projeto tiveram como objetivo principal aproximar o grupo da realidade enfrentada pela APADA e propor soluções viáveis para suas demandas. A partir do diagnóstico inicial, foram identificados problemas como o site desatualizado, a baixa presença digital da instituição, a dificuldade de organização nos cadastros de alunos e colaboradores, além da necessidade de maior visibilidade junto à comunidade.

Para enfrentar essas questões, as seguintes ações foram conduzidas:

- Levantamento das necessidades da instituição, por meio da análise do funcionamento atual e identificação de fragilidades.
- Proposta de criação de uma landing page moderna, com design responsivo, para promover maior alcance na divulgação de atividades e eventos da APADA.
- Desenvolvimento de um sistema de cadastro e banco de dados relacional, voltado à organização de informações de alunos,



professores e colaboradores, possibilitando maior controle e acessibilidade.

- Produção de artefatos de engenharia de software, tais como BPMN, casos de uso, documentação de requisitos e modelo de dados, que constituem a base técnica para a implementação futura do sistema.

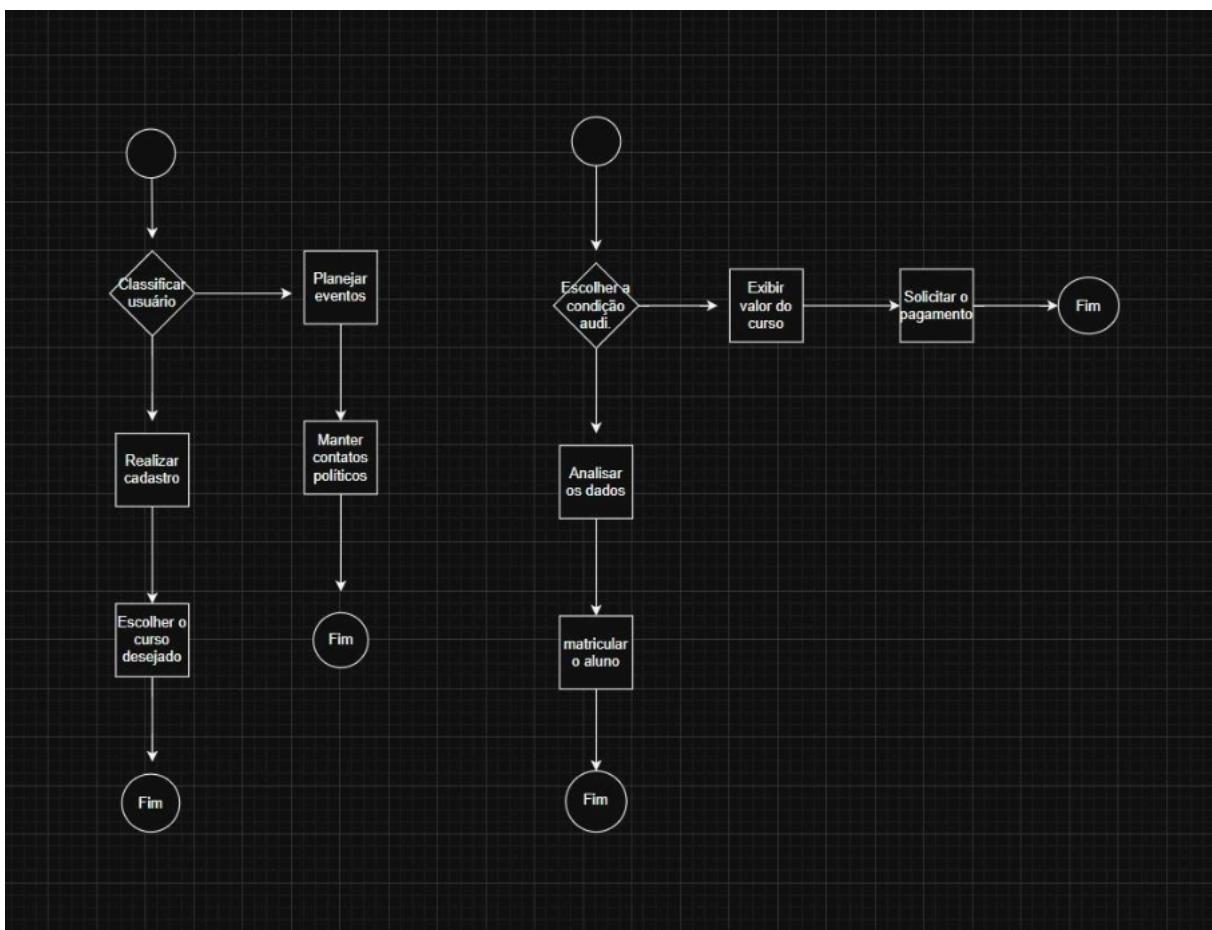
Essas intervenções foram planejadas de forma a garantir não apenas a entrega de um produto tecnológico, mas também o fortalecimento da inclusão digital da instituição. Assim, espera-se que a APADA obtenha maior eficiência em seus processos internos, além de ampliar sua visibilidade e impacto social.

## 6 PRINCIPAIS RESULTADOS

### DIAGRAMA DE ATIVIDADE

O diagrama de atividade é uma ferramenta de modelagem da UML (Unified Modeling Language) utilizada para representar, de forma gráfica, o fluxo de atividades de um processo ou sistema. Seu principal objetivo é descrever a sequência lógica de ações, as decisões envolvidas, as condições de transição e os possíveis caminhos alternativos que um processo pode seguir. Esse tipo de diagrama oferece uma visão geral do comportamento dinâmico de um sistema, sendo amplamente utilizado em projetos de desenvolvimento de software, modelagem de processos de negócio e análise de workflows.

A seguir, apresenta-se a ) elaborada com base nas atividades realizadas no desenvolvimento do sistema para a APADA.

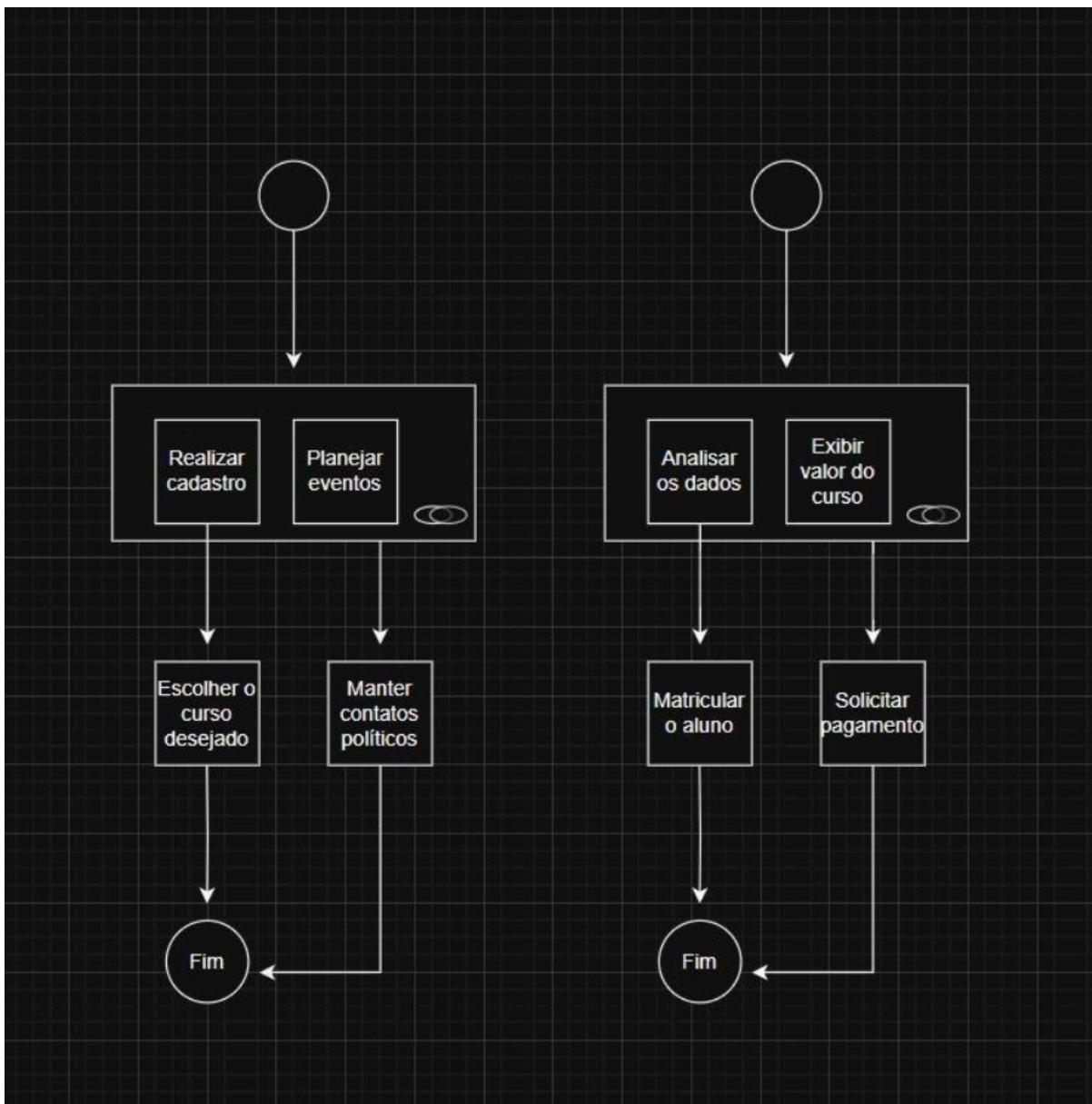


### DIAGRAMA DE MÁQUINA DE ESTADOS

O diagrama de máquina de estados é uma ferramenta de modelagem utilizada para representar os diferentes estados que um sistema ou objeto pode assumir ao longo de seu ciclo de vida. Ele descreve, de forma clara e objetiva, as transições entre os estados em resposta a determinados eventos ou condições. Esse tipo de diagrama

é bastante aplicado em projetos que envolvem controle de fluxo, comportamento de sistemas e definição de regras de transição entre estados.

A seguir, apresenta-se o diagrama de máquina de estados, elaborado com base nos comportamentos e transições identificados durante o desenvolvimento do sistema para a APADA.



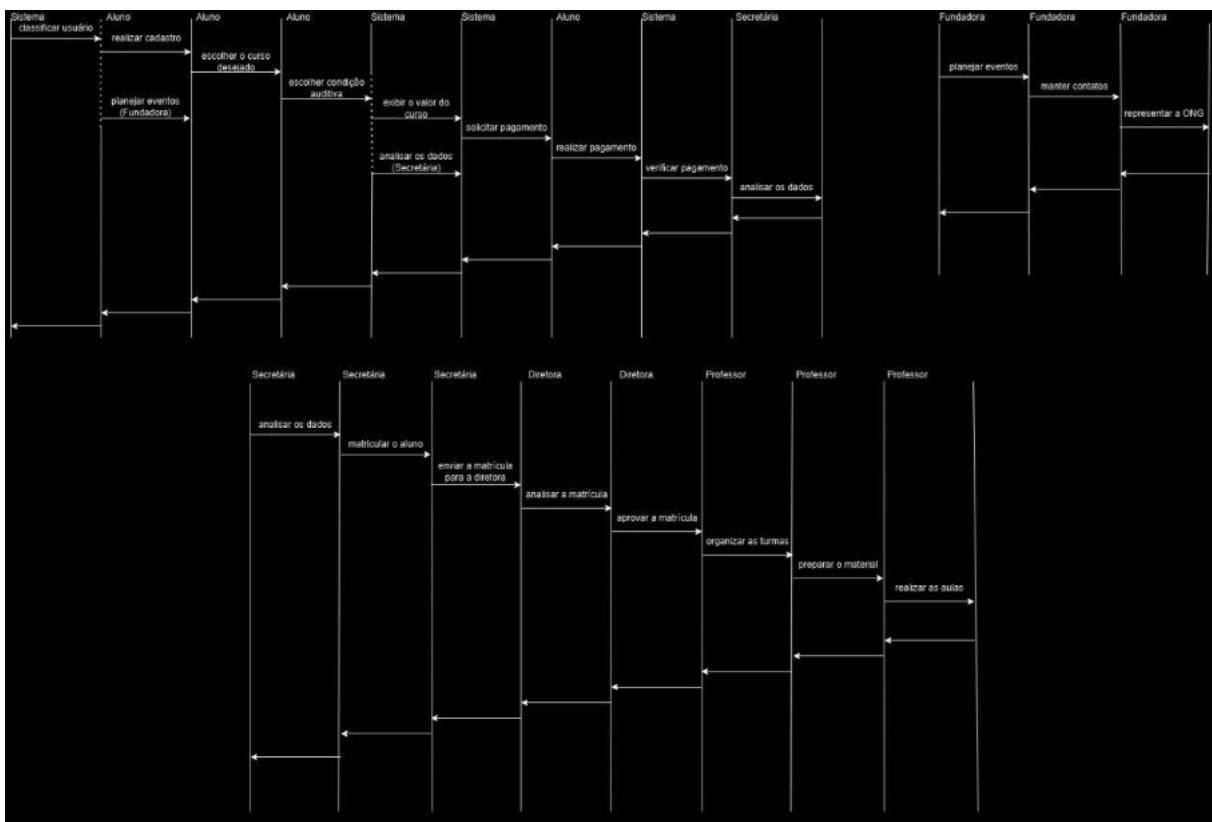
### DIAGRAMA DE SEQUÊNCIA

O diagrama de sequência é um dos principais diagramas da UML (Unified Modeling Language), utilizado para representar a interação entre os objetos de um sistema ao longo do tempo. Ele descreve de forma clara e sequencial como as mensagens são trocadas entre os componentes envolvidos em um determinado processo ou caso de uso. Seu principal objetivo é demonstrar a ordem das interações e o fluxo de

execução, facilitando a compreensão da lógica de funcionamento do sistema e auxiliando na identificação de responsabilidades e dependências entre os objetos.

A seguir, apresenta-se o Diagrama de Sequência, elaborado com base nos comportamentos identificados durante o desenvolvimento do sistema para a APADA.

O diagrama representa a troca de mensagens entre os objetos envolvidos, evidenciando a sequência das operações no processo modelado.



## PORTABILIDADE

A seguir, apresenta-se uma descrição da infraestrutura e recursos disponíveis na APADA, baseada nas informações coletadas durante o levantamento inicial. Esses dados são fundamentais para avaliar a portabilidade e adequação do sistema às condições atuais da organização.

A seguir, apresenta-se a análise da portabilidade do sistema, destacando os Claro! Aqui está o texto sobre portabilidade sem os números:

## Dados da Empresa

**Razão Social:** Associação dos Pais e Amigos do Deficiente Auditivo de Franca – APADA

**Nome Fantasia:** Associação dos Surdos de Franca (ASF)

**CNPJ:** 01.960.221/0001-74

**Endereço:** Rua Tibúrcio Barbosa Sandoval, 266, Vila Nicácio, Franca – SP, CEP 14405-108

**Telefone de contato:** (16) 99184-3482 ou (16) 3701-4762

**E-mail:** [apadafranca@gmail.com](mailto:apadafranca@gmail.com)

## Infraestrutura

Rede Física

Estrutura do prédio com cabeamento estruturado — detalhes internos a confirmar.

Energia elétrica suportada por nobreaks (capacidade a confirmar).

## Rede de Dado

Provável conexão via cabo e Wi-Fi para escritório e salas de atendimento. Velocidade do link e detalhes de roteamento a serem auditados – não disponíveis publicamente.

## Computadores

Utilização de PCs para aulas, administração e atendimento — especificações exatas a verificar (RAM, HD, processador, sistema operacional).

## Periféricos

Impressoras e scanners para material didático e administrativo — quantidades e modelos a confirmar.

Estabilizadores e possível nobreak para continuidade de energia — a identificar.

## Servidores

Ambiente possivelmente na nuvem (e-mail, arquivos) ou servidor local — sem dados públicos detalhados.

Hospeda serviços como e-mail, armazenamento central, backups de material e gestão de cursos.

## Licenças:

Softwares para administração e comunicação (ex.: Office, e-mail, navegador).

Antivírus em estações de trabalho — fornecedor e quantidade de licenças a verificar.

ERP ou sistema de gestão digital — confirmar se existe. Licença do sistema operacional — Windows ou Linux — verificar contratos ou aquisições.

## Recursos Humanos

Público-alvo: funcionários, instrutores e equipe administrativa — estimativa de 10 a 30 pessoas.

Nível de alfabetização digital: básico a intermediário (uso diário de computador e email).

Atividades: gestão de cursos LIBRAS, administração, comunicação e marketing comunitário.

## Configuração Mínima Requerida

**Arquitetura:** x64

**Processador:** Intel Core i3 mínimo ou equivalente

**Memória RAM:** 4 GB (ideal 8 GB)

**Armazenamento:** 256 GB SSD

**Sistema Operacional:** Windows 10 ou Linux Ubuntu LTS

**Rede:** conexão com velocidade mínima de 10 Mbps para uso (streaming, upload de O Modelo Entidade-Relacionamento (MER) apresenta, de forma conceitual, as principais entidades envolvidas no domínio da APADA e as relações entre elas. O MER especifica apenas as entidades e seus atributos essenciais, sem definir ainda os detalhes físicos do banco (tipos e constraints), que são apresentados posteriormente no PM3.vídeos, backup).

**Backup:** área Backup mínima de 500 GB em nuvem ou servidor local.

## Observações Gerais

Necessário fazer inventário completo de hardware e software.

Avaliar atualizações de licenças (antivírus, sistema operacional).

Garantir continuidade de atividades durante a migração — cursos LIBRAS e atendimentos não podem ser interrompidos.

Verificar planos de contingência e backup: frequência de backup, local (nuvem ou servidor), e política de rotação.

aspectos da infraestrutura, recursos humanos e requisitos técnicos da APADA, que influenciam na implementação e operação do sistema.

## Descrição Geral do Sistema e Estrutura do Banco de Dados

O sistema da APADA foi desenvolvido para gerenciar as atividades da instituição, incluindo alunos, cursos, matrículas e pagamentos. O objetivo do banco de dados é organizar e estruturar todas as informações, garantindo integridade, consistência e facilidade de acesso aos dados.

O modelo conceitual (**MER**) descreve as entidades principais e seus relacionamentos, permitindo compreender como alunos se relacionam com cursos, matrículas e pagamentos. A partir do MER, foi gerado o DER, que apresenta as cardinalidades e a estrutura lógica do banco de dados.

O Modelo Físico (**PM3**) detalha como as tabelas serão implementadas no PostgreSQL, especificando tipos de dados, chaves primárias e estrangeiras. O script SQL de criação do banco inclui exemplos de registros para ilustrar o funcionamento do sistema.

As operações de **CRUD (Create, Read, Update, Delete)** permitem a manipulação segura e consistente dos dados, garantindo que informações de alunos, cursos, matrículas e pagamentos possam ser inseridas, consultadas, atualizadas ou removidas conforme necessário. No sistema atual, essas operações são realizadas

por meio de menus interativos no terminal, simulando a interface da aplicação. Embora o sistema ainda não possua uma interface gráfica, os menus do terminal cumprem a função de interação com o usuário, permitindo a execução completa das funcionalidades do banco de dados.

Segue abaixo os artefatos gerados no projeto, incluindo o modelo conceitual (MER), o diagrama lógico (DER), o modelo físico (PM3), scripts SQL de criação e exemplos de CRUD, demonstrando a implementação completa do banco de dados da APADA.

### **MER – Modelo Entidade-Relacionamento**

O Modelo Entidade-Relacionamento (MER) apresenta de forma conceitual a estrutura dos dados utilizados no sistema da APADA. Este modelo descreve as entidades existentes no banco de dados — Aluno, Curso, Matrícula e Pagamento — e os relacionamentos entre elas, representando como as informações são organizadas e conectadas dentro da aplicação.

O MER serve como base para a modelagem lógica e física, permitindo visualizar a organização dos dados antes da implementação das tabelas no banco. A partir dele, é possível compreender como um aluno se relaciona com seus cursos, como as matrículas são registradas e de que maneira os pagamentos são associados a cada inscrição.

A seguir, apresenta-se o diagrama correspondente ao MER, evidenciando suas entidades, atributos principais e cardinalidades.

## ALUNO

### Atributos:

- id\_aluno (PK)
  - nome
  - email
  - data\_nascimento
  - telefone
  - endereco
  - cpf
  - status\_matricula
- 

## CURSO

### Atributos:

- id\_curso (PK)
  - nome
  - valor
  - duracao
- 

## MATRÍCULA

### Atributos:

- id\_matricula (PK)
  - data
  - status
  - id\_aluno (FK → ALUNO)
  - id\_curso (FK → CURSO)
- 

## PAGAMENTO

### Atributos:

- id\_pagamento (PK)

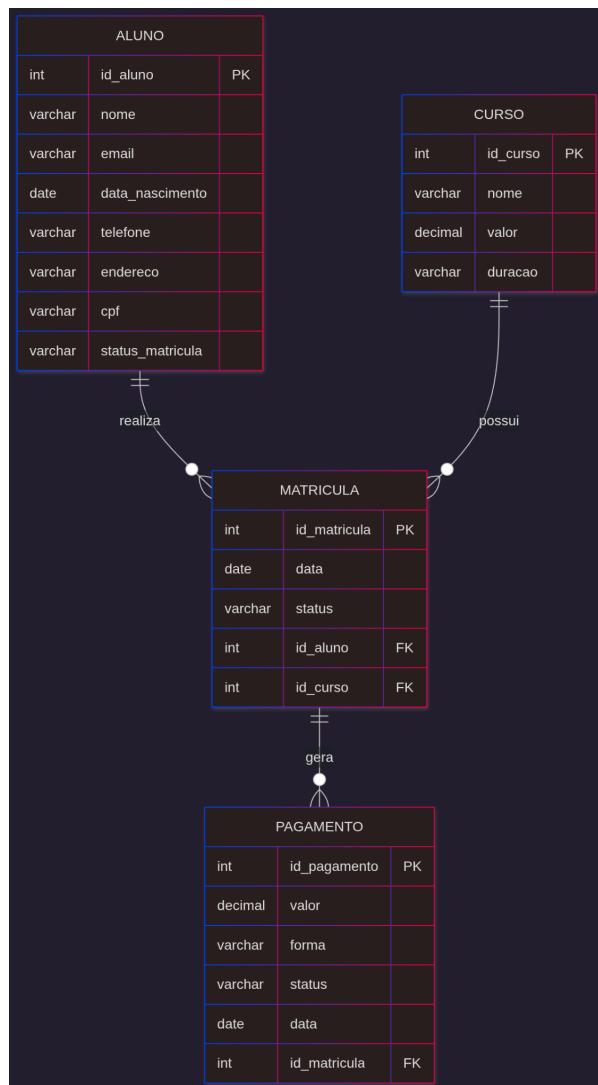
- valor
- forma
- status
- data
- id\_matricula (FK → MATRÍCULA)

### Relações principais

- Um **ALUNO** possui várias **MATRÍCULAS** (1:N)
- Um **CURSO** possui várias **MATRÍCULAS** (1:N)
- Uma **MATRÍCULA** possui um **PAGAMENTO** (1:1)

### DER – Diagrama Entidade-Relacionamento

O Diagrama Entidade-Relacionamento (DER) representa o modelo lógico do banco de dados, mostrando graficamente as entidades e suas cardinalidades. A seguir apresenta-se o DER gerado a partir das tabelas implementadas no PostgreSQL.



### PM3 - Modelo físico do banco de dados

O Modelo Físico (PM3) descreve a implementação do banco de dados da APADA no PostgreSQL, apresentando cada tabela com seus atributos, tipos de dados, chaves primárias e relacionamentos definidos por chaves estrangeiras. Este modelo representa a forma como as informações são efetivamente armazenadas no sistema

**Tabela: ALUNO**

Campo	Tipo	Restrições
<b>id_aluno</b>	<b>SERIAL</b>	<b>PK</b>
<b>nome</b>	<b>VARCHAR(255)</b>	<b>NOT NULL</b>
<b>email</b>	<b>VARCHAR(255)</b>	<b>NOT NULL</b>
<b>data_nascimento</b>	<b>DATE</b>	—
<b>telefone</b>	<b>VARCHAR(20)</b>	—
<b>endereco</b>	<b>VARCHAR(255)</b>	—
<b>cpf</b>	<b>VARCHAR(14)</b>	—
<b>status_matricula</b>	<b>VARCHAR(20)</b>	—

**Tabela: CURSO**

Campo	Tipo	Restrições
<b>id_curso</b>	<b>SERIAL</b>	<b>PK</b>
<b>nome</b>	<b>VARCHAR(255)</b>	<b>NOT NULL</b>
<b>valor</b>	<b>DECIMAL(10,2)</b>	—
<b>duracao</b>	<b>VARCHAR(100)</b>	—

**Tabela: MATRÍCULA**

Campo	Tipo	Restrições
<b>id_matricula</b>	SERIAL	<b>PK</b>
<b>data</b>	DATE	<b>NOT NULL</b>
<b>status</b>	VARCHAR	—
<b>id_aluno</b>	INT	<b>FK → ALUNO(id_aluno)</b>
<b>id_curso</b>	INT	<b>FK → CURSO(id_curso)</b>

**Tabela: PAGAMENTO**

Campo	Tipo	Restrições
<b>id_pagamento</b>	SERIAL	<b>PK</b>
<b>valor</b>	DECIMAL(10,2)	—
<b>forma</b>	VARCHAR	—
<b>status</b>	VARCHAR	—
<b>data</b>	DATE	—
<b>id_matricula</b>	INT	<b>FK → MATRÍCULA(id_matrícula)</b>

## ScripSQL

-- Criação do banco de dados  
CREATE DATABASE apada;

-- Conexão com o banco  
\c apada;

-- Tabela ALUNO

```
CREATE TABLE aluno (
    id_aluno SERIAL PRIMARY KEY,
    nome VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL,
    data_nascimento DATE,
    telefone VARCHAR(20),
    endereco VARCHAR(255),
    cpf VARCHAR(14),
    status_matricula VARCHAR(20)
);
```

-- Tabela CURSO

```
CREATE TABLE curso (
    id_curso SERIAL PRIMARY KEY,
    nome VARCHAR(255) NOT NULL,
    valor DECIMAL(10,2),
    duracao VARCHAR(100)
);
```

-- Tabela MATRÍCULA

```
CREATE TABLE matricula (
    id_matricula SERIAL PRIMARY KEY,
    data DATE NOT NULL,
    status VARCHAR(50),
    id_aluno INT REFERENCES aluno(id_aluno),
    id_curso INT REFERENCES curso(id_curso)
);
```

-- Tabela PAGAMENTO

```
CREATE TABLE pagamento (
    id_pagamento SERIAL PRIMARY KEY,
    valor DECIMAL(10,2),
    forma VARCHAR(50),
    status VARCHAR(50),
    data DATE,
    id_matricula INT REFERENCES matricula(id_matricula)
);
```

-- Exemplos de registros

```
INSERT INTO aluno (nome, email, status_matricula)
```

```
VALUES ('João Silva', 'joao@email.com', 'ativo'),  
       ('Maria Souza', 'maria@email.com', 'ativo');  
  
INSERT INTO curso (nome, valor)  
VALUES ('Libras Básico', 500.00),  
       ('Português para Surdos', 700.00);  
  
INSERT INTO matricula (data, status, id_aluno, id_curso)  
VALUES ('2025-10-03', 'ativa', 1, 1),  
       ('2025-10-03', 'ativa', 2, 2);  
  
INSERT INTO pagamento (valor, forma, status, id_matricula)  
VALUES (500.00, 'Cartão', 'pago', 1),  
       (700.00, 'Boleto', 'pendente', 2);
```

### CRUD – Banco de Dados APADA

O sistema da APADA possui operações de CRUD (Create, Read, Update, Delete) para gerenciar alunos, cursos, matrículas e pagamentos. A seguir, são apresentados exemplos de implementação do CRUD para a tabela ALUNO, demonstrando como os dados são manipulados no banco PostgreSQL.

#### Conexão com o banco de dados

```
public class Conexao {  
    private static final String URL = "jdbc:postgresql://localhost:5432/apada";  
    private static final String USER = "postgres";  
    private static final String PASSWORD = "@isaac2017";  
  
    public static Connection getConnection() {  
        try {  
            Connection conn = DriverManager.getConnection(URL, USER,  
                PASSWORD);  
            System.out.println("Conexão estabelecida com sucesso!");  
            return conn;  
        } catch (SQLException e) {  
            System.err.println("Erro ao conectar: " + e.getMessage());  
            return null;  
        }  
    }  
}
```

### **CRUD de Alunos (AlunoDAO.java)**

```
// CREATE - Inserir aluno
AlunoDAO.inserirAluno("João Silva", "2000-05-10", "joao@email.com",
                      "11999999999", "Rua A, 123", "ativo", 1,
                      "123.456.789-00");

// READ - Listar alunos
AlunoDAO.listarAlunos();

// UPDATE - Atualizar dados de um aluno
AlunoDAO.atualizarAluno(1,          "João        Silva      Atualizado",
                      "novoemail@email.com");

// DELETE - Remover aluno
AlunoDAO.deletarAluno(1);
```

### **AppAluno**

```
package com.apada.aluno;

import java.util.Scanner;

public class AppAluno {
    public static void main(String[] args) {

        Scanner entrada = new Scanner(System.in);
        int opcao;

        do {
            System.out.println("\n==== MENU ALUNO ====");
            System.out.println("1 - Inserir");
            System.out.println("2 - Atualizar");
            System.out.println("3 - Remover");
            System.out.println("4 - Listar");
            System.out.println("0 - Sair");
            System.out.print("Opção: ");
            opcao = entrada.nextInt();

            switch (opcao) {
                case 1:
                    entrada.nextLine(); // limpar buffer
                    System.out.print("Nome: ");
```

```
String nome = entrada.nextLine();
```

```
System.out.print("Data de Nascimento (AAAA-MM-DD): ");
String dataNascimento = entrada.next();
```

```
System.out.print("Email: ");
String email = entrada.next();
```

```
System.out.print("Telefone: ");
String telefone = entrada.next();
```

```
entrada.nextLine(); // limpar buffer
System.out.print("Endereço: ");
String endereco = entrada.nextLine();
```

```
System.out.print("Status da Matrícula: ");
String statusMatricula = entrada.next();
```

```
System.out.print("ID da Turma: ");
int turmald = entrada.nextInt();
```

```
System.out.print("CPF: ");
String cpf = entrada.next();
```

```
AlunoDAO.inserirAluno(nome, dataNascimento, email,
telefone, endereco, statusMatricula, turmald, cpf);
break;
```

case 2:

```
System.out.print("ID do aluno: ");
int idAluno = entrada.nextInt();
```

```
entrada.nextLine(); // limpar buffer
System.out.print("Novo Nome: ");
String novoNome = entrada.nextLine();
```

```
System.out.print("Novo Email: ");
String novoEmail = entrada.next();
```

```
AlunoDAO.atualizarAluno(idAluno, novoNome, novoEmail);
break;
```

case 3:

```

        System.out.print("ID do aluno: ");
        int idRemover = entrada.nextInt();
        AlunoDAO.deletarAluno(idRemover);
        break;

    case 4:
        AlunoDAO.listarAlunos();
        break;

    case 0:
        System.out.println("Saindo...");
        break;

    default:
        System.out.println("Opção inválida!");
        break;
    }

} while (opcao != 0);

entrada.close();
}
}

```

### **CRUD do curso(CursoDAO.java)**

```

package com.apada.curso;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import com.apada.database.Conexao;
import java.util.Scanner;

public class CursoDAO {
    public static void inserirCurso(String nome, double valor, String duracao) {
        String sql = "INSERT INTO curso (nome, valor, duracao) VALUES (?, ?, ?)";
        try (Connection conn = Conexao.getConnection()) {
            PreparedStatement stmt = conn.prepareStatement(sql)) {
                stmt.setString(1, nome);
                stmt.setDouble(2, valor);
                stmt.setString(3, duracao);
            }
        }
    }
}

```

```

        stmt.executeUpdate();
        System.out.println("Curso inserido com sucesso!");
    } catch (SQLException e) {
        System.err.println("Erro ao inserir curso: " + e.getMessage());
    }
}

public static void listarCursos() {
    String sql = "SELECT * FROM curso";
    try (Connection conn = Conexao.getConnection();
         PreparedStatement stmt = conn.prepareStatement(sql);
         ResultSet rs = stmt.executeQuery()) {
        while (rs.next()) {
            System.out.println("ID: " + rs.getInt("id_curso") +
                ", Nome: " + rs.getString("nome") +
                ", Valor: " + rs.getDouble("valor") +
                ", Duração: " + rs.getString("duracao"));
        }
    } catch (SQLException e) {
        System.err.println("Erro ao listar cursos: " + e.getMessage());
    }
}

public static void atualizarCurso(int id, String nome, double valor,
String duracao) {
    String sql = "UPDATE curso SET nome = ?, valor = ?, duracao = ?"
        WHERE id_curso = ?";
    try (Connection conn = Conexao.getConnection();
         PreparedStatement stmt = conn.prepareStatement(sql)) {
        stmt.setString(1, nome);
        stmt.setDouble(2, valor);
        stmt.setString(3, duracao);
        stmt.setInt(4, id);
        stmt.executeUpdate();
        System.out.println("Curso atualizado com sucesso!");
    } catch (SQLException e) {
        System.err.println("Erro ao atualizar curso: " + e.getMessage());
    }
}

public static void deletarCurso(int id) {
    String sql = "DELETE FROM curso WHERE id_curso = ?";
    try (Connection conn = Conexao.getConnection();

```

```
        PreparedStatement stmt = conn.prepareStatement(sql)) {
            stmt.setInt(1, id);
            stmt.executeUpdate();
            System.out.println("Curso deletado com sucesso!");
        } catch (SQLException e) {
            System.err.println("Erro ao deletar curso: " + e.getMessage());
        }
    }
}
```

# AppCurso

```
package com.apada.curso;  
import java.util.Scanner;
```

```
public class AppCurso {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        int opcao;  
        do {  
            System.out.println("\n==== MENU CURSO ====");  
            System.out.println("1 - Inserir");  
            System.out.println("2 - Atualizar");  
            System.out.println("3 - Remover");  
            System.out.println("4 - Listar");  
            System.out.println("0 - Sair");  
            System.out.print("Opção: ");  
            opcao = sc.nextInt();  
            sc.nextLine();  
            switch (opcao) {  
                case 1:  
                    System.out.print("Nome: ");  
                    String nome = sc.nextLine();  
                    System.out.print("Valor: ");  
                    double valor = sc.nextDouble();  
                    sc.nextLine();  
                    System.out.print("Duração: ");  
                    String duracao = sc.nextLine();  
                    com.apada.curso.CursoDAO.inserirCurso(nome, valor,  
duracao);  
                    break;  
                case 2:  
                    System.out.print("ID do curso: ");  
                    int idAtualizar = sc.nextInt();  
            }  
        } while (opcao != 0);  
    }  
}
```

```
        sc.nextLine();
        System.out.print("Novo nome: ");
        String novoNome = sc.nextLine();
        System.out.print("Novo valor: ");
        double novoValor = sc.nextDouble();
        sc.nextLine();
        System.out.print("Nova duração: ");
        String novaDuracao = sc.nextLine();
        com.apada.curso.CursoDAO.atualizarCurso(idAtualizar,
novoNome, novoValor, novaDuracao);
        break;
    case 3:
        System.out.print("ID do curso: ");
        int idDeletar = sc.nextInt();
        com.apada.curso.CursoDAO.deletarCurso(idDeletar);
        break;
    case 4:
        com.apada.curso.CursoDAO.listarCursos();
        break;
    case 0:
        System.out.println("Saindo...");
        break;
    default:
        System.out.println("Opção inválida!");
    }
} while (opcao != 0);
sc.close();
}
```

## CRUD matricula(matriculaDAO.java)

```
package com.apada.Matricula;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import com.apada.database.Conexao;

public class MatriculaDAO {
    public static void inserirMatricula(int idAluno, int idCurso, String data,
String status) {
```

```

String sql = "INSERT INTO matricula (id_aluno, id_curso, data,
status) VALUES (?, ?, ?, ?)";
try (Connection conn = Conexao.getConnection();
PreparedStatement stmt = conn.prepareStatement(sql)) {
stmt.setInt(1, idAluno);
stmt.setInt(2, idCurso);
stmt.setString(3, data);
stmt.setString(4, status);
stmt.executeUpdate();
System.out.println("Matrícula inserida com sucesso!");
} catch (SQLException e) {
System.err.println("Erro ao inserir matrícula: " + e.getMessage());
}
}

public static void listarMatriculas() {
String sql = "SELECT m.id_matricula, a.nome AS aluno, c.nome AS
curso, m.data, m.status " +
"FROM matricula m " +
"JOIN aluno a ON m.id_aluno = a.id_aluno " +
"JOIN curso c ON m.id_curso = c.id_curso";
try (Connection conn = Conexao.getConnection();
PreparedStatement stmt = conn.prepareStatement(sql);
ResultSet rs = stmt.executeQuery()) {
while (rs.next()) {
System.out.println("ID: " + rs.getInt("id_matricula") +
", Aluno: " + rs.getString("aluno") +
", Curso: " + rs.getString("curso") +
", Data: " + rs.getDate("data") +
", Status: " + rs.getString("status"));
}
} catch (SQLException e) {
System.err.println("Erro ao listar matriculas: " + e.getMessage());
}
}

public static void atualizarMatricula(int idMatricula, String status) {
String sql = "UPDATE matricula SET status = ? WHERE
id_matricula = ?";
try (Connection conn = Conexao.getConnection();
PreparedStatement stmt = conn.prepareStatement(sql)) {
stmt.setString(1, status);
stmt.setInt(2, idMatricula);
}
}

```

```

        stmt.executeUpdate();
        System.out.println("Matrícula atualizada com sucesso!");
    } catch (SQLException e) {
        System.err.println("Erro ao atualizar matrícula: " +
e.getMessage());
    }
}

public static void deletarMatricula(int idMatricula) {
    String sql = "DELETE FROM matricula WHERE id_matricula = ?";
    try (Connection conn = Conexao.getConnection()) {
        PreparedStatement stmt = conn.prepareStatement(sql)) {
            stmt.setInt(1, idMatricula);
            stmt.executeUpdate();
            System.out.println("Matrícula deletada com sucesso!");
        } catch (SQLException e) {
            System.err.println("Erro ao deletar matrícula: " +
e.getMessage());
        }
    }
}

```

## AppMatricula

```

package com.apada.Matricula;
import java.util.Scanner;

public class AppMatricula {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int opcao;

        do {
            System.out.println("\n==== MENU MATRÍCULA ===");
            System.out.println("1 - Inserir");
            System.out.println("2 - Atualizar status");
            System.out.println("3 - Remover");
            System.out.println("4 - Listar");
            System.out.println("0 - Sair");
            System.out.print("Opção: ");
            opcao = sc.nextInt();
            sc.nextLine();
        }
    }
}

```

```
switch (opcao) {
    case 1:
        System.out.print("ID do Aluno: ");
        int idAluno = sc.nextInt();
        System.out.print("ID do Curso: ");
        int idCurso = sc.nextInt();
        sc.nextLine();
        System.out.print("Data (AAAA-MM-DD): ");
        String data = sc.nextLine();
        System.out.print("Status: ");
        String status = sc.nextLine();
        MatriculaDAO.inserirMatricula(idAluno, idCurso, data,
status);
        break;

    case 2:
        System.out.print("ID da Matrícula: ");
        int idAtualizar = sc.nextInt();
        sc.nextLine();
        System.out.print("Novo status: ");
        String novoStatus = sc.nextLine();
        MatriculaDAO.atualizarMatricula(idAtualizar, novoStatus);
        break;

    case 3:
        System.out.print("ID da Matrícula: ");
        int idDeletar = sc.nextInt();
        MatriculaDAO.deletarMatricula(idDeletar);
        break;

    case 4:
        MatriculaDAO.listarMatriculas();
        break;

    case 0:
        System.out.println("Saindo...");
        break;

    default:
        System.out.println("Opção inválida!");
}

} while (opcao != 0);
```

```
        sc.close();
    }
}
```

## Menu

```
package com.apada;
```

```
import java.util.Scanner;
import com.apada.aluno.AppAluno;
import com.apada.curso.AppCurso;
import com.apada.Matricula.AppMatricula;
import com.apada.Pagamento.AppPagamento;
```

```
public class App {
```

```
    public static void main(String[] args) {
```

```
        Scanner entrada = new Scanner(System.in);
        int opcao;
```

```
        do {
```

```
            System.out.println("\n===== MENU PRINCIPAL =====");
```

```
            System.out.println("1 - Gerenciar Alunos");
```

```
            System.out.println("2 - Gerenciar Cursos");
```

```
            System.out.println("3 - Gerenciar Matrículas");
```

```
            System.out.println("4 - Gerenciar Pagamentos");
```

```
            System.out.println("0 - Sair");
```

```
            System.out.print("Escolha: ");
```

```
            opcao = entrada.nextInt();
```

```
            switch (opcao) {
```

```
                case 1 -> AppAluno.main(null);
```

```
                case 2 -> AppCurso.main(null);
```

```
                case 3 -> AppMatricula.main(null);
```

```
                case 4 -> AppPagamento.main(null);
```

```
                case 0 -> System.out.println("Encerrando o sistema... ");
```

```
                default -> System.out.println("Opção inválida. Tente novamente.");
```

```
            }
```

```
        } while (opcao != 0);
```

```
    entrada.close();
}
}
```

## 7 CONSIDERAÇÕES FINAIS

Em resumo, este projeto representa uma valiosa aplicação dos conhecimentos adquiridos no curso de Análise e Desenvolvimento de Sistemas, visando oferecer uma solução eficaz e personalizada para as necessidades da APADA. O objetivo geral do projeto foi o desenvolvimento de um sistema para otimizar o processo de cadastro, melhorar a organização interna da associação e fornecer uma base tecnológica robusta para as operações do dia a dia.

Durante a execução do projeto, foram realizados diversos procedimentos fundamentais, começando com a análise de requisitos a partir das informações fornecidas pela APADA. Através de entrevistas e interações contínuas com os membros da associação, foi possível entender as demandas específicas e as dificuldades enfrentadas pela instituição, como a falta de um sistema estruturado para o controle de cadastros e dados relevantes.

Com base nas informações coletadas, foram desenvolvidos artefatos como a EAP e o BPMN, que ajudaram a organizar e estruturar os processos internos da associação, além de identificar as funcionalidades essenciais do sistema. A metodologia de engenharia de software adotada permitiu que o projeto segue uma linha bem estruturada e alinhada com as necessidades da APADA.

A implementação está em andamento e, até o momento, o foco foi na construção da infraestrutura do sistema, garantindo que a base para os processos administrativos seja sólida e eficiente. O uso do PostgreSQL como banco de dados proporciona uma base confiável para o gerenciamento de dados da associação.

No entanto, existem algumas fragilidades e desafios a serem enfrentados, como a adaptação dos funcionários da APADA ao novo sistema e a falta de recursos tecnológicos disponíveis para a utilização completa da plataforma. Além disso, o treinamento dos colaboradores será um aspecto essencial para garantir uma transição suave e o pleno aproveitamento das funcionalidades do sistema. Em termos de possíveis futuras melhorias, o sistema pode ser expandido para incluir funcionalidades adicionais, como relatórios detalhados de estoque, integração com outras plataformas e a automação de processos administrativos. O acompanhamento contínuo das necessidades da APADA permitirá que o sistema evolua de acordo com as novas demandas, garantindo que a solução se mantenha eficaz e alinhada aos objetivos da associação.

O progresso alcançado até o momento demonstra a viabilidade do projeto e sua importância para melhorar a gestão e a organização interna da APADA. Estamos confiantes de que, ao concluir o desenvolvimento e a implementação do sistema, a associação terá uma ferramenta poderosa para otimizar seus processos, fortalecer sua operação e impactar positivamente a comunidade que ela atende.

## **8 CONTRIBUIÇÕES DA UCE PARA A FORMAÇÃO DISCENTE**

O presente relatório tem como objetivo apresentar o desenvolvimento e os resultados obtidos durante a realização do projeto de extensão vinculado à Unidade Curricular de Extensão (UCE). As atividades foram realizadas com foco na análise, modelagem e desenvolvimento de um sistema para atender às necessidades da Associação dos Pais e Amigos do Deficiente Auditivo de Franca – APADA. Ao longo do projeto, foram aplicados os conhecimentos adquiridos durante o curso, utilizando diversas ferramentas e metodologias de engenharia de software. Este documento reúne todas as etapas do processo, incluindo o levantamento de requisitos, elaboração de diagramas, planejamento e demais atividades executadas, evidenciando o aprendizado prático e as contribuições dos alunos envolvidos.

### **REFERÊNCIAS**

Os conteúdos utilizados para a elaboração deste relatório foram baseados nos materiais e orientações fornecidos pelo professor durante as aulas da disciplina, incluindo apresentações, atividades em sala e materiais de apoio disponibilizados ao longo do semestre.