**Documentação do Sistema de Avaliação Institucional**

**Versão 1.0**

**Autores:**

**Revisor: Msc. Marcos Gomes da Silva Rocha**

**Caxias-MA**

**2024**

# ÍNDICE DETALHADO

**Sumário**

[ÍNDICE DETALHADO 1](#_Toc753552664)

[Prefácio 3](#_Toc1139989158)

[1. Introdução 4](#_Toc2103196039)

[1.1 Tema 4](#_Toc636153137)

[1.2 Objetivo do Projeto 4](#_Toc1191245174)

[1.3 Delimitação do Problema 5](#_Toc1351836608)

[1.4 Método de Trabalho 5](#_Toc760662303)

[1.5 Organização do Trabalho: 5](#_Toc666015289)

[2 Descrição Geral do Sistema 6](#_Toc1916465792)

[2.1 Descrição do Problema 6](#_Toc106232811)

[2.2 Principais Envolvidos e suas Características 7](#_Toc100224846)

[2.2.1 Usuários do Sistema 7](#_Toc307070363)

[2.2.2 Desenvolvedores do Sistema 7](#_Toc525355732)

[2.3 Regras de Negócio 8](#_Toc665430890)

[3 Requisitos do Sistema 9](#_Toc1829767538)

[3.1 Requisitos Funcionais 10](#_Toc377610915)

[3.2 Requisitos Não-Funcionais 11](#_Toc1500679904)

[3.3 Protótipo 12](#_Toc2139292796)

[3.3.1 Diagrama de Navegação do Aluno 13](#_Toc1999703616)

[3.3.2 Diagrama de Navegação do Administrador 14](#_Toc1325329998)

[3.4 Métricas e Cronograma\* 15](#_Toc252587122)

[4 Análise e Design 15](#_Toc36819783)

[4.1 Arquitetura do Sistema 15](#_Toc1974087457)

[4.2 Modelo do Domínio \* 16](#_Toc557633067)

[4.3 Diagrama de Classes \* 16](#_Toc460553086)

[4.4 Diagrama de Caso de Uso 17](#_Toc210163114)

[4.5 Diagrama de Entidade e Relacionamento\* 18](#_Toc338431913)

[4.6 Modelo de Dados\* 19](#_Toc287233838)

[4.6.1 Modelo Lógico da Base de Dados \* 19](#_Toc966701982)

[4.6.2 Criação Física do Modelo de Dados \* 19](#_Toc61419905)

[4.6.3 Dicionário de Dados\* 19](#_Toc492828311)

[4.7 Ambiente de Desenvolvimento \* 20](#_Toc1502877783)

[4.8 Sistemas e componentes externos utilizados\* 21](#_Toc385068158)

[5. Implementação \* 21](#_Toc1483489643)

[6 Implantação \* 21](#_Toc397750853)

[6.1 Manual de Implantação 21](#_Toc224636797)

[7 Manual do Usuário\* 22](#_Toc1326375915)

[7.1 Introdução 22](#_Toc854627800)

[7.2 Tela de Login\* 22](#_Toc1998827397)

[7.3 Cadastro de Usuário\* 22](#_Toc1294086398)

[7.4 Recursos Adicionais\* 23](#_Toc322136547)

[8 Conclusões e Considerações Finais \* 23](#_Toc289809384)

[Bibliografia \* 24](#_Toc944634330)

# Prefácio

Este documento visa fornecer um guia detalhado para o desenvolvimento de um Sistema de Avaliação Institucional utilizando a linguagem de programação TypeScript, o framework NextJs. O projeto está sendo desenvolvido pela equipe da fábrica de software do Unifacema e é direcionado a todos os funcionários envolvidos no processo.

Baseado nos princípios da engenharia de software orientada a objetos e utilizando a notação UML (Unified Modeling Language), este documento estrutura e organiza o desenvolvimento do sistema. Embora esta seja a versão 1.0 e ainda esteja em revisão para integrar plenamente a notação UML, ela já oferece uma base sólida para o desenvolvimento do sistema.

Este projeto adota TypeScript e o framework NextJs devido à sua popularidade e robustez no desenvolvimento de interfaces de usuário dinâmicas e responsivas. Ao optar por essas tecnologias, buscamos alinhar-nos com as melhores práticas e tendências do desenvolvimento web atual, garantindo assim a longevidade e escalabilidade do sistema de avaliação institucional.

A metodologia escolhida para guiar o desenvolvimento deste projeto será baseada em práticas ágeis, que se adequam melhor à dinâmica e às necessidades da nossa equipe. Isso nos permitirá adaptar rapidamente às mudanças e garantir entregas contínuas e incrementais, assegurando que o Sistema de Avaliação Institucional atenda aos mais altos padrões de qualidade e funcionalidade.

Acreditamos que este documento servirá como um recurso valioso para guiar a equipe de desenvolvimento e assegurar que o Sistema de Avaliação Institucional atenda aos mais altos padrões de qualidade e funcionalidade.

# 1. Introdução

Este Sistema de Avaliação Institucional é desenvolvido especificamente para atender às necessidades do Grupo UniFacema, uma instituição dedicada à excelência educacional e ao desenvolvimento integral de seus alunos e colaboradores. A missão do Grupo UniFacema é proporcionar um ambiente acadêmico de alta qualidade, incentivando a inovação, a pesquisa e o crescimento pessoal e profissional.

Este documento visa fornecer um roteiro detalhado para o desenvolvimento do Sistema de Avaliação Institucional, utilizando os princípios da engenharia de software orientada a objetos com notação UML (Unified Modeling Language). Ele é destinado a todos os membros da equipe de desenvolvimento, incluindo funcionários, desenvolvedores e gestores de projeto, que estarão envolvidos nas fases de análise, design, implementação, testes e implantação do sistema.

Nossa dedicação principal é para os funcionários do Grupo UniFacema, que trabalham incansavelmente para assegurar que a instituição continue a ser um farol de educação e progresso na comunidade. Este sistema é projetado para facilitar a gestão e a execução das atividades institucionais, garantindo que os serviços prestados sejam eficientes e eficazes.

## Tema

Desenvolvimento de um software para gestão eficiente das atividades do Grupo UniFacema. O sistema de avaliação institucional permitirá que alunos e professores acessem avaliações e feedbacks, enquanto administradores terão dados essenciais para monitorar e melhorar a qualidade dos serviços educacionais. Utilizando princípios de engenharia de software orientada a objetos com notação UML, o sistema foca na melhoria contínua da educação e no desenvolvimento integral da comunidade acadêmica.

## Objetivo do Projeto

O objetivo do projeto do Sistema de Avaliação Institucional é desenvolver um software de gestão que apoie as atividades educacionais da instituição. O software visa otimizar a administração de avaliações, melhorar a coordenação de feedbacks e atividades acadêmicas, e facilitar o acompanhamento do desempenho de alunos e professores através de emissões de relatórios detalhados das avaliações realizadas. Contribuindo para a melhoria contínua da qualidade do ensino e o desenvolvimento integral da comunidade acadêmica.

## Delimitação do Problema

A delimitação do problema reside na necessidade de criar um sistema que atenda às demandas complexas e diversificadas ao emitir as avaliações institucionais realizadas durante todo o semestre, considerando a necessidade de emitir relatórios detalhados dos resultados dessas pesquisas. Com isso, surge a necessidade de criação do sistema que automatiza e trabalha melhor os dados emitidos nessas avaliações.

## Método de Trabalho

O software de Avaliação Institucional é desenvolvido usando TypeScript a framework NextJs, adotando uma abordagem orientada a componentes. O PostgreSQL é usado como sistema de gerenciamento de banco de dados. Essa combinação oferece uma solução completa para o desenvolvimento de aplicações web, garantindo eficiência, flexibilidade e manutenibilidade.

## Organização do Trabalho:

Este documento visa apresentar o roteiro de trabalho para o desenvolvimento do Sistema de Avaliação Institucional, que consiste em um Software criado para o Centro Universitário de Ciência e Tecnologia do Maranhão- UniFacema voltada à gestão educacional da universidade, juntamente com a participação dos alunos e professores.

Iniciamos com o planejamento do tema do projeto, destacando sua importância e proposito na assistência dos gestores educacionais do UniFacema. Logo após, apresentamos os objetivos gerais e específicos do projeto Sistema de Avaliação Institucional, traçando o que é esperado alcançar com sua implementação. Além disso, identificamos os desafios enfrentados no desenvolvimento do projeto, destacando sua área de atuação e sentido.

Exploramos os motivos para a escolha por trás do tema do projeto Sistema de Avaliação Institucional, enfatizando a sua relevância na implementação do sistema na universidade Unifacema. Em seguida, apresentamos a metodologia e abordagem adotadas para o desenvolvimento do projeto, fornecendo uma visão geral do processo de trabalho. Além disso, descrevemos a estrutura do documento, visando apresentar como cada seção está disposta e sua função no contexto geral do projeto.

Por conseguinte, disponibilizamos um glossário para esclarecer os termos técnicos utilizados ao longo do documento. A documentação segue com uma descrição geral do Sistema de Avaliação Institucional, tendo análise do problema, os principais envolvidos e suas características, bem como as regras de negócio envolvidas. Em seguida, abordamos os requisitos do sistema, tanto funcionais quanto não funcionais, e apresentamos um protótipo, métricas e cronograma.

Na seção de análise e design, discutimos a arquitetura do sistema, o modelo do domínio, os diagramas de interação, classes, atividades, estados e componentes, além do modelo de dados. Também descrevemos o ambiente de desenvolvimento e os sistemas e componentes externos utilizados.

A implementação do sistema é abordada em seguida, seguida pelo plano e execução de testes. Finalmente, discutimos a implantação do sistema, fornecendo um diagrama e manual de implantação, seguido pelo manual do usuário. O documento conclui com conclusões e considerações finais sobre o Sistema de Avaliação Institucional.

# Descrição Geral do Sistema

Este capítulo tem como objetivo descrever de forma geral o sistema, o escopo e as principais funções. A descrição geral do sistema deve abrange os itens a seguir.

## Descrição do Problema

O desenvolvimento do Sistema de Avaliação Institucional visa resolver o problema que diz respeito à necessidade de uma gestão eficiente e abrangente dos dados obtidos através das demandas complexas e diversificadas ao emitir as avaliações institucionais realizadas durante todo o semestre, considerando a necessidade de emitir relatórios detalhados dos resultados dessas pesquisas. Sendo essas avaliações institucionais voltadas aos polos de ensino do Grupo UniFacema Caxias, Codó e Presidente Dutra.

O Sistema de Avaliação Institucional tem como impacto significativo devido proporcionar uma melhor gestão dos dados obtidos através dessas avaliações. Tendo em vista desde a criação dos formulários personalizados até mesmo à emissão de relatórios detalhados com gráficos diversificados. Isso possibilitará uma abordagem mais eficaz no tratamento dos dados trabalhados pelos administradores das pesquisas, contribuindo para um melhor entendimento das avaliações respondidas.

Uma boa solução para esse problema é justamente a criação de um sistema que gerencie de forma eficiente as informações obtidas através dessas avaliações realizadas pela universidade, atendendo às necessidades especificas do Grupo Unifacema. Esse sistema deve ser capaz de centralizar e automatizar processos, facilitar o acompanhamento das avaliações, fornecer dados e relatórios para tomada de decisões, e promover a comunicação eficaz entre todos os envolvidos. Além disso, deve ser flexível o suficiente para se adaptar às mudanças e evoluções das atividades do Grupo UniFacema ao longo do tempo.

## Principais Envolvidos e suas Características

### Usuários do Sistema

O Sistema de Avaliação Institucional destina-se a ser utilizado por o Grupo UniFacema voltado à gestão eficiente dos dados das avaliações criadas pelos administradores do sistema. Os principais envolvidos do sistema são os alunos ao responderem as perguntas, e aos administradores que irão trabalhar os dados obtidos dessas respostas.

Para a utilização do sistema sendo administrador é necessário ter essas características:

* Conhecimento no tratamento dos dados através de relatórios;
* Boa interpretação dos gráficos emitidos;
* Conhecimento das normas de formalidade na emissão das perguntas;

### Desenvolvedores do Sistema

Os desenvolvedores do Sistema de Avaliação Institucional podem incluir programadores, analistas de sistemas, designers de interface e outros profissionais envolvidos no processo de desenvolvimento de software. Esses indivíduos são responsáveis por projetar, implementar, testar e manter o sistema, garantindo sua funcionalidade, segurança e desempenho adequados.

As características dos desenvolvedores do sistema incluem:

* Conhecimento em linguagens de programação, frameworks e tecnologias específicas, como TypeScript e NextJS.
* Habilidade para analisar requisitos de sistema e traduzi-los em soluções de software.
* Capacidade de trabalhar em equipe e comunicar-se efetivamente com outros membros da equipe de desenvolvimento.
* Necessidade de atualização constante em relação às melhores práticas e tendências em desenvolvimento de software.

Essas informações são importantes para a distribuição de responsabilidades e a definição de pontos focais de desenvolvimento ao longo do ciclo de vida do projeto.

## Regras de Negócio

**RN001: Acompanhamento dos dados por administradores**

**Descrição**: Os administradores do sistema devem trabalhar os dados obtidos das pesquisas com informações detalhadas, incluindo data de obtenção das respostas, histórico formulários, necessidades específicas e progresso no trabalho dos relatórios.

**Critérios de Implementação:**

* **Cadastro Inicial:**

Os administradores devem ser capazes de se cadastrarem no sistema utilizando seu login padrão disponível para os servidores do Grupo UniFacema envolvidos no sistema.

* **Atualizações dos Formulários:**
  + O sistema deve permitir a atualização regular dos dados dos formulários, incluindo a emissão dos relatórios.
  + Qualquer alteração no formulário deve ser registrada com data e nome social do responsável pela atualização.
* **Acompanhamento dos dados:**
  + O sistema deve permitir o acompanhamento em tempo real das respostas obtidas nos formulários emitidos pelos administradores.
* **Acesso restrito:** 
  + Apenas usuários autorizados (administradores do sistema) devem ter acesso às informações emitidas dos formulários.
  + Mecanismos de autenticação e autorização devem ser implementados para garantir que apenas pessoal autorizado possa visualizar ou modificar os dados.

**RN002: Gestão dos Dados e Agendamento de Formulários**

**Descrição**: O sistema deve incluir funcionalidades para a gestão eficiente dos dados obtidos das avaliações institucionais realizadas com os envolvidos do Grupo UniFacema e para o agendamento dos formulários.

**Critérios de Implementação:**

* **Gestão dos dados:**
  + O sistema deve permitir o registro e monitoramento de todos os dados disponíveis, incluindo edição, remoção e atualização de formulários.
  + Deve haver funcionalidades para a alocação eficiente desses dados conforme as necessidades de cada evento ocorrente na instituição que exija algum tipo de avaliação do mesmo.
* **Agendamento de formulários:**
  + O sistema deve oferecer uma interface intuitiva para o agendamento dos formulários, permitindo que os administradores registrem detalhes como data, hora, local, e objetivos do formulário.

# Requisitosdo Sistema

Os requisitos funcionais são especificações de comportamento ou funções que um sistema deve realizar. Eles geralmente descrevem o que o sistema deve fazer e suas características essenciais. As seções de requisitos funcionais no livro de Pressman² para mais informações

## Requisitos Funcionais

**RF001: Tela de Login**

* + O sistema deve permitir que o aluno ou administrador entre com um login padrão disponibilizado pela instituição.

**RF002: Tela de formulários**

* + O sistema deve permitir que o aluno consiga responder as perguntas disponíveis dos formulários de avaliação institucional.
  + O sistema não deve permitir ao aluno pular as perguntas ou ficar sem respondê-las.
  + O sistema deve permitir que o administrador consiga criar, atualizar, deletar e visualizar os formulários disponíveis no sistema.

**RF003: Tela de gráficos e relatórios**

* + O sistema deve permitir ao administrador visualizar os gráficos detalhados sobre os formulários respondidos.
  + O sistema deve permitir ao administrador gerar relatórios sobre os formulários respondidos.

**RF004: Agendamento de Formulários**

* + O sistema deve permitir o agendamento de formulários com detalhes como data, hora, local de onde será aplicado esse formulário a ser respondido.

## Requisitos Não-Funcionais

**RNF001: Segurança**

* + O sistema deve garantir a confidencialidade e privacidade dos dados das beneficiárias, utilizando criptografia e outras medidas de segurança.

**RNF002: Desempenho**

* + O sistema deve ser capaz de lidar com um grande número de usuários simultaneamente para processar solicitações de forma rápida e eficiente.

**RNF003: Escalabilidade**

* + O sistema deve ser escalável para lidar com um aumento no número usuários, formulários e relatórios.

**RNF004: Usabilidade**

* + O sistema deve ser intuitivo e fácil de usar. A interface amigável, clara e bem projetada garante que os usuários possam interagir efetivamente com o sistema.

**RNF005: Privacidade**

* + O sistema deve estar em conformidade com as regulamentações de proteção de dados, como a Lei Geral de Proteção de Dados (LGPD) no Brasil, garantindo que as informações pessoais dos usuários sejam tratadas com privacidade e confidencialidade adequadas.

**RNF006: Compatibilidade**

* + O sistema deve ser compatível com diferentes dispositivos e navegadores, garantindo acesso a partir de computadores e dispositivos móveis.

**RNF007:Manutenção**

* + O sistema deve estar em constante manutenção e recebendo atualizações, permitindo a correção de bugs, aplicação de patches de segurança e melhorias contínuas.

**RNF008: Backup e Recuperação**

* + O sistema deve incluir mecanismos de backup regular dos dados e procedimentos de recuperação em caso de falhas.

## Protótipo

O protótipo do Sistema de Avaliação Institucional será desenvolvido de forma incremental, evoluindo até a concordância completa dos requisitos com os usuários. A seguir, apresento algumas telas preliminares com suas respectivas descrições e funcionalidades.

**Tela 1: Tela de Login**

**Objetivo**: Permitir que os usuários autorizados (alunos ou administradores) façam login no sistema para acessar o formulário a ser respondido (aluno) ou acessar os dados dos formulários (administradores).

**Descrição das regras:**

* + **Campos:**
    1. **Nome de Usuário**: Campo de texto para inserir o nome de usuário (tamanho máximo de 50 caracteres).
    2. **CPF:** Campo de texto para inserir o Cadastro de Pessoa Física.
  + **Botões:**
    1. **Campus:** Permite a escolha do Campus onde o Aluno ou administrador pertence (seja campus Caxias, Codó ou Presidente Dutra).
    2. **Continuar:** Valida as credenciais e direciona para a Tela de Dashboard.

**Tela 2: Tela de pergunta para alunos**

**Objetivo da tela:** Fornecer a possibilidade de os alunos visualizarem e responderem as perguntas disponíveis no formulário.

**Descrição das regras:**

* + **Componentes:**
    1. **Menu de Navegação:** Visualizar e responder a pergunta disponível
  + **Botões:**
    1. **Próximo:** Botão que permite ao aluno avançar a pergunta já respondida.
    2. **Voltar:** Permite ao aluno voltar à pergunta anterior caso deseje mudar sua resposta.

**Tela 2.1: Tela de Dashboard do administrador**

**Objetivo da tela:** Fornecer ao administrador uma visão geral das funcionalidades e informações dos dados dos formulários.

**Descrição das regras:**

* + **Componentes:**
    1. **Menu de Navegação:** Links para diferentes funcionalidades do sistema.
    2. **Widgets de Informações:** Mostra informações resumidas como número de respostas, gráficos do formulário, novos formulários e novos eventos.

**Tela 3: Tela de Criação dos formulários**

**Objetivo da tela:** Fornecer ao administrador a possibilidade de criar os formulários de avaliação institucional.

**Descrição das regras:**

* + **Componentes:**
    1. **Criação de perguntas:** Links para diferentes modelos de perguntas.
    2. **Widgets de Informações:** Mostra informações das perguntas finais criadas.
  + **Botões:**
    1. **Adicionar pergunta:** Botão que permite ao administrador adicionar pergunta.
    2. **Remover pergunta:** Permite ao administrador remover a pergunta criada.

### Diagrama de Navegação do Aluno

### Diagrama de Navegação do Administrador

## Métricas e Cronograma\*

Para o desenvolvimento utilizou-se o modelo MVC (Models-Views-Controllers) seguindo o diagrama desta versão de acordo com o ano da documentação temos a seguir as informações:

|  |  |
| --- | --- |
| **Semana (Maio - Julho)** | **Atividades** |
| Semana 4 (Maio) | Melhoria na UX-UI |
| Semana 1 (Junho) | Reunião para discutir métricas. |
| Semana 2 (Junho) | Relatórios do Software (Inicio da documentação) |
| Semana 3 (Junho) | Melhoria na UX-UI |
| Semana 4 (Junho) | Testes de performance |
| Semana 1(Julho) | Testes de performance |
| Semana 2 (Julho) | Relatórios do Software (Término da documentação) |
| Semana 3 (Julho) | Reunião para discutir a finalização do projeto |
| Semana 4 (Julho) | Finalização |

# Análise e *Design[[1]](#footnote-1)*

## Arquitetura do Sistema

A infraestrutura do Sistema de Avaliação Institucional UniFacema será baseada em uma abordagem cliente/servidor de três camadas (3-tier) utilizando TypeScript, NextJs e NodeJs, PostgreSQL, e hospedado em um serviço de hospedagem disponibilizado pela instituição. A seguir, detalhamos a arquitetura, a configuração de hardware, rede, software e o dimensionamento mínimo de conexões necessárias para o funcionamento eficiente do sistema.

**Arquitetura de Três Camadas (3-Tier Architecture)**

A arquitetura de três camadas divide o sistema em três camadas principais: a camada de apresentação (front-end), a camada de lógica de negócios (back-end) e a camada de dados (banco de dados).

**Camada de Apresentação (Front-End):**

* + **Responsabilidade:** Interface do usuário.
  + **Tecnologias Utilizadas:** TypeScript**.**
  + **Acesso:** Via navegadores web pelos usuários**.**

**Camada de Lógica de Negócios (Back-End):**

* + **Responsabilidade:** Lógica de negócios e processamento**.**
  + **Tecnologias Utilizadas:** NodeJs e NextJs

**Camada de Dados (Banco de Dados):**

* + **Responsabilidade:** Armazenamento dos dados da aplicação.
  + **Tecnologias Utilizadas:** PostgreSQL**.**
  + **Função:** Persistência e recuperação de dados.

## Modelo do Domínio \*

No PU, o termo “Modelo de Domínio” significa uma representação de classes conceituais do mundo real, não de objetos de software. O termo não significa um conjunto de diagramas que descreve classes de software, a camada de domínio de uma arquitetura de software ou objetos de software com responsabilidades (Larman, 2007)

## Diagrama de Classes \*

Um diagrama de classes descreve os tipos de objetos presentes no sistema e os vários tipos de relacionamentos estáticos existentes entre eles. Os diagramas de classes também mostram as propriedades e as operações de uma classe e as restrições que se aplicam à maneira como os objetos estão conectados. A UML utiliza a palavra característica como um termo geral que cobre as propriedades e operações de uma classe (Folwer, 2005)

## Diagrama de Caso de Uso

O diagrama de casos de uso faz parte do conjunto de diagramas de comportamento propostos pela UML. Esse diagrama representa o usuário e suas interações com o sistema, portanto ele é uma ferramenta útil quando o sistema em desenvolvimento é um sistema de informação, ou seja, quando existem diversas interações desse tipo para serem projetadas (Reinehr, 2020). A figura 1 detalha o diagrama desenvolvido baseado no sistema.

Diagrama

Descrição gerada automaticamente

**Figura 1:** Diagrama de Caso de Uso do Sistema

## Diagrama de Entidade e Relacionamento\*

* **Traits**:
  + **Session e User**: Um usuário pode ter várias sessões, facilitando a autenticação e segurança no sistema.
* **Model**:
  + **Entidades Principais**: User, Course, Activity, Form, Question, QuestionOption, Answer, e QuestionAnswer formam a base do sistema. User é essencial para autenticação, enquanto Course e Activity organizam as atividades. Form, Question, e Answer lidam com a coleta e armazenamento de dados.
* **User**:
  + **Autenticação**: O usuário possui atributos como login e senha, gerenciando várias sessões (Session) para acessar o sistema de forma segura.
* **Relacionamentos**:
  + **User e Session**: Um usuário pode ter várias sessões.
  + **Course e Activity**: Cursos podem ter várias atividades.
  + **Form e Answer**: Formulários coletam respostas de usuários, ligadas a atividades específicas.

Diagrama, Desenho técnico

Descrição gerada automaticamente

## Modelo de Dados\*

### Modelo Lógico da Base de Dados \*

### Criação Física do Modelo de Dados \*

### Dicionário de Dados\*

## Ambiente de Desenvolvimento \*

**Linguagem de Programação:**

* **TypeScript:** foi escolhida devido seu aumento na produtividade devido envolver o desenvolvimento Backend e Front-end, assim como afirma Souza (2022):

**“**TypeScript é uma linguagem de programação fortemente tipada, multi-paradigma, construída como um superset do JavaScript. TypeScript foi escolhida como a linguagem a ser utilizada no Backend pelo fato de ser mais escalável que o JavaScript puro e também por ser a mais utilizadas em Frameworks FrontEnd, o que aumenta a produtividade de desenvolvedores FullStack, já que podem desenvolver uma aplicação completa utilizando somente uma linguagem de programação. Além disso o TypeScript possui uma quantidade enorme de bibliotecas que solucionam dos mais diversos problemas[...].”(Souza, 2022, p 47)

**Framework:**

* **NextJS**: Como Framework optamos pelo NextJS devido sua versatilidade na disponibilização das ferramentas de desenvolvimento assim como afirma Souza (2022):

“NestJS é um framework opinionated utilizado para o desenvolvimento de aplicações executadas no lado do servidor, pode ser utilizado para desenvolver Rest API, MicroServices e também GraphQL. O NestJS vem com padrões de código pré-definidos que devem ser seguidos nahora de desenvolver, além disso, conta com uma grande variedade de ferramentas para auxiliar no desenvolvimento de aplicações BackEnd complexas e escaláveis[...].” (Souza, 2022, p 47)

**Banco de Dados**:

* **PostgreSQL:**

De acordo com o livro PostgreSQL - Guia do Programador “O PostgreSQL é um Sistema Gerenciador de Banco de Dados (SGBD) Relacional, utilizado para armazenar informações de soluções de informática em todas as áreas de negócios existentes, bem como administrar o acesso a estas informações.” Com isso, optamos por utilizar esse SGBD para um melhor armazenamento dos dados dos envolvidos no sistema. Garantindo assim segurança e eficiência nas informações guardadas.

## Sistemas e componentes externos utilizados\*

# Implementação \*

# Implantação \*

## Manual de Implantação

1. **Configuração do Ambiente:**
2. Instale e configure
3. Instale o servidor de banco de dados MariaDB (versão 10.4.28) no dispositivo escolhido para hospedar o banco de dados.
4. **Importação do Banco de Dados:**
5. **Configuração do Servidor Web:**
6. **Acesso ao Sistema:**

# Manual do Usuário\*

Bem-vindo ao sistema de academia! Este manual do usuário fornecerá informações sobre como usar o sistema, incluindo as funcionalidades de login e cadastro de usuários.

## Introdução

O software foi criado para auxiliar os gestores educacionais do UniFacema no mapeamento das informações obtidas através das avaliações institucionais da universidade. Fornece vantagens tanto os alunos da instituição quanto aos gestores, seja nas respostas dos alunos quanto no trabalho no que diz respeito a disponibilidade dos diversos tipos de relatórios emitidos na página das pesquisas.

## Tela de Login\*

•

## Cadastro de Usuário\*

•

## Recursos Adicionais\*

# Conclusões e Considerações Finais \*

**Aplicabilidade dos Resultados Obtidos:**

**Limitações do Sistema:**

**Inovações Realizadas:**

**Possíveis Integrações com Outros Projetos:**

# Bibliografia \*

Sousa, Priscila. (29 de fevereiro de 2024). **Design - O que é, conceito e definição**. Conceito.de. https://conceito.de/design.

Souza, Francisco Moreira Calado; LIMA, Edilson Carlos Silva; CARIDADE, Elda Regina de Sena. CRIANDO SISTEMA ESCALÁVEL DE AGENDAMENTOS UTILIZANDO TYPESCRIPT COM NESTJS NO BACKEND E NEXTJS NO FRONTEND. **Revista Ibero-Americana de Humanidades**, Ciências e Educação, [S. l.], v. 8, n. 12, p. 43–57, 2022. DOI: 10.51891/rease.v8i12.7986. Disponível em: https://periodicorease.pro.br/rease/article/view/7986. Acesso em: 3 set. 2024.

MILANI, André. **PostgreSQL-Guia do Programador**. Novatec Editora, 2008.

REINEHR, S. **Engenharia de requisitos**. Porto Alegre: SAGAH, 2020.

1. [↑](#footnote-ref-1)