**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA**

**ETEC DA ZONA LESTE**

**NOVOTEC Desenvolvimento de Sistemas**

**Andrei Luiz Florêncio Matias**

**Danilo Costa Rodrigues**

**Gabriel da Silva Mendes**

**EASY HOURS: site para contabilização de horas extracurriculares.**

**São Paulo**

**2023**

**Andrei Luiz Florêncio Matias**

**Danilo Costa Rodrigues**

**Gabriel da Silva Mendes**

**EASY HOURS: site para contabilização de horas extracurriculares.**

Trabalho de Conclusão de Curso apresentado ao Curso Técnico em Desenvolvimento de Sistemas da ETEC da Zona Leste, orientado pelo professor Jeferson Roberto de Lima, como requisito parcial para a obtenção do título de técnico em Desenvolvimento de Sistemas.

**São Paulo**

**2023**

**Resumo em português**

O P-TECH é um programa de parcerias entre empresas e escolas globais que busca fornecer aos estudantes acesso às demandas do mercado de trabalho e a rápida adaptação a ele. Na ETEC da Zona Leste, o programa oferece atividades para o desenvolvimento profissional dos alunos, como cursos, criação de apresentações e currículos. Todas essas atividades são contabilizadas para uma meta anual de horas que o aluno deve ter. O objetivo do projeto é criar um sistema web que facilite o gerenciamento das horas por parte do aluno, e otimize o tempo dos coordenadores ao ter que corrigir alguma atividade ou fazer algum relatório. O sistema permitirá que os alunos vejam e entreguem suas atividades do programa P-TECH, além de enviarem reclamações aos coordenadores. Os coordenadores, por sua vez, poderão atribuir tarefas aos alunos, corrigi-las, adicionar horas extras e gerar relatórios de desempenho individual. O sistema será desenvolvido utilizando tecnologias como HTML, CSS, TailwindCSS, JavaScript, React, Next.js, MySQL e Prisma ORM. Durante a fase de planejamento e documentação, foram utilizados diagramas da Linguagem de Modelagem Unificada (UML).

Palavras-chave: Site. Horas. P-TECH.

**Resumo em inglês**

P-TECH is a partnership program between global companies and schools that aims

to provide students with access to the demands of the job market and enable them to

quickly adapt to it. At ETEC Zona Leste, the program offers activities for students’

professional development, such as courses, creating presentations, and resumes. All

these activities are recorded to meet the student's annual hour target. The project's

objective is to create a web system that simplifies hour management for students and

optimizes coordinators' time when correcting activities or generating reports. The

system will allow students to view and submit their P-TECH program activities, as

well as send complaints to coordinators. Coordinators, in turn, will be able to assign

tasks to students, review them, add extra hours, and generate individual performance

reports. The system will be developed using technologies such as HTML, CSS,

TailwindCSS, JavaScript, React, Next.js, MySQL and Prisma ORM. During the

planning and documentation phase, Unified Modeling Language (UML) diagrams

were used.

Keywords: Site. Hours. P-TECH.

## LISTA DE FIGURAS

[**Figura 1 - Exemplo de sintaxe HTML com uma tag com conteúdo** 11](#_Toc138706543)

[**Figura 2 - Exemplo de sintaxe HTML com uma tag vazia** 11](#_Toc138706544)

[**Figura 3 - Exemplo da estrutura básica HTML** 12](#_Toc138706545)

[**Figura 4 - Exemplo de código HTML para a criação de um formulário** 13](#_Toc138706546)

[**Figura 5 - Resultado do código da Figura 4** 13](#_Toc138706547)

[**Figura 6 - Exemplo de atributos HTML** 14](#_Toc138706548)

[**Figura 7 - Sintaxe CSS** 15](#_Toc138706549)

[**Figura 8 - Exemplo de estilização CSS** 17](#_Toc138706550)

[**Figura 9 – Resultado da estilização** 18](#_Toc138706551)

[**Figura 10 - Exemplo de código Javascript** 19](#_Toc138706552)

[**Figura 11 - Exemplo de estilização com Tailwind** 20](#_Toc138706553)

[**Figura 12 - Exemplo de formulário em React** 21](#_Toc138706554)

[**Figura 13 – Diagrama de Caso de Uso da aplicação** 24](#_Toc138706555)

[**Figura 14 - Diagrama de Classe do projeto** 25](#_Toc138706556)

[**Figura 15 - Diagrama de Sequência Login Usuário** 26](#_Toc138706557)

[**Figura 16 - Diagrama de Sequência Criar Aluno** 26](#_Toc138706558)

[**Figura 17 - Diagrama de Sequência Apagar Aluno** 27](#_Toc138706559)

[**Figura 18 - Diagrama de Sequência Atualizar Aluno** 27](#_Toc138706560)

[**Figura 19 - Diagrama de Sequência Visualizar Aluno** 28](#_Toc138706561)

[**Figura 20 - Diagrama de Sequência Visualizar Horas Aluno** 28](#_Toc138706562)

[**Figura 21 - Diagrama de Sequência Criar Atividade** 29](#_Toc138706563)

[**Figura 22 - Diagrama de Sequência Apagar Atividade** 29](#_Toc138706564)

[**Figura 23 - Diagrama de Sequência Atualizar Atividade** 30](#_Toc138706565)

[**Figura 24 - Diagrama de Sequência Visualizar Atividade** 30](#_Toc138706566)

[**Figura 25 - Diagrama de Sequência Visualizar Atividade Aluno** 31](#_Toc138706567)

[**Figura 26 - Diagrama de Sequência Enviar Atividade** 31](#_Toc138706568)

[**Figura 27 - Diagrama de Sequência Corrigir Atividade** 32](#_Toc138706569)

[**Figura 28 - Diagrama de Sequência Gerar Relatório de Desempenho** 32](#_Toc138706570)

[**Figura 29 - Diagrama de Sequência Criar Reclamação** 33](#_Toc138706571)

[**Figura 30 - Diagrama de Sequência Visualizar Reclamação** 33](#_Toc138706572)

[**Figura 31 - Diagrama de Atividade Visualizar reclamação** 34](#_Toc138706573)

[**Figura 32 - Diagrama de Atividade Visualizar horas** 35](#_Toc138706574)

[**Figura 33 - Diagrama de Atividade Visualizar horas por turma** 35](#_Toc138706575)

[**Figura 34 - Diagrama de Atividade Visualizar horas gerais** 36](#_Toc138706576)

[**Figura 35 - Diagrama de Atividade Visualizar atividades** 36](#_Toc138706577)

[**Figura 36 - Diagrama de Atividade Visualizar atividade do aluno** 37](#_Toc138706578)

[**Figura 37 - Diagrama de Atividade Visualizar alunos** 37](#_Toc138706579)

[**Figura 38 - Diagrama de Atividade Responder reclamações** 38](#_Toc138706580)

[**Figura 39 - Diagrama de Atividade Login** 39](#_Toc138706581)

[**Figura 40 - Diagrama de Atividade Gerar relatório de desempenho** 40](#_Toc138706582)

[**Figura 41 - Diagrama de Atividade Excluir atividade** 41](#_Toc138706583)

[**Figura 42 - Diagrama de Atividade Excluir aluno** 42](#_Toc138706584)

[**Figura 43 - Diagrama de Atividade Enviar atividade** 43](#_Toc138706585)

[**Figura 44 - Diagrama de Atividade Criar reclamação** 44](#_Toc138706586)

[**Figura 45 - Diagrama de Atividade Criar atividade** 45](#_Toc138706587)

[**Figura 46 - Diagrama de Atividade Corrigir atividade** 46](#_Toc138706588)

[**Figura 47 - Diagrama de Atividade Cadastrar aluno** 47](#_Toc138706589)

[**Figura 48 - Diagrama de Atividade Visualizar atividade** 48](#_Toc138706590)

[**Figura 49 - Diagrama de Atividade Alterar aluno** 49](#_Toc138706591)

[**Figura 50 - Diagrama de Atividade Acrescentar horas** 50](#_Toc138706592)

## LISTA DE ABREVIATURAS E SIGLAS

Cascading Style Sheets (CSS)

EMR (Electronic Medical Record)

HyperText Markup Language (HTML)

Object Relational Mapper (ORM)

Structured Query Language (SQL)

Linguagem de Modelagem Unificada (UML)

SUMÁRIO

[**1 INTRODUÇÃO 9**](#_Toc138711108)

[**2 DESENVOLVIMENTO 10**](#_Toc138711109)

[**2.1 Referencial teórico 10**](#_Toc138711110)

[**2.2 Tecnologias utilizadas 10**](#_Toc138711111)

[**2.2.1 HTML 11**](#_Toc138711112)

[**2.2.1.1 Sintaxe 11**](#_Toc138711113)

[**2.2.1.3 Atributos 15**](#_Toc138711114)

[**2.2.2 CSS 15**](#_Toc138711115)

[**2.2.2.1 Sintaxe 16**](#_Toc138711116)

[**2.2.2.3 Cascata e especificidade 16**](#_Toc138711117)

[**2.2.3 JavaScript 19**](#_Toc138711118)

[**2.2.3.1 Variáveis 19**](#_Toc138711119)

[**2.2.3.2 Funções 20**](#_Toc138711120)

[**2.2.4 TailwindCSS 20**](#_Toc138711121)

[**2.2.5 React 21**](#_Toc138711122)

[**2.2.6 Next.js 22**](#_Toc138711123)

[**2.2.7 MySQL 22**](#_Toc138711124)

[**2.2.8 Prisma ORM 23**](#_Toc138711125)

[**2.3 Requisitos do sistema e diagramação 23**](#_Toc138711126)

[**2.3.1 Requisitos 23**](#_Toc138711127)

[**2.3.2 Diagrama de Casos de Uso 24**](#_Toc138711128)

[**2.3.3 Diagrama de Classe 25**](#_Toc138711129)

[**2.3.4 Diagramas de Sequência 28**](#_Toc138711130)

[**2.3.5 Diagramas de Atividade 36**](#_Toc138711131)

[**3 CONSIDERAÇÕES FINAIS 53**](#_Toc138711132)

[**REFERÊNCIAS 54**](#_Toc138711133)

# 1 INTRODUÇÃO

O P-TECH é um programa que realiza parcerias entre empresas e escolas do mundo todo, fazendo com que os estudantes tenham acesso a demanda do mercado de trabalho e se adaptem rapidamente. Na ETEC da Zona Leste, o programa fornece atividades para desenvolver o aluno profissionalmente. Os alunos fazem cursos, criam apresentações, currículos e um perfil profissional. Todas essas atividades são contabilizadas para uma meta anual de horas que o aluno deve ter.

As empresas estão sempre tentando melhorar a eficiência de suas operações [...]. Das ferramentas de que os administradores dispõem, as tecnologias e os sistemas de informação estão entre as mais importantes para atingir altos níveis de eficiência e produtividade nas operações (...). (Laudon e Laudon, 2014, p.11).

Como um sistema web pode auxiliar na gestão administrativa das horas extracurriculares?

O Objetivo do projeto é criar um sistema web que facilite o gerenciamento das horas de cada aluno, otimizando o tempo dos coordenadores do programa e dos alunos, para que a entrega e a visualização das atividades não sejam um problema.

O projeto consiste num sistema que permitirá ao aluno ver e entregar atividades relativas ao P-TECH, e enviar reclamações aos coordenadores. Os coordenadores poderão atribuir as atividades do programa aos alunos, corrigir as atividades, acrescentar horas para os alunos e gerar relatórios de desempenho de um aluno. Para isso, serão usadas as seguintes tecnologias: HyperText Markup Language (HTML), Cascading Style Sheets (CSS), TailwindCSS, JavaScript, React, Next.js, MySQL, Prisma ORM. Durante a fase de planejamento e documentação, recorremos também aos diagramas da Linguagem de Modelagem Unificada (UML).

# 2 DESENVOLVIMENTO

Neste capítulo será abordado o processo de criação do projeto. O desenvolvimento consiste na documentação dos diagramas e das tecnologias usadas no projeto, e ilustrações das telas da aplicação.

## 2.1 Referencial teórico

Atualmente, os sistemas de informação contribuem cada vez mais para a solução de problemas organizacionais, tanto no âmbito público como privado.

Segundo Laudon e Laudon (2014), em 2012, os gastos dos Estados Unidos com registros médicos representavam 13% dos 3 trilhões de dólares gastos por ano com saúde.

Para melhorar a qualidade do serviço prestado, foi criado o EMR (*Electronic Medical Record*), um sistema de prontuário eletrônico que contém todos os dados médicos vitais de uma pessoa, com informações pessoais, histórico médico, resultado de exames, diagnósticos, tratamentos, prescrição de medicamentos e os efeitos desses tratamentos. Um sistema eletrônico permite que o médico acesse as informações médicas de um paciente imediatamente, sem precisar procurar essas informações em um papel.

No P-TECH, as horas costumavam ser contabilizadas através de um programa chamado *AirTable*. Nessa plataforma os alunos entregam suas atividades para que as horas sejam contabilizadas em seu perfil. Mas alguns problemas de organização se manifestaram. As horas eram mostradas individualmente, e as horas aprovadas eram mostradas junto com as horas que não foram aprovadas, e os alunos não conseguiam acompanhar seu desempenho e progresso com eficiência.

## 2.2 Tecnologias utilizadas

Neste capítulo serão abordadas as tecnologias que foram usadas no desenvolvimento do projeto.

## 2.2.1 HTML

A sigla HTML significa *Hypertext Markup Language* (Linguagem de marcação de hipertexto), e é responsável pela estrutura de qualquer página web, como textos, links e imagens.

O HTML utiliza de marcação e hipertextos para formar uma estrutura web. De acordo com Flatschart (2011), hipertexto é um documento formado por blocos de informação, ligados por *links*. Para deixar o texto da página semântico, o HTML utiliza marcação para exibir textos e imagens na tela do usuário através de *tags*.

## 2.2.1.1 Sintaxe

Geralmente, uma tag HTML possui abertura e fechamento. A abertura é representada por “<tag>” e o fechamento “</tag>”. Existem tags vazias, que não possuem conteúdo nem fechamento, sendo compostas apenas pela abertura. As Figuras 1 e 2 mostram como são as *tags* simples e vazias no HTML.

**Figura 1 - Exemplo de sintaxe HTML com uma tag com conteúdo**

Texto

Descrição gerada automaticamente com confiança baixa

Fonte: do próprio autor, 2023.

**Figura 2 - Exemplo de sintaxe HTML com uma tag vazia**

Texto, Logotipo, nome da empresa

Descrição gerada automaticamente

Fonte: do próprio autor, 2023.

**2.2.1.2 Estrutura de um arquivo HTML**

Todo arquivo HTML possui uma estrutura base que define as configurações da página e a hierarquia dos elementos. A seguir, a Figura 3 mostra a estrutura básica de um documento HTML.

**Figura 3 - Exemplo da estrutura básica HTML**

Texto

Descrição gerada automaticamente

Fonte: do próprio autor, 2023.

* *DOCTYPE*: É uma declaração que diz ao navegador qual versão do HTML será usada.
* Elemento raiz: o “<html>” é o elemento raiz da página. Ele envolve todos os elementos do site.
* *Head*: é o cabeçalho, onde ficam todas as configurações da página, como a disponibilidade de caracteres, descrição do site, entre outros.
* *Body*: no “<body>” está todo o conteúdo que será visível para o usuário.

As Figuras 4 e 5 mostram como seria um formulário em HTML.

**Figura 4 - Exemplo de código HTML para a criação de um formulário**

**Texto

Descrição gerada automaticamente**

Fonte: do próprio autor, 2023.

**Figura 5 - Resultado do código da Figura 4**

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: do próprio autor, 2023.

* *Header*: é uma tag semântica que representa o cabeçalho da página, com o título e barra de navegação.
* H1: é o título principal do site, ou seja, só pode haver apenas um por página.
* *Main*: essa tag representa o conteúdo principal do site, ou seja, todo o conteúdo que se relaciona com o tópico principal da página.
* *Form*: representa uma seção do documento direcionado ao envio de informações do usuário para um servidor web.
* *Input*: é usado para receber dados do usuário.
* *Button*: Representa um botão clicável, geralmente usados para o envio ou limpeza das informações do usuário.
* *Footer*: Representa o rodapé da página, contendo informações adicionais sobre o autor, direitos autorais, links e contato.

## 2.2.1.3 Atributos

Os elementos HTML possuem atributos, que são conjuntos de propriedade e valor que geralmente alteram o comportamento do elemento ou adicionam conteúdo nele.

“Dentro de cada atributo é indicado um valor que pode ser textual, numérico ou booleano.” (FLATSCHART, 2011, p. 43).”

A Figura 6 mostra sintaxe dos atributos HTML.

**Figura 6 - Exemplo de atributos HTML**

Linha do tempo

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

## 2.2.2 CSS

“CSS ou *Cascading Style Sheet* é uma linguagem voltada para a criação de folhas de estilos em páginas web.” (SCHEIDT, 2015).

Enquanto o HTML definirá o conteúdo da página, o CSS definirá o estilo da mesma, como fontes, cores, animações e localização dos elementos.

## 2.2.2.1 Sintaxe

O CSS é aplicado no HTML através de seletores, que são formas de o CSS selecionar o elemento HTML que será estilizado.

“Uma regra CSS é composta de duas partes: o *seletor* e a *declaração*. A declaração compreende uma *propriedade* e um *valor*.” (SILVA, 2012).

A seguir, a figura 7 mostra um exemplo da sintaxe do CSS.

**Figura 7 - Sintaxe CSS**

Logotipo, nome da empresa

Descrição gerada automaticamente

Fonte: do próprio autor, 2023.

## 2.2.2.3 Cascata e especificidade

O *Cascading* de CSS significa Cascata, e indica que o código sempre será interpretado de cima para baixo, ou seja, se um parágrafo for estilizado mais de uma vez em um arquivo, a estilização que estiver após a outra prevalecerá. Mas, ao usar diferentes tipos de seletores, a cascata pode ser quebrada. Por exemplo, se o parágrafo foi estilizado através do seu atributo de identificação única (ID), essa estilização prevalecerá sobre a maioria das estilizações, exceto estilização *inline*.

Segundo (SCHEIDT, 2015), a ordem seguida pelo CSS é:

1. Estilos *inline*.
2. Seletores por ID.
3. Seletores por classe.
4. Demais seletores.

As Figuras 8 e 9 mostram um exemplo CSS, estilizando o formulário feito anteriormente.

**Figura 8 - Exemplo de estilização CSS**

Texto

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 9 – Resultado da estilização**

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

## 2.2.3 JavaScript

O JavaScript é uma linguagem de programação, geralmente usada para a criação de páginas web.

Enquanto o HTML (Hypertext Markup Language) determina qual será o conteúdo da página e o CSS (Cascading Style Sheet) determina como esse conteúdo será visto, o JavaScript determina como o conteúdo vai interagir com o usuário.

“JavaScript foi criada pela NetScape em parceria com a Sun Microsystems, com a finalidade de fornecer um meio de adicionar interatividade a uma página web.” (SILVA, 2010).

## 2.2.3.1 Variáveis

Variáveis são espaços na memória que armazenam valores, que podem ser textos, números, e outros tipos de dados.

## 2.2.3.2 Funções

Funções são blocos de códigos que serão usados mais de uma vez em seu código. Uma função apenas é executada quando é chamada no código, e permite que o código seja fragmentado em vários pedaços, facilitando o entendimento e a manutenção do sistema.

De acordo com SILVA (2010), uma vez que a função está criada, com nome ou uma variável para representá-la, é possível usá-la em qualquer parte do programa, sem precisar repetir o código da função.

A Figura 8 mostra o exemplo de um código JavaScript com variáveis e funções.

**Figura 10 - Exemplo de código Javascript**

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

## 2.2.4 TailwindCSS

O Tailwind é um *framework* CSS que permite que os elementos HTML sejam estilizados através das suas classes.

Imagine que em um projeto exista uma estilização que será usada várias vezes no código, ao invés de colocar essa estilização toda vez que ela for usada, torna-se mais prático criar uma única classe CSS para aquela estilização, e é isso que o Tailwind faz.

A Figura 11 mostra a estilização do formulário com *TailwindCSS*.

**Figura 11 - Exemplo de estilização com Tailwind**

Texto

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

## 2.2.5 React

O React é uma biblioteca Javascript para a criação de interfaces web. Ele facilita a criação de páginas web, possibilitando a divisão do site em pequenos componentes reutilizáveis.

“As ferramentas da biblioteca foram criadas baseadas nas mais modernas técnicas de desenvolvimento *frontend*. [...] é uma das mais, senão a mais, popular das bibliotecas JavaScript existentes” (Silva, 2021, p.37).

A Figura 12 mostra o exemplo de um simples componente em React.

**Figura 12 - Exemplo de formulário em React**

Texto

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

## 2.2.6 Next.js

O Next é um *framework*React que tem o objetivo de agilizar a criação de sistemas web com React. Para isso, o Next fornece recursos de roteamento, *pre-rendering* e renderização no lado do servidor.

## 2.2.7 MySQL

O SQL (Linguagem de Consulta Estruturada, em português) é uma linguagem de programação usada para gerenciar e manipular bancos de dados relacionais. O SQL permite a criação, modificação e exclusão de estruturas de banco de dados, como tabelas, índices, visões e procedimentos armazenados, bem como a recuperação e manipulação de dados armazenados nessas estruturas.

Por outro lado, de acordo com Milani (2007), o MySQL, inicialmente desenvolvido para atender a aplicações de médio e pequeno porte, é um sistema de gerenciamento de banco de dados relacional. No entanto, ao longo do tempo, ele evoluiu para incluir todas as funcionalidades necessárias para suportar bancos de dados de grande escala. Essa flexibilidade e robustez fizeram com que o MySQL fosse reconhecido por várias entidades como o principal banco de dados de código aberto, capaz de competir com programas similares de código fechado.

## 2.2.8 Prisma ORM

O Prisma é uma ORM (*Object Relational Mapper*), feita com o objetivo de abstrair as entidades do banco de dados para classes JavaScript e TypeScript. Além de deixar o desenvolvimento com banco de dados mais rápido, o Prisma fornece recursos para a edição das entidades do banco de dados online.

## 2.3 Requisitos do sistema e diagramação

Essa é a parte anterior ao processo de desenvolvimento da aplicação. Antes do desenvolvimento é necessário coletar os requisitos dos usuários e representar as suas interações com o sistema em diagramas UML.

## 2.3.1 Requisitos

Antes da criação de qualquer diagrama, modelo ou design, é necessário coletar requisitos funcionais e não funcionais que dirão quais são as funcionalidades da aplicação e seus atores.

Requisitos funcionais do aluno:

* RF01 – O sistema deverá permitir que o aluno faça login;
* RF02 – O aluno poderá enviar atividades em arquivos;
* RF03 – O aluno poderá ver as horas que ele deve ter ao final do ano letivo;
* RF04 – O sistema deverá mostrar as horas que o aluno já possui;
* RF05 – O sistema deverá mostrar todas as atividades que o aluno precisa concluir;
* RF06 – O aluno poderá ver todas as suas atividades concluídas.

Requisitos funcionais dos coordenadores do programa:

* RF01 – O sistema deverá permitir que o coordenador faça login;
* RF02 – O coordenador poderá gerar relatórios de desempenho de um aluno;
* RF03 – O coordenador poderá ver as horas que cada aluno precisa ter ao final do ano letivo;
* RF04 – O coordenador poderá ver quantas horas cada aluno possui;
* RF05 – O sistema permitirá ao coordenador ver as atividades pendentes de cada aluno;
* RF06 – O sistema permitirá ao coordenador ver as atividades que cada aluno concluiu;
* RF07 – O coordenador poderá corrigir a atividade de cada aluno;
* RF08 – O sistema deverá permitir que o coordenador acrescente horas ao aluno;
* RF09 - O sistema deverá permitir que o coordenador remova horas do aluno.

Requisitos não funcionais do sistema:

* RNF01 – O sistema deverá melhorar a visualização das horas e atividades para os alunos e coordenadores do P-TECH;
* RNF02 – As informações dos alunos e coordenadores (senhas e tokens) devem ser protegidas;
* RNF03 – O cadastro dos alunos deverá ser feito pelo coordenador.

## 2.3.2 Diagrama de Casos de Uso

Com os requisitos coletados, é possível adicionar as funcionalidades e os atores do sistema em um Diagrama de Caso de Uso, que vai descrever como cada usuário irá interagir com o sistema.

**Figura 13 – Diagrama de Caso de Uso da aplicação**

Diagrama

Descrição gerada automaticamenteFonte: do próprio autor, 2023

## 2.3.3 Diagrama de Classe

O Diagrama de Classe é utilizado para representar as entidades do sistema, seus atributos e métodos.

**Figura 14 - Diagrama de Classe do projeto**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

## 2.3.4 Diagramas de Sequência

O Diagrama de Sequência permite que as interações entre os objetos do sistema sejam representadas ao longo do tempo.

**Figura 15** **- Diagrama de Sequência Login Usuário**

Gráfico, Gráfico de caixa estreita

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 16 - Diagrama de Sequência Criar Aluno**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 17 - Diagrama de Sequência Apagar Aluno**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 18 - Diagrama de Sequência Atualizar Aluno**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 19 - Diagrama de Sequência Visualizar Aluno**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 20 - Diagrama de Sequência Visualizar Horas Aluno**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 21 - Diagrama de Sequência Criar Atividade**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 22 - Diagrama de Sequência Apagar Atividade**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 23 - Diagrama de Sequência Atualizar Atividade**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 24 - Diagrama de Sequência Visualizar Atividade**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 25 - Diagrama de Sequência Visualizar Atividade Aluno**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 26 - Diagrama de Sequência Enviar Atividade**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 27 - Diagrama de Sequência Corrigir Atividade**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 28 - Diagrama de Sequência Gerar Relatório de Desempenho**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 29 - Diagrama de Sequência Criar Reclamação**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 30 - Diagrama de Sequência Visualizar Reclamação**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

## 2.3.5 Diagramas de Atividade

Com os Diagramas de Atividade, é possível descrever as etapas sequenciais e paralelas do sistema, incluindo atividades, decisões, bifurcações e junções.

**Figura 31 - Diagrama de Atividade Visualizar reclamação**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 32 - Diagrama de Atividade Visualizar horas**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 33 - Diagrama de Atividade Visualizar horas por turma**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 34 - Diagrama de Atividade Visualizar horas gerais**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 35 - Diagrama de Atividade Visualizar atividades**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 36 - Diagrama de Atividade Visualizar atividade do aluno**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 37 - Diagrama de Atividade Visualizar alunos**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 38 - Diagrama de Atividade Responder reclamações**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 39 - Diagrama de Atividade Login**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 40 - Diagrama de Atividade Gerar relatório de desempenho**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 41 - Diagrama de Atividade Excluir atividade**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 42 - Diagrama de Atividade Excluir aluno**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 43 - Diagrama de Atividade Enviar atividade**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 44 - Diagrama de Atividade Criar reclamação**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 45 - Diagrama de Atividade Criar atividade**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 46 - Diagrama de Atividade Corrigir atividade**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 47 - Diagrama de Atividade Cadastrar aluno**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 48 - Diagrama de Atividade Visualizar atividade**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 49 - Diagrama de Atividade Alterar aluno**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

**Figura 50 - Diagrama de Atividade Acrescentar horas**

Diagrama

Descrição gerada automaticamente

Fonte: do próprio autor, 2023

# 3 CONSIDERAÇÕES FINAIS

O sistema a ser desenvolvido tem a intenção de auxiliar e automatizar o gerenciamento das horas extracurriculares do programa P-TECH.

Ainda não se pode comprovar a eficácia do sistema, pois ele ainda não foi construído. Porém, espera-se um resultado satisfatório, tendo em vista as pesquisas realizadas e a importância de um sistema gerenciador dentro de um projeto como o P-TECH, que preza pela organização e eficiência. Alguns dos resultados esperados incluem maior agilidade no gerenciamento das horas extracurriculares, redução de erros, otimização do tempo dos coordenadores e mentores, entre outros aspectos relevantes para o programa.

O sistema web visa melhorar a organização, gerando assim mais engajamento por parte dos alunos; apresentar uma transparência no processo de registro das horas extracurriculares, entre outros aspectos que contribuam para o sucesso do programa.

Por fim, os agradecimentos aos coordenadores, mentores, professores, colegas de classe, e claro, à IBM e todos os envolvidos no programa P-TECH.

## REFERÊNCIAS

FLATSCHART, Fábio. HTML - Embarque imediato. 1. ed. Rio de Janeiro: BRASPORT, 2011. 256 p.

GUEDES, Gilleanes T.A. UML 2 uma abordagem prática. São Paulo: Novatec Editora, 2011. ISBN 978-85-7522-281-2.

LAUDON, Kenneth C.; LAUDON, Jane P. Sistemas de Informação Gerenciais. 11. ed. São Paulo: Pearson Education, 2014. 484 p. ISBN 978-85-430-0585-0.

MILANI, André. MySQL - Guia do Programador. São Paulo: Novatec Editora, 2007. ISBN 85-7522-10-5.

SCHEIDT, Felippe Alex. Fundamentos de CSS: Criando Design para Sistemas Web. 1. ed. Foz do Iguaçu: Outbox Livros Digitais, 2015. ISBN 978-85-66664-04-1.

SILVA, Maurício Samy. Desenvolva aplicações web profissionais com uso dos poderosos recursos de estilização das CSS3. São Paulo: Novatec Editora, 2012. ISBN 978-85-7522-797-8.

SILVA, Maurício Samy. JavaScript - Guia do Programador: Guia completo das funcionalidades de linguagem JavaScript. São Paulo: Novatec Editora, 2010. ISBN 978-85-7522-794-7.

SILVA, Maurício Samy. React Aprenda Praticando: Desenvolva aplicações web reais com uso da biblioteca React e de seus módulos auxiliares. São Paulo: Novatec Editora, 2021. 240 p. ISBN 978-65-86057-39-3.