

# learning-management-system

Repositório: learning-management-system

Nome	RA
Enzo Benedetto Proença	10418579
Gabriel Ken Kazama Geronazzo	10418247
Jéssica Bispo	10410798
Lucas Fernandes	10419400
Lucas P. C. Sarai	10418013
Vitor Alves Pereira	10410862

## Capítulo 1: Introdução

O projeto visa não apenas modernizar a experiência de usuário através de um design consistente e responsivo, mas também expandir o papel do LMS, transformando-o de um repositório de conteúdo em um ecossistema integrado que promove o desenvolvimento do aluno.

Para isso, propõe-se a implementação de funcionalidades de melhoria, como uma ferramenta de busca inteligente, e inovações como um sistema de recomendação de oportunidades acadêmicas e extracurriculares.

## Capítulo 2: Detalhamento do Projeto

### 2.1. Motivação

A motivação central deste projeto nasce da experiência diária como estudante universitário e da observação das dificuldades operacionais causadas pelas ferramentas de gestão de aprendizagem atualmente em uso.

A dificuldade em localizar materiais de estudo, a inconsistência na organização de conteúdo entre diferentes disciplinas e a instabilidade da plataforma resultam em interrupções no acesso ao conteúdo de aprendizagem.

Adicionalmente, identifica-se uma lacuna na centralização de informações relevantes para a jornada do aluno, como oportunidades de estágio, projetos de pesquisa e atividades de extensão, que hoje são divulgadas de forma descentralizada e assimétrica.

O objetivo é, portanto, criar uma solução que resolva esses pontos práticos e organize a experiência acadêmica, tornando-a mais fluida, integrada e personalizada.

### 2.2. Cenário Atual

O cenário de Learning Management Systems é dominado por plataformas robustas, mas que frequentemente carregam legados técnicos e de design. A análise a

seguir combina a perspectiva do usuário final com tendências de mercado.

**Perspectiva de Aluno** Como usuário direto do Moodle, os seguintes pontos são recorrentes:

- **Design e Usabilidade:** A interface é percebida como não alinhada aos padrões visuais atuais, com navegação pouco intuitiva. A falta de um padrão na estruturação do conteúdo por parte dos docentes demanda um esforço adicional do aluno para a localização de materiais em cada nova disciplina. A experiência em dispositivos móveis apresenta limitações, o que dificulta o acesso rápido a informações.
- **Desempenho e Confiabilidade:** Episódios de lentidão, indisponibilidade em períodos de alta demanda (como semanas de prova) e falhas no upload ou download de arquivos são reportados com frequência por alunos e professores. Tais ocorrências podem impactar a continuidade das atividades acadêmicas.

**Análise de Mercado** As observações pessoais são validadas por análises mais amplas do setor de EdTech.

- **Confiabilidade como Fator Crítico:** Relatórios como o da Educause apontam consistentemente que a confiabilidade e a facilidade de uso são os fatores mais importantes para a satisfação de alunos e professores com um LMS (Educause Horizon Report, 2023). Plataformas mais modernas, nativas em nuvem, como o Canvas, utilizam essa estabilidade como um forte diferencial competitivo.
- **A Tendência de “Consumerização”:** Usuários esperam que o software educacional tenha a mesma qualidade de design e experiência de aplicativos de consumo (e.g., Spotify, Netflix). A expectativa por interfaces limpas, personalizadas e intuitivas é uma tendência de mercado que plataformas mais antigas têm dificuldade em acompanhar (Beyond Functionality: How Consumerization of Enterprise UX is Revolutionizing Work).
- **Ecossistemas Integrados:** A direção do mercado aponta para a criação de plataformas que servem como um “hub” central na vida do estudante, integrando-se a outras ferramentas (agendas, ferramentas de anotação) e oferecendo uma visão holística da jornada acadêmica, algo que sistemas tradicionais raramente oferecem de forma nativa.

### 2.3. Estruturação da Demanda

Com base na análise, a demanda para o novo sistema é estruturada em três esferas de atuação:

1. **Problemas (Fundamentais):** Questões centrais que afetam a usabilidade e a funcionalidade do sistema e que devem ser solucionadas na fundação do novo produto.

- **Design Defasado e Inconsistente:** A necessidade de uma interface de usuário (UI) moderna, intuitiva e padronizada.
  - **Responsividade Deficiente:** O requisito de que a plataforma seja totalmente funcional em dispositivos móveis (mobile-first).
  - **Indisponibilidade e Instabilidade:** Embora seja um desafio de arquitetura complexo, o projeto deve ser concebido sobre uma arquitetura que priorize a escalabilidade e a confiabilidade.
2. **Melhorias (Incrementais):** Funcionalidades que aprimoram processos existentes, gerando ganhos de eficiência para o usuário.
- **Ferramenta de Busca Difusa (Fuzzy Search):** Implementação de um sistema de busca avançado que permita encontrar termos dentro do conteúdo de documentos (PDFs, slides), mesmo com pequenas variações ou erros de digitação.
  - **Integração com Ecossistema de Ferramentas:** Capacidade de integrar-se com ferramentas externas populares entre estudantes, como Google Calendar (para sincronizar prazos) e Notion.
3. **Inovações (Diferenciais):** Novas funcionalidades que expandem o propósito do LMS e criam um valor único para o produto.
- **Sistema de Recomendação de Oportunidades:**
    - **Perfil de Interesses do Aluno:** Área para o aluno cadastrar suas habilidades e interesses através de tags.
    - **Cadastro de Oportunidades:** Formulário para cadastro de oportunidades, também utilizando tags.
    - **Algoritmo de Recomendação (Content-Based):** Implementação inicial do motor que cruza as tags do perfil do aluno com as das oportunidades para gerar um feed personalizado.

## 2.4. Funcionalidades Fora do Escopo do MVP (Próximos Passos)

As seguintes funcionalidades, embora importantes, serão planejadas para fases futuras do projeto:

- **Ferramenta de Busca Difusa (Fuzzy Search):** Sistema de busca avançado para encontrar termos dentro do conteúdo dos documentos.
- **Integração** com APIs de ferramentas externas (Google Calendar, Notion).

## Capítulo 3: Requisitos do Sistema

### 3.1. Requisitos Funcionais

Módulo: Gestão de Usuários

ID	Requisito Funcional	Prioridade
USU-RF1	O sistema deve permitir criar, editar e remover usuários	Alta
USU-RF2	O sistema deve permitir listar usuários com paginação e filtros básicos	Média
USU-RF3	O sistema deve permitir vincular usuários a disciplinas (roles: aluno/professor)	Média

#### **Módulo: Autenticação e Autorização**

ID	Requisito Funcional	Prioridade
AUTH-RF1	O sistema deve autenticar usuários via e-mail mackenzista e senha	Alta
AUTH-RF2	O sistema deve permitir logout	Alta
AUTH-RF3	O sistema deve aplicar controle de acesso baseado em roles (middleware na API)	Baixo

#### **Módulo: Gestão de Disciplinas**

ID	Requisito Funcional	Prioridade
DISC-RF1	O sistema deve permitir criar, editar e remover disciplinas	Alta
DISC-RF2	O sistema deve permitir listar disciplinas com paginação	Alta
DISC-RF3	O sistema deve permitir upload/download de documentos	Média
DISC-RF4	O sistema deve permitir acessar os detalhes de uma disciplina	Baixo

#### **Módulo: Armazenamento de Arquivos**

ID	Requisito Funcional	Prioridade
FILE-RF1	O sistema deve permitir upload de arquivos (PDF, slides, vídeo)	Alta
FILE-RF2	O sistema deve permitir download de arquivos	Média
FILE-RF3	O sistema deve permitir listar arquivos com metadados	Baixo

ID	Requisito Funcional	Prioridade
FILE-RF4	O sistema deve permitir excluir arquivos	Baixo

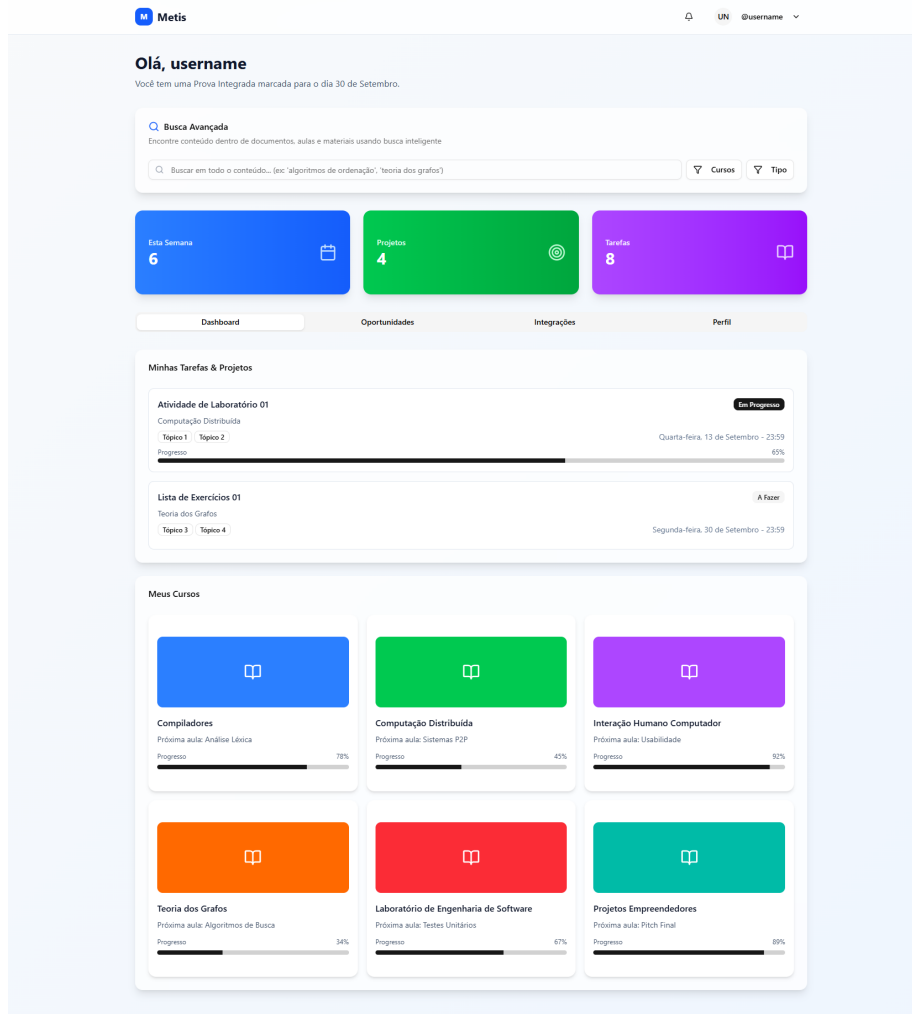
### Módulo: Oportunidades

ID	Requisito Funcional	Prioridade
OPP-RF1	O sistema deve permitir cadastrar oportunidades (estágio, complementar, extensão)	Alta
OPP-RF2	O sistema deve permitir listar e filtrar oportunidades	Alta
OPP-RF4	O sistema deve suportar um motor de recomendações (content-based filtering)	Média

### 3.2. Requisitos Não Funcionais

ID	Categoria	Requisito Não Funcional	Critério / Métrica	Prioridade
SEC-NF2	Segurança	Armazenamento seguro de dados sensíveis	Senhas com hashing seguro e salting (bcrypt ou Argon2)	Alta
SEC-NF3	Segurança	Controle de acesso	RBAC (Controle de Acesso Baseado em Função) aplicado nas APIs e aplicação web	Média
PERF-NF1	Desempenho	Latência de API	95% das respostas < 200 ms sob cenário de carga	Alta
PERF-NF2	Desempenho	Upload de arquivos	Suportar uploads até 10 MB no MVP	Média
MON-NF1	Monitoramento	Logs	Logs de autenticação/erros com retenção de 30 dias	Baixa
USAB-NF1	Usabilidade e Acessibilidade	Responsividade e compatibilidade	Mobile-first; suportar navegadores modernos	Alta
USAB-NF2	Usabilidade e Acessibilidade	Acessibilidade	Avaliação heurística e correções críticas; meta: WCAG 2.1 AA em iterações futuras	Baixa

## Capítulo 4: Protótipo da Interface



Para acessar o protótipo clique [aqui](#).

Para essa aplicação, decidimos utilizar uma interface interativa invés do wireframe para demonstrar como os componentes e outros elementos podem ser utilizados pelo usuário e, acima de tudo, como essa nova proposta contrasta com os outros sistemas de aprendizagem disponibilizados no mercado. Com isso, pretendemos demonstrar logo no início os nossos objetivos com esse projeto e quais funções farão a diferença no dia a dia dos usuários.

## Capítulo 5: Modelagem do Sistema

### Casos de Uso: Visão Geral

Aluno:

- Fazer sign-in e sign-up (e-mail mackenzista e senha)
- Acessar disciplinas
- Cadastrar oportunidade (estágio, complementar e extensão)
- Feed de recomendações de oportunidades

(Opcional)

- Busca difusa (fuzzy search) em documentos
- Baixar documentos

Professor:

- Fazer upload de documentos (PDFs, slides, vídeos), com metadados (título, descrição, tags)
- Cadastrar oportunidade (estágio, complementar e extensão)

Administrador:

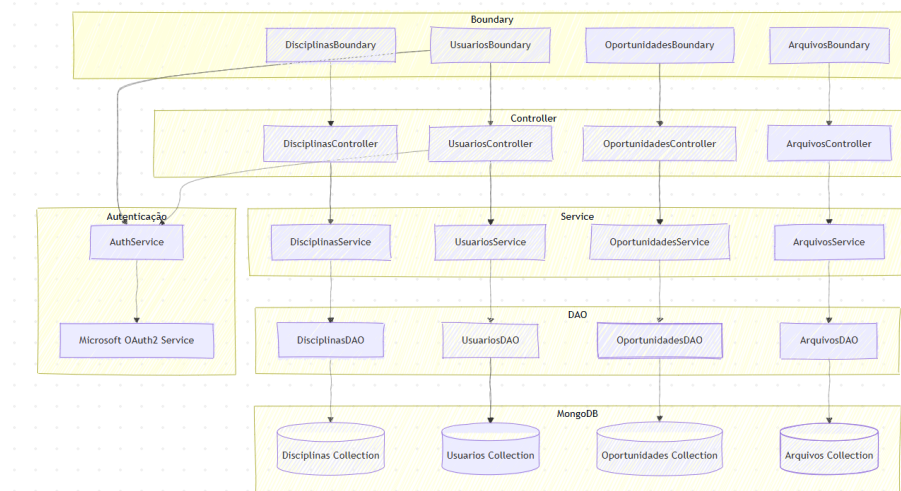
- Gerenciar usuários (criar, editar, remover)
- Gerenciar disciplinas (criar, editar, remover)
- Vincular professores e alunos às disciplinas

### Módulos

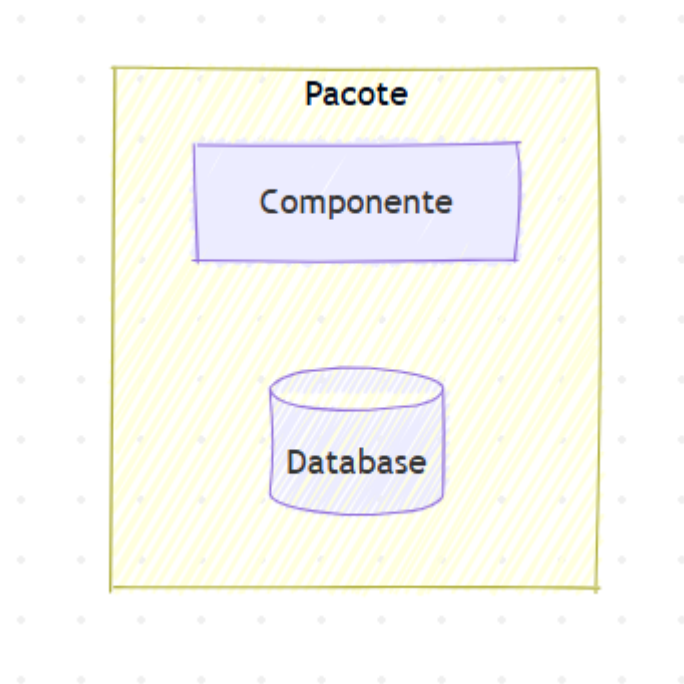
- Autenticação e Autorização
- Usuários
- Arquivos
- Disciplinas
- Oportunidades

## Capítulo 6: Arquitetura do Sistema

### Visão Geral



### Legenda:





## Tecnologias

- **Frontend:** Next.js 15 e Tailwind CSS
- **Backend:** Java 21 com Spring
- **Banco de Dados:** MongoDB e MongoDB GridFS
- **Cloud:** AWS + Docker
- **CI/CD:** GitHub Actions

## Pipelines

### Frontend

`name: Frontend CI/CD`

`on:`

`push:`

`branches:`

`- master`

`paths:`

`- "web/**"`

`pull_request:`

`branches:`

`- master`

`paths:`

`- "web/**"`

`env:`

`AWS_REGION: ${ secrets.AWS_REGION }`

`ECR_REGISTRY: ${ secrets.AWS_ACCOUNT_ID }.dkr.ecr.${ secrets.AWS_REGION }.amazonaws.com`

`ECR_REPOSITORY: ${ secrets.ECR_REPOSITORY_WEB }`

`ECS_CLUSTER: ${ secrets.ECS_CLUSTER_WEB }`

`ECS_SERVICE: ${ secrets.ECS_SERVICE_WEB }`

`ECS_TASK_DEFINITION_PATH: web/ecs-task-definition.json`

`CONTAINER_NAME: web-app`

`jobs:`

`build:`

`runs-on: ubuntu-latest`

`permissions:`

`contents: read`

`id-token: write`

`steps:`

`- name: Checkout Code`

`uses: actions/checkout@v4`

```

- name: Setup Node.js 20
  uses: actions/setup-node@v4
  with:
    node-version: "20"
    cache: "npm"
    cache-dependency-path: web/package-lock.json

- name: Install Dependencies
  run: npm ci
  working-directory: ./web

- name: Build Next.js Project
  run: npm run build
  working-directory: ./web

- name: Configure AWS Credentials
  uses: aws-actions/configure-aws-credentials@v4
  with:
    aws-access-key-id: ${ secrets.AWS_ACCESS_KEY_ID }
    aws-secret-access-key: ${ secrets.AWS_SECRET_ACCESS_KEY }
    aws-region: ${ env.AWS_REGION }

- name: Login to Amazon ECR
  id: login-ecr
  uses: aws-actions/amazon-ecr-login@v2

- name: Build and Push Docker Image to ECR
  uses: docker/build-push-action@v5
  with:
    context: ./web
    push: true
    tags: ${ env.ECR_REGISTRY }/${ env.ECR_REPOSITORY }:${ env.github.sha }
    cache-from: type=gha
    cache-to: type=gha,mode=max

deploy:
  needs: build
  runs-on: ubuntu-latest
  environment: Production
  if: github.ref == 'refs/heads/master'

steps:
- name: Checkout Code
  uses: actions/checkout@v4

```

```

- name: Configure AWS Credentials
  uses: aws-actions/configure-aws-credentials@v4
  with:
    aws-access-key-id: ${ secrets.AWS_ACCESS_KEY_ID }
    aws-secret-access-key: ${ secrets.AWS_SECRET_ACCESS_KEY }
    aws-region: ${ env.AWS_REGION }

- name: Render ECS Task Definition
  id: render-task
  uses: aws-actions/amazon-ecs-render-task-definition@v1
  with:
    task-definition: ${ env.ECS_TASK_DEFINITION_PATH }
    container-name: ${ env.CONTAINER_NAME }
    image: ${ env.ECR_REGISTRY }/${ env.ECR_REPOSITORY }:${ github.sha }

- name: Deploy to ECS Service
  uses: aws-actions/amazon-ecs-deploy-task-definition@v1
  with:
    task-definition: ${ steps.render-task.outputs.task-definition }
    service: ${ env.ECS_SERVICE }
    cluster: ${ env.ECS_CLUSTER }
    wait-for-service-stability: true

```

## Backend

name: Backend CI/CD

on:

```

push:
  branches:
    - master
  paths:
    - "backend/**"
pull_request:
  branches:
    - master
  paths:
    - "backend/**"

```

env:

```

AWS_REGION: ${ secrets.AWS_REGION }
ECR_REGISTRY: ${ secrets.AWS_ACCOUNT_ID }.dkr.ecr.${ secrets.AWS_REGION }.amazonaws.com
ECR_REPOSITORY: ${ secrets.ECR_REPOSITORY_BACKEND }

ECS_CLUSTER: ${ secrets.ECS_CLUSTER_BACKEND }
ECS_SERVICE: ${ secrets.ECS_SERVICE_BACKEND }

```

```
ECS_TASK_DEFINITION_PATH: backend/ecs-task-definition.json
CONTAINER_NAME: backend-app
```

```
jobs:
  build_and_push:
    runs-on: ubuntu-latest

    permissions:
      contents: read
      id-token: write

    steps:
      - name: Checkout Code
        uses: actions/checkout@v4

      - name: Setup Java 21
        uses: actions/setup-java@v4
        with:
          distribution: "zulu"
          java-version: "21"

      - name: Configure Maven Cache
        uses: actions/cache@v4
        with:
          path: ~/.m2
          key: ${{ runner.os }}-maven-${{ hashFiles('**/pom.xml') }}
          restore-keys: |
            ${{ runner.os }}-maven-

      - name: Build Spring Boot Project (Maven)
        run: mvn -B package --file backend/pom.xml -DskipTests

      - name: Configure AWS Credentials
        uses: aws-actions/configure-aws-credentials@v4
        with:
          aws-access-key-id: ${{ secrets.AWS_ACCESS_KEY_ID }}
          aws-secret-access-key: ${{ secrets.AWS_SECRET_ACCESS_KEY }}
          aws-region: ${{ env.AWS_REGION }}

      - name: Login to Amazon ECR
        id: login-ecr
        uses: aws-actions/amazon-ecr-login@v2

      - name: Build and Push Docker Image to ECR
        uses: docker/build-push-action@v5
        with:
```

```

    context: ./backend
    push: true
    tags: ${ env.ECR_REGISTRY }}/${ env.ECR_REPOSITORY }}:${ github.sha }}
    cache-from: type=gha
    cache-to: type=gha,mode=max

deploy:
  needs: build_and_push
  runs-on: ubuntu-latest
  environment: Production
  if: github.ref == 'refs/heads/master'

  steps:
    - name: Checkout Code
      uses: actions/checkout@v4

    - name: Configure AWS Credentials
      uses: aws-actions/configure-aws-credentials@v4
      with:
        aws-access-key-id: ${ secrets.AWS_ACCESS_KEY_ID }}
        aws-secret-access-key: ${ secrets.AWS_SECRET_ACCESS_KEY }}
        aws-region: ${ env.AWS_REGION }}

    - name: Render ECS Task Definition
      id: render-task
      uses: aws-actions/amazon-ecs-render-task-definition@v1
      with:
        task-definition: ${ env.ECS_TASK_DEFINITION_PATH }}
        container-name: ${ env.CONTAINER_NAME }}
        image: ${ env.ECR_REGISTRY }}/${ env.ECR_REPOSITORY }}:${ github.sha }}

    - name: Deploy to ECS Service
      uses: aws-actions/amazon-ecs-deploy-task-definition@v1
      with:
        task-definition: ${ steps.render-task.outputs.task-definition }}
        service: ${ env.ECS_SERVICE }}
        cluster: ${ env.ECS_CLUSTER }}
        wait-for-service-stability: true

```

## Capítulo 7: Desenvolvimento

### Processo de Construção

**Etapa 1: Modularização e separação de responsabilidades** Como módulos principais do projeto, resolvemos separá-lo em 5 principais:

- Autenticação e Autorização: Responsável pelo controle de acesso dos

usuários e pelo controle de permissões, os quais foram divididos entre Aluno, Professor e Administrador. Este módulo ficou sob a responsabilidade de Lucas Fernandes.

- **Usuários:** Responsável pelas operações básicas do CRUD (*Create, Read, Update, Delete*) com a entidade de usuários. Este módulo ficou sob a responsabilidade de Enzo.
- **Arquivos:** Responsável pelas operações básicas do CRUD com a entidade de arquivos, além dos serviços de hospedagem e acesso a eles em ambiente de nuvem. Este módulo ficou sob a responsabilidade de Jéssica.
- **Disciplinas:** Responsável pelas operações básicas do CRUD com as entidades de disciplinas e tarefas. Este módulo ficou sob a responsabilidade de Lucas Sarai.
- **Oportunidades:** Responsável pelas operações básicas do CRUD com entidades associadas ao sistema de recomendação de oportunidades, além de trabalhar no próprio motor de recomendação. Este módulo ficou sob a responsabilidade de Gabriel.

Para desenvolvê-los, cada um desses integrantes trabalharia no lado do *back-end* da aplicação. Em paralelo, Vitor trabalharia no *front-end* de todos os módulos.

**Etapla 2: Escolha de tecnologias** Para implementar o projeto, as seguintes tecnologias foram utilizadas:

- **Frontend:** Next.js 15, por permitir separar as telas em componentes, o que garante melhor manutenibilidade e maior facilidade para integrar com o *back-end*, uma vez que os componentes poderiam ser associados aos módulos (ex: um componente para exibir os detalhes de uma tarefa pode ser associado aos dados extraídos de uma entidade tarefa). O Tailwind foi utilizado para auxiliar no design, por permitir aplicar estilos CSS apenas atribuindo classes de estilo nos objetos. Por fim, o TypeScript foi aplicado para garantir a atribuição de tipo aos dados, o que facilita o tratamento de erros.
- **Backend:** Java 21 com Spring: O Java pode ser executado em qualquer arquitetura de máquina caso essa possua uma JVM instalada, permitindo executar a aplicação em ambientes diversos. O Java de forma geral, mas principalmente em sua versão 21 e com o framework Spring são muito utilizadas no mercado e já de conhecimento dos integrantes, o que facilitou sua escolha.
- **Banco de Dados:** MongoDB e MongoDB GridFS são banco de dados não relacionais que garantem maior escalabilidade horizontal e performance em relação aos banco de dados relacionais, além de maior flexibilidade de armazenamento dos dados, dado que não precisam ter uma estrutura fixa. Foram aplicados no projeto, pois várias informações poderiam ser associadas às entidades e removidas com maior facilidade conforme o projeto ia se desenvolvendo. Maior performance acaba sendo necessário

no motor recomendação, uma vez que o algoritmo não pode demorar para exibir as oportunidades ao aluno, seguindo os requisitos não funcionais especificados.

- **Cloud:** Para hospedar e disponibilizar a aplicação ao público, pretendemos utilizar a Azure, por ser gratuita e não conseguirmos acesso à AWS, como inicialmente especificado. Para containerizar a aplicação e facilitar a sua gestão nesse ambiente de nuvem e no de desenvolvimento, foi utilizado o Docker.
- **CI/CD:** Por fim, pipelines para seguir com os princípios de CI/CD foram implementadas pelo *GitHub Actions*. Para subir as alterações na master, devem ser criadas pull requests e, nesse processo, as pipelines executam testes tanto no *back-end* como no *front-end*, de forma a manter uma integração contínua e um *deploy* constante e seguro.

**Etapa 3: Início da implementação e revisão da documentação** A primeira tecnologia aplicada foi a criação de arquivos '*docker compose*', os quais permitem gerenciar as imagens usadas no projeto (java, node e mongo DB), além de conter as configurações e credenciais para subir a aplicação na Azure. Em seguida, foram criadas as pipelines. Após, as implementações foram iniciadas no *back-end* e, em paralelo, também iniciou-se a montagem das telas. Conforme a implementação ia avançando, a documentação era revisada e, em especial, os diagramas, buscando adequá-los às melhorias e ajustes feitos durante o desenvolvimento.

**Etapa 4: O trabalho até aqui** Todos os módulos já possuem uma implementação base, com entidades, serviços e testes de integração. Além disso, os protótipos de telas já foram montados e disponibilizados para visualização. Sendo assim, falta realizar a integração de ambos para que o sistema passe a ser funcional. Além disso, precisa-se disponibilizar a aplicação completa na Azure.

## Capítulo 8: Resultados (Parcial)

O objetivo deste capítulo é demonstrar como a interface implementa a proposta de valor do projeto, focando na **Usabilidade Superior** e no conceito de **Hub da Vida Acadêmica**. O uso de tecnologias modernas no *frontend* (Next.js e Tailwind CSS) e um *Design System* padronizado garantem a consistência e a responsividade da aplicação.

### Página Inicial

#### 8.1. Visão Geral da Arquitetura Frontend

A arquitetura de *frontend* foi construída para superar as limitações de usabilidade e responsividade dos sistemas legados.

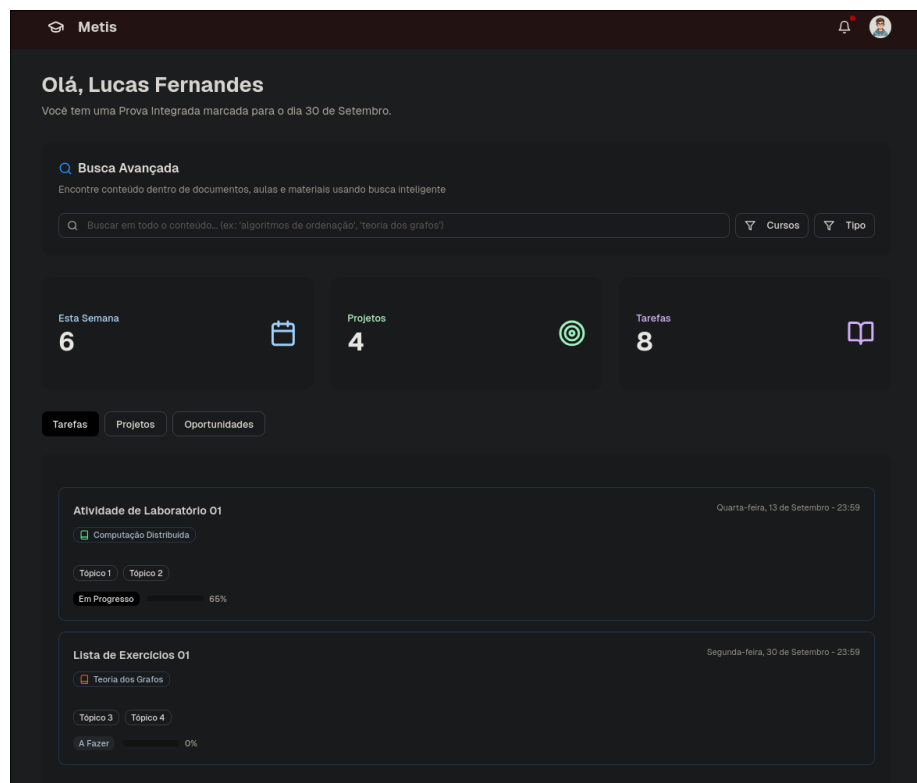


Figure 1: Página Inicial do LMS



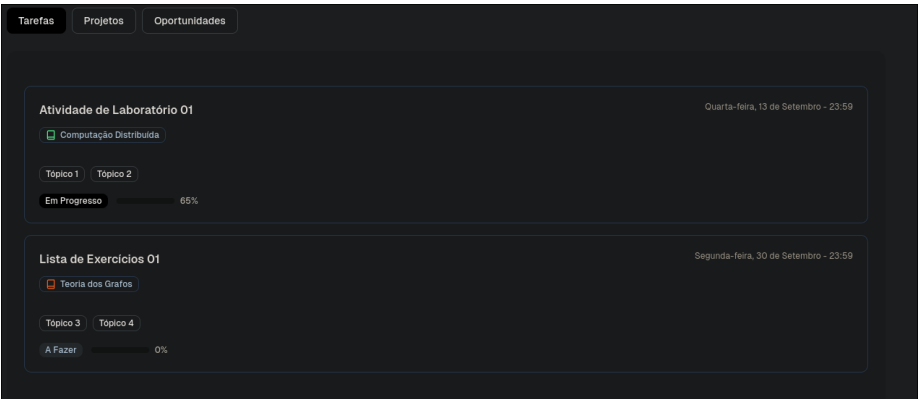
- **Tecnologia:** A interface foi desenvolvida utilizando **Next.js 15** e **Tailwind CSS**.
- **Design System:** Foi adotado um padrão de componentes reutilizáveis (**Shadcn UI**) para garantir uma interface moderna, intuitiva e padronizada. Isso resolve o problema de “**Design Defasado e Inconsistente**” dos sistemas legados.
- **Responsividade:** O uso de componentes e a filosofia de design **mobile-first** garante que a plataforma seja totalmente funcional em dispositivos móveis, cumprindo o requisito não funcional de alta prioridade (**USAB-NF1**).

## 8.2. Dashboard e Hub Acadêmico

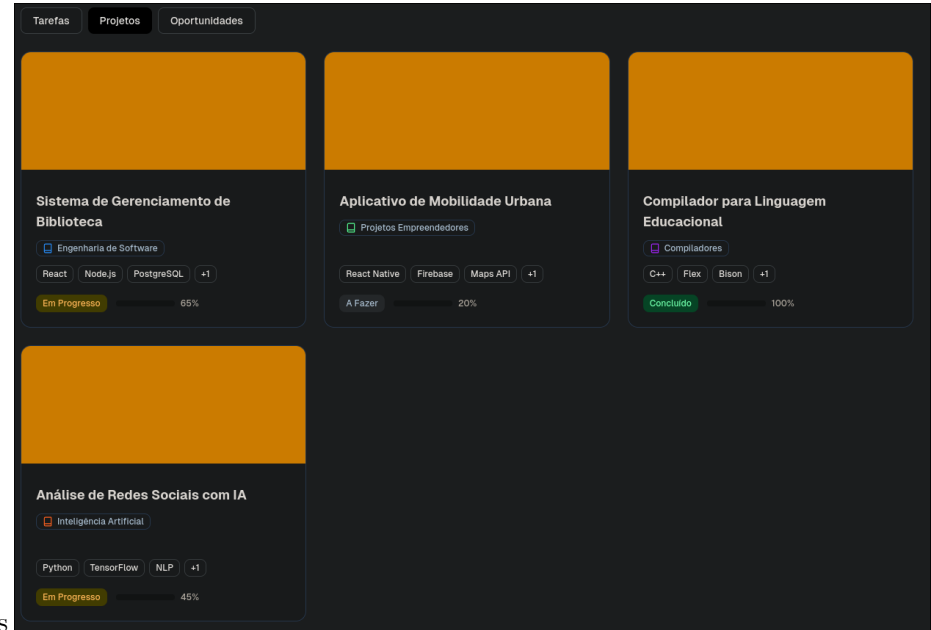
O *Dashboard* demonstra a centralização das informações, transformando o LMS em um “**Hub da Vida Acadêmica**”.

**8.2.1. Centralização de Tarefas e Projetos** As guias **Tarefas** e **Projetos** oferecem uma visão rápida e acionável das pendências do aluno.

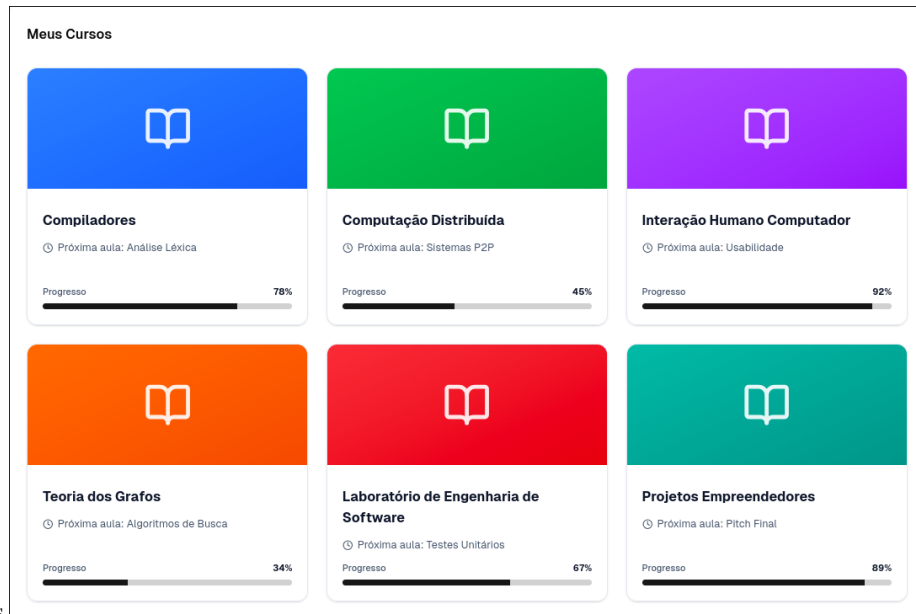
Imagem de Referência	
Característica/Funcionalidade	Impacto/Diferencial
<b>Gerenciamento de Tarefas</b>	Permite ao usuário visualizar os próximos prazos, o progresso e o <i>status</i> ( <b>Em Progresso</b> , <b>A Fazer</b> ) diretamente na tela principal, promovendo eficiência e controle sobre as atividades.
<b>Visão de Projetos</b>	A <b>guia Projetos</b> exibe os trabalhos em andamento em um formato visual de cartões, indicando o progresso percentual e as tecnologias ( <b>tags</b> ), o que facilita a localização e o acompanhamento do status.
<b>Visualização de Cursos</b>	A seção “ <b>Meus Cursos</b> ” exibe o progresso visual de cada disciplina e a próxima aula agendada em cartões claros, contrastando com interfaces legadas.



Tarefas



Projetos

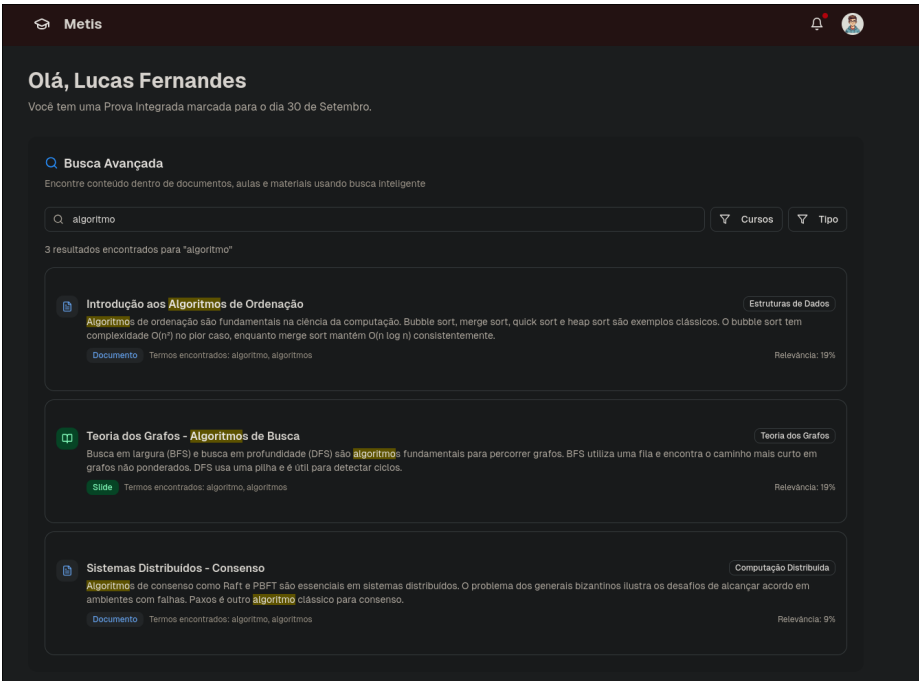


Cursos

Caso use alguma extensão de fundo preto os cursos perdem as diferenciação de cores.

**8.2.2. Busca Avançada (Fuzzy Search)** Este recurso implementa a **Busca Difusa**, abordando a dor de “perder tempo procurando materiais” ao buscar dentro do conteúdo.

	Imagem
	de
	Refer-
Característica/Funcionalidade	Diferencial
<b>Busca por Conteúdo</b>	<b>busca avançada</b> O sistema busca dentro do conteúdo dos documentos, slides e vídeos (simulação de <i>fuzzy matching</i> ), retornando resultados ranqueados por <b>Relevância</b> e destacando os termos encontrados para dar contexto imediato.
<b>Filtros</b>	<b>busca avançada</b> Permite filtrar por <b>Cursos</b> e <b>Tipo</b> de material (Documento, Slides, Vídeo), refinando a busca de forma intuitiva.
<b>Con-textuais</b>	



Busca Avançada

8.3. Sistema de Recomendação de Oportunidades (Inovação)

Este é o principal diferencial do projeto, cumprindo o requisito de ter um **Motor de Recomendações** (OPP-RF4).

	Imagem de Referência Funcional	Racional/Diferencial
Match Personalizado		Exibe a porcentagem de <b>Match</b> para cada oportunidade e o progresso (círculo de progresso). A seção <b>“Por que recomendamos:”</b> lista os fatores de correspondência (habilidades, interesses, nível de dificuldade), dando transparência ao algoritmo.
Tipos e Detalhes		Oportunidades são categorizadas de forma clara (Estágio, Hackathon, Bolsa) e fornecem detalhes essenciais (Localização, Prazo, Salário/Prêmio).
Ações e Métricas		O painel de resumo exibe métricas importantes (Total de Oportunidades, <i>Match</i> Médio e Salvos), e os cartões oferecem ações diretas de <b>“Salvar”</b> e <b>“Candidatar-se”</b> .

Oportunidades

Tarefas

Projetos

Oportunidades

Oportunidades Recomendadas

Sistema inteligente que cruza seu perfil acadêmico com oportunidades disponíveis

6

88%

1

Estágio

Intermediário

87%

Estágio em Desenvolvimento Full-Stack

TechCorp

Oportunidade para trabalhar com React, Node.js e PostgreSQL em projetos inovadores.

Por que recomendamos:

• Conhecimento em JavaScript, React, Node.js, TypeScript

• Nível de dificuldade adequado (Intermediário)

• Localização Ideal (São Paulo, SP)

React

Node.js

TypeScript

PostgreSQL

Full-Stack

São Paulo, SP

Prazo: 6 meses

R\$ 2.000 - R\$ 3.000

Salvar

Candidatar-se

Emprego

Intermediário

91%

Programa de Trainee - Data Science

DataLab

Programa completo de formação em Data Science com mentoria especializada.

Por que recomendamos:

• Conhecimento em Python, SQL

• Interesse em Data Science

• Nível de dificuldade adequado (Intermediário)

Python

Machine Learning

SQL

Data Science

Analytics

São Paulo, SP

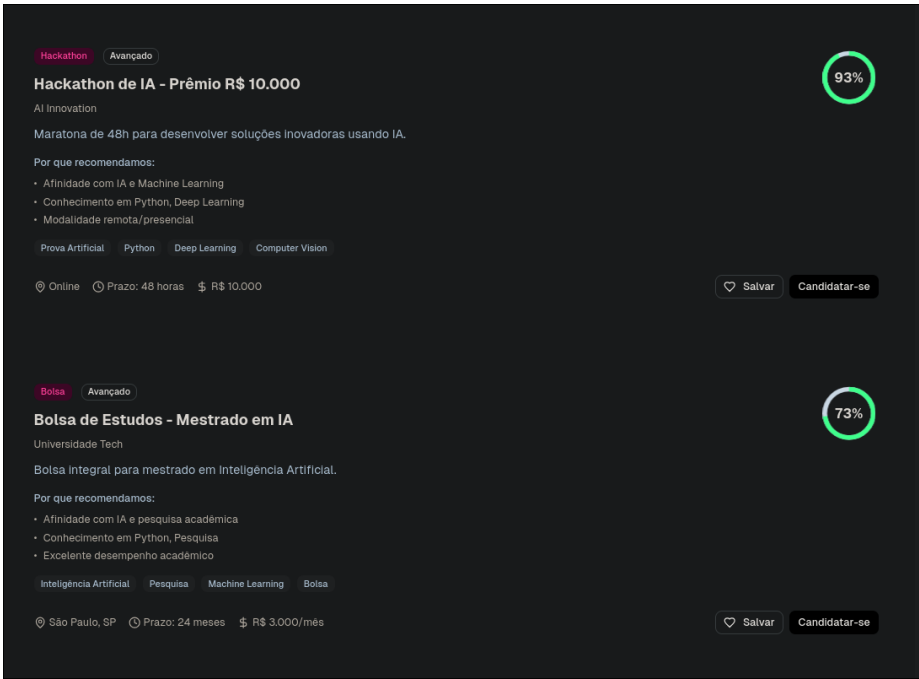
Prazo: 12 meses

R\$ 4.000 - R\$ 6.000

Salvar

Candidatar-se

21



Oportunidades 2

8.4. Gestão de Perfil

A tela de perfil serve como o painel de controle do aluno e a fonte de dados primária para o motor de recomendação.

Imagem de Referência	Características de Funcionalidade/Diferencial
Entrada de Tags	Perfil_interesses.png e gerencia suas Áreas de Interesse, Habilidades Técnicas e Objetivos de Carreira por meio de tags. Essas entradas são a base do sistema de recomendação por Content-Based Filtering.
Desempenho Consolidado	Desempenho.png “Desempenho Acadêmico” lista as notas e créditos por disciplina em um formato visualmente organizado, e a seção Estatísticas (em Perfil.png) resume o CRA, Disciplinas e Conquistas.
Conquistas e Certificações	Desempenho.png de Conquistas permite registrar prêmios e certificações (ex: Hackathon, Certificação AWS), enriquecendo o perfil do aluno e melhorando a precisão do match.

Imagem de Referência	
Características/Funções/Qualidade/Diferencial	
<b>Configurações</b>	A seção <b>Configurações</b> permite ao aluno gerenciar informações pessoais e, crucialmente, definir as preferências de <b>Notificações</b> , incluindo alertas sobre <b>Novas Oportunidades</b> .

## Perfil de Interesse

**Interesses**

Áreas de Interesse

Desenvolvimento Web x Inteligência Artificial x Data Science x Mobile x DevOps x

Nova área de interesse +

Habilidades Técnicas

React x Node.js x Python x TypeScript x PostgreSQL x Docker x Java x

C +

Objetivos de Carreira

Full-Stack Developer x Data Scientist x Tech Lead x

Novo objetivo de carreira +

Figure 2: Seção de Interesses dentro do Perfil

## Desempenho

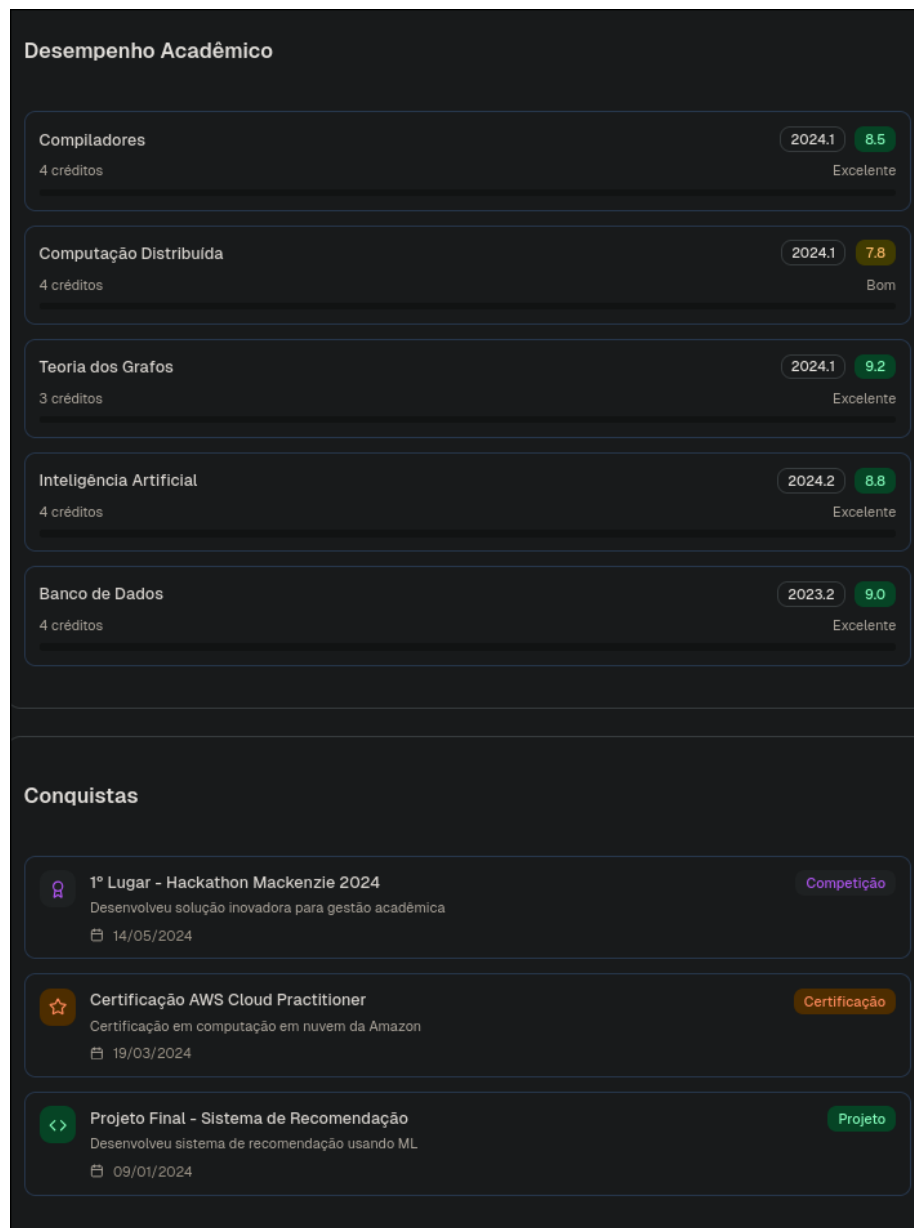
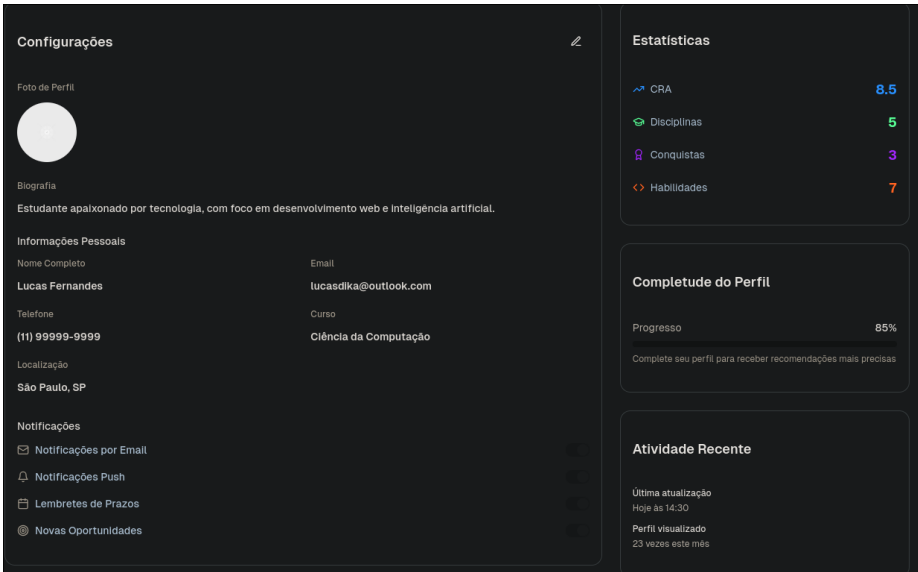


Figure 3: Desempenho do Usuário






Perfil

8.5. Detalhes da Disciplina

A tela detalhada do curso garante a organização padronizada e a centralização de todo o conteúdo e agenda do professor.

Imagem de Referência	Características/Funcionalidades/Diferencial
Organização dos cursos de conteúdo	O conteúdo é dividido em blocos claros (“Sobre o Curso”, “Cronograma de Aulas”, “Tarefas”), resolvendo o problema de “inconsistência na organização de conteúdo”.
Estrutura	O Cronograma de Aulas
Cronograma Visual	O Cronograma de Aulas
Informações do Professor	O Calendário de Entregas



## Compiladores

Terças e Quintas, 19h00 - 20h40

### Sobre o Curso

Estudo dos princípios e técnicas de construção de compiladores, incluindo análise léxica, sintática, semântica e geração de código.

Conclusão 78%

### Cronograma de Aulas

- Introdução a Compiladores  
05/09 - 2h
- Análise Léxica  
10/09 - 2h
- Análise Sintática  
17/09 - 2h
- Análise Semântica  
24/09 - 2h
- Geração de Código  
01/10 - 2h

### Tarefas

Lista de Exercícios 01 15/09

Análise Léxica Expressões Regulares

Concluído 100%

Projeto de Analisador Léxico 22/09

Análise Léxica Implementação


### Calendário de Entregas

Setembro 2025

D	S	T	Q	Q	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Entregas e Provas

### Professor



**Prof. Dr. Carlos Silva**  
Doutor em Ciência da Computação com especialização em Compiladores e Linguagens de Programação.

**Sala**  
Prédio 6, Sala 302

**Horário de Atendimento**  
Terças e Quintas, 17h00 - 18h30

**Áreas de Pesquisa**  
Compiladores Linguagens de Programação  
Otimização de Código

**Publicações**  
15+ artigos publicados em conferências internacionais

Enviar Email

Página do Curso

Extra:

Pagina para Login no Metis

Login Ecossistema Microsoft

Validação dos Login

Todas as imagens podem ser encontradas aqui: `./docs/assets/`

## Capítulo 9:

### Trabalhos futuros

- Uso de MongoDB baseado em grafos para auxiliar no motor de recomendação.

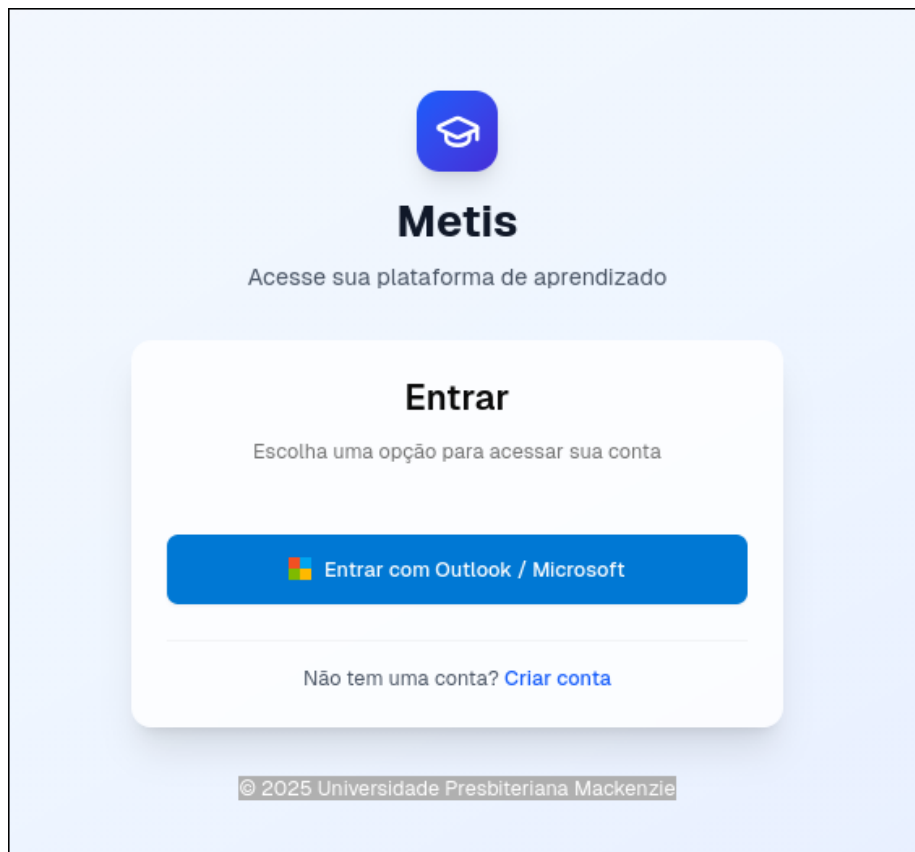


Figure 4: Login Metis

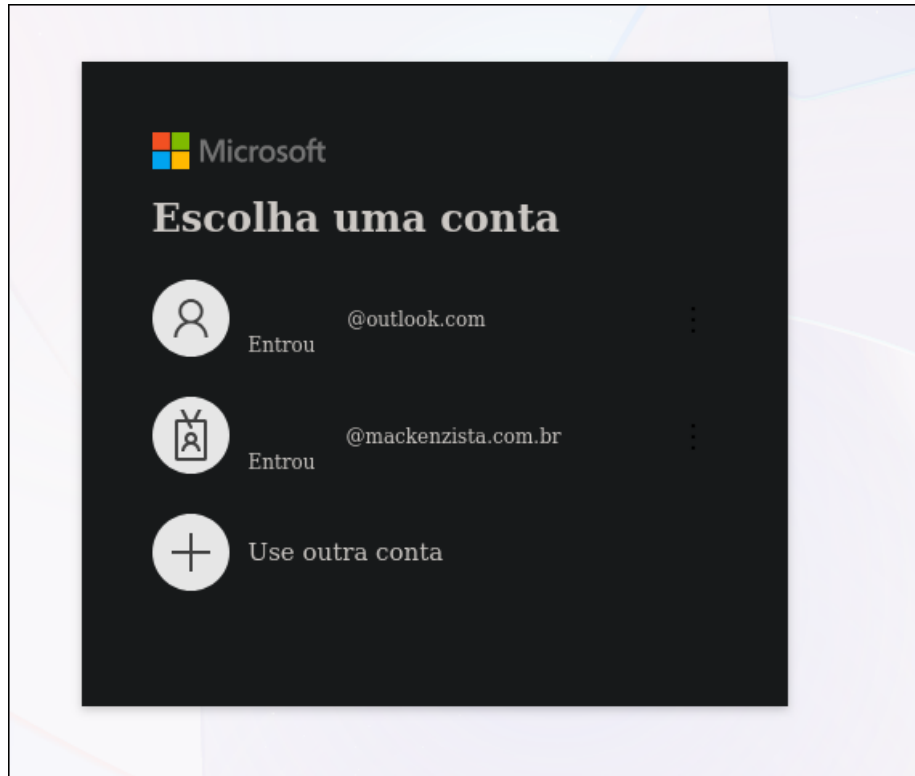


Figure 5: Login

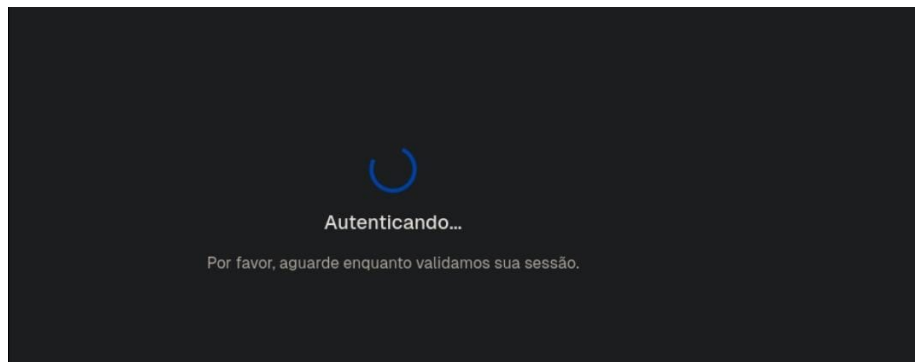


Figure 6: Val\_Metis