

Metis - Learning Management System

Sistema de gerenciamento de aprendizado desenvolvido para a Universidade Presbiteriana Mackenzie.

Equipe: | Nome | RA | | Enzo Benedetto Proen  sa | 10418579 | | Gabriel Ken Kazama Geronazzo | 10418247 | | J  ssica Bispo | 10410798 | | Lucas Fernandes | 10419400 | | Lucas P. C. Sarai | 10418013 | | Vitor Alves Pereira | 10410862 |

   Documenta  o T  cnica

Para instru  es de instala  o, configura  o, autentica  o e troubleshooting, consulte o [Guia T  cnico \(TECHNICAL GUIDE.md\)](#).

  – Sobre o Projeto

Cap  tulo 1: Introdu  o

O projeto visa n  o apenas modernizar a experi  ncia de usu  rio atrav  s de um design consistente e responsivo, mas tamb  m expandir o papel do LMS, transformando-o de um reposit  rio de conte  do em um ecossistema integrado que promove o desenvolvimento do aluno.

Para isso, prop  ue-se a implementa  o de funcionalidades de melhoria, como uma ferramenta de busca inteligente, e inova  es como um sistema de recomenda  o de oportunidades acad  micas e extracurriculares.

Cap  tulo 2: Detalhamento do Projeto

2.1. Motiva  o

A motiva  o central deste projeto nasce da experi  ncia di  ria como estudante universit  rio e da observa  o das dificuldades operacionais causadas pelas ferramentas de gest  o de aprendizagem atualmente em uso.

A dificuldade em localizar materiais de estudo, a inconsist  ncia na organização de conte  do entre diferentes disciplinas e a instabilidade da plataforma resultam em interrup  es no acesso ao conte  do de aprendizagem.

Adicionalmente, identifica-se uma lacuna na centraliza  o de informa  es relevantes para a jornada do aluno, como oportunidades de est  gio, projetos de pesquisa e atividades de extens  o, que hoje s  o divulgadas de forma descentralizada e assim  trica.

O objetivo   , portanto, criar uma solu  o que resolva esses pontos pr  ticos e organize a experi  ncia acad  mica, 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" é:** 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:** Form 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
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	Suporitar uploads até 10 MB no MVP	Média
MON- NF1	Monitoramento	Logs	Logs de autenticação/erros com retenção de Baixa 30 dias	
USAB- NF1	Usabilidade e Acessibilidade	Responsividade e compatibilidade	Mobile-first; suporitar 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

Olá, username

Você tem uma Prova Integrada marcada para o dia 30 de Setembro.

Q Busca Avançada

Encontre conteúdo dentro de documentos, aulas e materiais usando busca avançada.

Q Buscar em todo o conteúdo... (ex: 'algoritmos de ordenação', 'teoria dos grafos')

Esta Semana

6



Projeto

4

Dashboard

Oportunida

Minhas Tarefas & Projetos

Atividade de Laboratório 01

Computação Distribuída

Tópico 1

Tópico 2

Progresso

Lista de Exercícios 01

Teoria dos Grafos

Tópico 3

Tópico 4

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

Para essa aplicação, decidimos utilizar uma interface interativa invocada 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 aprendizado 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:

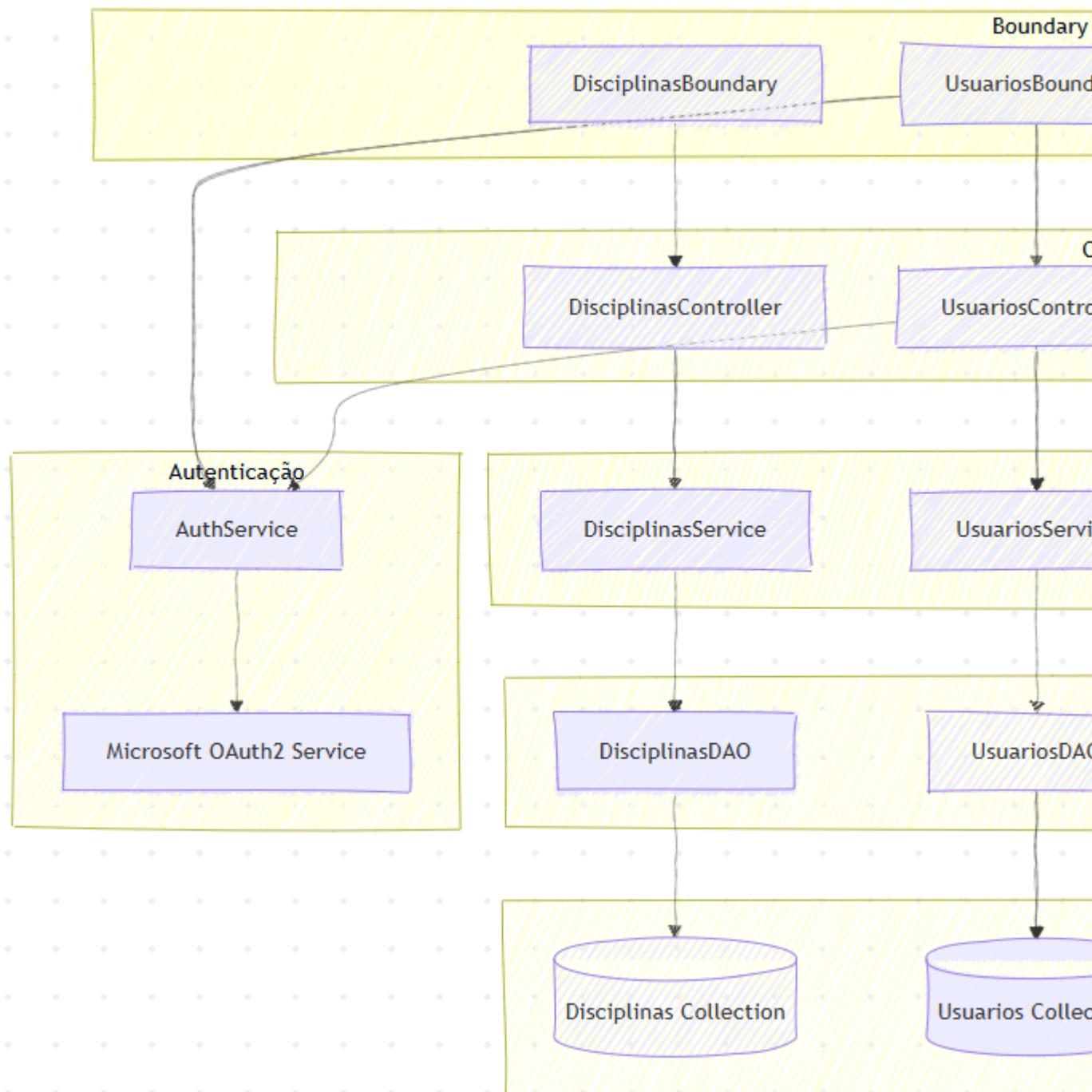
- Gerenciar usuários (criar, editar, remover)
- Gerenciar disciplinas (criar, editar, remover)
- Vincular professores e alunos às disciplinas

Módulos

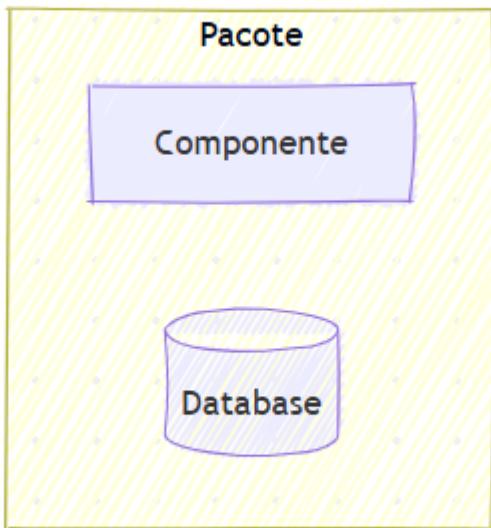
- [Autenticação e Autorização](#)
- [Usuários](#)
- [Arquivos](#)
- [Disciplinas](#)
- [Oportunidades](#)

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: ${{ env.ECR_REGISTRY }}/{{ env.ECR_REPOSITORY }}:${{ 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-
        ${{ 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 Jússica.
- 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.

Etapa 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 recomendado, 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.

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.

PÃ¡gina Inicial

Olá, Lucas Fernandes

Você tem uma Prova Integrada marcada para o dia 30 de Setembro.

🔍 Busca Avançada

Encontre conteúdo dentro de documentos, aulas e materiais usando busca inteligente

🔍 Buscar em todo o conteúdo... (ex: 'algoritmos de ordenação', 'teoria dos grafos')

Esta Semana

6



Projetos

4

Tarefas

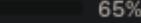
Projetos

Oportunidades

Atividade de Laboratório 01

 Computação Distribuída

 Tópico 1  Tópico 2

Em Progresso  65%

Lista de Exercícios 01

 Teoria dos Grafos

 Tópico 3  Tópico 4

A Fazer  0%

8.1. Visão Geral da Arquitetura Frontend

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

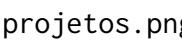
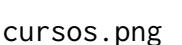
- **Tecnologia:** A interface foi desenvolvida utilizando **Next.js 15** e **Tailwind CSS**.
- **Design System:** Foi adotado um padrão de componentes reutilizáveis (**Shaden UI**) para garantir uma interface moderna, intuitiva e padronizada. Isso resolve o problema de “**Design Defasado e Inconsistente**” 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.

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

Tarefas

Tarefas

Projetos

Oportunidades

Atividade de Laboratório 01

Computação Distribuída

Tópico 1

Tópico 2

Em Progresso

65%

Lista de Exercícios 01

Teoria dos Grafos

Tópico 3

Tópico 4

A Fazer

0%

Projetos

Tarefas

Projetos

Oportunidades

Sistema de Gerenciamento de Biblioteca

Engenharia de Software

React Node.js PostgreSQL +1

Em Progresso

65%

Aplicativo de Mobilidade Urbana

Projetos Empreendedores

React Native Firebase Maps API +1

A Fazer

20%

Análise de Redes Sociais com IA

Inteligência Artificial

Python TensorFlow NLP +1

Em Progresso

45%

Cursos

Meus Cursos



Compiladores

⌚ Próxima aula: Análise Léxica

Progresso

78%



Computação Distribuída

⌚ Próxima aula: Sistemas P2P

Progresso

45%



Teoria dos Grafos

⌚ Próxima aula: Algoritmos de Busca

Progresso

34%



Laboratório de Engenharia de Software

⌚ Próxima aula: Testes Unitários

Progresso

67%

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

8.2.2. Busca Avançada (Fuzzy Search)

Este recurso implementa a **Busca Difusa**, abordando a dor de â€œperder tempo procurando materiaisâ€œ ao buscar dentro do conteúdo.

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

Busca Avançada

Olá, Lucas Fernandes

Você tem uma Prova Integrada marcada para o dia 30 de Setembro.

Q Busca Avançada

Encontre conteúdo dentro de documentos, aulas e materiais usando busca inteligente

Q algoritmo

3 resultados encontrados para "algoritmo"



Introdução aos Algoritmos de Ordenação

Algoritmos de ordenação são fundamentais na ciência da computação. Bubble sort, merge complexidade $O(n^2)$ no pior caso, enquanto merge sort mantém $O(n \log n)$ consistentemente.

Documento

Termos encontrados: algoritmo, algoritmos



Teoria dos Grafos - Algoritmos de Busca

Busca em largura (BFS) e busca em profundidade (DFS) são algoritmos fundamentais para grafos não ponderados. DFS usa uma pilha e é útil para detectar ciclos.

Slide

Termos encontrados: algoritmo, algoritmos



Sistemas Distribuídos - Consenso

Algoritmos de consenso como Raft e PBFT são essenciais em sistemas distribuídos. O problema é que os sistemas devem ser resilientes a ambientes com falhas. Paxos é outro algoritmo clássico para consenso.

Documento

Termos encontrados: algoritmo, algoritmos

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).

Característica/ Funcionalidade	Imagen de Referência	Racional/Diferencial
Match Personalizado	oprotunidades.png, oprotunidades2.png	Exibe a porcentagem de Match para cada oportunidade (círculo de progresso). A seção Por que recomendamos: enumera os fatores de correspondência (habilidades, interesses, nível de dificuldade), dando transparência ao algoritmo.
Tipos e Detalhes	oprotunidades.png, oprotunidades2.png	Oportunidades são categorizadas de forma clara (Estágio, Hackathon, Bolsa) e fornecem detalhes essenciais (Localização, Prazo, Salário/Prêmio).
Ações e Metrícias	oprotunidades.png	O painel de resumo exibe métricas importantes (Total de Oportunidades, Match Médio e Salvos), e os cartões oferecem ações diretas de Salvar e Candidatar-se .

Oportunidades

Tarefas

Projetos

Oportunidades

⌚ Oportunidades Recomendadas

Sistema inteligente que cruza seu perfil acadêmico com oportunidades disponíveis

Estágio

Intermediário

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

Emprego

Intermediário

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

Oportunidades 2

[Hackathon](#) [Avançado](#)

Hackathon de IA - Prêmio R\$ 10.000

AI Innovation

Maratona de 48h para desenvolver soluções inovadoras usando IA.

Por que recomendamos:

- Afinidade com IA e Machine Learning
- Conhecimento em Python, Deep Learning
- Modalidade remota/presencial

[Prova Artificial](#) [Python](#) [Deep Learning](#) [Computer Vision](#)

⌚ Online ⏰ Prazo: 48 horas \$ R\$ 10.000

[Bolsa](#) [Avançado](#)

Bolsa de Estudos - Mestrado em IA

Universidade Tech

Bolsa integral para mestrado em Inteligência Artificial.

Por que recomendamos:

- Afinidade com IA e pesquisa acadêmica
- Conhecimento em Python, Pesquisa
- Excelente desempenho acadêmico

[Inteligência Artificial](#) [Pesquisa](#) [Machine Learning](#) [Bolsa](#)

⌚ São Paulo, SP ⏰ Prazo: 24 meses \$ R\$ 3.000/mês

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.

Característica/ Funcionalidade

Imagen de Referência Racional/Diferencial

Entrada de Tags

Perfil_interesses.png

O usuário insere 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

A seção **Desempenho** é o **Dashboard** do usuário, que 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

A área de **Conquistas** permite registrar prêmios e certificações (ex: Hackathon, Certificações AWS), enriquecendo o perfil do aluno e melhorando a precisão do *match*.

Configurações Essenciais

Perfil.png

A seção **Configurações** permite ao aluno gerenciar informações pessoais e, crucialmente, definir as preferências de **Notificações**, incluindo alertas sobre **Novas Oportunidades**.

Perfil de Interesse

Interesses

Áreas de Interesse

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

Nova área de interesse

Habilidades Técnicas

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

C

Objetivos de Carreira

Full-Stack Developer × Data Scientist × Tech Lead ×

Novo objetivo de carreira

Seção de Interesses dentro do Perfil

Desempenho

Desempenho Acadêmico

Compiladores

4 créditos

2024.1

8

Excelente

Computação Distribuída

4 créditos

2024.1

7

Bom

Teoria dos Grafos

3 créditos

2024.1

9

Excelente

Inteligência Artificial

4 créditos

2024.2

8

Excelente

Banco de Dados

4 créditos

2023.2

9

Excelente

Conquistas



1º Lugar - Hackathon Mackenzie 2024

Competição

Desenvolveu solução inovadora para gestão acadêmica

14/05/2024



Certificação AWS Cloud Practitioner

Certificação

Certificação em computação em nuvem da Amazon

19/03/2024



Projeto Final - Sistema de Recomendação

Projeto

Desenvolveu sistema de recomendação usando ML

09/01/2024

Desempenho do Usuário

Perfil

Configurações

Foto de Perfil



Biografia

Estudante apaixonado por tecnologia, com foco em desenvolvimento web e inteligência artificial.

Informações Pessoais

Nome Completo	Email
Lucas Fernandes	lucasdika@outlook.com
Telefone	Curso
(11) 99999-9999	Ciência da Computação
Localização	
São Paulo, SP	

Notificações

- Notificações por Email
- Notificações Push
- Lembretes de Prazos
- Novas Oportunidades

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.

CaracterÃstica/ Imagem de Funcionalidade ReferÃncia**Racional/Diferencial****OrganizaÃ§Ã£o Estruturada**

O conteÃºdo Ã© dividido em blocos claros (â€œSobre o Cursoâ€, â€œCronograma de Aulasâ€, â€œTarefasâ€), resolvendo o problema de â€œinconsistÃ¢ncia na organizaÃ§Ã£o de conteÃºdosâ€.

Cronograma Visual

 O **Cronograma de Aulas** exibe a ordem, status de conclusÃ£o e duraÃ§Ã£o de cada aula, com fÃ;til acesso ao material. O **CalendÃ¡rio de Entregas** centraliza os prazos de provas e tarefas na lateral/[page.tsx].

InformaÃ§Ãµes do Professor

 Centraliza informaÃ§Ãµes de contato, biografia e horÃ;jrios de atendimento do professor em um painel dedicado, melhorando a comunicaÃ§Ã£o/[page.tsx].



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

Cale

D

7

14

21

28

Ent

Profe



Sala

Prédio

Horári

Terças

Áreas

Com

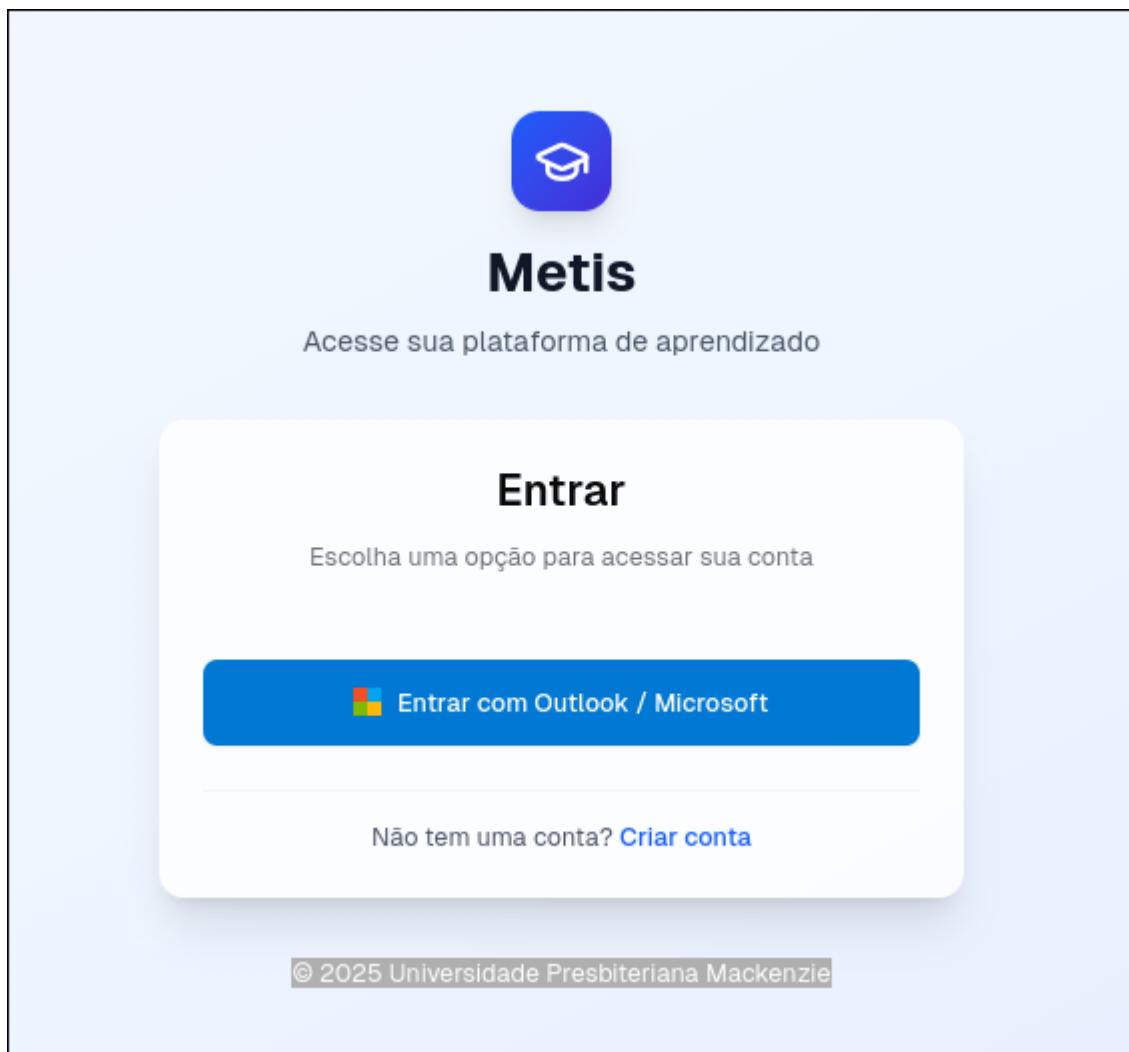
Otim

Public

15+ art

Extra:

Pagina para Login no Metis



Login Metis

Login Ecossistema Microsoft



Escolha uma conta



Entrou

@outlook.com



Entrou

@mackenzista.com.br



Use outra conta

Login

ValidaÃ§Ã£o dos Login



Autenticando...

Por favor, aguarde enquanto validamos sua sessão.

Val_Metis

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.