

# CURSO DE BACHARELADO EM CIENCIA DA COMPUTAÇÃO

João Vitor Deliberador

Mario Henrique Akihiko da Costa Adaniya

PROJETO COLMEIA CUBO

Londrina

2024

---

## SUMÁRIO

- 2024
  - SUMÁRIO
  - AGRADECIMENTOS
  - INTRODUÇÃO
  - DESCRIÇÃO DO SISTEMA
    - FRONT-END
    - BACK-END
    - BANCO DE DADOS
    - DIAGRAMAÇÃO
  - MODELAGEM DO SISTEMA
    - DIAGRAMA DE CASO DE USO
    - DIAGRAMA DE CLASSE
    - DIAGRAMA DE ENTIDADE E RELACIONAMENTO
    - CASOS DE USO
      - UC001\_GERENCIAR\_CURSOS
        - Telas
      - UC002\_GERENCIAR\_AREAS\_DE\_CONHECIMENTO
        - Telas
      - UC003\_GERENCIAR\_AREAS\_FUNCIONAIS
        - Telas
  - CONCLUSÕES E TRABALHOS FUTUROS
  - REFERÊNCIAS
  - APÊNDICE
    - ARTEFATOS DE REQUISITOS
      - Documento de visão
      - Especificação Suplementar
      - Especificações de Caso de Uso:
      - Glossário
      - Pedido do Investidor
    - DIAGRAMAS
  - Diagrama de Caso de Uso
  - Diagrama de Classes

- Diagrama de Entidade e Relacionamento
- Diagramas de Estado:
- Diagrama de Implantação
  - Diagramas de Sequência:
- Workflow AS-IS

## AGRADECIMENTOS

Gostaria de expressar minha gratidão a todos que contribuíram direta ou indiretamente para a realização deste projeto.

Agradeço ao meu professor orientador Mario pelo suporte e principalmente pela paciência na orientação em todas as fases do projeto. Sempre me ajudando a entender cada vez mais do projeto e me ajudando a ser melhor tanto como pessoa mas como profissional

Aos colegas que compartilharam conhecimentos e experiências, que enriqueceram meu aprendizado e me incentivaram a seguir em frente.

Agradeço à UniFil por oferecer um ambiente de prática e desenvolvimento, proporcionando a oportunidade de desenvolver um projeto real e aplicável à instituição.

Por fim, À Deus e à minha família e amigos próximos que em momentos difíceis do projeto sempre estiveram comigo

## INTRODUÇÃO

O projeto **Colmeia CUBO** foi desenvolvido pelo Núcleo de Práticas em Informática (NPI) do Centro Universitário Filadélfia (UniFil), com o objetivo de digitalizar e centralizar a gestão curricular da instituição.

O NPI, como uma Empresa Júnior, visa consolidar conhecimentos acadêmicos através de projetos práticos. Este projeto surgiu para facilitar o gerenciamento de cursos, áreas de conhecimento e unidades curriculares, oferecendo uma interface intuitiva e acessível para alunos, orientadores e coordenadores.

## DESCRIÇÃO DO SISTEMA

O sistema **Colmeia CUBO** centraliza informações curriculares, proporcionando uma visão clara e organizada da estrutura de cursos. Funciona 24/7 e oferece funcionalidades como:

- **Gerenciamento de Cursos:** Adicionar, atualizar, e remover cursos.
- **Gerenciamento de Áreas de Conhecimento:** Definir e personalizar áreas de conhecimento.
- **Gerenciamento de Áreas Funcionais:** Adicionar, atualizar, e remover áreas funcionais .
- **Gerenciamento de Unidades Curriculares (UCs):** Organizar UCs por carga horária e conteúdo.
- **Visualização da Colmeia Curricular:** Interface de consulta para alunos e professores.

### FRONT-END

Tecnologias utilizadas no *front-end*:

- **HTML5:** Estrutura de conteúdo.
- **CSS3 (Tailwind CSS):** Estilização e responsividade.
- **JavaScript (Alpine.js):** Adição de interatividade leve.

- **Livewire**: Componentização de interfaces no Laravel.

## BACK-END

Tecnologias no *back-end*:

- **Laravel 11**: Framework PHP para lógica de negócios e gerenciamento de dados.
- **Laravel Sanctum**: Autenticação de usuários para Single Page Applications (SPA).
- **Livewire**: Para componentização, facilitando a criação de interfaces reativas sem necessidade de uma API separada.

## BANCO DE DADOS

- **MySQL**: Sistema de gerenciamento de banco de dados.

## DIAGRAMAÇÃO

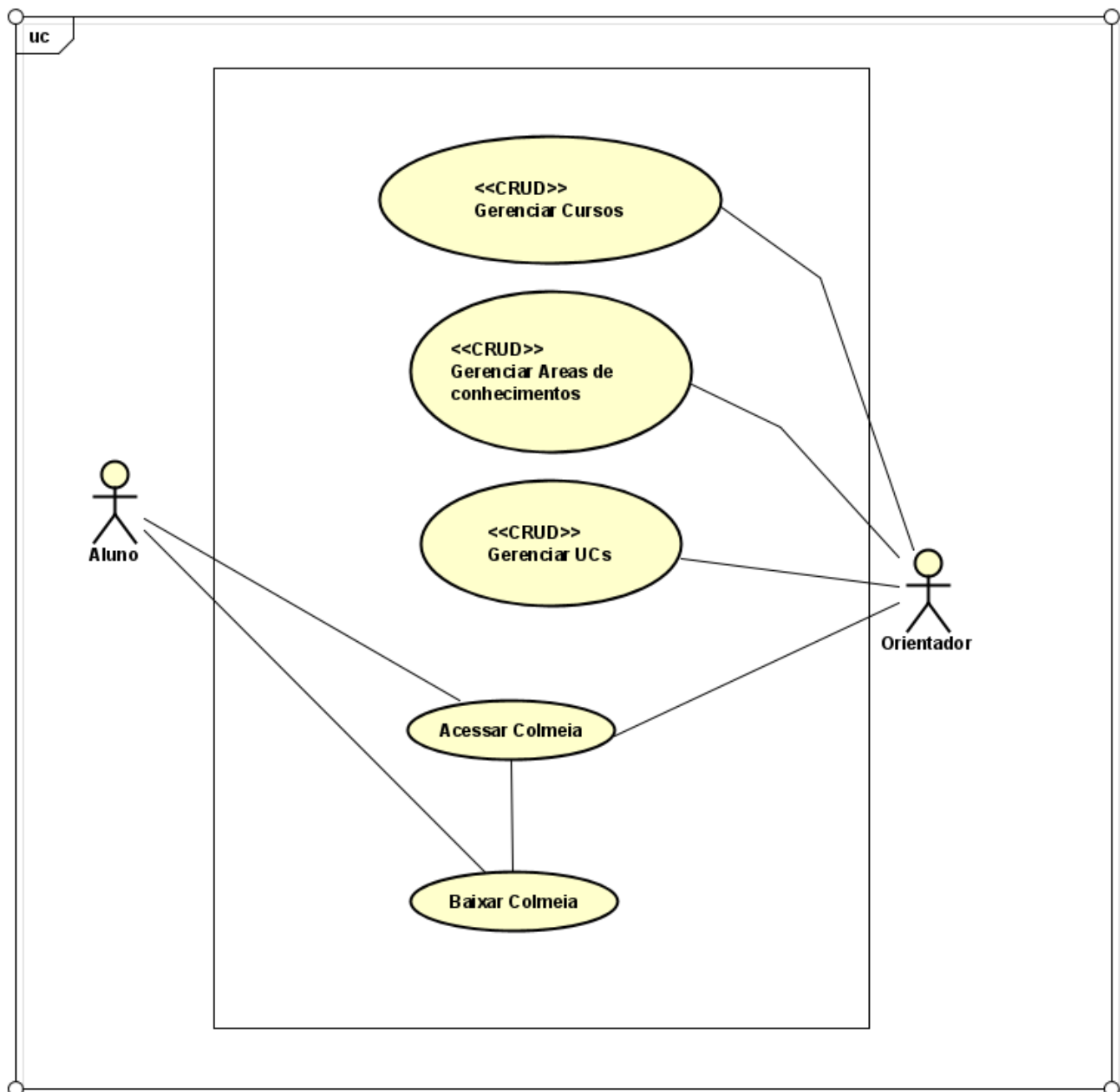
Para a criação dos diagramas, foram utilizadas as ferramentas:

- **draw.io**
- **Mermaid**

## MODELAGEM DO SISTEMA

### DIAGRAMA DE CASO DE USO

O diagrama de caso de uso detalha os requisitos funcionais do sistema, abordando as interações entre usuários e funcionalidades do sistema **Colmeia CUBO**.



## DIAGRAMA DE CLASSE

O diagrama de classes representa as principais entidades e seus relacionamentos no sistema.

```
classDiagram
class Aluno {
    -Nome: string
    -CPF: string
    +acessarColmeia(): void
}

class Coordenador {
    -nome: string
    -ID: int
    +gerenciarCursos(): void
    +gerenciarAreasConhecimento(): void
}
```

```
+gerenciarAreasFuncionais(): void
+gerenciarUnidadesCurriculares(): void
}

class Curso {
    -nome: string
    -codigo: int
    +adicionarCurso(): void
    +atualizarCurso(): void
    +removerCurso(): void
    +visualizarCurso(): void
}

class AreasDeConhecimento {
    -titulo: string
    -descricao: string
    +adicionarArea(): void
    +atualizarArea(): void
    +removerArea(): void
    +visualizarArea(): void
}

class AreasFuncionais {
    -nome: string
    -descricao: string
    +adicionarAreaFuncional(): void
    +atualizarAreaFuncional(): void
    +removerAreaFuncional(): void
    +visualizarAreaFuncional(): void
}

class UnidadeCurricular {
    -titulo: string
    -descricao: string
    -cargaHoraria: int
    +adicionarUC(): void
    +atualizarUC(): void
    +removerUC(): void
    +visualizarUC(): void
}

%% Definição das Relações entre as Classes
Curso "1" -- "0..*" AreasDeConhecimento : contém
AreasDeConhecimento "1" -- "0..*" AreasFuncionais : contém
AreasFuncionais "1" -- "0..*" UnidadeCurricular : contém
Coordenador "1" -- "1" Curso : gerencia
Coordenador "1" -- "1" AreasDeConhecimento : gerencia
Coordenador "1" -- "1" AreasFuncionais : gerencia
Coordenador "1" -- "1" UnidadeCurricular : gerencia
Aluno "0..*" -- "1" Curso : acessa
```

## DIAGRAMA DE ENTIDADE E RELACIONAMENTO

```
erDiagram
CURSO {
    int codigo PK
    string nome
    string descricao
}

AREAS_DE_CONHECIMENTO {
    int id PK
    int curso_id FK
    string titulo
    string descricao
}

AREAS_FUNCIONAIS {
    int id PK
    int area_conhecimento_id FK
    string nome
    string descricao
}

UNIDADES_CURRICULARES {
    int id PK
    int area_funcional_id FK
    string titulo
    string descricao
    int carga_horaria
}

COORDENADOR {
    int id PK
    string nome
}

ALUNO {
    int id PK
    string nome
    string cpf
}

COLMEIA {
    int id PK
    int aluno_id FK
    int curso_id FK
}

%% Relacoes
CURSO ||--o| AREAS_DE_CONHECIMENTO : contem
AREAS_DE_CONHECIMENTO ||--o| AREAS_FUNCIONAIS : contem
```

```
AREAS_FUNCIONAIS ||--o| UNIDADES_CURRICULARES : contem
COORDENADOR ||--o| CURSO : gerencia
COORDENADOR ||--o| AREAS_DE_CONHECIMENTO : gerencia
COORDENADOR ||--o| AREAS_FUNCIONAIS : gerencia
COORDENADOR ||--o| UNIDADES_CURRICULARES : gerencia
ALUNO ||--o| COLMEIA : acessa
CURSO ||--o| COLMEIA : organiza
```

CASOS DE USO

UC001\_GERENCIAR\_CURSOS

Este caso de uso permite que o Coordenador gerencie os cursos.

Telas

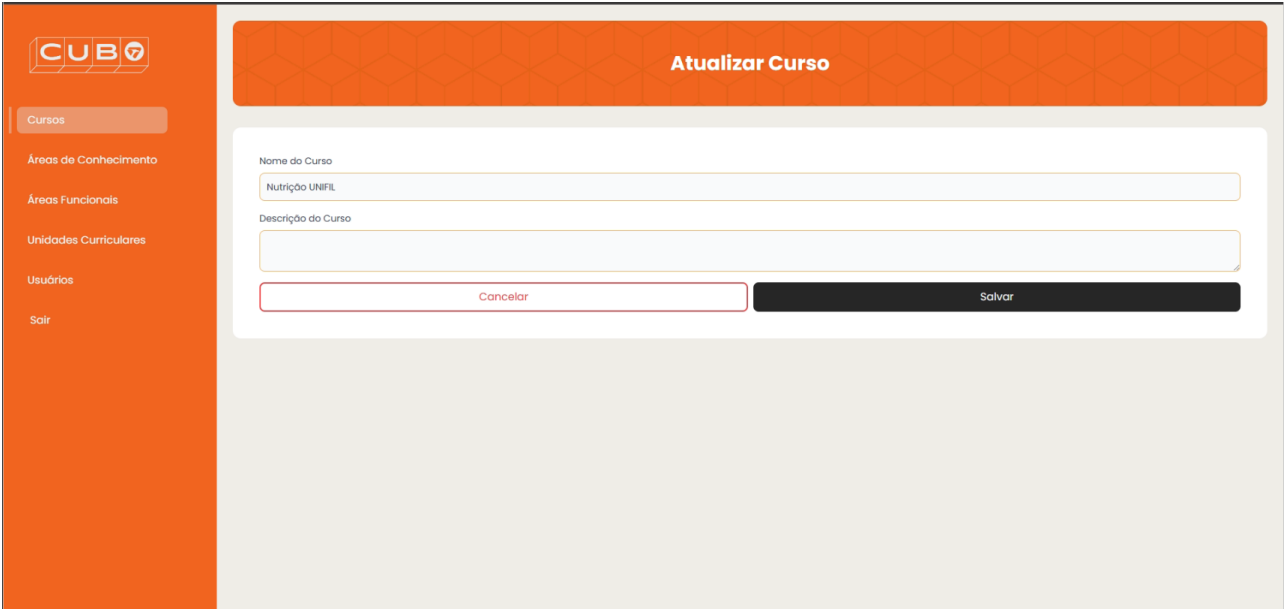
- Tela de Listagem de Cursos



- Tela de Criação de Novo Curso



• Tela de Edição de Curso Existente



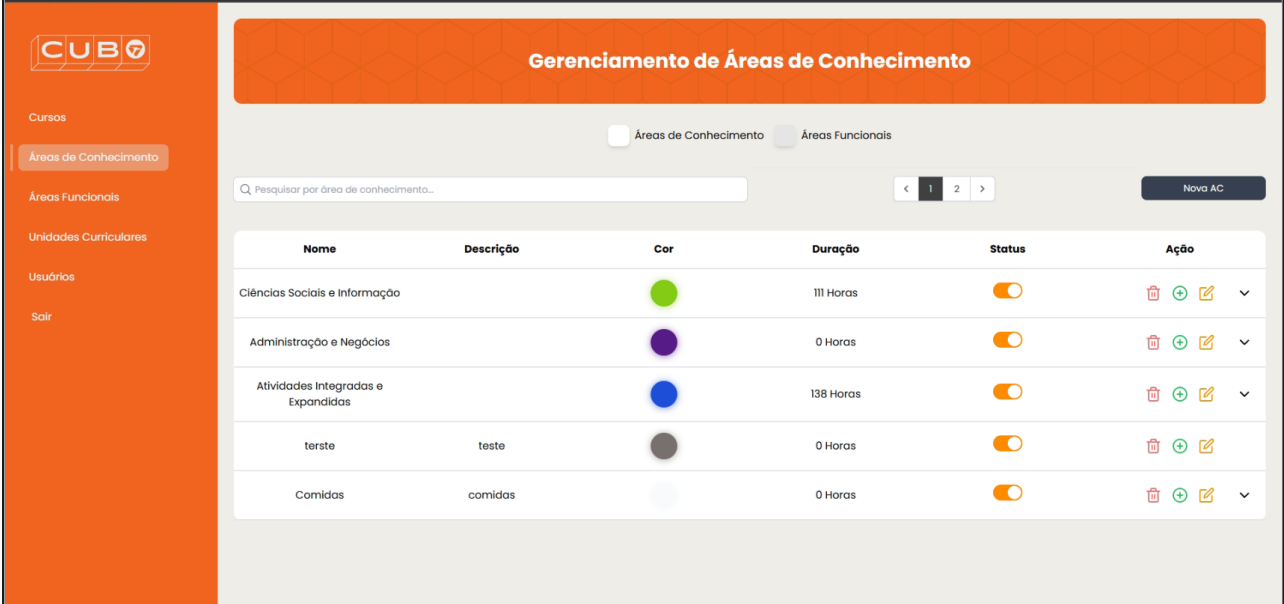
UC002\_GERENCIAR\_AREAS\_DE\_CONHECIMENTO

Permite que o Coordenador gerencie as áreas de conhecimento, incluindo a personalização de cor.

Telas

• Tela de Listagem de Áreas de Conhecimento

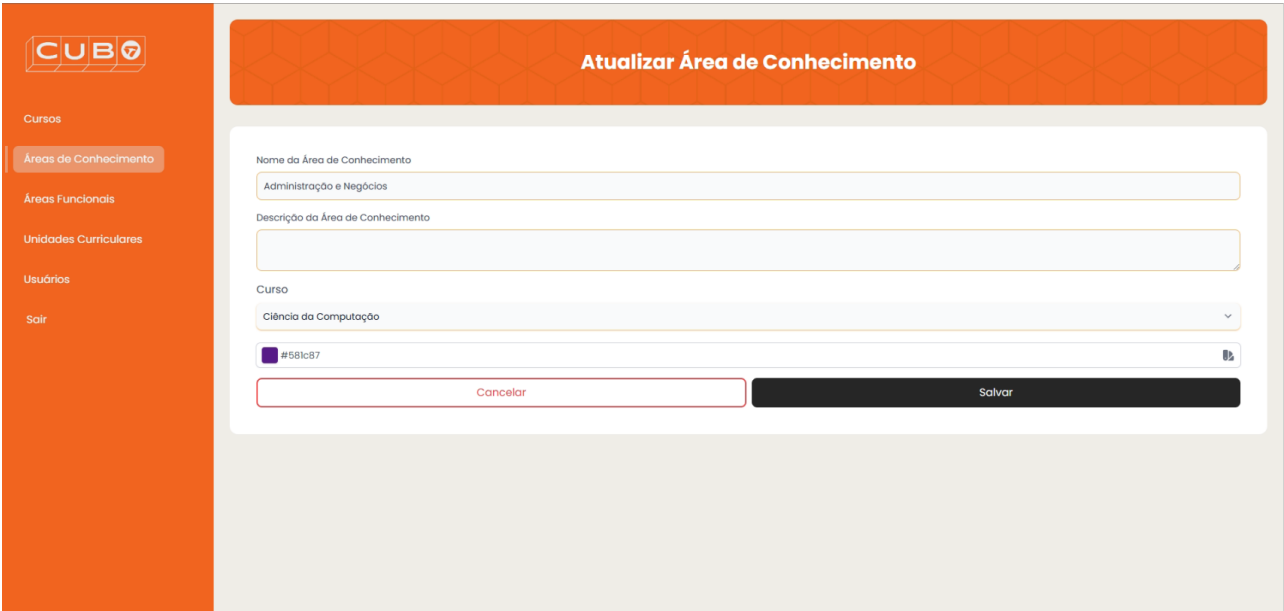




• Tela de Criação de Nova Área de Conhecimento



• Tela de Edição de Área de Conhecimento

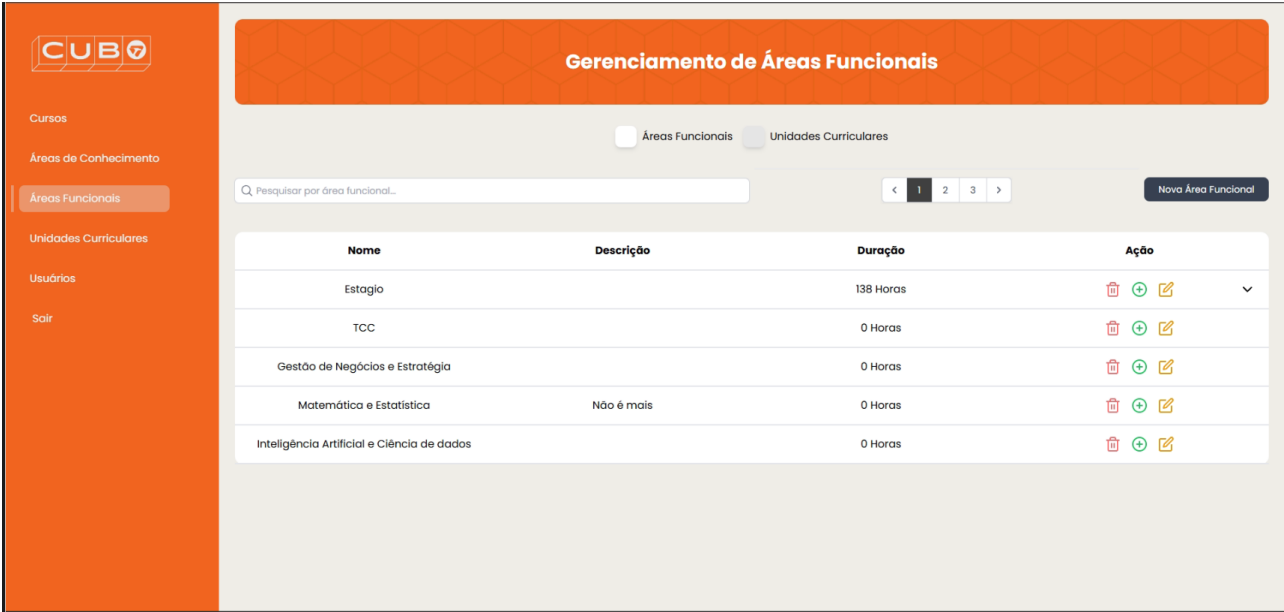


UC003\_GERENCIAR\_AREAS\_FUNCIONAIS

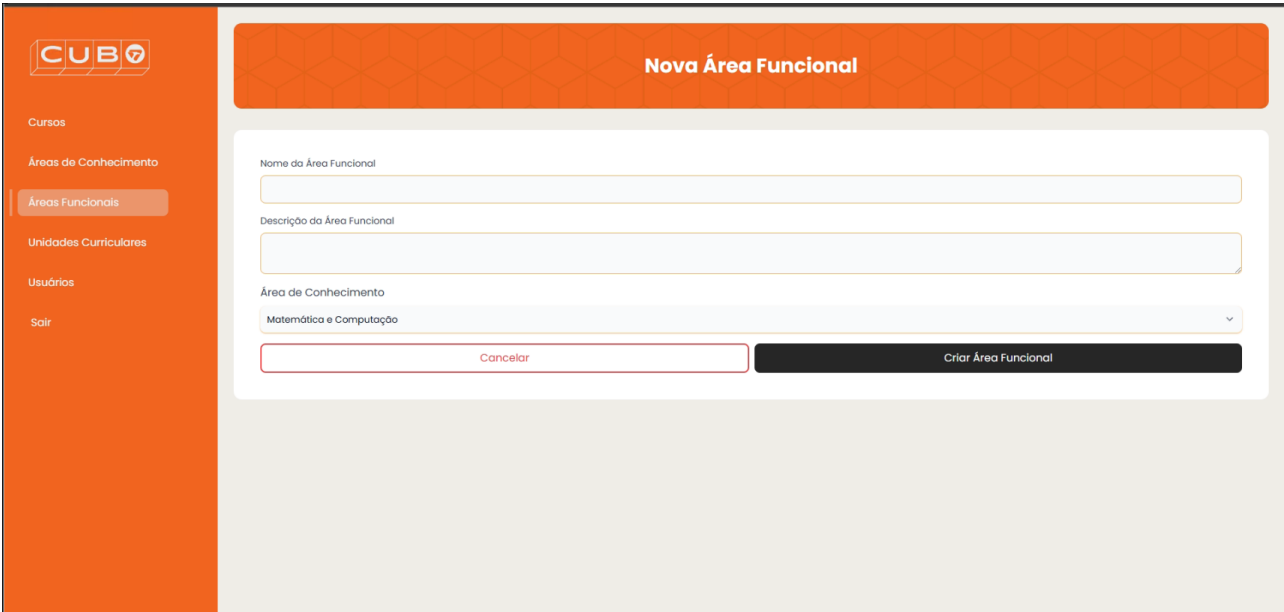
O Coordenador pode gerenciar as áreas funcionais, definindo e organizando as UCs.

Telas

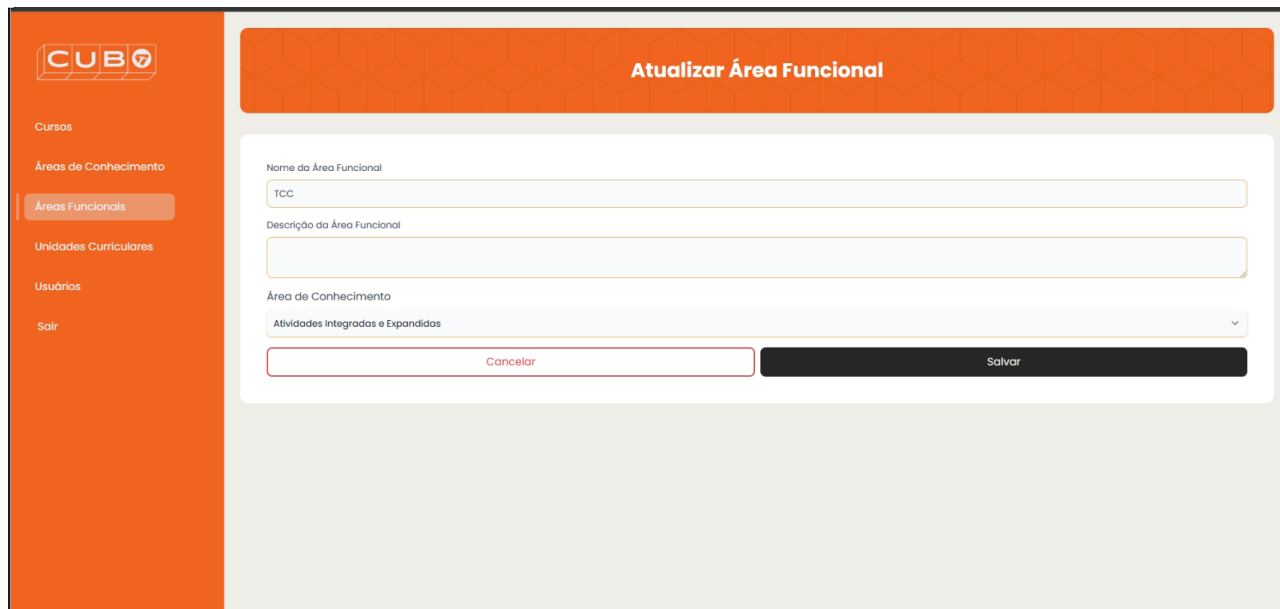
- Tela de Listagem de Áreas Funcionais



- Tela de Criação de Nova Área Funcional



- Tela de Edição de Área Funcional



The screenshot displays the CUBO system interface. On the left is a vertical orange sidebar with a menu containing: CUBO (logo), Cursos, Áreas de Conhecimento, Áreas Funcionais (highlighted), Unidades Curriculares, Usuários, and Sair. The main content area has a white background with an orange header bar that reads 'Atualizar Área Funcional'. Below this header is a form with the following fields: 'Nome da Área Funcional' (text input with 'TCC'), 'Descrição da Área Funcional' (text input), 'Área de Conhecimento' (dropdown menu with 'Atividades Integradas e Expandidas' selected), and a 'Salvar' button. At the bottom of the form are two buttons: 'Cancelar' and 'Salvar'.

## CONCLUSÕES E TRABALHOS FUTUROS

O projeto **Colmeia CUBO** atendeu aos requisitos iniciais, proporcionando uma solução digital eficiente para a gestão curricular da UniFil. Durante o desenvolvimento, foram consolidados conhecimentos em Laravel, Livewire e Alpine.js, além da experiência prática com MYSQL para gerenciamento de banco de dados e Git para controle de versão.

Para melhorias futuras, sugere-se a implementação de novas funcionalidades, como notificações automáticas para atualização de currículos e integração com outros sistemas acadêmicos. A experiência foi enriquecedora e o sistema desenvolvido contribuirá para a organização e acessibilidade das informações curriculares.

## REFERÊNCIAS

1. Colmeia CUBO. Disponível em: <https://unifil.br/cubo/>
2. Yggdrasil2. Disponível em: <https://akafts.github.io/yggdrasil2/>
3. Honeycomb - Hexagons. Disponível em: <https://abbekeultjes.nl/honeycomb/>
4. Documentação do Laravel. Disponível em: <https://laravel.com/docs/11.x>
5. Documentação do Livewire. Disponível em: <https://laravel-livewire.com/>
6. Documentação do Tailwind CSS. Disponível em: <https://tailwindcss.com/>

## APÊNDICE

### ARTEFATOS DE REQUISITOS

#### Documento de visão

#### Especificação Suplementar

#### Especificações de Caso de Uso:

- **UC001 GERENCIAR CURSOS**
- **UC002 GERENCIAR AREAS DE CONHECIMENTO**

- UC003 GERENCIAR AREAS FUNCIONAIS
- UC003 GERENCIAR AREAS DE CONHECIMENTO

Glossário

Pedido do Investidor

DIAGRAMAS

Diagrama de Caso de Uso

```
%%{init: {'theme': 'base'}}%%
graph TD
    Coordenador --> |Gerenciar Cursos| Curso
    Coordenador --> |Gerenciar Áreas de Conhecimento| AreaConhecimento
    Coordenador --> |Gerenciar Áreas Funcionais| AreaFuncional
    Coordenador --> |Gerenciar Unidades Curriculares| UnidadeCurricular
    Aluno --> |Visualizar Colmeia| Colmeia
    Coordenador --> |Acessar Colmeia| Colmeia

    classDef classStyle fill:#f9f,stroke:#333,stroke-width:2px;
    Curso:::classStyle
    AreaConhecimento:::classStyle
    AreaFuncional:::classStyle
    UnidadeCurricular:::classStyle
    Colmeia:::classStyle
```

Diagrama de Classes

```
classDiagram
    class Curso {
        +id: Int
        +nome: String
        +descricao: String
        +status: Boolean
        +index()
        +create()
        +store()
    }

    class AreaConhecimento {
        +id: Int
        +nome: String
        +cor: String
        +duracao: Int
        +status: Boolean
        +index()
```

```

        +create()
        +store()
    }

    class AreaFuncional {
        +id: Int
        +nome: String
        +descricao: String
        +duracao: Int
        +status: Boolean
        +index()
        +create()
        +store()
    }

    class UnidadeCurricular {
        +id: Int
        +nome: String
        +descricao: String
        +duracao: Int
        +areaFuncional: AreaFuncional
        +index()
        +create()
        +store()
    }

    Curso "1" -- "0..*" AreaConhecimento : gerencia
    AreaConhecimento "1" -- "0..*" AreaFuncional : define
    AreaFuncional "1" -- "0..*" UnidadeCurricular : organiza

```

## Diagrama de Entidade e Relacionamento

```

erDiagram
    CURSO {
        int id PK
        varchar nome
        varchar descricao
        boolean status
    }

    AREA_CONHECIMENTO {
        int id PK
        varchar nome
        varchar cor
        int duracao
        boolean status
    }

    AREA_FUNCIONAL {
        int id PK
    }

```

```

        varchar nome
        varchar descricao
        int duracao
        boolean status
    }

    UNIDADE_CURRICULAR {
        int id PK
        varchar nome
        varchar descricao
        int duracao
        int area_funcional_id FK
    }

    CURSO ||--o{ AREA_CONHECIMENTO : gerencia
    AREA_CONHECIMENTO ||--o{ AREA_FUNCIONAL : define
    AREA_FUNCIONAL ||--o{ UNIDADE_CURRICULAR : organiza

```

## Diagramas de Estado:

stateDiagram-v2

```

[*] --> CriandoCurso : Início do Cadastro de Curso
CriandoCurso --> CursoCriado : Curso salvo com sucesso
CriandoCurso --> ErroCurso : Falha no cadastro de Curso

```

```

CursoCriado --> EditandoCurso : Edição solicitada
CursoCriado --> AtivoCurso : Curso publicado
AtivoCurso --> InativoCurso : Curso desativado
InativoCurso --> AtivoCurso : Curso reativado
CursoCriado --> ExcluindoCurso : Excluir curso
ExcluindoCurso --> [*] : Curso excluído

```

```

[*] --> CriandoAreaConhecimento : Início do Cadastro de Área de Conhecimento
CriandoAreaConhecimento --> AreaConhecimentoCriada : Área de Conhecimento
salva com sucesso
CriandoAreaConhecimento --> ErroAreaConhecimento : Falha no cadastro de Área
de Conhecimento

```

```

AreaConhecimentoCriada --> EditandoAreaConhecimento : Edição solicitada
AreaConhecimentoCriada --> AtivoAreaConhecimento : Área de Conhecimento
publicada
AtivoAreaConhecimento --> InativoAreaConhecimento : Área de Conhecimento
desativada
InativoAreaConhecimento --> AtivoAreaConhecimento : Área de Conhecimento
reativada
AreaConhecimentoCriada --> ExcluindoAreaConhecimento : Excluir Área de
Conhecimento
ExcluindoAreaConhecimento --> [*] : Área de Conhecimento excluída

```

```

[*] --> CriandoAreaFuncional : Início do Cadastro de Área Funcional

```

```

CriandoAreaFuncional --> AreaFuncionalCriada : Área Funcional salva com
sucesso
CriandoAreaFuncional --> ErroAreaFuncional : Falha no cadastro de Área
Funcional

AreaFuncionalCriada --> EditandoAreaFuncional : Edição solicitada
AreaFuncionalCriada --> AtivoAreaFuncional : Área Funcional publicada
AtivoAreaFuncional --> InativoAreaFuncional : Área Funcional desativada
InativoAreaFuncional --> AtivoAreaFuncional : Área Funcional reativada
AreaFuncionalCriada --> ExcluindoAreaFuncional : Excluir Área Funcional
ExcluindoAreaFuncional --> [*] : Área Funcional excluída

[*] --> CriandoUnidadeCurricular : Início do Cadastro de Unidade Curricular
CriandoUnidadeCurricular --> UnidadeCurricularCriada : Unidade Curricular
salva com sucesso
CriandoUnidadeCurricular --> ErroUnidadeCurricular : Falha no cadastro de
Unidade Curricular

UnidadeCurricularCriada --> EditandoUnidadeCurricular : Edição solicitada
UnidadeCurricularCriada --> AtivoUnidadeCurricular : Unidade Curricular
publicada
AtivoUnidadeCurricular --> InativoUnidadeCurricular : Unidade Curricular
desativada
InativoUnidadeCurricular --> AtivoUnidadeCurricular : Unidade Curricular
reativada
UnidadeCurricularCriada --> ExcluindoUnidadeCurricular : Excluir Unidade
Curricular
ExcluindoUnidadeCurricular --> [*] : Unidade Curricular excluída

```

## Diagrama de Implantação

```

graph TD
    subgraph "Usuário"
        browser["Navegador Web"]
    end

    subgraph "Servidor"
        backend["Backend (Laravel)"]
        frontend["Frontend (Livewire + Alpine.js)"]
        database["Banco de Dados (PostgreSQL)"]
    end

    subgraph "Servidor do Banco de Dados"
        storage["Storage (Local/Cloud)"]
    end

    browser --> frontend["Requisição de Página Web"]
    frontend --> backend["Requisição de Dados"]
    backend --> database["Operações CRUD"]

```

```
backend --> storage["Armazenamento de Arquivos"]
backend --> frontend["Comunicação com Frontend via Livewire"]
```

### Diagramas de Sequência:

- [UC001\_GERENCIAR\_CURSOS]

- ```
sequenceDiagram
    actor Coordenador
    Coordenador ->> CursosController: acessar página de listagem de cursos
    CursosController ->> CursosService: obter lista de cursos
    CursosService ->> Database: consulta de cursos
    Database -->> CursosService: retorna lista de cursos
    CursosService -->> CursosController: retorna lista de cursos
    CursosController -->> Coordenador: exibe lista de cursos

    Coordenador ->> CursosController: solicitar criação de novo curso
    CursosController ->> CursosService: criar curso (dados do novo curso)
    CursosService ->> Database: inserir novo curso
    Database -->> CursosService: confirmação de criação
    CursosService -->> CursosController: confirmação de criação
    CursosController -->> Coordenador: curso criado com sucesso

    Coordenador ->> CursosController: solicitar edição de curso
    CursosController ->> CursosService: editar curso (dados atualizados)
    CursosService ->> Database: atualizar curso
    Database -->> CursosService: confirmação de atualização
    CursosService -->> CursosController: confirmação de atualização
    CursosController -->> Coordenador: curso atualizado com sucesso

    Coordenador ->> CursosController: solicitar exclusão de curso
    CursosController ->> CursosService: excluir curso
    CursosService ->> Database: remover curso
    Database -->> CursosService: confirmação de remoção
    CursosService -->> CursosController: confirmação de remoção
    CursosController -->> Coordenador: curso excluído com sucesso
```

- [UC002\_GERENCIAR\_AREAS\_DE\_CONHECIMENTO]

```
sequenceDiagram
    actor Coordenador
    Coordenador ->> AreasConhecimentoController: acessar página de listagem de áreas de conhecimento
    AreasConhecimentoController ->> AreasConhecimentoService: obter lista de áreas de conhecimento
    AreasConhecimentoService ->> Database: consulta de áreas de conhecimento
    Database -->> AreasConhecimentoService: retorna lista de áreas
```



```

AreasConhecimentoService -->> AreasConhecimentoController: retorna lista de
áreas
AreasConhecimentoController -->> Coordenador: exibe lista de áreas de
conhecimento

Coordenador -->> AreasConhecimentoController: solicitar criação de nova área
AreasConhecimentoController -->> AreasConhecimentoService: criar área de
conhecimento (dados da nova área)
AreasConhecimentoService -->> Database: inserir nova área
Database -->> AreasConhecimentoService: confirmação de criação
AreasConhecimentoService -->> AreasConhecimentoController: confirmação de
criação
AreasConhecimentoController -->> Coordenador: área de conhecimento criada com
sucesso

Coordenador -->> AreasConhecimentoController: solicitar edição de área de
conhecimento
AreasConhecimentoController -->> AreasConhecimentoService: editar área (dados
atualizados)
AreasConhecimentoService -->> Database: atualizar área
Database -->> AreasConhecimentoService: confirmação de atualização
AreasConhecimentoService -->> AreasConhecimentoController: confirmação de
atualização
AreasConhecimentoController -->> Coordenador: área de conhecimento atualizada
com sucesso

Coordenador -->> AreasConhecimentoController: solicitar exclusão de área de
conhecimento
AreasConhecimentoController -->> AreasConhecimentoService: excluir área
AreasConhecimentoService -->> Database: remover área
Database -->> AreasConhecimentoService: confirmação de remoção
AreasConhecimentoService -->> AreasConhecimentoController: confirmação de
remoção
AreasConhecimentoController -->> Coordenador: área de conhecimento excluída
com sucesso

```

## • [UC003\_GERENCIAR\_AREAS\_FUNCIONAIS]

```

sequenceDiagram
actor Coordenador
Coordenador -->> AreasFuncionaisController: acessar página de listagem de áreas
funcionais
AreasFuncionaisController -->> AreasFuncionaisService: obter lista de áreas
funcionais
AreasFuncionaisService -->> Database: consulta de áreas funcionais
Database -->> AreasFuncionaisService: retorna lista de áreas funcionais
AreasFuncionaisService -->> AreasFuncionaisController: retorna lista de áreas
AreasFuncionaisController -->> Coordenador: exibe lista de áreas funcionais

Coordenador -->> AreasFuncionaisController: solicitar criação de nova área

```

```

funcional
  AreasFuncionaisController ->> AreasFuncionaisService: criar área funcional
(dados da nova área)
  AreasFuncionaisService ->> Database: inserir nova área funcional
  Database -->> AreasFuncionaisService: confirmação de criação
  AreasFuncionaisService -->> AreasFuncionaisController: confirmação de criação
  AreasFuncionaisController -->> Coordenador: área funcional criada com sucesso

  Coordenador ->> AreasFuncionaisController: solicitar edição de área funcional
  AreasFuncionaisController ->> AreasFuncionaisService: editar área funcional
(dados atualizados)
  AreasFuncionaisService ->> Database: atualizar área funcional
  Database -->> AreasFuncionaisService: confirmação de atualização
  AreasFuncionaisService -->> AreasFuncionaisController: confirmação de
atualização
  AreasFuncionaisController -->> Coordenador: área funcional atualizada com
sucesso

  Coordenador ->> AreasFuncionaisController: solicitar exclusão de área
funcional
  AreasFuncionaisController ->> AreasFuncionaisService: excluir área funcional
  AreasFuncionaisService ->> Database: remover área funcional
  Database -->> AreasFuncionaisService: confirmação de remoção
  AreasFuncionaisService -->> AreasFuncionaisController: confirmação de remoção
  AreasFuncionaisController -->> Coordenador: área funcional excluída com
sucesso

```

## • [UC004\_GERENCIAR\_UNIDADES\_CURRICULARES]

```

sequenceDiagram
actor Coordenador
Coordenador ->> UnidadesCurricularesController: acessar página de listagem de
UCs
UnidadesCurricularesController ->> UnidadesCurricularesService: obter lista de
UCs
UnidadesCurricularesService ->> Database: consulta de UCs
Database -->> UnidadesCurricularesService: retorna lista de UCs
UnidadesCurricularesService -->> UnidadesCurricularesController: retorna lista
de UCs
UnidadesCurricularesController -->> Coordenador: exibe lista de UCs

Coordenador ->> UnidadesCurricularesController: solicitar criação de nova UC
UnidadesCurricularesController ->> UnidadesCurricularesService: criar UC
(dados da nova UC)
UnidadesCurricularesService ->> Database: inserir nova UC
Database -->> UnidadesCurricularesService: confirmação de criação
UnidadesCurricularesService -->> UnidadesCurricularesController: confirmação
de criação
UnidadesCurricularesController -->> Coordenador: UC criada com sucesso

Coordenador ->> UnidadesCurricularesController: solicitar edição de UC

```

```
UnidadesCurricularesController ->> UnidadesCurricularesService: editar UC
(dados atualizados)
UnidadesCurricularesService ->> Database: atualizar UC
Database -->> UnidadesCurricularesService: confirmação de atualização
UnidadesCurricularesService -->> UnidadesCurricularesController: confirmação
de atualização
UnidadesCurricularesController -->> Coordenador: UC atualizada com sucesso

Coordenador ->> UnidadesCurricularesController: solicitar exclusão de UC
UnidadesCurricularesController ->> UnidadesCurricularesService: excluir UC
UnidadesCurricularesService ->> Database: remover UC
Database -->> UnidadesCurricularesService: confirmação de remoção
UnidadesCurricularesService -->> UnidadesCurricularesController: confirmação
de remoção
UnidadesCurricularesController -->> Coordenador: UC excluída com sucesso
```

## Workflow AS-IS

```
graph TD
  A[Início] --> B[Login do Coordenador]
  B --> C[Selecionar Curso]
  C --> D[Gerenciar Áreas de Conhecimento]
  D --> E[Gerenciar Áreas Funcionais]
  E --> F[Gerenciar Unidades Curriculares]
  F --> G[Finalizar]
```

---

[Voltar](#)