

INF2102 - Documentação do Projeto

Sistema SGD - Sistema de Gerência de Disciplinas

Especificação do Programa

Escopo

O objetivo desse projeto é criar um sistema que poderá ser utilizado pela secretaria do departamento de informática para registrar e armazenar disciplinas, professores, turmas e currículos num sistema unificado para futura geração de informações e relatórios referentes a esses dados.

Hoje essas informações estão registradas em diferentes locais, alguns de acesso apenas pelo DAR, que deve ser requisitado pela secretaria para consulta. Muitas dessas informações hoje também se encontram em planilhas Excel que tornam difícil a consulta e geração de relatórios.

Com esse sistema, será possível em uma única plataforma consultar informações referentes a disciplinas, as turmas sendo lecionadas nesse e em outros períodos e por quais professores, os currículos existentes para diferentes cursos, os dados dos professores que hoje se encontram disponíveis apenas no site do departamento.

Dado que o sistema poderá conter dados sensíveis, ele também conta com um sistema de usuários, perfis e permissões que controla o acesso às suas diferentes funcionalidades.

Especificação dos Requisitos

Requisitos Funcionais:

- O sistema deve ser acessado somente por usuários autenticados.
- O usuário deve poder recuperar sua senha via email.
- O usuário deve apenas acessar o que seu perfil de acesso permite.
- Deve ser possível configurar diferentes perfis para usuários, cada um com suas regras de acesso.
- O menu de cada usuário deverá ser customizado a partir de suas permissões.
- Deve ser possível realizar operações CRUD (create, read, update, delete) para as seguintes tabelas: Disciplinas, Professores, Bibliografias, Currículos, Turmas.
- Deve ser possível visualizar Disciplinas e Currículos, de forma que se possa navegar pelas disciplinas e seus pré-requisitos.
- Para as disciplinas, deve-se registrar suas biografias e pré-requisitos, além de suas informações básicas (Título, Ementa, Nº créditos e código).
- Deve ser possível visualizar as turmas de um dado professor.

Requisitos Não-Funcionais:

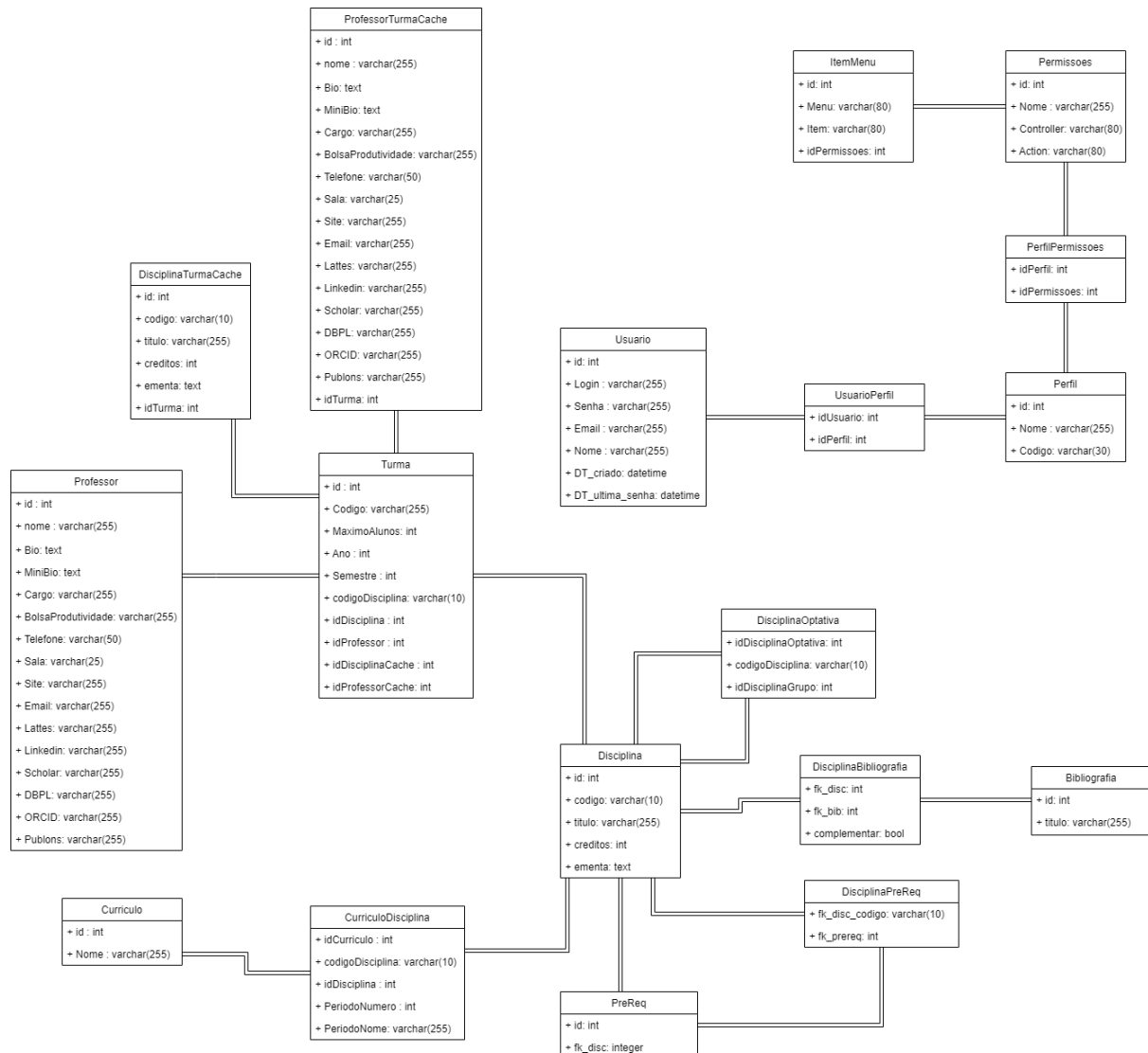
- O aplicativo será desenvolvido para web, portanto deve rodar em qualquer navegador.
- O banco deve ser desenvolvido em MySQL 8.0
- O aplicativo será desenvolvido em Laravel 9.0.
- O aplicativo será desenvolvido usando Bootstrap 5 como frontend toolkit.

Projeto do Programa

Arquitetura do Banco

O banco será implementado em MySQL. Ele tem duas partes separadas, as tabelas referentes ao sistema de usuário e as tabelas referentes aos dados do DI.

Para as tabelas do sistema de usuário, temos:



Arquitetura do Sistema

O sistema é desenvolvido em laravel 9.0, portanto segue a arquitetura desse framework.

Temos 3 principais tipos de arquivos que serão utilizados para a criação do sistema, esses são: Controller, Models e Views. Arquivos deste tipo serão criados para cada tabela principal do banco, e são responsáveis pelas funcionalidades do sistema.

Além desses, temos arquivos específicos de roteamento (web.php), configurações (app.php, database.php, mail.php e .env) além dos middlewares que serão utilizados (Authenticate.php) e um que será criado (Acesso.php).

Controllers

Controllers são arquivos .php responsáveis pelas funcionalidades backend do sistema, interagindo com o banco de dados e passando as variáveis necessárias para as views. Esse arquivo contém uma classe php que estende a superclasse Controller definida pelo laravel.

Dentro dessa classe é definido um __construct() que inicializa os middleware a serem utilizados pela classe. São definidas também as funções que são chamadas pelas rotas definidas no arquivo web.php.

Em geral serão definidas duas funções para uma mesma url, uma que lida com as rotas do tipo get e outra com os tipo post. As funções que lidam com request tipo post terão o mesmo nome das suas funções tipo get com o sufixo _post. É possível que uma única função lide com ambos get e post, como é o caso dos index's, para tratar com filtros e paginação.

Os controller têm acesso ao banco através dos Models, que são definidos em outro arquivo e utilizam o Eloquent ORM do laravel para construir as queries. Com exceção de funções auxiliares, toda função de um arquivo tipo controller deve retornar uma view para ser renderizada para o cliente ou redirecionar para outra URL passando parâmetros.

Models

Models são arquivos .php que são responsáveis pela definição das características de cada tabela para a criação de Eloquent ORM do laravel. Esse arquivo contém uma classe que estende a superclasse Model (ou Authenticatable no caso de usuários).

Dentro dessa classe são definidas as seguintes variáveis:

- \$table - Uma string que contém o nome da tabela do banco..
- \$fillable - Uma array que contém os campos que podem ser preenchidos em massa pelo Eloquent ORM.
- \$sortable - Uma array que contém os campos da tabela que podem ser ordenados.
- \$hidden - Uma array que contém os campos que devem ser escondidos para serialização.

Além dessas variáveis são também definidas funções que definem os relacionamentos entre as tabelas. Essas são funções públicas que retornam métodos de relacionamento definidos pelo Laravel. Esses são: hasOne, hasMany, belongsToMany, belongTo.

Views

View são arquivos .blade.php que são responsáveis pela geração de código HTML para renderização no navegador do cliente.

Esses arquivos utilizam a extensão .blade que é uma template engine incluída no Laravel. Esse template permite a inserção de código php através de {{ código php }}

 diretamente no arquivo.

Além disso é possível definir um arquivo template a ser utilizado em cada view, que é inicializado com @extends('path para o arquivo'). No arquivo template podemos definir sessões para serem preenchidas pelo arquivo que utilizará o template usando @section('nome da section'). Os arquivos template também são do tipo .blade.php

Web.php

Esse arquivo é responsável pelo roteamento das URL's. É um arquivo php contendo métodos do tipo:

```
Route::get('URL', [Controller, 'função']);
```

```
Route::post('URL', [Controller, 'função']);
```

Quando o URL requisitado da match com algum desses métodos a função do controller especificada é chamada e lida com o request.

Acesso.php

Middleware criado para lidar com o sistema de acesso de usuários. Cada usuário pode ter diversos perfis que por sua vez tem podem ter diversas permissões. Essas são contém um controller e action.

O arquivo Acesso.php contém uma função handle que é executada antes das funções de Controllers e testa se o usuário que está logado no sistema tem algum perfil com a permissão que corresponda ao controller e ação sendo requisitado. Caso não tenha, redirecione o usuário para a página de onde veio, levantando um erro de falta de acesso.

Dicionário de Dados

Usuario

Descrição:

Tabela para armazenar usuários registrados no sistema. Campo senha deve ser armazenado criptografado no banco. DT_ultima_senha deve ser atualizada sempre que a senha é alterada.

Itens de dados relacionados:

Perfil - através da tabela UsuarioPerifl

Faixa de Valores:

```
id int not null auto_increment,  
Login varchar(255) unique not null,  
Senha varchar(255) not null,  
Email varchar(255) unique not null,  
Nome varchar(255),  
DT_criado datetime not null default now(),
```

DT_ultima_senha datetime not null default now(),

Perfil

Descrição:

Tabela para armazenar perfis registrados no sistema. O perfil deve definir quais funcionalidades do sistema o usuário tem acesso. Cada ação do sistema é definida pela tabela Permissões, e um perfil pode ter (n) permissões associadas a ele. Um usuário pode ter (n) perfis.

Itens de dados relacionados:

Usuário - através da tabela UsuarioPerifl

Permissão - através da tabela PerfilPermissoes

Faixa de Valores:

id int not null auto_increment,
Nome varchar(100) not null,
Codigo varchar(30) not null,

Permissoes

Descrição:

Tabela para armazenar permissões registradas no sistema. Uma permissão define uma ação do sistema. Isso é feito a partir dos campos controller e action, onde o Controller representa o controller e Action representa a função dentro do controller. Obs: é importante que os valores de Controller e Action sejam iguais aos nomes definidos no código.

Itens de dados relacionados:

Perfil - através da tabela PerfilPermissoes

ItemMenu - através da FK idPermissoes

Faixa de Valores:

id int not null auto_increment,
Nome varchar(255) not null,
Controller varchar(80) not null,
Action varchar(80) not null,

ItemMenu

Descrição:

Tabela para armazenar os itens do menu registrados no sistema. Cada item menu se refere a uma permissão, onde a ação desta permissão é a função do item menu. Para um item menu único, o campo Menu deve ser deixado em branco. Para um item menu cascata, o campo Menu define o nome do menu que agrupa os itens (itens com mesmo nome de Menu serão agrupados em um único item cascata)..

Itens de dados relacionados:

Permissão - através da FK idPermissoes

Faixa de Valores:

Menu varchar(80),
Item varchar(80) not null,
id int not null auto_increment,
idPermissoes int not null,

UsuarioPefil

Descrição:

Tabela de relacionamentos entre usuário e perfil.

Itens de dados relacionados:

Usuário - através da FK idUsuario

Perfil - através da FK idPerfil

Faixa de Valores:

idUsuario int not null,

idPerfil int not null,

PerfilPermissoes

Descrição:

Tabela de relacionamentos entre perfil e permissões.

Itens de dados relacionados:

Perfil - através da FK idPerfil

Permissao - através da FK idPermissoes

Faixa de Valores:

idPerfil int not null,

idPermissoes int not null,

Disciplina

Descrição:

Tabela para armazenar disciplinas registradas no banco. Uma disciplina pode ter (n) bibliografias.

A relação com a tabela PreReq define um grupo de pré-requisitos da disciplina, onde todas as disciplinas desse grupo devem ser cumpridas. Uma disciplina pode ter (n) PreReq relacionados (basta um grupo de pre-requisitos ser cumpridos para cursar a disciplina).

A relação com a tabela DisciplinaPreReq define se a disciplina pertence a um grupo de pré-requisitos, isso é, se ela é pré-requisito de alguma disciplina.

A relação com a tabela DisciplinaOptativa define o grupo de disciplinas que compõem a optativa. Essa tabela só é utilizada no caso da disciplina ser uma optativa, que é representada pelo campo Optativa.

Itens de dados relacionados:

Bibliografia - através da tabela DisciplinaBibliografia

PreReq - define o grupo de pré-requisitos de uma disciplina

DisciplinaPreReq - tabela de relacionamento

DisciplinaOptativa - tabela de relacionamento

Faixa de Valores:

id int UNIQUE NOT NULL AUTO_INCREMENT,

Codigo varchar(10),

Titulo varchar(255),

Creditos int,

Ementa text,
Optativa bool default false,

Bibliografia

Descrição:

Tabela para armazenar bibliografias registradas no banco.

Itens de dados relacionados:

Disciplina - através da tabela DisciplinaBibliografia

Faixa de Valores:

id int UNIQUE NOT NULL AUTO_INCREMENT,
Titulo varchar(255),

DisciplinaBibliografia

Descrição:

Tabela de relacionamentos entre Disciplina e Bibliografia. O campo Complementar define se a bibliografia é ou não complementar de uma dada disciplina.

Itens de dados relacionados:

Disciplina - através da FK idDisciplina

Bibliografia - através da FK idBibliografia

Faixa de Valores:

idDisciplina int NOT NULL,
idBibliografia int NOT NULL,
Complementar bool,

DisciplinaPreReq

Descrição:

Tabela de relacionamentos entre Disciplina e PreReq. Para obter o grupo de disciplinas que compõem o pré-requisito, pegar a lista de idDisciplinas com um mesmo idPreReq.

Para obter a disciplina que tem o grupo de disciplinas como pré-requisito, pegar o idDisciplina na tabela PreReq.

Itens de dados relacionados:

Disciplina - através da FK idDisciplina

PreReq - através da FK idPreReq

Faixa de Valores:

codigoDisciplina varchar(10),
idDisciplina int,
idPreReq int NOT NULL,

PreReq

Descrição:

Tabela para armazenar PreReq registrados no banco. Um PreReq define um grupo de disciplinas que são pré-requisitos de outra. A disciplina apontada pela FK idDisciplina é a “dona” do PreReq, e as disciplinas do grupo são definidas pela tabela DisciplinaPreReq

Itens de dados relacionados:

O campo ListaCodigos é um campo auxiliar que tem a lista de códigos das disciplinas do grupo separados por vírgulas. Ele deve ser gerado dinamicamente na criação e atualização de um PreReq.

O campo NumeroCreditos define se o pré-requisito tem um número de créditos mínimo como requisito. O PreReq pode ter um grupo de disciplinas e um número mínimo de créditos.

Um PreReq pertence a uma única disciplina (a “dona” do pré-requisito) e pode se relacionar com (n) disciplinas através da tabela DisciplinaPreReq, essas compondo o grupo de requisitos da “dona”.

Disciplina - através da FK idDisciplina

DisciplinaPreReq - tabela de relacionamento

Faixa de Valores:

id int UNIQUE NOT NULL AUTO_INCREMENT,
idDisciplina int NOT NULL,
ListaCodigos varchar(255),
NumeroCreditos int default 0,

DisciplinaOptativa

Descrição:

Tabela de relacionamento de Disciplinas com Disciplinas,

A disciplina referente ao campo idDisciplinaOptativa é a “dona” do grupo de optativas (isso é, a disciplina optativa, que tem um grupo de disciplinas como opções).

A disciplina referente ao campo idDisciplinaGrupo, é parte do grupo de disciplinas da optativa.

O campo codigoDisciplina se refere ao código da disciplina definida pelo campo idDisciplinaGrupo.

Para obter o grupo de disciplinas de uma optativa, pegar a lista de idDisciplinaGrupo com um mesmo idDisciplinaOptativa .

Itens de dados relacionados:

Disciplina - através da FK idDisciplinaOptativa e idDisciplinaGrupo

Faixa de Valores:

idDisciplinaOptativa int not null,
codigoDisciplina varchar(10),
idDisciplinaGrupo int,

Professor

Descrição:

Tabela para armazenar professores registrados no banco.

Os campos da tabela armazenam dados referentes ao professor, os mesmos que podem ser encontrados no Site do departamento de informática.

Um professor pode ter (n) turmas. A turma é uma relação entre um professor e uma disciplina.

Itens de dados relacionados:

Turma - através da tabela Turma

Faixa de Valores:

id int not null auto_increment,
Nome varchar(255) not null,
Bio text,
MiniBio text,
Cargo varchar(255),
BolsaProdutividade varchar(255),
Telefone varchar(50),
Sala varchar(25),
Site varchar(255),
Email varchar(255),
Lattes varchar(255),
Linkedin varchar(255),
Scholar varchar(255),
DBPL varchar(255),
ORCID varchar(255),
Publons varchar(255),

Curriculo

Descrição:

Tabela para armazenar currículos registrados no banco.

Um currículo contém uma lista de disciplinas que o compõem. A relação com as disciplinas de um dado currículo é feita pela tabela CurriculoDisciplina, que contém informações sobre o período da disciplina no currículo.

Itens de dados relacionados:

Disciplina - através da tabela CurriculoDisciplina

Faixa de Valores:

id int not null auto_increment,
Nome varchar(255) not null,

Turma

Descrição:

Tabela para armazenar turmas registradas no banco.

Essa tabela cria uma relação entre professor e disciplina, que representa uma turma. As turmas têm um ano e semestre nas quais são lecionadas e o código da turma, além do número máximo de alunos.

As tabelas ProfessorCache e DisciplinaCache servem para gerar um espelho dos dados do professor e disciplina no momento de criação da turma. Assim, caso os dados do professor ou da disciplina venham a ser alterados no sistema, poderemos ainda ter um

histórico dos dados. Os valores armazenados nas tabelas de Cache podem ser alterados na edição da turma.

Caso uma turma tenha o professor ou a disciplina alterada, os cachês são atualizados para refletir a mudança.

Pode-se criar uma turma com o idProfessor = 0 para deixar a turma com professor a definir.

Itens de dados relacionados:

Professor - através da FK idProfessor

Disciplina - através da FK idDisciplina

ProfessorCache - através da FK idProfessorCache

DisciplinaCache - através da FK idDisciplinaCache

Faixa de Valores:

id int not null auto_increment,

Codigo varchar(255) not null,

MaximoAlunos int,

Ano int,

Semestre int,

codigoDisciplina varchar(10) not null,

idDisciplina int,

idProfessor int,

idDisciplinaCache int,

idProfessorCache int,

CurriculoDisciplina

Descrição:

Tabela de relacionamentos entre Disciplina e Curriculo. Os campos PeriodoNumero e PeriodoNome definem em que período do currículo a disciplina se encontra.

Itens de dados relacionados:

Curriculo - através da FK idCurriculo

Disciplina - através da FK idDisciplina

Faixa de Valores:

codigoDisciplina varchar(10) not null,

idCurriculo int not null,

idDisciplina int,

PeriodoNumero int,

PeriodoNome varchar(255),

ProfessorTurmaCache

Descrição:

Tabela que armazena o espelho dos dados do professor na criação de uma Turma. Os dados dessa tabela podem ser alterados na edição de uma turma e são registrados automaticamente na criação de uma turma, com os dados do professor no momento da criação.

Itens de dados relacionados:

Turma - através da FK idTurma

Faixa de Valores:

id int not null auto_increment,
Nome varchar(255) not null,
Bio text,
MiniBio text,
Cargo varchar(255),
BolsaProdutividade varchar(255),
Telefone varchar(50),
Sala varchar(25),
Site varchar(255),
Email varchar(255),
Lattes varchar(255),
Linkedin varchar(255),
Scholar varchar(255),
DBPL varchar(255),
ORCID varchar(255),
Publons varchar(255),
idTurma int

DisciplinaTurmaCache

Descrição:

Tabela que armazena o espelho dos dados da disciplina na criação de uma Turma. Os dados dessa tabela podem ser alterados na edição de uma turma e são registrados automaticamente na criação de uma turma, com os dados da disciplina no momento da criação.

Esse espelho não armazena as relações de Pré-Requisitos e Bibliografias, apenas os dados da disciplina.

Itens de dados relacionados:

Turma - através da FK idTurma

Faixa de Valores:

id int UNIQUE NOT NULL AUTO_INCREMENT,
Codigo varchar(10),
Titulo varchar(255),
Créditos int,
Ementa text,
Optativa bool default false,
idTurma int

Padrões de Programação Adotados

Não foi adotado nem um padrão específico de programação.

Roteiro de Testes

Critérios de Testes

O sistema contará apenas com testes manuais. Devem ser testadas todas as funcionalidades:

- A listagem da tabela, incluindo seus filtros e paginação.
- A inserção de um novo elemento, testando os casos de campos em branco e valores incorretos para um dado campo.
- A edição de um elemento, seguindo os mesmos princípios da inserção.
- A remoção de um elemento, verificando no banco de dados se tabelas de relacionamentos foram devidamente removidas.
- No caso de disciplinas e currículos, verificar se a visualização de ambos está de acordo com os valores do banco de dados, e se a navegação pelas disciplinas está correta.
- Para usuários, verificar se o menu está sendo propriamente construído dado as permissões do usuário.
- Verificar se um usuário sem permissão a uma dada funcionalidade consegue acessá-la
- Verificar a autenticação do usuário

O sistema foi amplamente testado e está de acordo com os requisitos apresentados neste documento.

Laudos dos Testes

Teste de CRUD

Os testes apresentados abaixo são da tabela Permissões. Foram efetuados os mesmo testes para todas as tabelas.

Teste de Listagem

Listagem sem filtros

Permissões

Nova Permissão

Filtro	Filtro	Categoria	Nome	▼	Filtrar	Resetar
#	Nome ↕	Controller ↕	Action	Opções		
1	Dashboard	dashboard	index	Editar	Apagar	
2	Ver usuarios	usuarios	index	Editar	Apagar	
3	Adicionar usuarios	usuarios	add	Editar	Apagar	
4	Editar usuarios	usuarios	edit	Editar	Apagar	
5	Apagar usuarios	usuarios	delete	Editar	Apagar	
6	Ver Perfil	perfil	index	Editar	Apagar	
7	Adicionar Perfil	perfil	add	Editar	Apagar	
8	Editar Perfil	perfil	edit	Editar	Apagar	
9	Apagar Perfil	perfil	delete	Editar	Apagar	
10	Ver Permissoes	permissoes	index	Editar	Apagar	

Previous12345Next

Filtrando por nome,utilizando “ver” como filtro.

Permissões

Nova Permissão

Filtro	Ver	Categoria	Nome	▼	Filtrar	Resetar
#	Nome ↕	Controller ↕	Action	Opções		
2	Ver usuarios	usuarios	index	Editar	Apagar	
6	Ver Perfil	perfil	index	Editar	Apagar	
10	Ver Permissoes	permissoes	index	Editar	Apagar	
14	Ver Item Menu	itemMenu	index	Editar	Apagar	
22	Ver Disciplinas	disciplina	index	Editar	Apagar	
23	Ver Bibliografias	bibliografia	index	Editar	Apagar	
27	Ver Professores	professor	index	Editar	Apagar	
31	Ver Curriculos	curriculo	index	Editar	Apagar	
35	Ver Turmas do Professor	professor	turmas	Editar	Apagar	
41	Ver Turmas	turma	index	Editar	Apagar	

Ordenando por Controller, em ordem ascendente alfabética

Permissões

Nova Permissão

Filtro	Ver	Categoria	Nome	▼	Filtrar	Resetar
#	Nome	Controller	Action	Opções		
23	Ver Bibliografias	bibliografia	index	Editar	Apagar	
31	Ver Curriculos	curriculo	index	Editar	Apagar	
22	Ver Disciplinas	disciplina	index	Editar	Apagar	
14	Ver Item Menu	itemMenu	index	Editar	Apagar	
6	Ver Perfil	perfil	index	Editar	Apagar	
10	Ver Permissoes	permissoes	index	Editar	Apagar	
27	Ver Professores	professor	index	Editar	Apagar	
35	Ver Turmas do Professor	professor	turmas	Editar	Apagar	
41	Ver Turmas	turma	index	Editar	Apagar	
2	Ver usuarios	usuarios	index	Editar	Apagar	

Navegando pela paginação

Permissões

Nova Permissão

Filtro	Filtro	Categoria	Nome	▼	Filtrar	Resetar
#	Nome	Controller	Action	Opções		
25	Editar Bibliografia	bibliografia	edit	Editar	Apagar	
26	Apagar Bibliografia	bibliografia	delete	Editar	Apagar	
27	Ver Professores	professor	index	Editar	Apagar	
28	Adicionar Professor	professor	add	Editar	Apagar	
29	Editar Professor	professor	edit	Editar	Apagar	
30	Apagar Professor	professor	delete	Editar	Apagar	
31	Ver Curriculos	curriculo	index	Editar	Apagar	
32	Adicionar Curriculos	curriculo	add	Editar	Apagar	
33	Editar Curriculos	curriculo	edit	Editar	Apagar	
34	Apagar Curriculos	curriculo	delete	Editar	Apagar	

Previous

1

2

3

4

5

Next

Teste de Inserção

Tentativa de inserção com campo obrigatório vazio.

Nova Permissão

Nome

Controller

Please fill out this field.

Action

Voltar

Salvar

Inserção com campos preenchidos corretamente

Nova Permissão

Nome

Teste

Controller

professor

Action

add

Voltar

Salvar

Dashboard

×

Permissao registrada com sucesso!

Permissões

Nova Permissão

Filtro	Filtro	Categoria	Nome		Filtrar	Resetar
#	Nome	Controller	Action	Opções		
45	Adicionar Disciplina	disciplina	add	<div>EditarApagar</div>		
46	Deletar Disciplina	disciplina	delete	<div>EditarApagar</div>		
47	Editar Disciplina	disciplina	edit	<div>EditarApagar</div>		
48	Teste	professor	add	<div>EditarApagar</div>		

Previous

1

2

3

4

5

Next

Teste de Edição

Editando permissão teste, criada no teste acima.

Permissões

Filtro

Filtro

Categoria

Nome

Filtrar

Resetar

#

Nome

Controller

Action

Opções

45

Adicionar Disciplina

disciplina

add

Editar

Apagar

46

Deletar Disciplina

disciplina

delete

Editar

Apagar

47

Editar Disciplina

disciplina

edit

Editar

Apagar

51

teste

professor

add

Editar

Apagar

Previous

1

2

3

4

5

Next

Teste de edição com campo obrigatório vazio.

Editar Permissão - Teste

Nome

Controller

professor

!

Please fill out this field.

Action

add

Voltar

Salvar

Edição da permissão alterando o nome.

Editar Permissão - teste

Nome

teste2

Controller

professor

Action

add

Voltar

Salvar

Dashboard

×

Permissao atualizada com sucesso!

Permissões

Nova Permissão

Filtro	Filtro	Categoria	Nome		Filtrar	Resetar
#	Nome	Controller	Action	Opções		
45	Adicionar Disciplina	disciplina	add	Editar	Apagar	
46	Deletar Disciplina	disciplina	delete	Editar	Apagar	
47	Editar Disciplina	disciplina	edit	Editar	Apagar	
51	teste2	professor	add	Editar	Apagar	

Previous

1

2

3

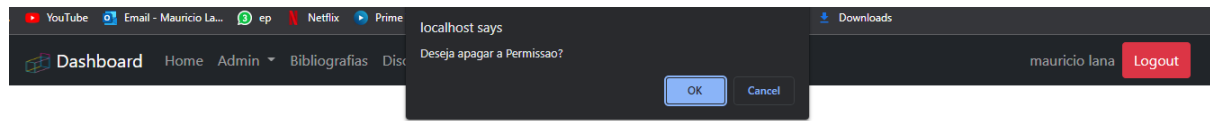
4

5

Next

Teste de Remoção

Efetuating a remoção da permissão criada.

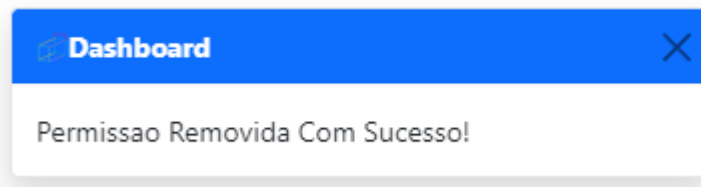


Permissões

[Nova Permissão](#)

Filtro	Filtro	Categoria	Nome		Filtrar	Resetar
#	Nome	Controller	Action	Opções		
45	Adicionar Disciplina	disciplina	add	Editar	Apagar	
46	Deletar Disciplina	disciplina	delete	Editar	Apagar	
47	Editar Disciplina	disciplina	edit	Editar	Apagar	
51	teste2	professor	add	Editar	Apagar	

[Previous](#) [1](#) [2](#) [3](#) [4](#) [5](#) [Next](#)



Permissões

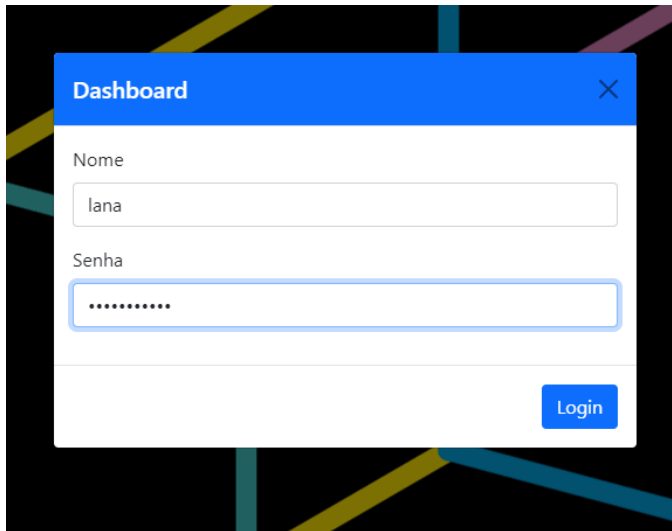
[Nova Permissão](#)

Filtro	Filtro	Categoria	Nome		Filtrar	Resetar
#	Nome	Controller	Action	Opções		
45	Adicionar Disciplina	disciplina	add	Editar	Apagar	
46	Deletar Disciplina	disciplina	delete	Editar	Apagar	
47	Editar Disciplina	disciplina	edit	Editar	Apagar	

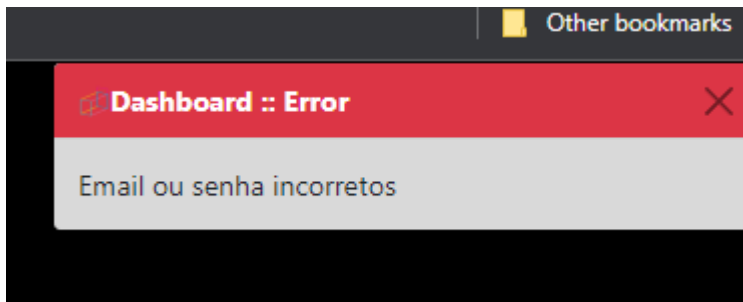
[Previous](#) [1](#) [2](#) [3](#) [4](#) [5](#) [Next](#)

Teste de Autenticação

Tentativa de acesso com a senha incorreta

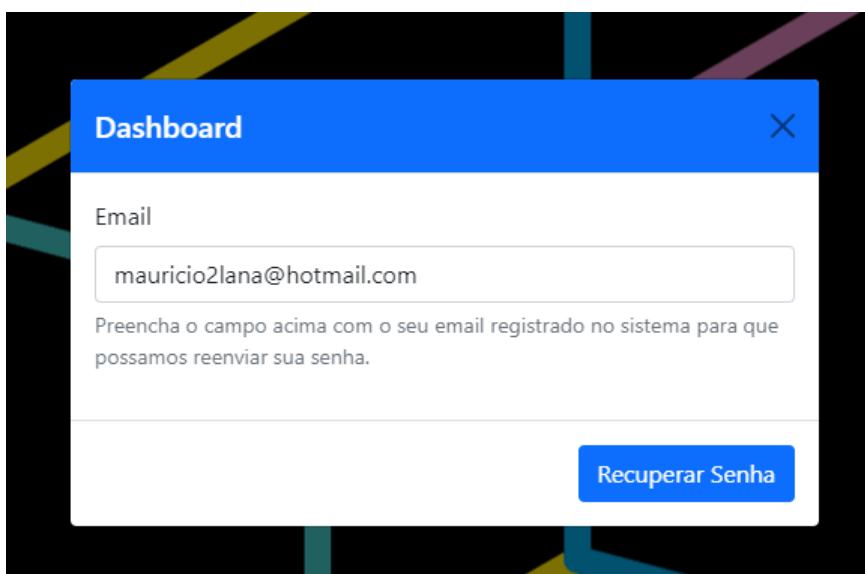


A screenshot of a web application's login interface. It features a white modal window titled "Dashboard" with a close button (X) in the top right corner. Inside the modal, there are two input fields: "Nome" (Name) containing the text "lana" and "Senha" (Password) which is masked with eight dots. A blue "Login" button is positioned at the bottom right of the modal. The background of the application is dark with abstract colorful lines.

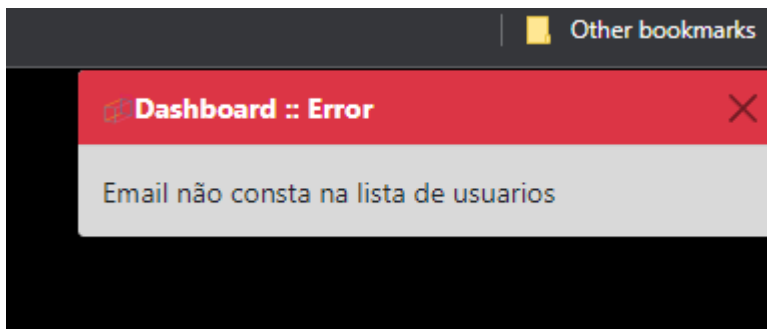


Teste de recuperação da senha

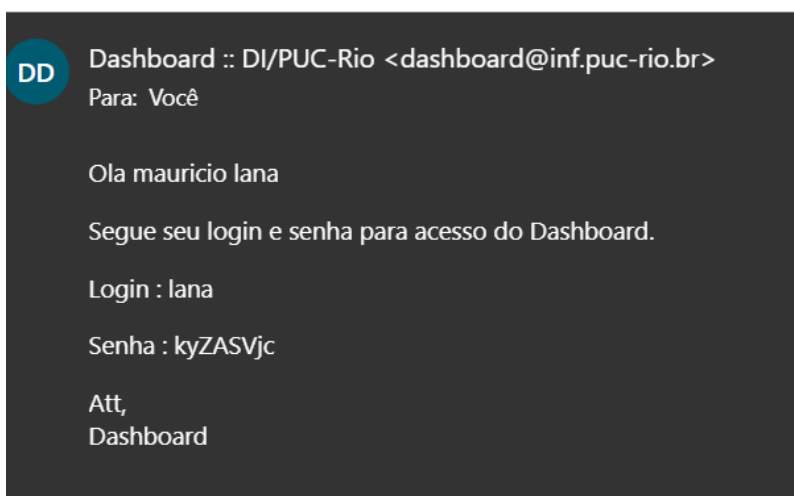
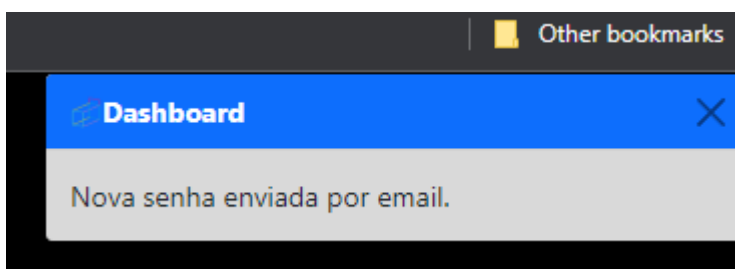
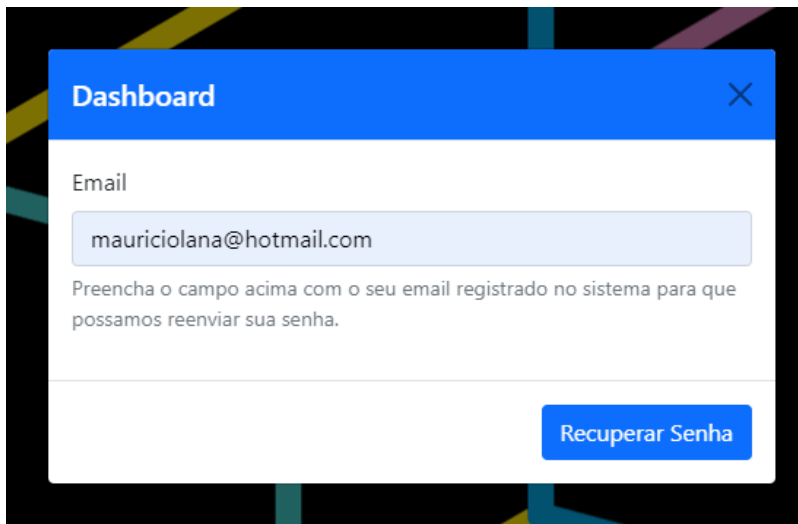
Passando um email incorreto



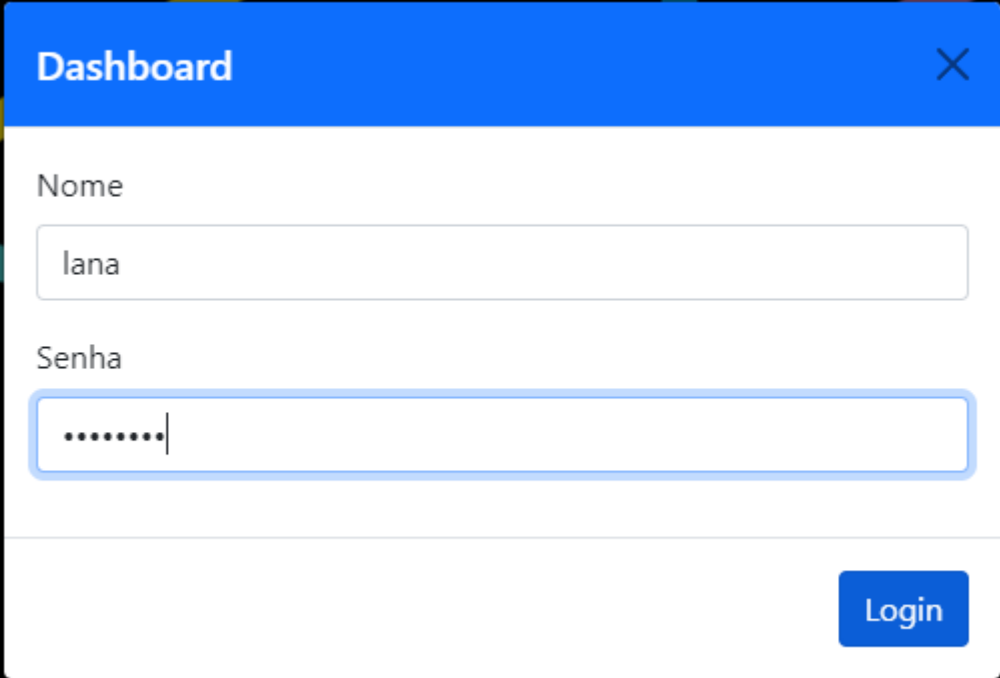
A screenshot of a web application's password recovery interface. It features a white modal window titled "Dashboard" with a close button (X) in the top right corner. Inside the modal, there is an "Email" input field containing the text "mauricio2lana@hotmail.com". Below the input field, there is a text instruction: "Preencha o campo acima com o seu email registrado no sistema para que possamos reenviar sua senha." (Fill the field above with your email registered in the system so we can resend your password). A blue "Recuperar Senha" (Recover Password) button is located at the bottom right of the modal. The background of the application is dark with abstract colorful lines.



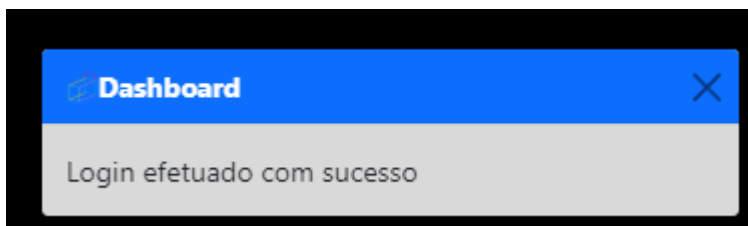
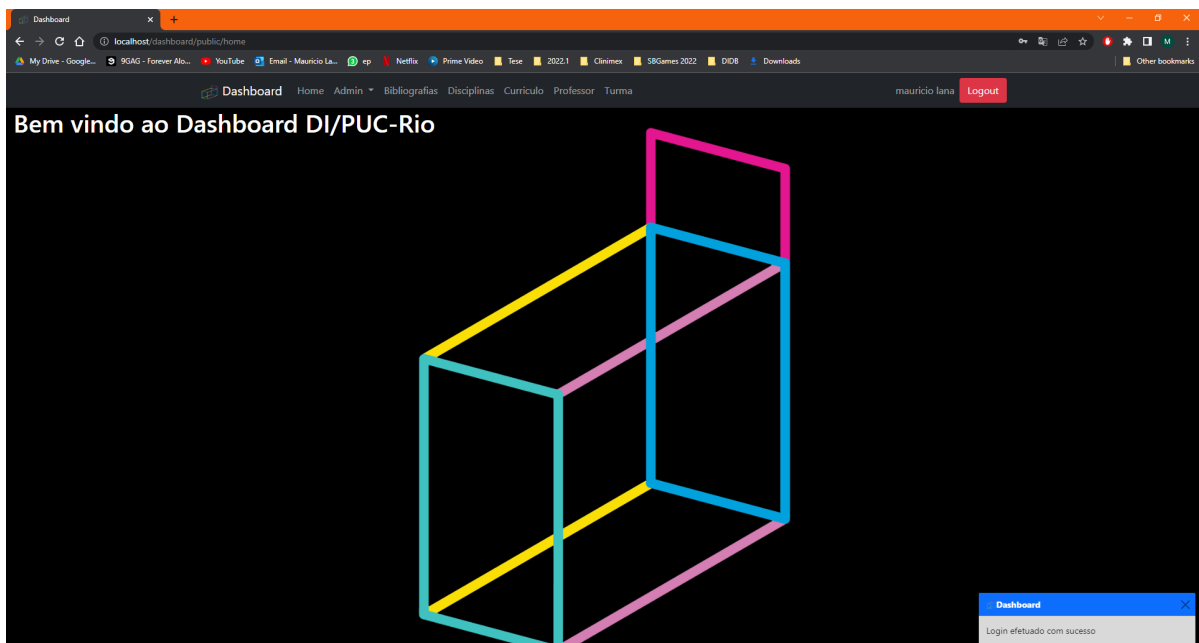
Passando o email registrado ao usuário



Acessando o sistema com a senha recuperada

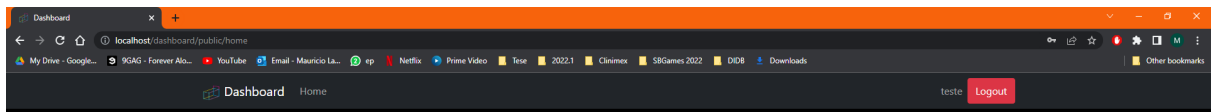


A login modal window titled "Dashboard" with a close button (X) in the top right corner. It contains two input fields: "Nome" with the text "lana" and "Senha" with masked characters ".....". A blue "Login" button is located at the bottom right of the modal.

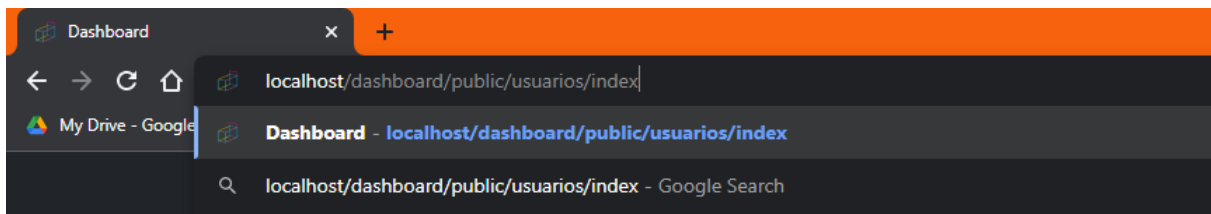


Teste de Acesso

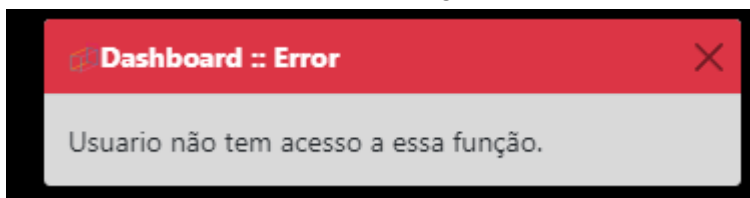
Login efetuado em usuário com permissão restrita (Pode acessar apenas a home).
O menu do usuário tem apenas a opção home, que é sua única permissão



Inserindo url para página que o usuário não tem acesso



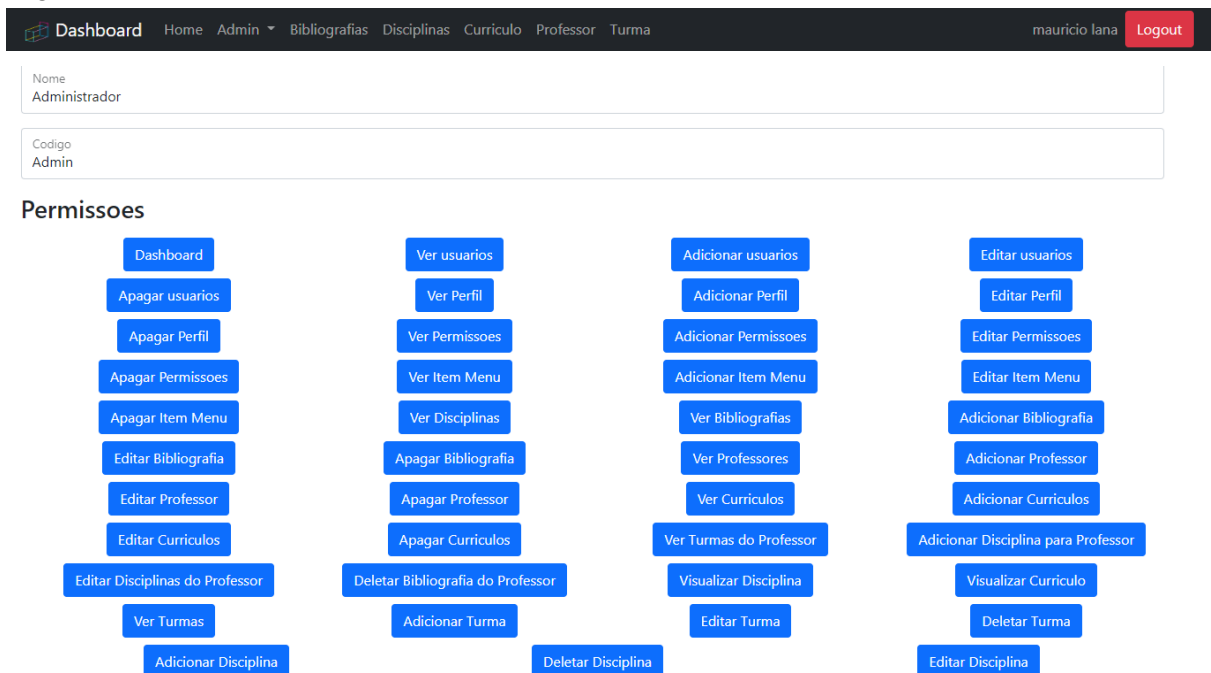
Retorna para a home com mensagem de erro



Teste de Menu

Testando a geração dos Menus

Login efetuado em usuário com todas as permissões. Menu completo com todos os itens.



Editando o perfil do usuário (Administrador) e removendo a permissão para ver currículos.
(Item menu Currículos referente a ação de ver currículos)

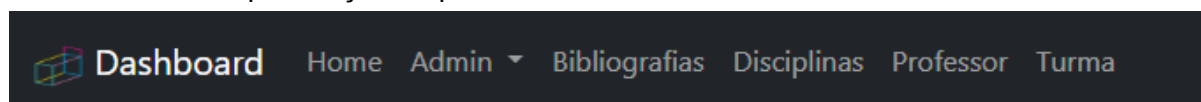
Editar Perfil - Administrador

Nome	Administrador
Código	Admin

Permissoes



Menu do usuário após edição do perfil.



Item Currículo não aparece no menu, pois usuario não tem acesso a função ver Currículos que está associada ao ItemMenu Currículos

Editar Item Menu - ### > Currículo

Menu
Item
Currículo
Ver Currículos
Voltar
Salvar

Documentação para Usuário

Esse sistema tem como usuário alvo a secretaria do departamento de informática. O objetivo principal é criar um ambiente onde se possa agrupar informações que hoje se encontram dispersas. Apesar de criar um trabalho grande inicial para povoar o banco de dados, a longo prazo o sistema virá a apresentar grandes melhoras para o trabalho de geração de relatórios e obtenção de informações referentes ao escopo do sistema.

O sistema ainda será posto em produção, e uma vez que estiver no ar não será necessário instalação. Como o sistema se trata de um site, bastará que os usuários acessem o link e tenham uma conta criada para acessar o sistema.

Uma vez que o sistema estiver no ar iremos continuar a adicionar funcionalidades e melhorar a interface onde for necessário. Uma vez que o usuário é a própria secretaria do departamento, o contato com os desenvolvedores será de fácil acesso.

O desenvolvimento do sistema continuará após o término desse trabalho, que tem como intenção criar a base inicial para que possa começar a ser utilizado, assim validando que o sistema trará benefícios.