



Instituto Politécnico de Viseu
Escola Superior de Tecnologia e Gestão de Viseu
Departamento de Informática

Unidade Curricular: Aplicações para a Internet II

Relatório Trabalho Prático de AI2

Tema: Criação de uma aplicação Web para o Projeto Integrado

Realizado por: Guilherme Bento - 25193

Tiago Portugal - 30816

Lucas Santos - 27440

César Cabral - 6802

Viseu, 2025

Instituto Politécnico de Viseu
Escola Superior de Tecnologia e Gestão de Viseu
Departamento de Informática

Relatório relativo ao Trabalho Prático de AI2

Curso de Licenciatura em Engenharia Informática

Unidade Curricular de Aplicações para a Internet II

Trabalho Prático de AI2

Ano Letivo 2024/25

Viseu, 2025

Índice

Introdução	6
Objetivos	7
Arquitetura da Aplicação Web	8
Desenvolvimento do Backend	9
Estrutura do Projeto	9
Funcionalidade principais da API	9
Exemplo Real de Endpoint REST	10
Exemplo de Rota Protegida para Listar Utilizadores	10
Integração com a Base de Dados (PostgreSQL)	11
Exemplo da criação de uma tabela	11
Segurança e Autenticação	12
Exemplo Tratamento de Erros e Autenticação	12
Validação e Tratamento de Erros	12
Integração com Serviços Externos	13
Integração com Azure Blob Storage	13
Integração com Firebase Cloud Messaging (FCM)	13
Integração com Mailgun (Envio de Emails)	14
Geração Automática de Modelos com Sequelize-Auto	14
Desenvolvimento do FrontEnd	15
Descrição geral da stack e ferramentas usadas	15
Estrutura do FrontEnd	15
Autenticação no FrontEnd	15
Criação dos formulários	15
FrontOffice	16
Descrição das Views do FrontOffice	16
BackOffice	26
Descrição das views do BackOffice	26

<i>Conclusões</i>	36
<i>Bibliografia</i>	37

Índice de Ilustrações

Figura 1 - Diagrama de Arquitetura.....	8
Figura 2 – Controlador ByEmail	10
Figura 3 - Rota para listar utilizadores.....	10
Figura 4 - Parte do código de criação tabela dos utilizadores	11
Figura 5 - Middleware de autenticação JWT	12
Figura 6 - Exemplo de geração de URL SAS para Azure Blob Storage.....	13
Figura 7 – Exemplo de envio de email com a API Mailgun	14
Figura 8 - Homepage	16
Figura 9 – Login	16
Figura 10 – Login com sucesso.....	17
Figura 11 – Criação de conta.....	17
Figura 12 – Registo com sucesso	18
Figura 13 – Email para finalizar a criação da conta	18
Figura 14- Definição de palavra-passe.....	19
Figura 15 - Palavra-passe definida com sucesso.....	19
Figura 16 - Perfil de utilizador.....	20
Figura 17 - Edição de perfil de utilizador	20
Figura 18 – Perfil alterado	21
Figura 19 – Página inicial sem conteúdo	22
Figura 20 - Página inicial com cursos.....	22
Figura 21 – Página inicial com aplicação do filtro de pesquisa.....	23
Figura 22 – Lista de todos os cursos	23
Figura 23 – Lista de cursos com filtros aplicados.....	24
Figura 24 - Curso para inscrição.....	24
Figura 25 – Inscrição no curso com sucesso	25
Figura 26 – Curso com inscrição ativa	25
Figura 27 – Página de login do BackOffice	26
Figura 28 – Login realizado com sucesso	27
Figura 29 – Página inicial do administrador	27
Figura 30 – Perfil do utilizador administrador.....	28
Figura 31 – Gestão de utilizadores.....	29
Figura 32 – Edição de utilizador.....	30
Figura 33 – Gestão de cursos	31
Figura 34 – Edição de curso	31
Figura 35 – Página de um curso específico.....	32
Figura 36 – Gestão da estrutura temática	32

<i>Figura 37 – Criação de nova área temática.....</i>	<i>33</i>
<i>Figura 38 – Criação de nova categoria</i>	<i>33</i>
<i>Figura 39 – Criação de novo tópico.....</i>	<i>33</i>
<i>Figura 40 – Edição de uma área existente</i>	<i>34</i>
<i>Figura 41 – Edição de uma categoria existente</i>	<i>34</i>
<i>Figura 42 – Tentativa de eliminação negada</i>	<i>35</i>

Introdução

Este relatório documenta o desenvolvimento da aplicação web realizada no âmbito da unidade curricular de **Aplicações para a Internet II**, integrada no Projeto Integrado do 2.º ano da Licenciatura em Engenharia Informática da ESTGV. Apesar de este relatório se focar na componente web, o projeto foi desenvolvido por uma equipa multidisciplinar e destinou-se a responder a um desafio real proposto pela empresa Softinsa.

A aplicação, intitulada **The SoftSkills**, visa facilitar a gestão e avaliação de competências interpessoais (soft skills) em ambiente académico e profissional. A vertente web, que constitui o foco deste relatório, permite a gestão administrativa dos utilizadores, competências, relatórios e interações registadas na aplicação móvel.

Objetivos

Os principais objetivos da componente web deste projeto foram:

- Desenvolver uma aplicação web funcional, responsiva e segura.
- Aplicar os conhecimentos adquiridos na unidade curricular de Aplicações para a Internet II, incluindo:
 - Estruturação de frontend com React.js;
 - Construção de APIs RESTful com Node.js e Express;
 - Integração com bases de dados relacionais (PostgreSQL) usando Sequelize.
- Permitir ao administrador e gestores da aplicação aceder e controlar os dados da plataforma.
- Implementar autenticação e autorização de utilizadores.

Arquitetura da Aplicação Web

A aplicação segue uma arquitetura baseada em cliente-servidor. O frontend React comunica com o backend Express através de uma API REST, que por sua vez interage com a base de dados que tem referências a ficheiros contidos object storage.

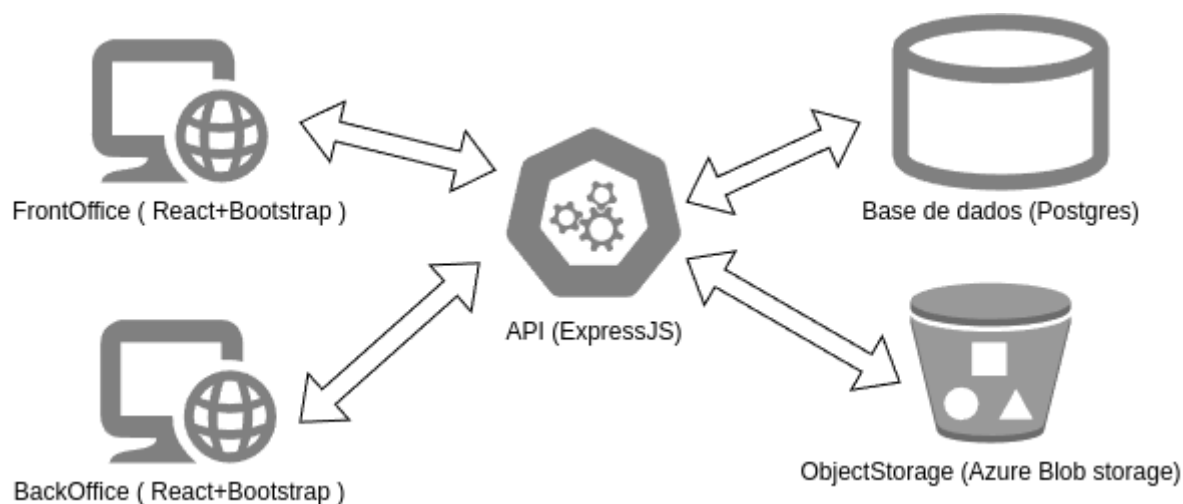


Figura 1- Diagrama de Arquitetura

Desenvolvimento do Backend

O backend da aplicação foi desenvolvido com Node.js e Express, estruturado segundo boas práticas de separação de responsabilidades, com o uso do ORM Sequelize para comunicação com uma base de dados PostgreSQL. A API REST desenvolvida serve como ponte entre a aplicação web e os dados persistidos, fornecendo todas as operações necessárias à gestão de conteúdos, utilizadores, notificações, cursos, avaliações e tópicos.

Estrutura do Projeto

A arquitetura do backend segue uma estrutura modular bem definida:

- **src/controllers/** – Controladores responsáveis por aplicar a lógica de negócio para cada entidade (ex.: `utilizadorController.js`, `cursoController.js`, `notificacaoController.js`).
- **src/routes/** – Define os endpoints públicos da API e associa-os aos controladores respetivos (ex.: `topicoRoute.js`, `categoriaRoute.js`).
- **src/models/** – Modelos Sequelize que representam as tabelas da base de dados e as suas relações.
- **src/database.js** – Responsável pela configuração da ligação à base de dados PostgreSQL.
- **src/middleware.js** – Middleware genérico para autenticação e tratamento de erros.
- **src/utils.js** – Funções auxiliares reutilizáveis.
- **src/app.js** – Ponto de entrada principal do servidor Express.

Funcionalidade principais da API

A API oferece suporte a uma vasta gama de operações, organizadas por entidade. Exemplos de endpoints implementados:

- **/api/utilizadores** – Gestão de contas de utilizador (criação, autenticação, edição).
- **/api/cursos** – Gestão de cursos assíncronos e síncronos.
- **/api/topicos** – Criação, listagem e associação de tópicos.
- **/api/notificacoes** – Envio, histórico e gestão de notificações (gerais ou pessoais).
- **/api/categorias** e **/api/areas** – Organização temática do conteúdo formativo.

-
- **/api/avaliacoes** – Avaliação contínua e final.
 - **/api/denuncias** – Registo e gestão de denúncias de conteúdos (comentários, posts, etc.).

Cada endpoint suporta operações padrão do tipo CRUD, implementadas segundo os princípios REST.

Exemplo Real de Endpoint REST

Abaixo apresenta-se um exemplo real de parte da lógica usada para **gestão de utilizadores** no backend. O controlador `byEmail` consulta o modelo dos utilizadores e procura aqueles que em que o email é igual ao passado na query (`\rota?email=random@email.com`) :



```
controllers.byEmail = async (req, res) => {  
  const email = req.query.email;  
  
  if (!email) {  
    return res.status(400).json({ error: 'Email is required' });  
  }  
  
  try {  
    const data = await models.utilizadores.findOne({  
      where: { email },  
      attributes: ["idutilizador", "email", "nome", "dataregisto", "foto", "ativo"]  
    });  
  
    if (!data) {  
      return res.status(404).json({ error: 'User not found' });  
    }  
  
    if (!data.ativo) {  
      return res.status(403).json({ error: 'User disabled' });  
    }  
  
    data.dataValues.roles = await findRoles(data.idutilizador);  
    data.dataValues.foto = data.dataValues.foto ? await generateSASUrl(data.dataValues.foto,  
      'userprofiles') : null;  
    res.status(200).json(data);  
  } catch (error) {  
    return res.status(500).json({ error: 'Something bad happened' });  
  }  
};
```

Figura 2 – Controlador ByEmail

Exemplo de Rota Protegida para Listar Utilizadores

A rota `/list` permite a um administrador autenticado obter a lista de utilizadores. A proteção é feita por middleware que verifica o token JWT e o papel do utilizador:

```
router.get('/list', authenticateJWT, authorizeRoles('admin'), utilizadorController.list);
```

Figura 3 - Rota para listar utilizadores

Este padrão de desenvolvimento é repetido em diversas rotas sensíveis da aplicação, garantindo segurança e separação de responsabilidades.

Integração com a Base de Dados (PostgreSQL)

A base de dados foi modelada com o Sequelize, que permitiu definir de forma clara:

- Tipos de dados e validações por campo;
- Relações complexas entre modelos (ex.: `hasMany`, `belongsTo`, `belongsToMany`);
- Geração automática de queries SQL.

O ficheiro **init-models.js** centraliza a inicialização de todas as entidades e as respetivas associações. A base de dados inclui mais de 30 tabelas, refletindo a complexidade e riqueza da aplicação, como por exemplo:

- utilizadores, formadores, formandos
- cursos, sessoes, licoes
- notificacoes, comentarios, posts, denuncias
- avaliacoes, submissoes, inscrições

Exemplo da criação de uma tabela

A definição da tabela utilizadores, por exemplo, é feita da seguinte forma:

```
const Sequelize = require('sequelize');
module.exports = function(sequelize, DataTypes) {
  return sequelize.define('utilizadores', {
    idutilizador: {
      autoIncrement: true,
      autoIncrementIdentity: true,
      type: DataTypes.BIGINT,
      allowNull: false,
      primaryKey: true
    },
    nome: {
      type: DataTypes.STRING(60),
      allowNull: false
    },
    email: {
      type: DataTypes.STRING(60),
      allowNull: false,
      unique: "unique_email"
    }
  });
};
```

Figura 4 - Parte do código de criação tabela dos utilizadores

Segurança e Autenticação

Foi implementado um sistema de autenticação com JSON Web Tokens (JWT) para proteger rotas sensíveis. O processo inclui:

- Autenticação inicial com email e palavra-passe;
- Geração de token JWT com tempo de expiração;
- Middleware para verificação e validação do token nas rotas protegidas.

Este sistema assegura que apenas utilizadores autenticados possam aceder aos dados sensíveis ou realizar operações administrativas.

Exemplo Tratamento de Erros e Autenticação

O excerto seguinte mostra o middleware `authenticateJWT`, responsável por validar o token incluído no cabeçalho `Authorization` dos pedidos. Caso o token esteja ausente ou inválido, é devolvida uma resposta adequada com o código HTTP correspondente.

```
function authenticateJWT(req, res, next) {
  const authHeader = req.headers.authorization;

  if (!authHeader || !authHeader.startsWith('Bearer ')) {
    return res.status(401).json({ message: 'Missing token' });
  }

  const token = authHeader.split(' ')[1];

  try {
    const decoded = jwt.verify(token, SECRET);
    req.user = decoded;
    next();
  } catch(error) {
    return res.status(403).json({ message: 'Invalid token' });
  }
}
```

Figura 5 - Middleware de autenticação JWT

Este middleware é aplicado nas rotas que exigem autenticação, garantindo que apenas utilizadores com um token válido (gerado no login) consigam aceder.

Validação e Tratamento de Erros

O backend inclui também mecanismos de validação de dados e tratamento de erros, assegurando robustez e clareza na comunicação com o cliente. Os principais aspetos incluem:

- Validação de entradas malformadas antes de serem processadas;
- Respostas padronizadas com códigos HTTP adequados (400, 401, 403, 500, etc.);

-
- Separação entre erros funcionais (ex: campos em falta) e erros técnicos (ex: falhas de base de dados).

O middleware apresentado no exemplo anterior (figura 5) exemplifica este padrão, devolvendo mensagens claras como "Missing token" ou "Invalid token" consoante o tipo de erro detetado.

Integração com Serviços Externos

Além da autenticação, lógica de negócio e gestão de dados, o backend comunica com vários serviços externos que suportam funcionalidades críticas da aplicação. Esta integração foi feita através de bibliotecas Node.js específicas e APIs fornecidas por cada serviço.

Integração com Azure Blob Storage

Para armazenar imagens e outros ficheiros enviados pelos utilizadores, foi utilizado o serviço Azure Blob Storage. A aplicação gera URLs temporários (SAS tokens) e permite fazer upload direto para o armazenamento externo.

```
const { BlobServiceClient, generateBlobSASQueryParameters } = require("@azure/storage-blob");

// Exemplo de função usada para gerar URL de upload com tempo limitado
function generateSASUrl(blobName) {
  const sasUrl = blobClient.getBlobClient(blobName).generateSasUrl({
    expiresOn: new Date(new Date().valueOf() + 3600 * 1000),
    permissions: BlobSASPermissions.parse("r"),
  });
  return sasUrl;
}
```

Figura 6 - Exemplo de geração de URL SAS para Azure Blob Storage

Integração com Firebase Cloud Messaging (FCM)

A aplicação está preparada para integrar com Firebase Cloud Messaging (FCM) para o envio de notificações push em tempo real. No entanto, atualmente, apenas o envio de emails está implementado. A configuração para integração futura com FCM já se encontra preparada no backend.

Integração com Mailgun (Envio de Emails)

Para o envio de emails automáticos, como recuperação de palavra-passe ou alertas administrativos, foi integrada a API da plataforma Mailgun.

```
const formData = require('form-data');
const Mailgun = require('mailgun.js');

const mg = Mailgun.client({ username: 'api', key: process.env.MAILGUN_API_KEY });

mg.messages.create('sandboxXXXX.mailgun.org', {
  from: "Sistema <noreply@exemplo.com>",
  to: ["utilizador@dominio.com"],
  subject: "Recuperação de Palavra-passe",
  text: "Clique no link seguinte para recuperar o acesso à sua conta..."
});
```

Figura 7 – Exemplo de envio de email com a API Mailgun

Geração Automática de Modelos com Sequelize-Auto

Os modelos utilizados no backend foram gerados automaticamente com recurso à ferramenta sequelize-auto, a partir de uma base de dados PostgreSQL já existente. Esta abordagem permitiu:

- Criar rapidamente os ficheiros de definição de modelos;
- Garantir consistência com a estrutura real da base de dados;
- Reduzir erros e acelerar o arranque do desenvolvimento.

Desenvolvimento do FrontEnd

Descrição geral da stack e ferramentas usadas

O frontend da aplicação foi desenvolvido com React.js, recorrendo a uma arquitetura baseada em componentes reutilizáveis e gestão de estado com hooks (useState, useEffect). Foi utilizado o React Router para navegação entre páginas e o Axios para comunicação com a API backend. Para o design visual, usou-se Bootstrap e CSS personalizado. Esta abordagem permitiu criar uma interface responsiva, moderna e fácil de manter.

Estrutura do FrontEnd

A estrutura do projeto React está organizada da seguinte forma:

- **/src/components** – Componentes reutilizáveis como Navbar, Footer, FormularioCurso, etc.
- **/src/pages** – Páginas principais como HomePage, Login, Registo, Perfil, Cursos, BackOffice.
- **/src/services** – Módulos que comunicam com a API backend via Axios.
- **/src/assets** – Imagens e ficheiros estáticos.
- **/src/App.js** – Definição de rotas principais e layout da aplicação.
- **/src/index.js** – Ponto de entrada da aplicação.

Autenticação no FrontEnd

Após o login com sucesso, o token JWT recebido do backend é guardado em LocalStorage e utilizado em cada requisição protegida, através de `axios.defaults.headers.common['Authorization']`. Isto garante que apenas utilizadores autenticados acedem a certas funcionalidades, como o BackOffice ou a edição de perfil.

Criação dos formulários

Todos os formulários foram criados com React e usam o estado local para controlo dos campos. A submissão é feita via Axios para endpoints REST da API. O feedback ao utilizador é exibido com mensagens de sucesso ou erro consoante a resposta do servidor.

FrontOffice

Descrição das Views do FrontOffice

Página inicial da aplicação.

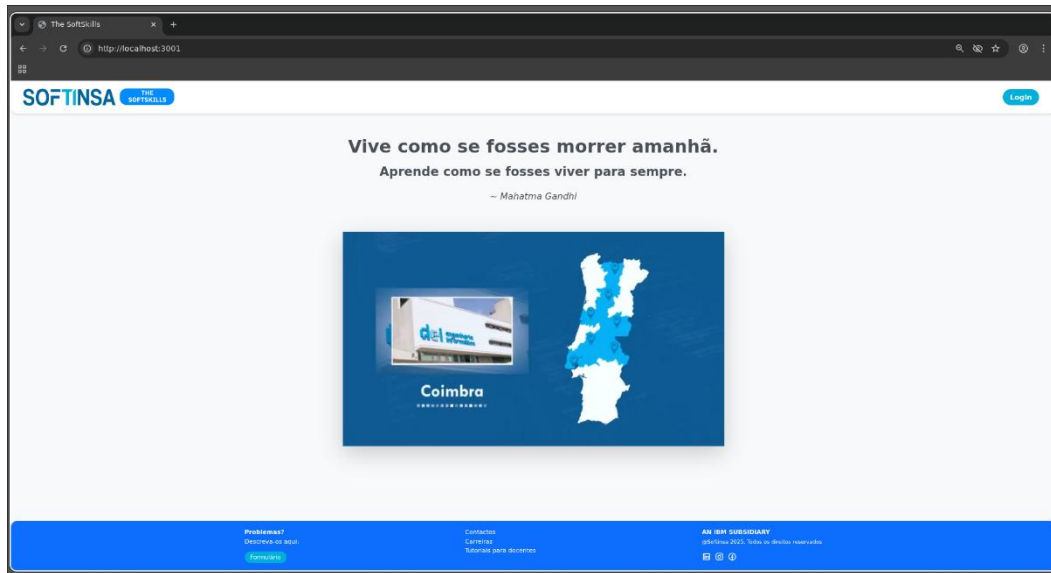


Figura 8 - Homepage

Página de autenticação onde o utilizador insere as credenciais de acesso.

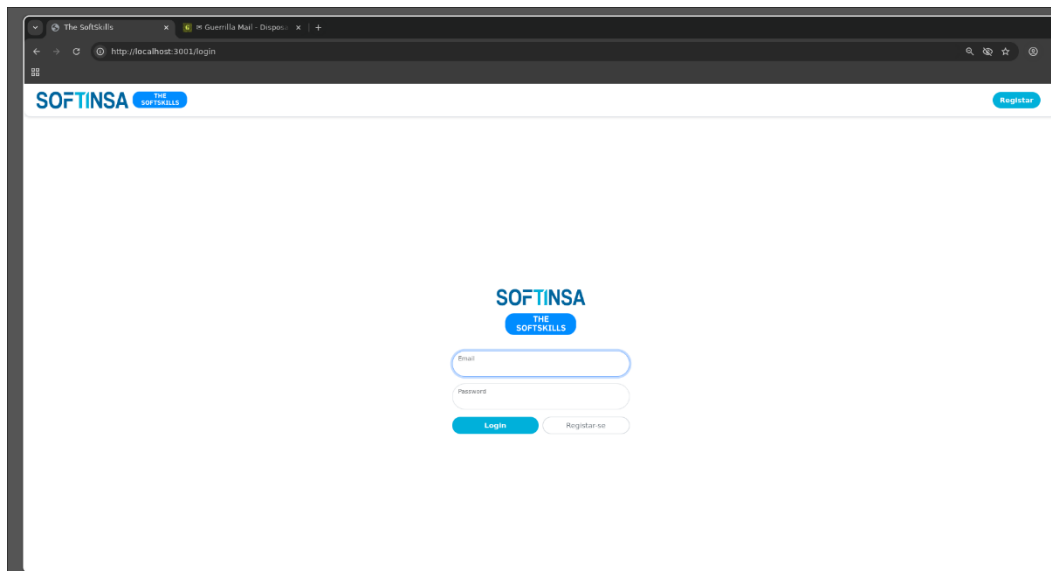


Figura 9 – Login

Confirmação visual de login feito com sucesso.

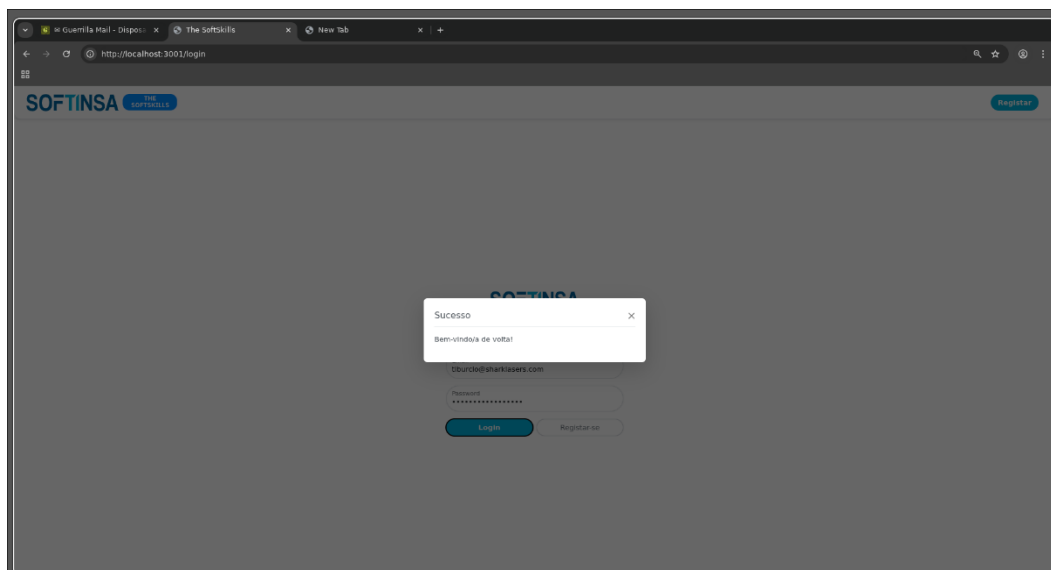


Figura 10 – Login com sucesso

Primeira etapa do formulário de registo de novo utilizador.

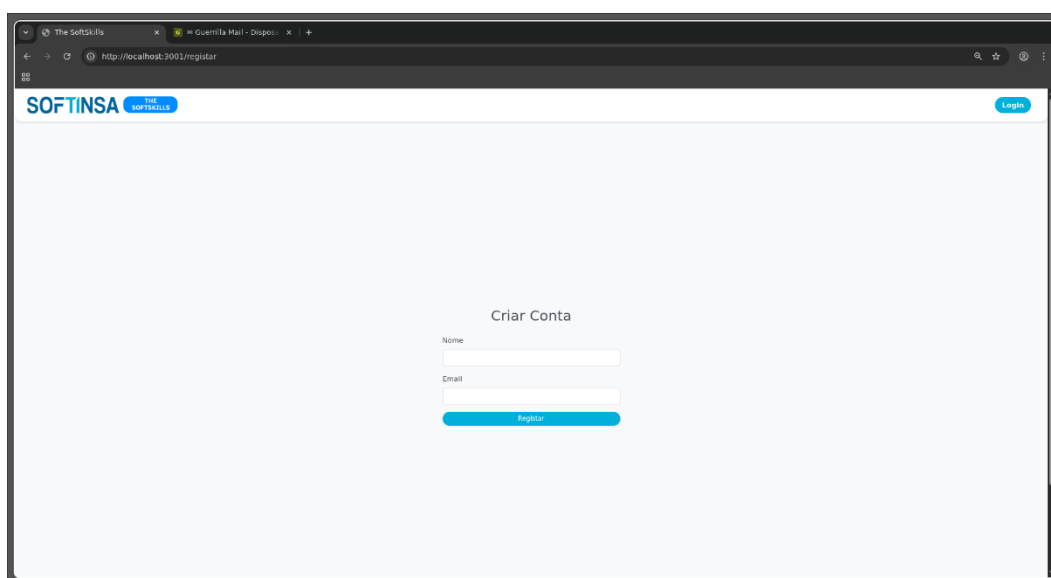


Figura 11 – Criação de conta

Confirmação visual do envio de email para criar uma password.

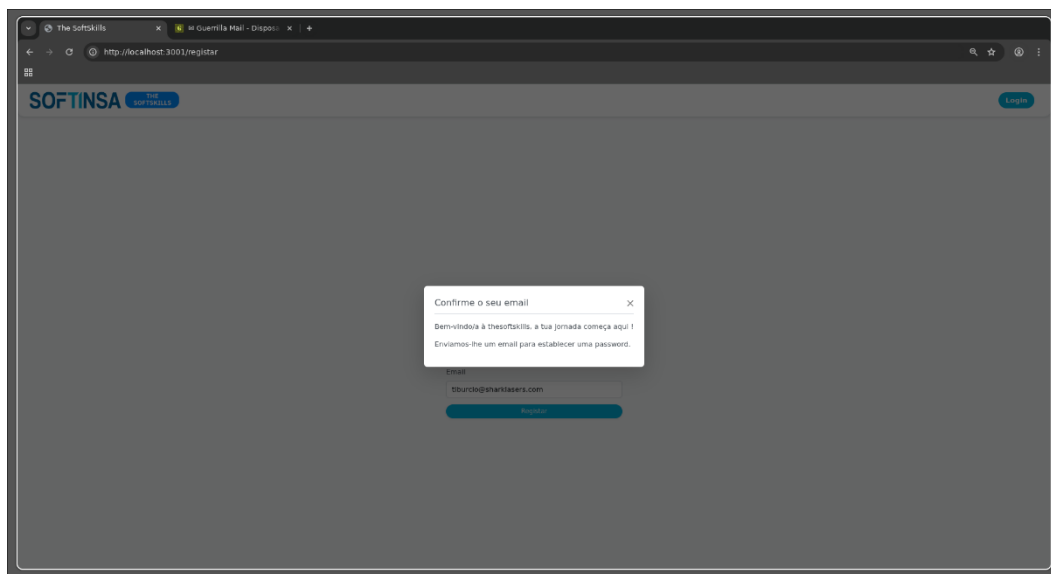


Figura 12 – Registo com sucesso

Email enviado para confirmação da criação da conta.

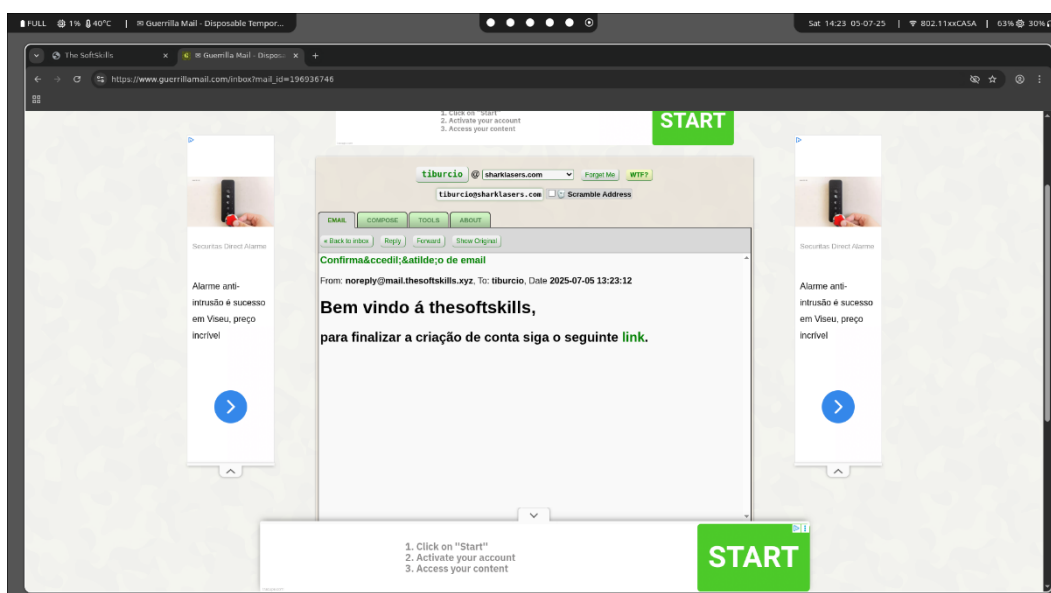


Figura 13 – Email para finalizar a criação da conta

Definição da palavra-passe após criação de conta.

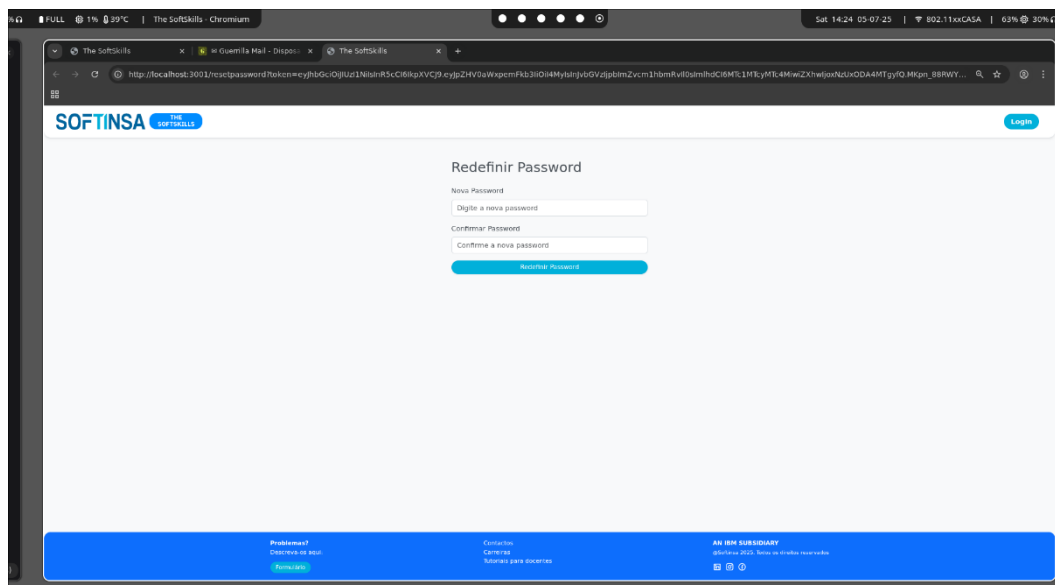


Figura 14- Definição de palavra-passe

Confirmação visual de que a palavra-passe foi definida com sucesso.

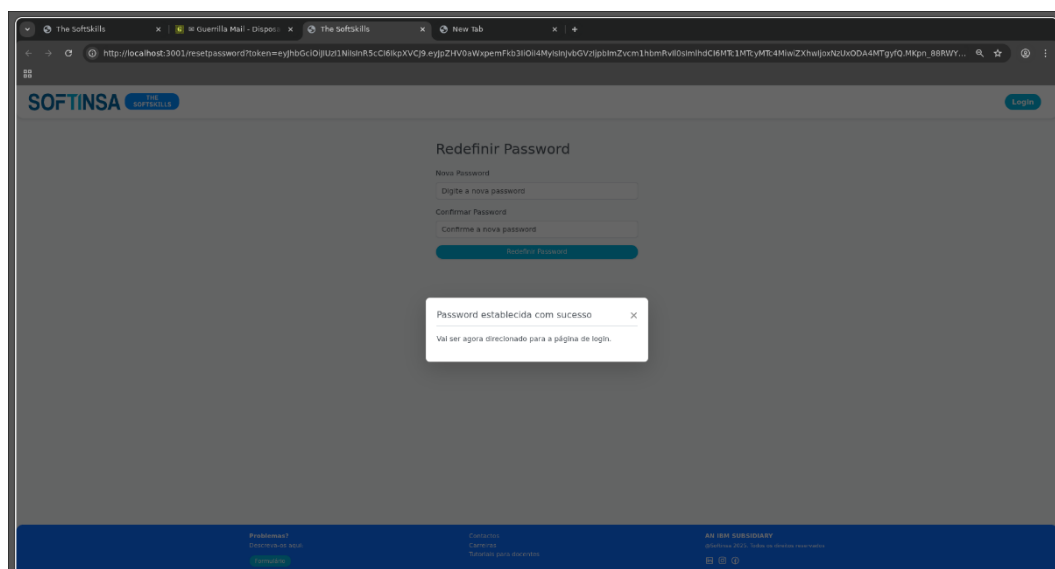


Figura 15 - Palavra-passe definida com sucesso

Estado inicial do perfil para novos utilizadores após o registo.

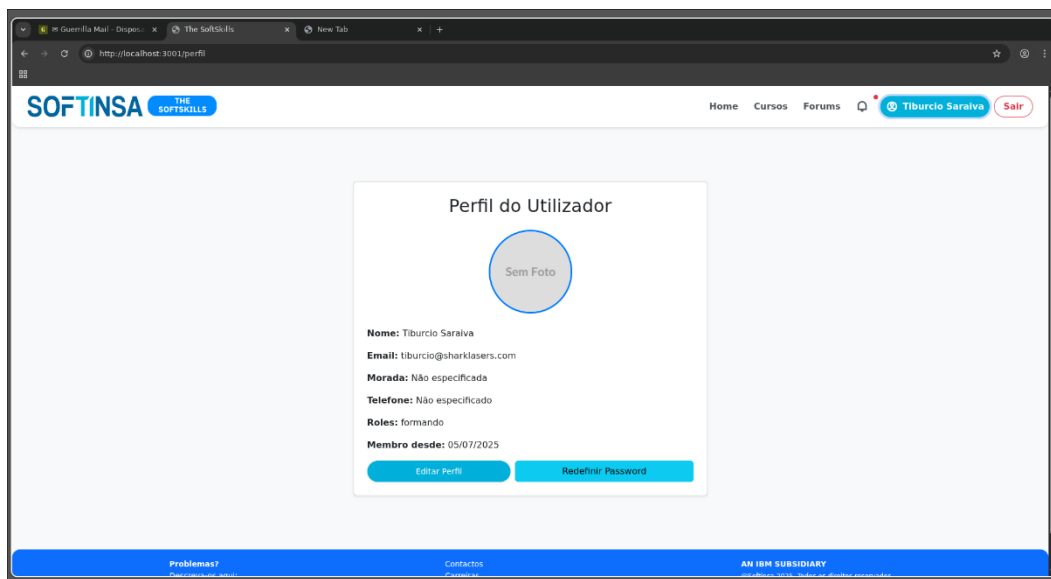


Figura 16 - Perfil de utilizador

Página de edição de perfil, permitindo alterar dados pessoais, email ou palavra-passe.

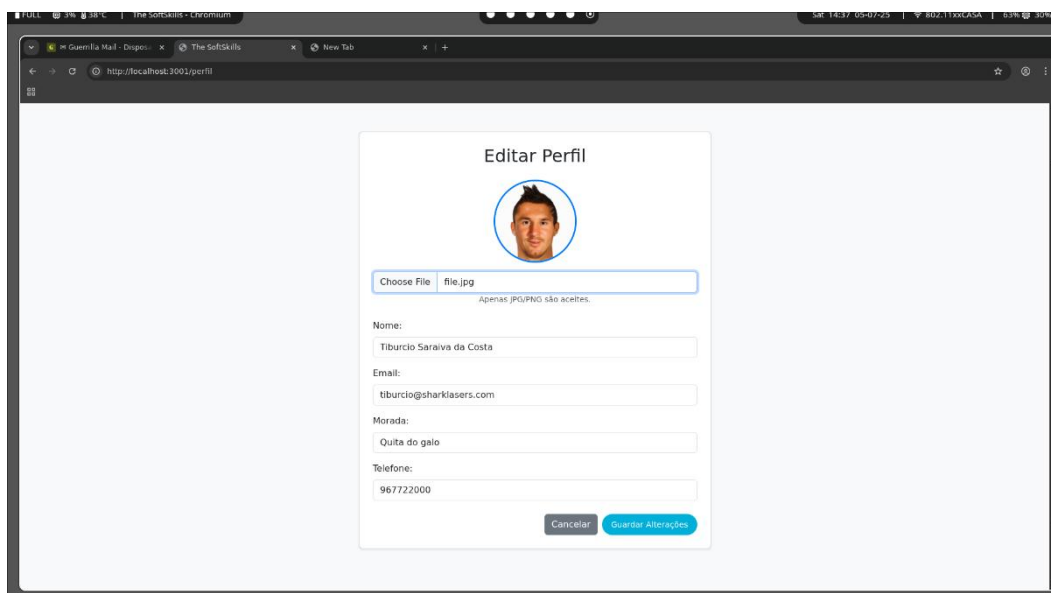
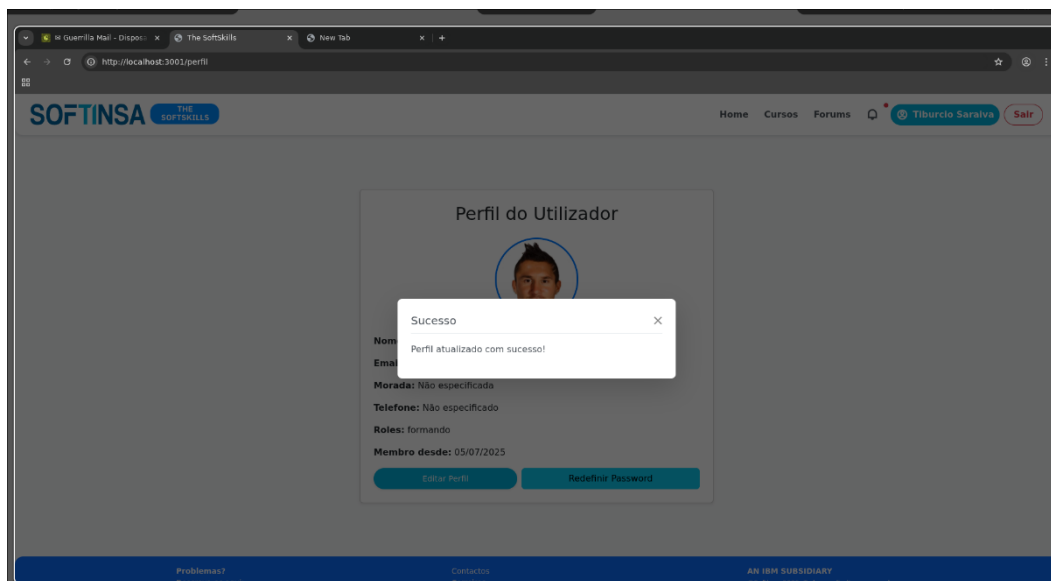


Figura 17 - Edição de perfil de utilizador

Confirmação visual de que o perfil foi alterado com sucesso.



Confirmação visual de que o perfil foi alterado com sucesso.

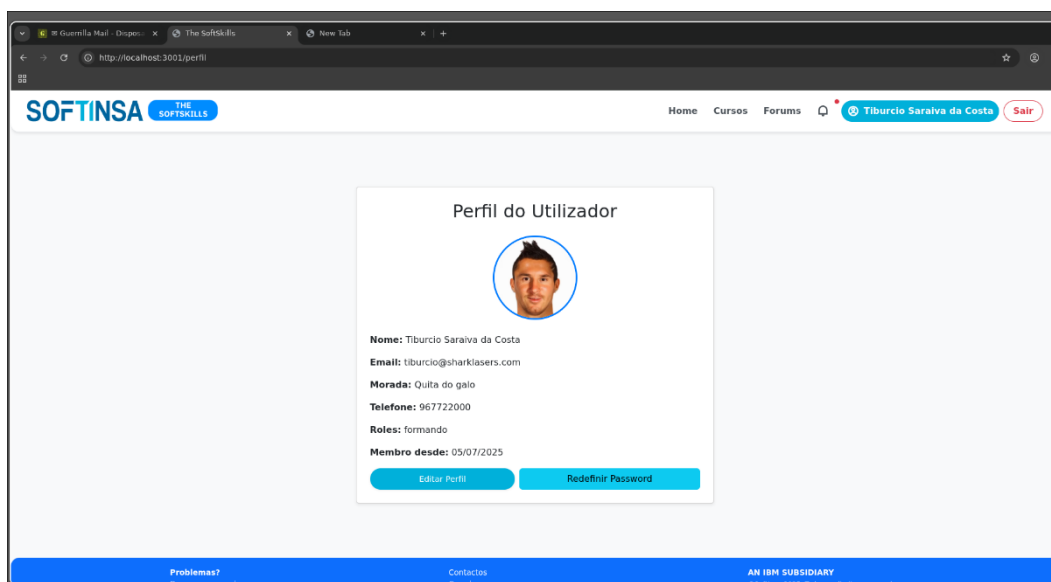


Figura 18 – Perfil alterado

Estado da página inicial sem conteúdo disponível para o utilizador.

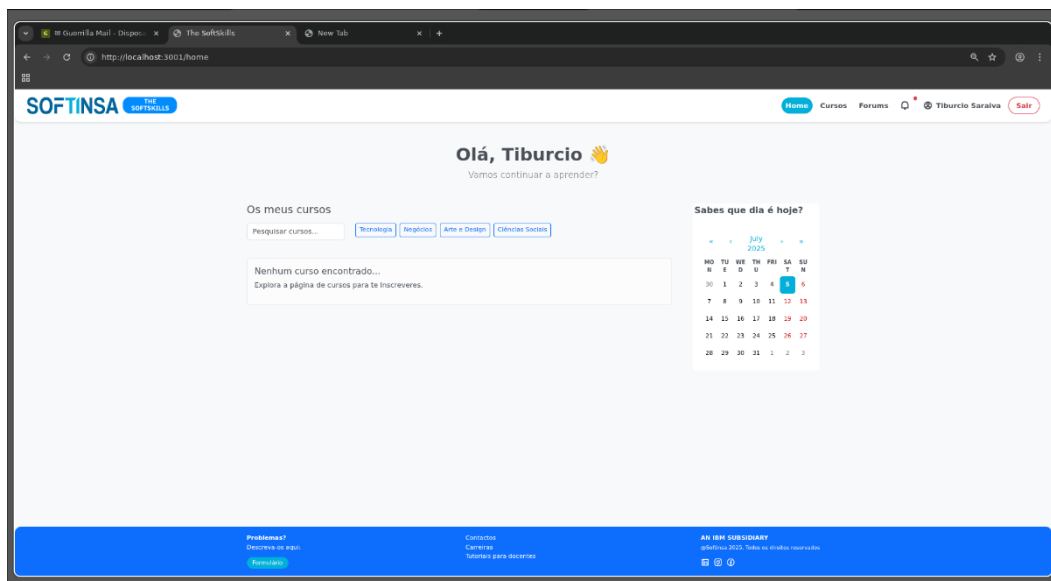


Figura 19 – Página inicial sem conteúdo

Página inicial com conteúdo apresentado ao utilizador, como os cursos inscritos.

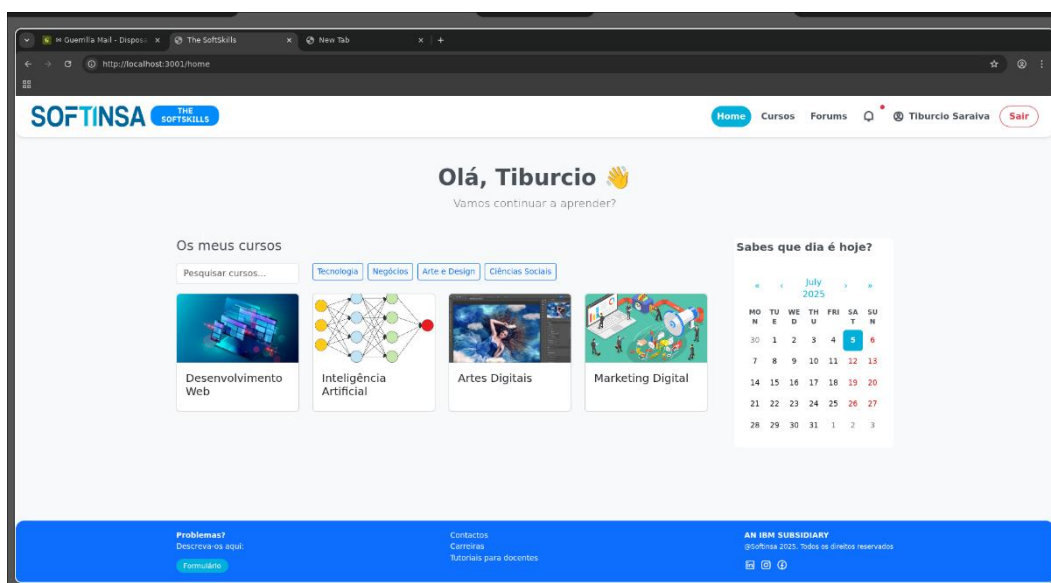


Figura 20 - Página inicial com cursos

Página inicial após aplicação de filtros de pesquisa ou categorias.

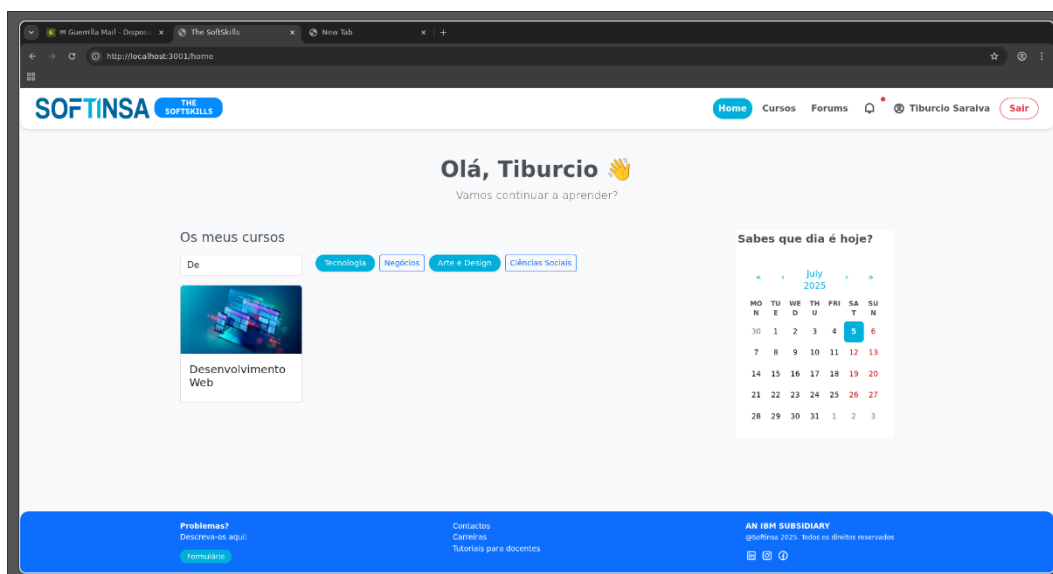


Figura 21 – Página inicial com aplicação do filtro de pesquisa

Listagem de todos os cursos disponíveis.

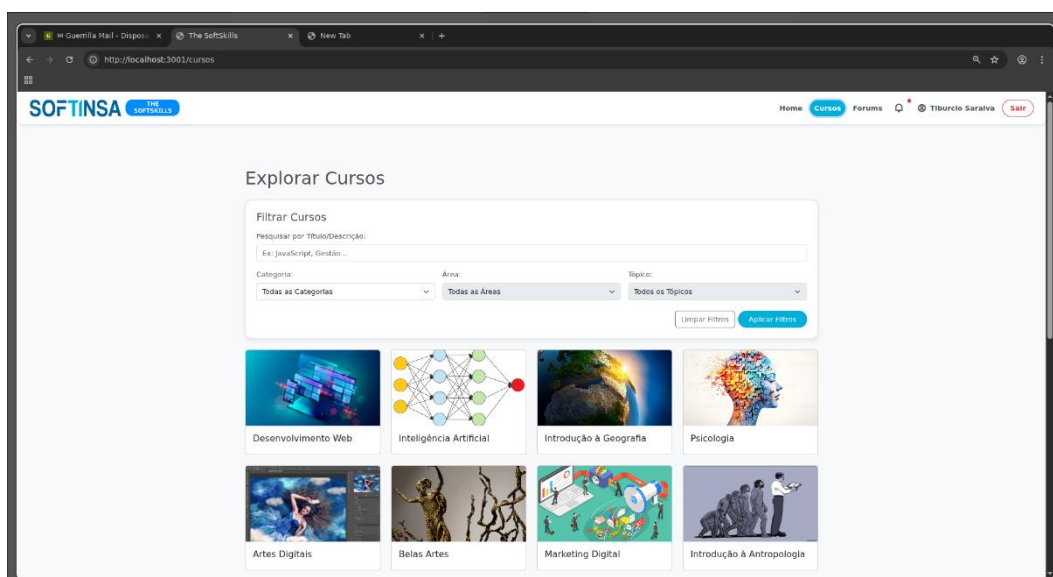


Figura 22 – Lista de todos os cursos

Listagem de cursos com filtros aplicados, como categoria ou disponibilidade.

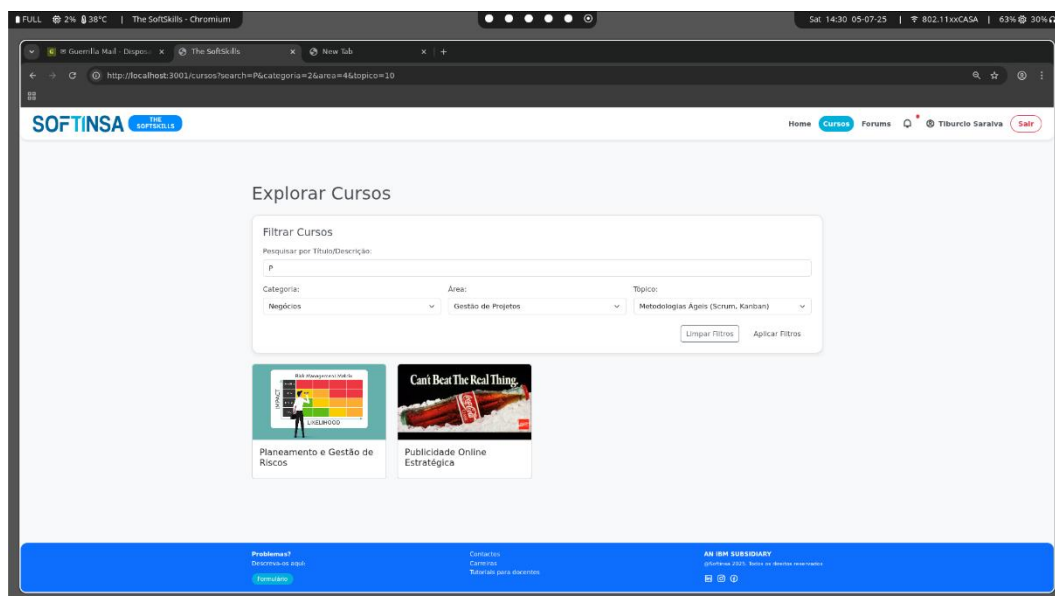


Figura 23 – Lista de cursos com filtros aplicados

Visualização de um curso ainda não iniciado pelo utilizador, com a opção de inscrição disponível.

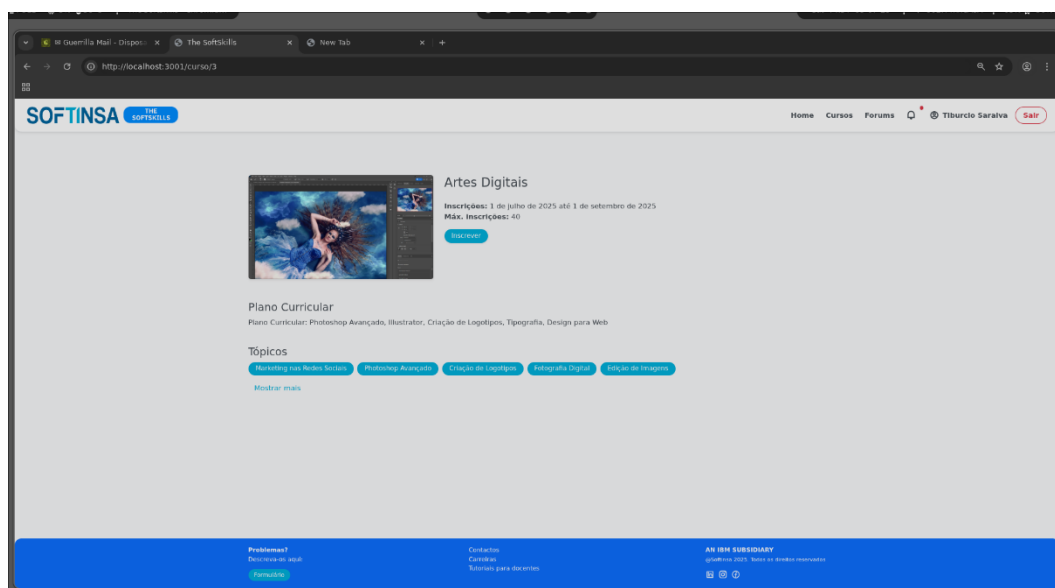


Figura 24 - Curso para inscrição

Confirmação visual da realização da inscrição no curso.

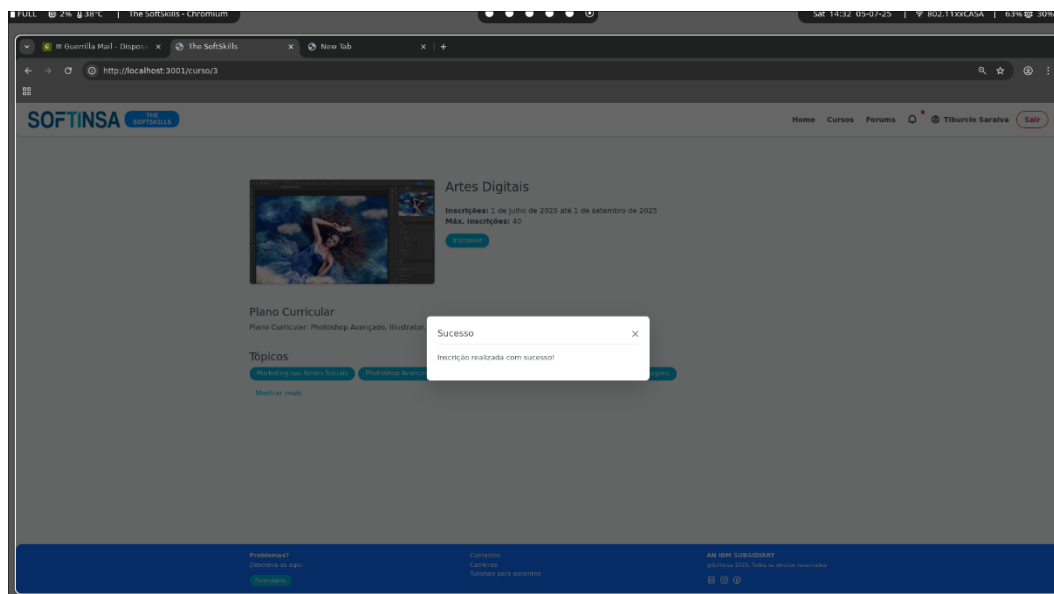


Figura 25 – Inscrição no curso com sucesso

Visualização de um curso no qual o utilizador já está inscrito, incluindo opções de visualização de conteúdos ou cancelamento.

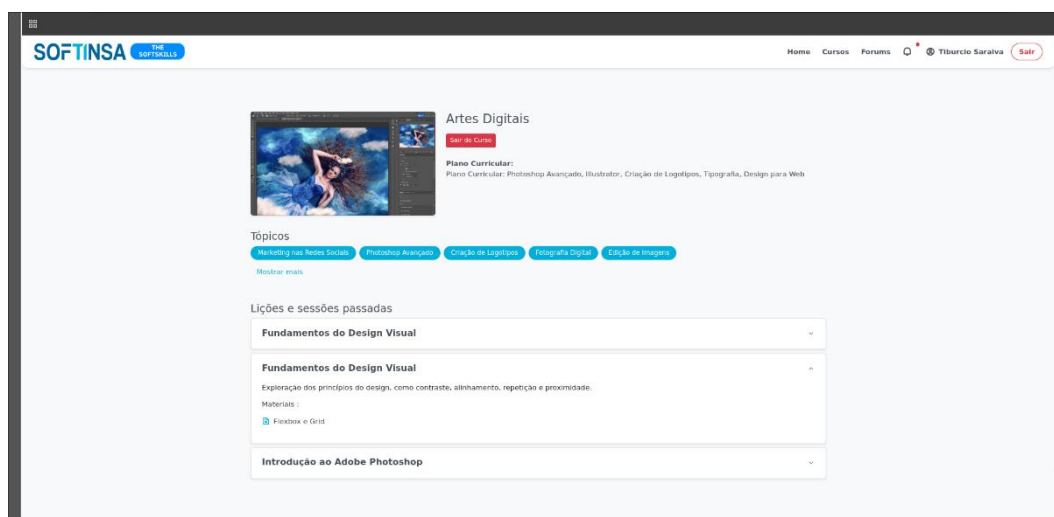


Figura 26 – Curso com inscrição ativa

BackOffice

Descrição das views do BackOffice

O BackOffice da aplicação permite aos utilizadores com perfil de administrador ou gestor aceder a funcionalidades de gestão e configuração da plataforma. Através de uma interface clara e estruturada, é possível gerir utilizadores, cursos, estrutura temática (áreas, categorias e tópicos), bem como editar perfis e validar permissões.

Ecrã de autenticação dedicado aos administradores, onde é necessário introduzir credenciais válidas para aceder à área de gestão.

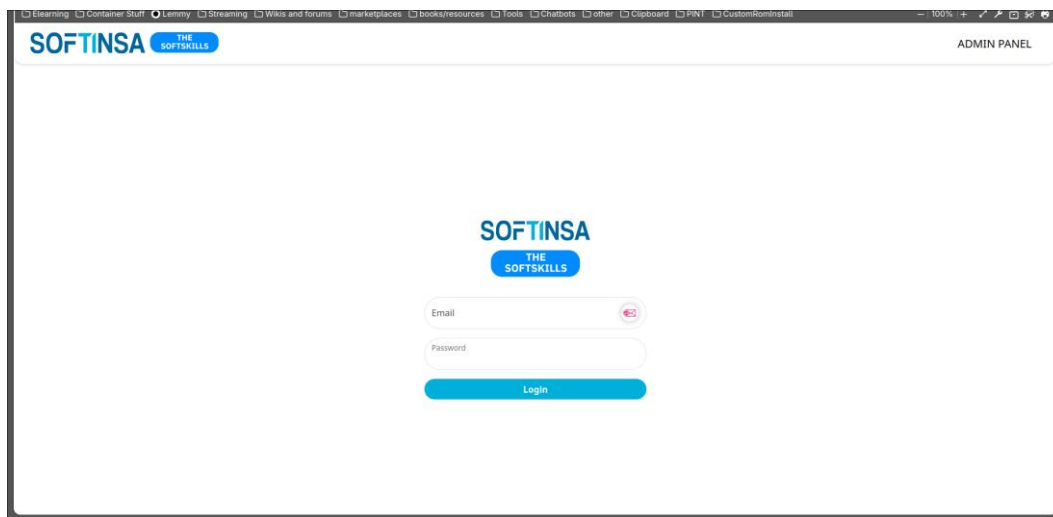


Figura 27 – Página de login do BackOffice

Confirmação visual de que o login foi efetuado com sucesso, redirecionando o utilizador para o painel de administração.

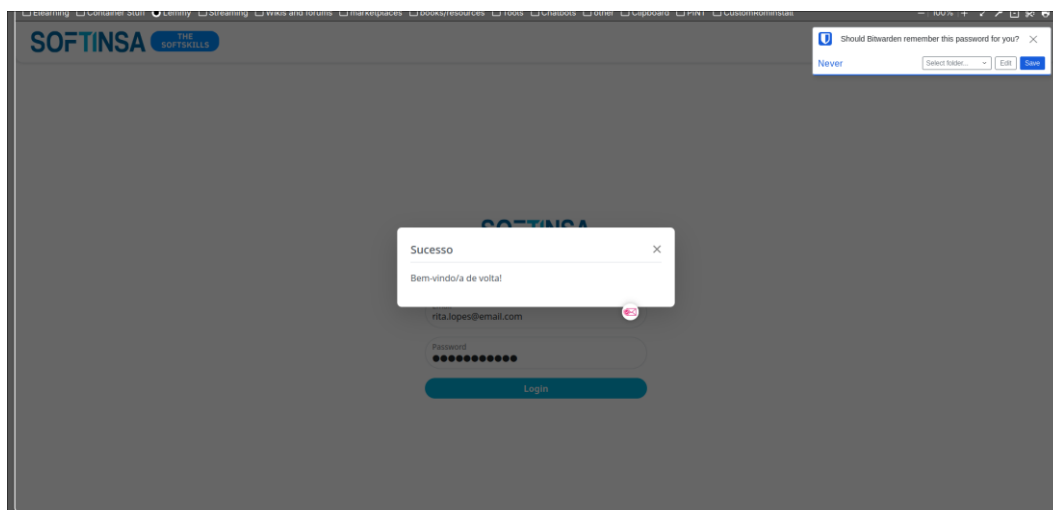


Figura 28 – Login realizado com sucesso

Painel inicial do BackOffice para aceder às áreas de gestão da aplicação.

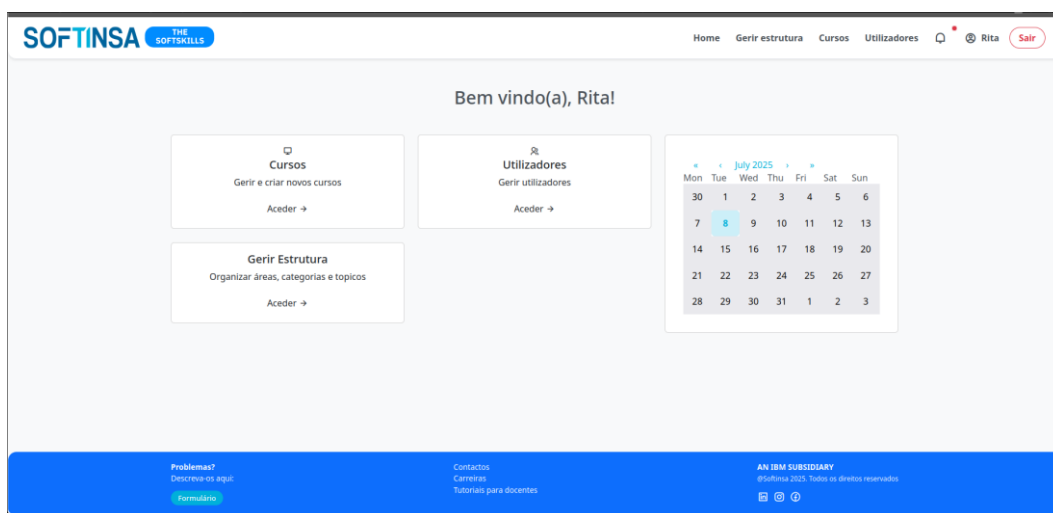


Figura 29 – Página inicial do administrador

Vista do perfil pessoal do administrador, onde é possível editar dados como nome, email ou palavra-passe.

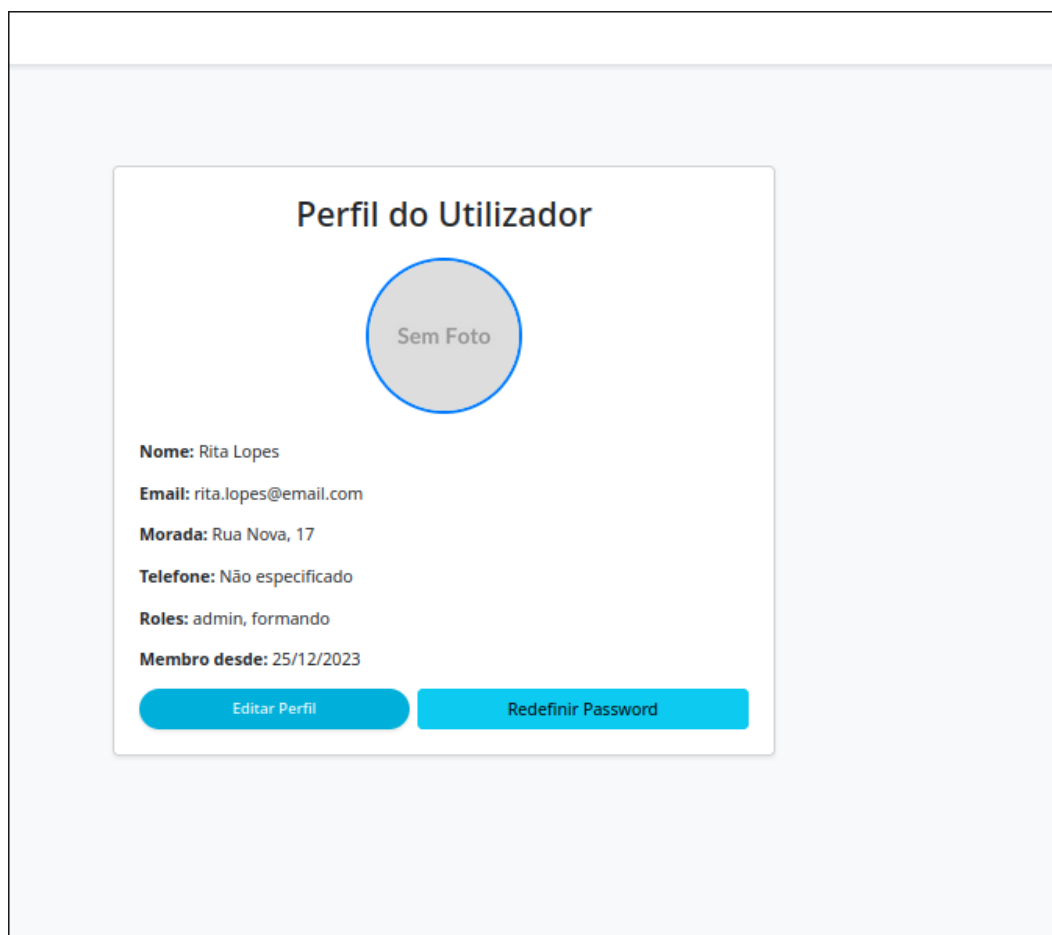


Figura 30 – Perfil do utilizador administrador

Lista de utilizadores registados na plataforma, com funcionalidades para editar ou desativar contas.


Gestão de Utilizadores

ID	Nome	Email	Data de Registo	Ativo	Roles	Ações
9	Bruno Almeida	bruno.almeida@email.com	9/10/2023	Sim	N/A	@ Ver/Editar Desativar
10	Inês Mendes	ines.mendes@email.com	8/1/2023	Sim	admin formador	@ Ver/Editar Desativar
1	João Silva	joao.silva@email.com	5/1/2024	Sim	admin formador	@ Ver/Editar Desativar
41	Daniel Costa	danielcosta@email.com	5/27/2025	Sim	admin formador	@ Ver/Editar Desativar
67	Teste De Criação	webspoue@sharklasers.com	6/11/2025	Sim	admin	@ Ver/Editar Desativar
38	Pedro Santos	pedrosantos@email.com	5/20/2025	Sim	admin formador	@ Ver/Editar Desativar
37	João Matos	joamatos@email.com	5/20/2025	Sim	formando formador	@ Ver/Editar Desativar
71	Nuno Luis	nunoluis@sharklasers.com	7/2/2025	Sim	formando	@ Ver/Editar Desativar
72	Marco	marcocarlos@sharklasers.com	7/2/2025	Sim	formando	@ Ver/Editar Desativar
4	Ana Costa	ana.costa@email.com	2/10/2024	Sim	formando	@ Ver/Editar Desativar
73	Hugo Medeiros	hugomedeiros@sharklasers.com	7/2/2025	Sim	formando	@ Ver/Editar Desativar
2	Maria Oliveira	maria.oliveira@email.com	4/15/2024	Sim	admin formador	@ Ver/Editar Desativar
79	Tiago Portugal	dalekbuffyfrsk@sharklasers.com	7/2/2025	Sim	formando	@ Ver/Editar Desativar
82	vwzeidrk	vwzeidrk@sharklasers.com	7/2/2025	Sim	formando	@ Ver/Editar Desativar
83	Tiburcio Saraiva da Costa	tiburcio@sharklasers.com	7/5/2025	Sim	formando	@ Ver/Editar Desativar
39	Sara Prata	saraprata@email.com	5/25/2025	Sim	N/A	@ Ver/Editar Desativar
7	Tiago Ferreira	tiago.ferreira@email.com	11/30/2023	Sim	formando	@ Ver/Editar Desativar
8	Sofia Rocha	sofia.rocha@email.com	10/18/2023	Sim	formando	@ Ver/Editar Desativar
68	César Cabral	ailfbubakjnlfk@sharklasers.com	6/11/2025	Sim	admin formador	@ Ver/Editar Desativar
3	Carlos Santos	carlos.santos@email.com	3/20/2024	Sim	formador	@ Ver/Editar Desativar
5	Pedro Martins	pedro.martins@email.com	1/5/2024	Sim	formando	@ Ver/Editar Desativar

Figura 31 – Gestão de utilizadores

Formulário de edição de utilizador, permitindo alterar nome, email, tipo de utilizador e estado da conta. Também é possível “desinscrever” o utilizador dos cursos já inscritos.

Editar Utilizador: Pedro Martins



Browse...

No file selected.

Apenas JPG/PNG são aceites.

Nome:

Pedro Martins

Email:

pedro.martins@email.com

Morada:

Rua aleatória de Coimbra

Telefone:

962344632

Roles:

☐ Admin ☐ Formador ☒ Formando

Cancelar

Guardar Alterações

Cursos Inscritos

Nome do Curso	
Artes Digitais	<div>Desinscrever</div>
Belas Artes	<div>Desinscrever</div>
Introdução à Antropologia	<div>Desinscrever</div>
Estratégias de Marketing Online	<div>Desinscrever</div>

Figura 32 – Edição de utilizador

Painel para visualizar, editar ou remover cursos existentes, bem como aceder aos seus detalhes.

Gestão de Cursos

+ Criar Novo Curso

Filtrar Cursos

Pesquisar por Título/Descrição:

Ex: JavaScript, Gestão...

Categoria:

Todas as Categorias

Área:

Todas as Áreas

Tópico:

Todos os Tópicos

Limpar Filtros

Aplicar Filtros

ID	Nome	Disponível	Início Inscrições	Fim Inscrições	Max Inscrições	Ações
27	Introdução à Geografia	Sim	7/23/2025	N/A		<div>Ver</div> <div>Editar</div> <div>Eliminar</div>
4	Psicologia	Sim	8/1/2025	9/1/2025	60	<div>Ver</div> <div>Editar</div> <div>Eliminar</div>
3	Artes Digitais	Sim	7/1/2025	9/1/2025	40	<div>Ver</div> <div>Editar</div> <div>Eliminar</div>
18	Belas Artes	Sim	7/10/2025	N/A		<div>Ver</div> <div>Editar</div> <div>Eliminar</div>
20	Criação de Sites com JavaScript	Não	7/16/2025	N/A		<div>Ver</div> <div>Editar</div> <div>Eliminar</div>
24	Técnicas de Fotografia Profissional	Não	7/20/2025	N/A		<div>Ver</div> <div>Editar</div> <div>Eliminar</div>
26	Psicologia do Desenvolvimento	Não	7/22/2025	N/A		<div>Ver</div> <div>Editar</div> <div>Eliminar</div>
30	Programação Front-End Moderna	Não	7/26/2025	N/A		<div>Ver</div> <div>Editar</div> <div>Eliminar</div>

Figura 33 – Gestão de cursos

Formulário para alterar os detalhes de um curso: nome, descrição, disponibilidade e categorias associadas.

Editar Curso: Artes Digitais

Detalhes do Curso

Nome do Curso:

Artes Digitais

Início das Inscrições:

07 / 01 / 2025, 10:00 AM

Fim das Inscrições:

09 / 01 / 2025, 11:59 PM

Máx. Inscrições:

40

Plano Curricular:

Plano Curricular: Photoshop Avançado, Illustrator, Criação de Logotipos, Tipografia, Design para Web

Disponível para Inscrição

Thumbnail:

Browse...

No file selected.

Thumbnail atual

Tópicos:

☐ JavaScript Avançado

☐ Frameworks Front-End (React, Angular, Vue)

☐ Redes Neurais Artificiais

☐ Machine Learning Básico

☐ Deep Learning e Redes Convolucionais

☐ Criptografia

☐ Testes de Penetração (Pen-Testing)

☐ Proteção contra Phishing

☐ Metodologias Ágeis (Scrum, Kanban)

☐ Planejamento de Projetos

☐ Gestão de Riscos

☐ SEO - Otimização para Motores de Busca

☐ Publicidade no Google Ads

☒ Marketing nas Redes Sociais

☒ Photoshop Avançado

☒ Criação de Logotipos

☐ Tipografia e Design Visual

☒ Fotografia Digital

☒ Edição de Imagens

☒ Técnicas de Fotografia Noturna

☒ Desenho Vetorial

☒ Pintura Digital em Photoshop

☐ Design de Personagens

☐ Psicologia Cognitiva

☐ Psicologia Clínica

☐ Psicologia do Desenvolvimento

☐ Teorias Sociológicas

☐ Desigualdade Social

☐ Comportamento Humano e Sociedade

☐ Antropologia Cultural

☐ Evolução Humana

☐ Antropologia Forense

☐ geografia

☐ Compiladores

☐ random topico

☐ random topico

Atualizar Curso

Lições do Curso


+ Adicionar Lição

Figura 34 – Edição de curso

31

Visualização detalhada de um curso selecionado, incluindo tópicos, participantes e ações disponíveis.

Artes Digitais



N/A

ID do Curso: 3

Disponível: sim

Síncrono: não

Duração (Horas): N/A

Início Inscrições: 1 de julho de 2025

Fim Inscrições: 1 de setembro de 2025

Máx. Inscrições: 40

Inscrições Atuais: N/A

Curso Assíncrono

Editar Curso

Eliminar Curso

Piano Curricular

Piano Curricular: Photoshop Avançado, Illustrator, Criação de Logotipos, Tipografia, Design para Web

Tópicos Abordados

Marketing nas Redes Sociais

Photoshop Avançado

Criação de Logotipos

Fotografia Digital

Edição de Imagens

Mostrar mais (3 mais)

Utilizadores Inscritos neste Curso

Nome do Utilizador	Email	Ações
Tibúrcio Saraiva da Costa	tiburcio@sharklasers.com	<div>Desinscrever</div>
Carlos Santos	carlos.santos@email.com	<div>Desinscrever</div>
Pedro Martins	pedro.martins@email.com	<div>Desinscrever</div>

Lições e Materiais do Curso

Fundamentos do Design Visual

Fundamentos do Design Visual

Figura 35 – Página de um curso específico

Secção que permite gerir a estrutura organizacional dos conteúdos: áreas, categorias e tópicos.

Gerir Estrutura

Criar Categoria

Criar Área

Criar Tópico

Negócios

Arte e Design

Áreas:

Design Gráfico

Fotografia

Tópicos:

Desenho Vetorial

Pintura Digital em Photoshop

Design de Personagens

Ilustração Digital

Ciências Sociais

Editar

Editar

Editar

Editar

Editar

Editar

Editar

Editar

Figura 36 – Gestão da estrutura temática

Formulário para introduzir uma nova área temática no sistema

SOFTINSA THE SOFTSKILLS

Home Gerir estrutura Cursos Utilizadores

Criar Nova Estrutura

[Criar Categoria](#) **[Criar Área](#)** [Criar Tópico](#)

Designação da Área

Categoria

Negócios

[Criar Área](#)

Figura 37 – Criação de nova área temática

Formulário para introduzir uma nova categoria dentro de uma área previamente definida.

SOFTINSA THE SOFTSKILLS

Home Gerir estrutura Cursos Utilizadores

Criar Nova Estrutura

[Criar Categoria](#) [Criar Área](#) [Criar Tópico](#)

Designação da Categoria

[Criar Categoria](#)

Figura 38 – Criação de nova categoria

Formulário para adicionar um novo tópico, que será associado a uma categoria.

SOFTINSA THE SOFTSKILLS

Home Gerir estrutura Cursos Utilizadores

Criar Nova Estrutura

[Criar Categoria](#) [Criar Área](#) **[Criar Tópico](#)**

Designação do Tópico

Áreas

- ☐ Segurança da Informação
- ☐ Gestão de Projetos
- ☐ Marketing Digital
- ☐ Empreendedorismo
- ☐ Design Gráfico

[Criar Tópico](#)

Figura 39 – Criação de novo tópico

Formulário para alterar o nome ou estado de uma área temática já criada.

The screenshot shows the 'Editar Área' (Edit Area) form in the SOFTINSA system. The header includes the SOFTINSA logo, 'THE SOFTSKILLS' tagline, and navigation links: Home, Gerir estrutura, Cursos, Utilizadores, a notification bell, a user profile icon for 'Rita', and a 'Sair' (Logout) button. The form itself has a title 'Editar Área' and two input fields: 'Designação da Área' (Area Designation) with the value 'Design Gráfico' and 'Categoria' (Category) with a dropdown menu showing 'Arte e Design'. Below the fields are two buttons: 'Atualizar Área' (Update Area) and 'Cancelar' (Cancel). The footer contains links for 'Problemas?' (Problems?), 'Contactos' (Contacts), and 'AN IBM SUBSIDIARY' information.

Figura 40 – Edição de uma área existente

Formulário para modificar uma categoria específica, incluindo nome e associação.

The screenshot shows the 'Editar Categoria' (Edit Category) form in the SOFTINSA system. The header is identical to the previous figure. The form has a title 'Editar Categoria' and a single input field 'Designação da Categoria' (Category Designation) with the value 'Negócios'. Below the field are two buttons: 'Atualizar Categoria' (Update Category) and 'Cancelar' (Cancel). The footer is also identical to the previous figure.

Figura 41 – Edição de uma categoria existente

Mensagem de erro exibida quando se tenta remover uma entidade que ainda tem dependências no sistema (ex.: tópicos associados).

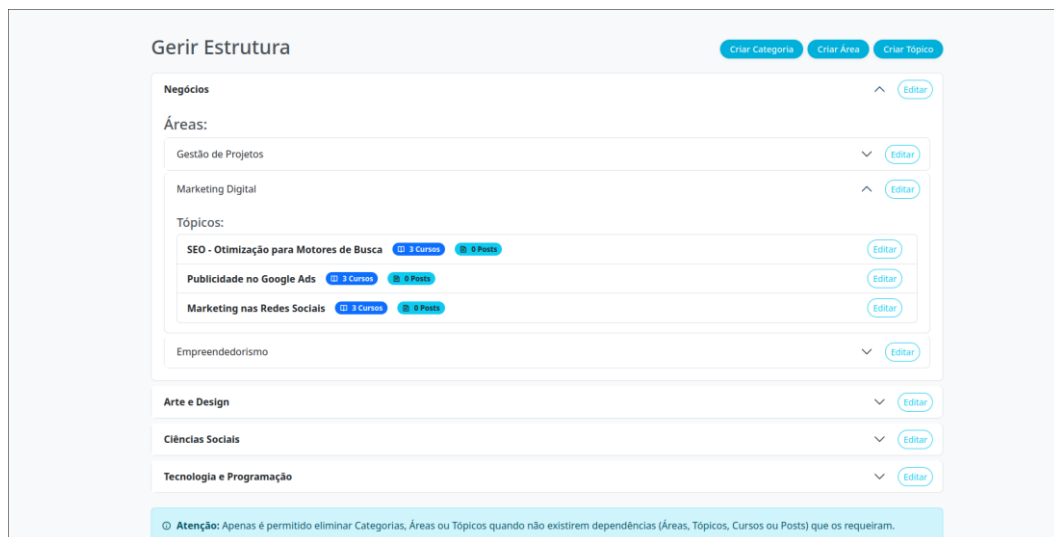


Figura 42 – Tentativa de eliminação negada

Conclusões

O desenvolvimento da aplicação The SoftSkills representou uma oportunidade de aplicar, de forma prática, os conhecimentos adquiridos na unidade curricular de Aplicações para a Internet II, integrando conceitos de desenvolvimento web moderno, segurança, integração com serviços externos e boas práticas de arquitetura de software.

Na componente backend, foi implementada uma API RESTful robusta com Node.js e Express, protegida por autenticação JWT e estruturada em camadas bem definidas. A comunicação com a base de dados PostgreSQL foi realizada através do ORM Sequelize, com os modelos gerados automaticamente a partir do esquema existente usando a ferramenta sequelize-auto. Foram ainda integrados serviços externos essenciais como o Azure Blob Storage para armazenamento de ficheiros, Mailgun para envio de emails e configuração futura para Firebase Cloud Messaging.

O frontend foi desenvolvido com React, utilizando componentes modulares, React Router para navegação e Axios para comunicação com a API. A gestão do estado foi feita com hooks, e a autenticação no frontend é persistida via LocalStorage. O frontend inclui tanto a vertente FrontOffice, destinada ao utilizador final, como o BackOffice, orientado para administradores, com funcionalidades de gestão avançada. A interface foi desenhada de forma responsiva e acessível, com apoio de CSS personalizado e Bootstrap.

Durante o desenvolvimento, enfrentaram-se desafios relevantes como a proteção de rotas, sincronização de tokens entre cliente e servidor, gestão de permissões e tratamento de erros. Estes foram superados com soluções técnicas sólidas, como middlewares personalizados, rotas protegidas e tratamento unificado de exceções no cliente e no servidor.

No seu todo, o projeto resultou numa aplicação completa, segura e escalável, preparada para evoluir e adaptar-se a novas funcionalidades. A colaboração entre diferentes áreas do projeto integrado e a utilização de ferramentas modernas permitiram desenvolver um produto funcional com impacto potencial em contextos reais.

Bibliografia

1. **MDN Web Docs**. [Documentação de JavaScript, HTML e APIs Web]. Disponível em: <https://developer.mozilla.org/>
2. **React – A JavaScript library for building user interfaces**. Meta. Disponível em: <https://reactjs.org/>
3. **Express – Fast, unopinionated, minimalist web framework for Node.js**. Disponível em: <https://expressjs.com/>
4. **Sequelize ORM**. Documentação oficial. Disponível em: <https://sequelize.org/>
5. **PostgreSQL**. Base de dados relacional. Disponível em: <https://www.postgresql.org/docs/>
6. **Azure Blob Storage Documentation**. Microsoft. Disponível em: <https://learn.microsoft.com/en-us/azure/storage/blobs/>
7. **W3schools**. Disponível em: <https://w3schools.com>
8. **Documentação fornecida pelo docente da unidade curricular** – Materiais da disciplina Aplicações para a Internet II, incluindo exemplos de código, guias de boas práticas e enunciado do trabalho prático. Escola Superior de Tecnologia e Gestão de Viseu, 2024/2025.