

**Relatório do Website Dinâmico para o Instituto Superior de
Comunicação e Imagem de Moçambique (ISCIM)**

Desenvolvedor: Nelo Madeira Machel

Maputo, 2026

Lista de figuras

Figura 1: Diagrama de classes do sistema	17
Figura 2: <i>Carousel</i> principal.....	18
Figura 3: Secção de notícias	19
Figura 4: Secção de testemunhos	20
Figura 5: Página de cursos.....	21
Figura 6: Página de notícias.....	22
Figura 7: Tela de <i>Login</i> do painel administrativo.....	23
Figura 8: Tela Inicial do Sistema (Painel de Controle)	24
Figura 9: Tela de visualização de cursos	25
Figura 10: Formulário para a criação de curso	26
Figura 11: <i>Modal</i> com mensagem de sucesso após a criação de curso	27
Figura 12: Formulário de edição de curso	28
Figura 13: <i>Modal</i> com mensagem de sucesso após actualização do curso	29
Figura 14: <i>Modal</i> para confirmação de exclusão de curso	30
Figura 15: <i>Modal</i> com mensagem de sucesso após exclusão de curso	31
Figura 16: Tela de gestão de utilizadores do sistema	32
Figura 17: <i>Modal</i> de adição de utilizadores do sistema	32

Lista de tabelas

Tabela 1: Comparação entre <i>frameworks</i> de desenvolvimento customizado (<i>Laravel</i> , <i>CodeIgniter</i> , <i>Symfony</i> e <i>Express.JS</i>).....	6
Tabela 2: Tecnologias utilizadas para o desenvolvimento	13
Tabela 3: Funcionalidades do sistema	14

Lista de abreviaturas

API	<i>Application Programming Interface</i>
CMS	<i>Content Management System</i>
CSRF	<i>Cross-Site Request Forgery</i>
CSS	<i>Cascading Style Sheets</i>
HTML	<i>Hyper Text Markup Language</i>
HTTP	<i>Hyper Text Transfer Protocol</i>
ISCIM	Instituto Superior de Comunicação e Imagem de Moçambique
I/O	<i>Input/Output</i>
JS	<i>JavaScript</i>
MVC	<i>Model, View, Controller</i>
MySQL	<i>My Structured Query Language</i>
ORM	<i>Object-Relational Mapping</i>
PDF	<i>Portable Document Format</i>
PHP	<i>Hypertext Preprocessor</i>
SGBD	Sistema de Gestão de Base de Dados
SQL	<i>Structured Query Language</i>
UML	<i>Unified Modeling Language</i>
WWW	<i>World Wide Web</i>
XSS	<i>Cross-Site Scripting</i>

RESUMO

A transformação digital no ensino superior exige soluções inovadoras que melhorem a comunicação e o acesso à informação. Este trabalho apresenta o desenvolvimento de um *website* dinâmico para o Instituto Superior de Comunicação e Imagem de Moçambique, com o objectivo de melhorar a comunicação institucional e facilitar o acesso à informação. O projecto visa criar um painel administrativo intuitivo que permita a actualização em tempo real do conteúdo do *site*. O desenvolvimento foi realizado utilizando o *framework* **Laravel** no *back-end* e o **Bootstrap** no *front-end*, proporcionando uma *interface* responsiva e de fácil usabilidade. Para a modelação do sistema recorreu-se a **diagramas UML**, que possibilitaram a representação estruturada dos casos de uso, fluxos de interacção e arquitectura da solução. Utiliza-se uma pesquisa aplicada, exploratória e qualitativa, com base em revisão bibliográfica, entrevistas e questionários dirigidos a estudantes e responsáveis pelo *site*. Os resultados demonstram que o *website* anterior possuía limitações na actualização de conteúdos e na interacção com os utilizadores, exigindo o uso excessivo de canais convencionais por parte dos estudantes. A nova solução proposta inclui uma arquitectura em camadas e uma *interface* responsiva. A unificação da gestão de conteúdos num painel administrativo melhora a eficiência e reduz a dependência de conhecimentos técnicos por parte dos funcionários, fazendo com que eles se foquem mais em tarefas mais produtivas.

Palavras-chave: *website* dinâmico, painel administrativo, sistema de gestão de conteúdo.

ABSTRACT

Digital transformation in higher education requires innovative solutions that enhance communication and access to information. This work presents the development of a dynamic website for the Instituto Superior de Comunicação e Imagem de Moçambique, aiming to improve institutional communication and facilitate access to information. The project seeks to create an intuitive administrative dashboard that allows real-time content updates on the website. Development was carried out using the Laravel framework for the back-end and Bootstrap for the front-end, providing a responsive and user-friendly interface. UML diagrams were used for system modeling, enabling structured representation of use cases, interaction flows, and system architecture. An applied, exploratory, and qualitative research methodology was adopted, based on literature review, interviews, and questionnaires addressed to students and site administrators. The results show that the previous website had limitations in content updates and user interaction, requiring excessive reliance on conventional communication channels by students. The proposed solution includes a layered architecture and a responsive interface. The unified content management through an administrative dashboard improves efficiency and reduces dependency on technical knowledge among staff, allowing them to focus on more productive tasks.

Keywords: dynamic website, administrative dashboard, content management system.

Índice

Lista de figuras	i
Lista de tabelas	ii
Lista de abreviaturas	iii
RESUMO	iv
ABSTRACT	v
CAPÍTULO I – INTRODUÇÃO.....	1
1.1. Contextualização.....	1
1.2. Problemática	1
1.3. Objectivos	3
1.3.1. Geral	3
1.3.2. Específicos.....	3
1.4. Justificativa	3
CAPÍTULO II – REVISÃO DA LITERATURA	5
2.1. Sistemas de Gestão de Conteúdo (CMS) em Instituições Acadêmicas	5
2.2. Limitações dos CMS para o problema em estudo	5
2.3. <i>Frameworks</i> para desenvolvimento <i>web</i>	6
2.3.1. <i>Framework</i>	6
2.4. Justificação da Abordagem	8
2.4.1. Porque não optar por CMS	8
2.5. Conceitos Fundamentais	9
2.5.1. <i>Website</i> dinâmico.....	9
2.5.2. Sistema de Gestão de Base de Dados (SGBD).....	9
2.5.3. APIs	9
CAPÍTULO III – METODOLOGIAS.....	10

3.	Classificação da pesquisa	10
3.1.	Quanto a natureza da pesquisa.....	10
3.2.	Quanto aos objectivos da pesquisa	10
3.3.	Quanto a abordagem da pesquisa.....	10
3.4.	Quanto ao procedimento da pesquisa	11
3.5.	Metodologia de Desenvolvimento	11
3.5.1.	Fases da Metodologia de Desenvolvimento	11
4.	Desenvolvimento do Sistema	13
4.1.	Arquitectura do Sistema.....	13
4.2.	Tecnologias Utilizadas.....	13
4.3.	Funcionalidades do Sistema.....	14
4.4.	Fluxo do Sistema	16
4.5.	Diagrama de classes do sistema (base de dados)	17
4.6.	Interfaces do Sistema	18
4.6.1.	Interfaces do <i>website</i> público (utilizador comum)	18
4.6.2.	Interfaces Privadas (dashboard).....	23
5.	Conclusão	33
6.	Recomendações	34
	Referências Bibliográficas.....	35

CAPÍTULO I – INTRODUÇÃO

1.1. Contextualização

Com o crescimento exponencial da informação a presença *online* é crucial para instituições de ensino que buscam oferecer serviços eficientes e acessíveis aos seus utilizadores, acresce a dificuldade de gerir e organizar, de forma a ser recuperada, a informação produzida. Por outro lado, a evolução tecnológica tem alterado as metodologias e processos de trabalho das organizações, resultando na necessidade de implementar novas ferramentas de gestão e organização da informação (Neves, 2010).

O Instituto Superior de Comunicação e Imagem de Moçambique (ISCIM) não é excepção a esses desafios. O *website* actual apresenta uma estrutura estática que limita a agilidade na actualização de informações e a personalização da experiência do utilizador. Com o aumento da procura por interacções *online* mais dinâmicas e acessíveis, surge a necessidade de uma solução tecnológica que optimize a gestão de conteúdo e forneça suporte automatizado aos utilizadores.

Diante disso, seria proposto o **desenvolvimento de um novo *website* dinâmico para o ISCIM** por meio da implementação de um ***dashboard* administrativo** permitindo que o *website* dinâmico facilite a actualização contínua de informações relevantes.

Este estudo tem como objectivo principal o desenvolvimento de um novo *website* dinâmico para o ISCIM. O estudo possui a capacidade de proporcionar uma interacção mais dinâmica e personalizada entre os utilizadores e o *website* do ISCIM. A utilização de técnicas avançadas do *website* dinâmico não apenas modernizaria a infraestrutura digital da instituição, mas também potencializaria a acessibilidade às informações e serviços oferecidos, tornando-os mais acessíveis e eficientes para um público diversificado.

1.2. Problemática

Diante da crescente demanda por informações rápidas e precisas, o Instituto Superior de Comunicação e Imagem de Moçambique (ISCIM) enfrenta o desafio de otimizar seu *website* para melhor atender às necessidades dos utilizadores. Surge, então,

a seguinte questão: **Como melhorar o acesso dos estudantes e encarregados às informações do ISCIM por meio de um *website* dinâmico?**

- Actualmente, estudantes e encarregados do ISCIM enfrentam dificuldades de acesso e localização de informações no *website*. A estrutura estática do *website* dificulta uma navegação eficiente e intuitiva, tornando-se um obstáculo para encontrar informações essenciais, como detalhes de matrícula, calendário acadêmico, regulamentos internos e outros comunicados importantes.

E com isso a falta de organização e disponibilidade otimizada dos conteúdos impacta negativamente na experiência do utilizador, gerando frustrações.

- Os funcionários enfrentam dificuldades significativas para actualizar o conteúdo de forma rápida e eficiente. A ausência de um *dashboard* administrativo para a gestão de conteúdo impede actualizações em tempo real, resultando na manutenção de informações desactualizadas. Esse cenário propicia a disseminação de dados imprecisos ou defasados, enquanto o processo de actualização da informação do *website* depende de intervenções manuais e de conhecimentos técnicos específicos, dificultando a comunicação efectiva entre a instituição e sua comunidade.

Tendo em conta que do momento não existe ninguém responsável pela actualização regular de todos dados do *website*, o que faz com que haja problemas na Instituição no Geral, como:

- A falta de interactividade no *website*;
- A ausência de mecanismos que ofereçam suporte imediato aos utilizadores que buscam assistência.
- O uso excessivo de canais de atendimento convencionais, como telefone ou *email*, leva à sobrecarga dos serviços de suporte e pode resultar em atrasos no atendimento.

E com tudo isto, também possui um impacto na eficiência operacional e na imagem institucional, pela falta de agilidade e precisão na disponibilização de informações e prejudica a eficiência operacional o que traz consequências negativas tanto na eficiência interna quanto na satisfação do público externo.

1.3. Objectivos

1.3.1. Geral

Desenvolver um *website* dinâmico com *dashboard* administrativo que possibilite a actualização em tempo real das informações do ISCIM para a melhoria da comunicação institucional.

1.3.2. Específicos

- ❖ Descrever o *website* actual do Instituto Superior de Comunicação e Imagem de Moçambique;
- ❖ Modelar e codificar o *website* dinâmico e o *dashboard* administrativo, que seja responsivo e ofereça uma experiência de uso adequada para diferentes dispositivos;
- ❖ Apresentar uma solução integrada em que o *dashboard* administrativo gere o conteúdo do *website* em tempo real.

1.4. Justificativa

A escolha deste tema justificou-se pela necessidade de um *website* dinâmico com um *dashboard* administrativo voltado exclusivamente para responder questões relacionadas à instituição. Essa abordagem visa facilitar o acesso à informação pelos utilizadores e proporcionar maior flexibilidade na disponibilização de dados importantes e actualizados por parte dos funcionários, reduzindo consideravelmente o tempo de resposta para a obtenção de informações básicas.

Para os estudantes, o *website* dinâmico permitirá que eles tenham acesso à informação em tempo real, pois com o *dashboard* os funcionários poderão fazer actualizações constantes da informação, os estudantes encontrarão todas as informações necessárias em um único lugar (centralização de dados), o que facilita o acesso a consulta. Com acesso rápido a informações e suporte automatizado, os estudantes podem resolver suas dúvidas de forma independente, promovendo a autonomia no processo de aprendizagem. A facilidade de acesso a dados essenciais reduz o tempo gasto em buscas, permitindo que os estudantes se concentrem mais em suas actividades académicas.

Para os encarregados, o *website* dinâmico proporciona maior **transparência e acompanhamento académico**. Eles podem aceder rapidamente a informações importantes sobre o calendário escolar, comunicados oficiais, propinas e eventos da instituição. Isso reduz a necessidade de deslocamentos à faculdade para obter informações básicas e facilita a comunicação entre instituição e encarregados. Com informações centralizadas e actualizadas em tempo real, os encarregados podem apoiar melhor os seus educandos, acompanhando de forma mais próxima o percurso académico.

Para os funcionários, a optimização do *website* do ISCIM traria benefícios significativos, pois iria proporcionar um sistema dinâmico no qual as actualizações de conteúdo poderiam ser feitas de forma rápida e eficiente sem precisar de um conhecimento amplamente técnico por conta da facilidade de uso. Com essa ferramenta os funcionários teriam uma melhoria na comunicação (informações actualizadas em tempo real), e o aumento da produtividade, ao automatizar processos e simplificar a gestão de conteúdo, os funcionários poderiam se concentrar em tarefas mais estratégicas e produtivas.

Por fim, haveria a **necessidade de uma solução tecnológica robusta que facilite a gestão interna de conteúdo e melhore a experiência do utilizador**. A implementação de um *dashboard* administrativo para a gestão de conteúdo permitiria que a equipe administrativa do ISCIM tenha mais autonomia para actualizar informações sem depender de técnicos especializados, agilizando o processo de publicação de novos conteúdos.

CAPÍTULO II – REVISÃO DA LITERATURA

2.1. Sistemas de Gestão de Conteúdo (CMS) em Instituições Acadêmicas

Estudos como o de Neves (2010) demonstram que, o crescente volume de informação, associado à desmaterialização da informação, tem revelado dificuldades acrescidas às organizações em geral, na gestão da informação publicada na *web*. A implementação de sistemas de gestão de conteúdos permite descentralizar todos os processos de gestão de conteúdo, desde a criação até à publicação, revertendo na optimização dos serviços prestados.

Diferentes instituições acadêmicas utilizam plataformas de CMS, como *WordPress*, *Joomla* e *Drupal*, para gerir o conteúdo de seus *websites*.

2.2. Limitações dos CMS para o problema em estudo

Apesar da popularidade e facilidade de uso de sistemas de gestão de conteúdos (CMS), como *WordPress*, *Joomla* e *Drupal*, estes apresentam algumas limitações que comprometem a sua adequação às necessidades do ISCIM. Entre as principais limitações destacam-se:

- **Dependência de *plugins* externos para funcionalidades adicionais:** A maioria dos CMS, como *WordPress* e *Joomla*, necessita de *plugins* ou extensões para oferecer funcionalidades específicas além do que vem por padrão. Essa dependência implica que, para atender às necessidades do ISCIM (como *dashboards* personalizados ou gestão de utilizadores diferenciada), seria necessário instalar e manter vários *plug-ins*, podendo aumentar a complexidade da manutenção e limitar a personalização;
- **Risco de vulnerabilidades devido ao uso massivo e *plugins* de terceiros:** Sendo os CMS amplamente utilizados no mundo, tornam-se alvos frequentes de ataques cibernéticos. Muitos desses ataques exploram falhas de segurança em *plugins* ou temas desenvolvidos por terceiros. Isso significa que a segurança do *website* do ISCIM poderia ficar comprometida, principalmente se dependesse de extensões não oficiais ou pouco actualizadas;
- **Estrutura menos flexível para customizações profundas:** Embora os CMS permitam certa personalização, a sua arquitectura limita alterações profundas,

pois muitas funcionalidades estão rigidamente ligadas ao núcleo do sistema. Isso dificulta a criação de soluções sob medida para o ISCIM, como módulos administrativos específicos ou fluxos de trabalho personalizados, sem recorrer a modificações complexas e pouco sustentáveis;

2.3. Frameworks para desenvolvimento web

2.3.1. Framework

Um *framework* é uma estrutura de *software* baseada nos princípios da programação orientada a objectos, que utiliza abstração, herança e polimorfismo para fornecer um conjunto de classes genéricas e esqueletos de código reutilizáveis. Esses "esqueletos" definem quais funções e características serão implementadas de forma específica na aplicação, permitindo que o programador adicione apenas o código exclusivo do projecto. Dessa forma, o *framework* facilita a construção de aplicações robustas, promovendo a reutilização e a flexibilidade no desenvolvimento (Assis, 2021). São exemplos de *frameworks web*: o *Laravel*, o *CodeIgniter*, o *Symfony* e o *Express.JS*.

i. Framework selecionado para a solução

Tabela 1: Comparação entre *frameworks* de desenvolvimento customizado (*Laravel*, *CodeIgniter*, *Symfony* e *Express.JS*)

Característica	<i>Laravel</i>	<i>CodeIgniter</i>	<i>Symfony</i>	<i>Express.js</i>
Arquitetura	MVC.	MVC, mais simples e menos estruturado.	MVC, mais robusto e modular.	<i>Event-driven, single-threaded, non-blocking.</i>
Facilidade de uso	Fácil de aprender, boa documentação e suporte.	Mais difícil de aprender, menos recursos e comunidade.	Mais complexo e detalhado, requer mais configuração.	Requer conhecimento em <i>JavaScript</i> , leve e flexível.

Performance	Bom desempenho, otimizado com <i>cache</i> , <i>Eloquent ORM</i> .	Desempenho superior em alguns casos devido à sua leveza.	Pode ser mais lento em tarefas complexas devido à sua flexibilidade.	Muito rápido devido à sua arquitetura assíncrona.
Escalabilidade	Muito escalável, ideal para projectos grandes.	Menos escalável em grandes projectos.	Altamente escalável, mas complexo para configurar.	Extremamente escalável, ideal para aplicações com alta carga de I/O.
Recursos e funcionalidades	Grande conjunto de funcionalidades integradas.	Menos funcionalidades integradas, mais voltado para soluções simples.	Funcionalidades muito extensas e flexíveis, mas complexas.	Focado em aplicações <i>JavaScript</i> , ideal para I/O em tempo real.

Para este problema de pesquisa, foi selecionado o *Laravel* para o desenvolvimento do *website* para o ISCIM, pelos seguintes motivos:

- Facilidade de aprendizado, sua sintaxe clara e uma documentação clara e extensa;
- Oferece uma API que facilita a gestão da base de dados;
- Possui funcionalidades integradas que deixam o desenvolvimento mais eficiente;
- Possui suporte a testes, tornando o desenvolvimento mais robusto e seguro;
- Possui uma segurança robusta (já vem com protecção contra ataques *CSRF*, *SQL Injection* e *XSS*);
- Tem mais suporte e soluções prontas em comparação com os concorrentes.

2.4. Justificação da Abordagem

2.4.1. Porque não optar por CMS

Tendo em conta as limitações identificadas, para este problema de pesquisa seria mais adequado optar pelo desenvolvimento do *website* a partir de um *framework* ao invés de um CMS já existente. Essa escolha justifica-se pelos seguintes motivos:

- ❖ **Personalização total e maior flexibilidade:** Com um *framework* pode-se desenvolver um sistema exactamente conforme as necessidades do ISCIM, criando um *dashboard* administrativo sob medida para gerir notícias, cursos e outras informações sem as limitações de um CMS como o facto de algumas funcionalidades não serem gratuitas.
- ❖ **Melhor performance e eficiência:** Um CMS carrega muitos recursos que podem não ser necessários para a faculdade, tornando o *site* mais pesado e menos eficiente. Com um *framework*, pode-se criar apenas o que é essencial, o que garante um *site* mais rápido e optimizado.
- ❖ **Segurança reforçada:** CMSs populares, como o *WordPress*, são alvos frequentes de ataques devido ao uso massivo e à dependência de *plugins* de terceiros, que podem ter falhas de segurança. Um *framework* permite que sejam implementadas práticas de segurança personalizadas, reduzindo riscos.
- ❖ **Estrutura escalável para crescimento futuro:** Se o ISCIM precisar de novas funcionalidades ou uma integração com outras plataformas, um *framework* facilitaria essa expansão, sem as restrições que um CMS pode impor.
- ❖ **Melhor gestão de utilizadores e permissões:** Com um *framework*, você pode criar um sistema de gestão de utilizadores personalizado, permitindo que diferentes administradores tenham permissões específicas para actualizar conteúdos, evitando problemas como exclusão acidental de páginas ou conteúdos importantes.

2.5. Conceitos Fundamentais

2.5.1. *Website* dinâmico

Para Miguel (2019), o termo “*website* dinâmico” não se refere apenas a efeitos visuais ou animações, mas à capacidade do *site* de gerar e apresentar conteúdos de forma personalizada e interactiva. Em contraste com um *site* estático, cujo conteúdo permanece fixo, um *site* dinâmico constrói suas páginas em tempo real com base nas interações do utilizador e nas informações armazenadas em um banco de dados. Dessa forma, as selecções ou acções realizadas pelos visitantes influenciam directamente os dados exibidos na página. Para desenvolver esse tipo de *site*, é necessário ter conhecimento em linguagens de programação e gestão de bancos de dados ou utilizar *softwares* que integrem essas tecnologias de forma automatizada.

2.5.2. Sistema de Gestão de Base de Dados (SGBD)

Um sistema gestor de banco de dados (SGBD) é uma colecção de programas que permite aos utilizadores criar e manter um banco de dados. O SGBD é, portanto, um sistema de *software* de propósito geral que facilita os processos de definição, construção, manipulação e compartilhamento de bancos de dados entre vários utilizadores e aplicações. A definição de um banco de dados implica especificar os tipos de dados, as estruturas e as restrições para os dados a serem armazenados em um banco de dados (Elmasri & Navathe, 2005).

Vantagens do uso: restringir acesso não autorizado, garantir o armazenamento persistente para objectos em programas, garantir o armazenamento de estruturas para o processamento eficiente de consultas, garantir *backup* e restauração e fornecer múltiplas *interfaces* para os utilizadores.

2.5.3. APIs

As APIs ou *Interfaces* de Programação de Aplicativos (*Application Programming Interface*) são uma ferramenta que facilita a criação de um aplicativo com base em um esboço pré-programado pelo seu idealizador. Na actualidade, pode-se encontrar muitas destas aplicações disponíveis, cada uma com características e funcionalidades genéricas ou bem específicas (Junior & Carvalho, 2018).

CAPÍTULO III – METODOLOGIAS

3. Classificação da pesquisa

É sabido que toda e qualquer classificação se faz mediante algum critério. Com relação às pesquisas, é usual a classificação com base em seus objectivos gerais (Gil, 2002).

A pesquisa científica é uma actividade humana, cujo objectivo é conhecer e explicar os fenómenos, fornecendo respostas às questões significativas para a compreensão da natureza. Para essa tarefa, o pesquisador utiliza o conhecimento anterior acumulado e manipula cuidadosamente os diferentes métodos e técnicas para obter resultado pertinente às suas indagações (Prodanov & Freitas, 2013).

3.1. Quanto a natureza da pesquisa

Quanto à natureza a pesquisa que melhor se aplica a realização deste projecto é **aplicada**, pois busca solucionar um problema real (a dificuldade na actualização de informações no *website* da instituição, devido à falta de um sistema intuitivo que permita aos funcionários gerir conteúdos de forma eficiente e sem necessidade de conhecimento técnico avançado) enfrentado pelo Instituto Superior de Comunicação e Imagem de Moçambique (ISCIM).

3.2. Quanto aos objectivos da pesquisa

O presente projecto quanto aos objectivos é **exploratório**, pois busca compreender o contexto actual do *website* do ISCIM, identificando suas limitações e problemas, como a dificuldade de actualização das informações pelos funcionários. Além de, analisar possíveis soluções e discutir tecnologias apropriadas para o desenvolvimento de um novo *website* dinâmico com *dashboard* administrativo

3.3. Quanto a abordagem da pesquisa

O presente projecto adopta uma abordagem de **pesquisa qualitativa**, devido à natureza do problema investigado, que está relacionada à disponibilidade da informação actualizada e à facilidade de uso do *dashboard* administrativo pelos funcionários. Assim, será possível tomar decisões mais informadas e equilibradas sobre o desenvolvimento do *website* para o ISCIM e a implementação do *chatbot*.

3.4. Quanto ao procedimento da pesquisa

Os procedimentos técnicos utilizados na realização deste projecto de pesquisa foram a pesquisa **bibliográfica** e o **estudo de caso**. A pesquisa bibliográfica foi fundamental para a recolha de informações em livros, artigos científicos, teses e dissertações, com o objectivo de aprofundar o conhecimento sobre Inteligência Artificial, Processamento de Linguagem Natural, *Chatbots*, API's, sistemas de gestão de conteúdo e desenvolvimento de *websites*. O estudo de caso teve como foco o *website* do ISCIM, permitindo uma análise detalhada do seu funcionamento, das suas limitações e da experiência dos seus utilizadores. Para complementar essa análise, foram aplicados questionários aos estudantes que já tiveram acesso ao *website*, com o intuito de recolher percepções e opiniões que ajudassem a compreender as principais dificuldades enfrentadas e a propor melhorias adequadas.

3.5. Metodologia de Desenvolvimento

Diversas metodologias foram criadas para sistematizar o desenvolvimento de *software*. Estas podem ser divididas em **tradicionais**, que enfatizam a documentação e o planeamento detalhado de cada etapa, e **ágeis**, que priorizam a flexibilidade e entregas incrementais.

Para este projecto, adoptou-se a metodologia tradicional em **cascata** (*Waterfall*), que organiza o desenvolvimento em fases sequenciais e bem definidas. Segundo Almeida (2017), esta metodologia apresenta, entre outras características, documentação detalhada de cada etapa, papéis claramente definidos para os membros da equipe e padronização do processo de desenvolvimento.

3.5.1. Fases da Metodologia de Desenvolvimento

- i. **Análise de Requisitos:** Nesta fase foi feita uma entrevista com o responsável pelo *website* no Departamento de Informática, foram feitos questionários para levantar dificuldades no acesso à informação aos estudantes do ISCIM, foi feita a análise do *website* actual (baseado em *WordPress*) e o levantamento de funcionalidades desejadas (página inicial, notícias, cursos, gestor de utilizadores e permissões de acesso, área administrativa para gestão de conteúdo).

- ii. **Projecto (*design*) e Implementação (codificação):** Nesta fase foram feitos protótipos das páginas usando *Bootstrap* (*layouts* responsivos), definição da arquitectura da aplicação com *Laravel MVC*, *design* do banco de dados (tabelas, chaves, relacionamentos), desenvolvimento do *front-end* com HTML, CSS, JS e *Bootstrap* e o desenvolvimento do *back-end* com *Laravel*.

- iii. **Testes:** Nesta fase se verificou individualmente cada componente desenvolvido, para garantir que funcionem correctamente e avaliar o comportamento do sistema completo em relação aos requisitos iniciais, foram feitos testes funcionais para verificar se cada funcionalidade está correcta, testes de usabilidade (navegação amigável, *layout* adaptado) e correcções de *bugs* detectados.

- iv. **Implantação:** Nesta fase, como o sistema foi configurado e disponibilizado em um **ambiente de desenvolvimento local**, permitindo a simulação do funcionamento completo do *website* dinâmico com servidor local (*XAMPP*), foram feitos testes do sistema em condições controladas para verificar funcionalidade e *interface*.

- v. **Manutenção:** Nesta fase foram feitas actividades de acompanhamento e melhorias para futuras versões: actualização de conteúdos do *site* (notícias, cursos, *carousels*, testemunhos) e correcção de erros identificados após uso contínuo e testes adicionais.

4. Desenvolvimento do Sistema

4.1. Arquitectura do Sistema

Laravel é um *framework* baseado no padrão MVC (*Model-View-Controller*), o que significa que: *controller* gere todas as requisições do sistema, *model* processa a lógica de negócio e manipula os dados e a *view* é responsável pela apresentação dos resultados. Essa separação facilita a manutenção, a reutilização de código e a escalabilidade do projecto.

Como os componentes interagem:

- ❖ *Models* → base de dados;
- ❖ *Views* → interface (*Blade*);
- ❖ *Controllers* → lógica.

4.2. Tecnologias Utilizadas

Tabela 2: Tecnologias utilizadas para o desenvolvimento

Tecnologia	Papel
PHP (<i>Laravel</i>)	<i>Back-end</i> do projecto
HTML, CSS, <i>JavaScript</i> , <i>Bootstrap</i>	<i>Front-end</i> do projecto
<i>XAMPP</i>	Ambiente de desenvolvimento local
<i>MySQL</i>	Base de dados via <i>PHPMyAdmin</i>
<i>Visual Studio Code</i>	Editor de código

4.3. Funcionalidades do Sistema

Tabela 3: Funcionalidades do sistema

	Funcionalidade	Descrição
Sistema de autenticação	<i>Login</i>	<ul style="list-style-type: none"> • O actor acessa a tela de <i>login</i>. • Informa <i>email</i> e senha. • O sistema valida credenciais no banco de dados. a. Em caso de credenciais incorrectas o sistema exhibe mensagem de erro e permanece na tela de <i>login</i>. b. Se válidas, cria sessão autenticada e redirecciona para o <i>dashboard</i> conforme perfil.
	Registro de utilizadores (<i>admin</i>)	<ul style="list-style-type: none"> • Criar Utilizador: <ul style="list-style-type: none"> a. Administrador clica em “Gestão de utilizadores” menú. b. Selecciona “Criar utilizador”. c. Preenche formulário com nome, <i>email</i>, senha e selecciona função (Admin, Gestor de conteúdo) e clica em “salvar”. d. Sistema grava dados no banco, envia <i>email</i> com instruções de <i>login</i> e senha. • Editar Utilizador: <ul style="list-style-type: none"> a. Administrador visualiza lista de utilizadores cadastrados. b. Clica em “Editar” ao lado do utilizador desejado. c. Altera campos (nome, <i>email</i>, função) e Clica em “Actualizar”. d. Sistema salva alterações. • Excluir Utilizador: <ul style="list-style-type: none"> a. Na mesma tela de lista de utilizadores, administrador clica em “excluir” para remover o utilizador. b. Sistema exhibe <i>pop-up</i> de confirmação. c. Se confirmado, sistema remove registro. • Atribuir permissões: <ul style="list-style-type: none"> a. No menú de funções, administrador selecciona <i>checkboxes</i> de permissões para cada função.

		Ao salvar, sistema relaciona perfil ao utilizador.
Gestão de Conteúdo	Criação e publicação de conteúdo	<ul style="list-style-type: none"> • Criar Conteúdo: <ol style="list-style-type: none"> a. No menú do <i>dashboard</i>, selecciona a página a qual deseja criar conteúdo. b. Preenche os campos do formulário e, opcionalmente, faz <i>upload</i> de imagen. c. Clica em “Publicar”. d. Sistema valida campos obrigatórios e persiste registro no banco e exhibe mensagem de sucesso. • Editar Conteúdo: <ol style="list-style-type: none"> a. Selecciona no menú o conteúdo da página a qual deseja editar. b. Sistema carrega formulário preenchido com dados anteriores. c. Utilizador altera campos e clica em “Actualizar” e o sistema grava a versão actualizada. • Excluir Conteúdo: <ol style="list-style-type: none"> a. Lista de conteúdos: utilizador clica em “deletar” no item desejado. b. Sistema mostra caixa de confirmação (“Deseja realmente remover este conteúdo?”). c. Se confirmado, exclui. • Visualizar Conteúdo: Utilizador clica no menú a página a qual deseja para visualizar.

4.4. Fluxo do Sistema

Ao aceder ao sistema existem duas vertentes, a do utilizador comum e a do administrador ou do gestor de conteúdo. O utilizador comum apenas pode acessar as páginas públicas do *website* onde são disponibilizadas todas as informações para o público em geral ver e nas interfaces privadas apenas o administrador e o gestor de conteúdo tem acesso, em que precisam de credenciais para fazer o *login*. O fluxo do sistema das interfaces privadas segue-se de forma seguinte:

1. O utilizador (administrador ou gestor de conteúdo) acessa o *site*;
2. Preenche o formulário de *login* com as suas credenciais (*email* e *password*);
3. É redirecionado a tela inicial do sistema;
4. Seleciona o conteúdo que deseja gerir (*carousel*, testemunho, curso ou notícias);
5. Preenche o formulário com os dados;
6. Clica o botão para adicionar/actualizar;
7. É redirecionado para a tela de listagem do conteúdo a gerir;
8. Retorna com uma *modal* de confirmação.

4.5. Diagrama de classes do sistema (base de dados)

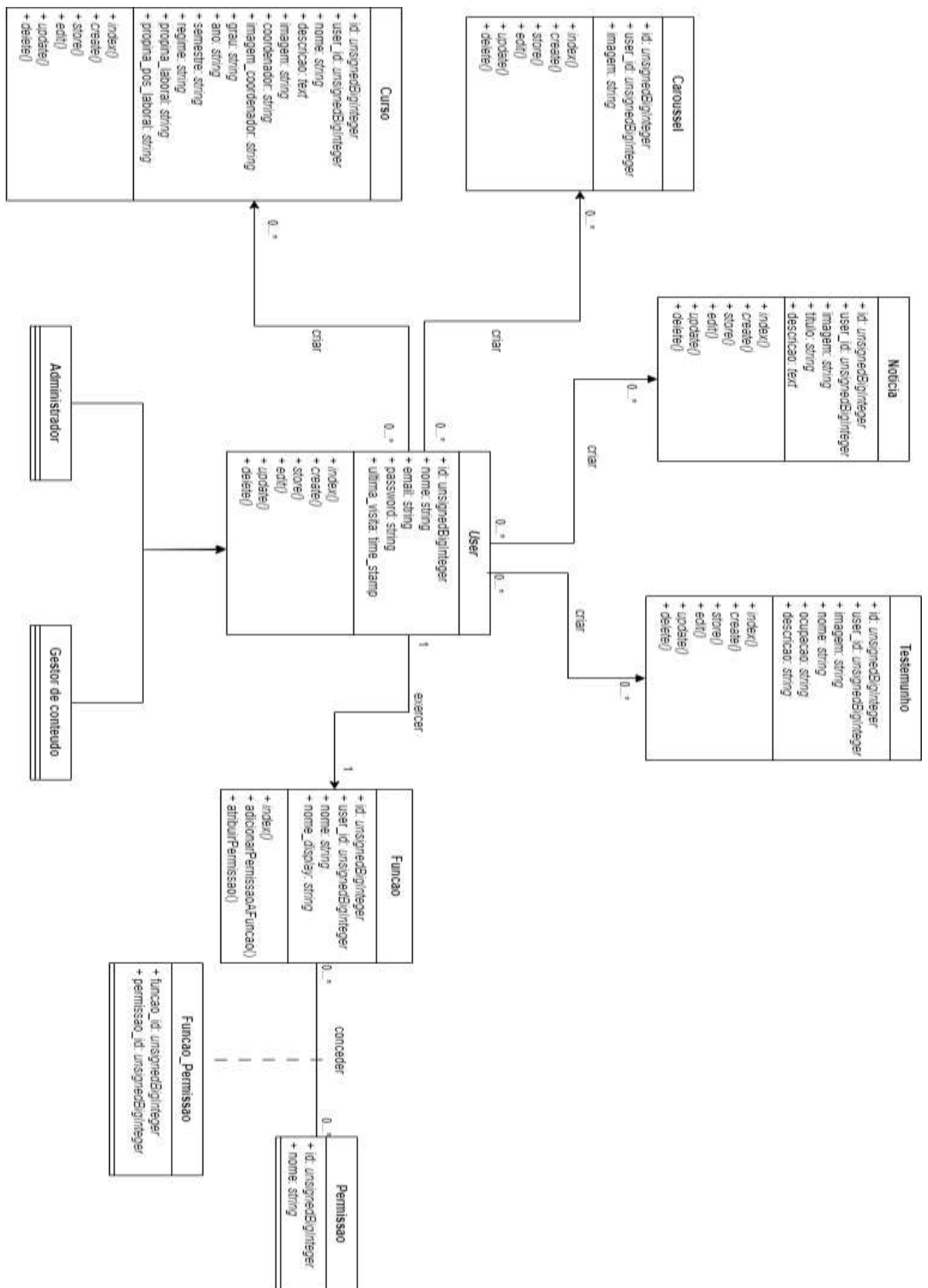


Figura 1: Diagrama de classes do sistema

4.6. Interfaces do Sistema

4.6.1. Interfaces do *website* público (utilizador comum)

Esta secção apresenta as páginas acessíveis ao visitante do *website*, sem autenticação. As páginas existentes nesta secção são:

Página inicial: A página inicial do *website* apresenta os principais elementos informativos e visuais destinados a facilitar o acesso imediato às informações mais relevantes para os utilizadores. Esta página foi organizada em diferentes secções, cada uma com um propósito específico, conforme descrito a seguir.

- ❖ **Carousel principal:** A figura 2 apresenta o *carousel* principal da página inicial. Este componente exibe imagens institucionais em rotação automática, destacando anúncios, campanhas, eventos académicos e mensagens importantes dirigidas aos estudantes e visitantes do *website*.

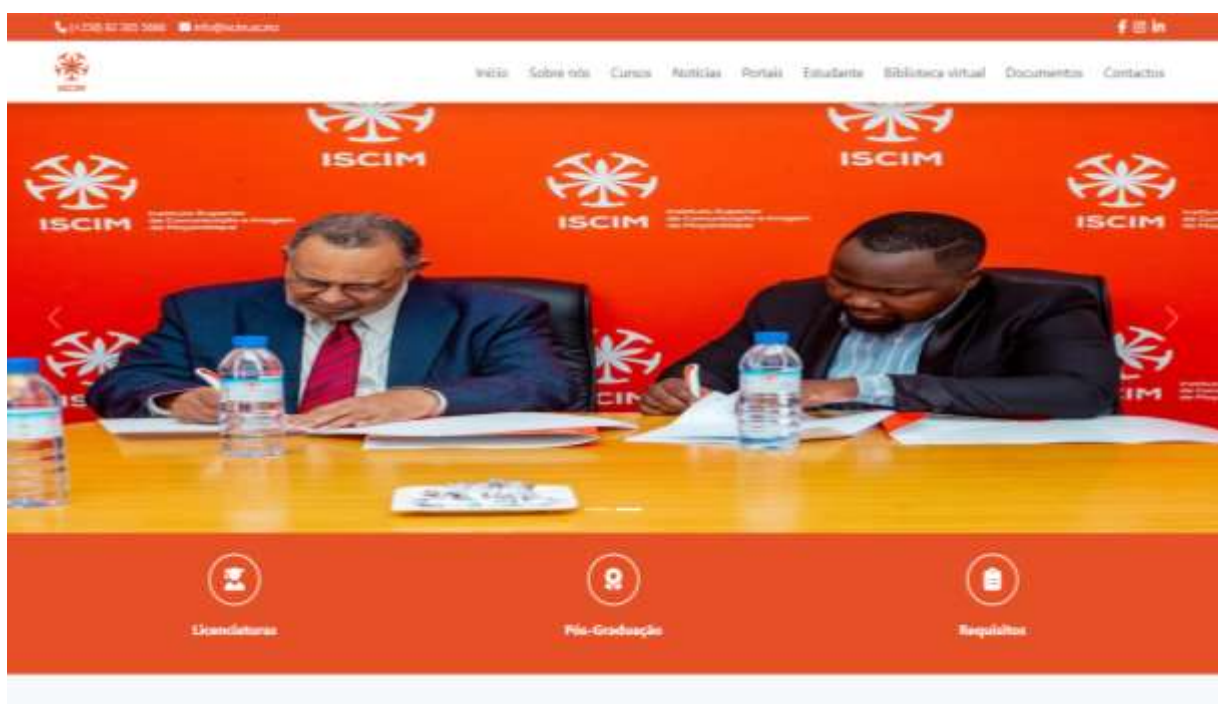


Figura 2: *Carousel* principal

- ❖ **Secção de notícias:** Na Figura 3 observa-se a secção de notícias, localizada imediatamente após o *carousel*. Esta área apresenta de forma dinâmica as **três últimas notícias publicadas** pela instituição, permitindo ao utilizador acompanhar rapidamente actualizações recentes, eventos académicos e comunicados.



Figura 3: Secção de notícias

- ❖ **Secção de testemunhos:** A Figura 4 ilustra a secção de testemunhos, composta por depoimentos de estudantes que partilham as suas experiências e percepções sobre a instituição. Esta secção contribui para reforçar a credibilidade e reputação do ISCIM, apresentando opiniões reais que ajudam a criar uma conexão mais próxima com futuros candidatos e visitantes do *website*.



Figura 4: Secção de testemunhos

Página de cursos: A página de cursos (Figura 5) apresenta a listagem completa de todos os cursos oferecidos pelo ISCIM. Os utilizadores podem navegar facilmente entre os cursos, acedendo a detalhes específicos de cada um. Esta página tem como objectivo fornecer informações claras e organizadas sobre a oferta académica, auxiliando tanto estudantes actuais quanto potenciais candidatos a tomar decisões informadas.

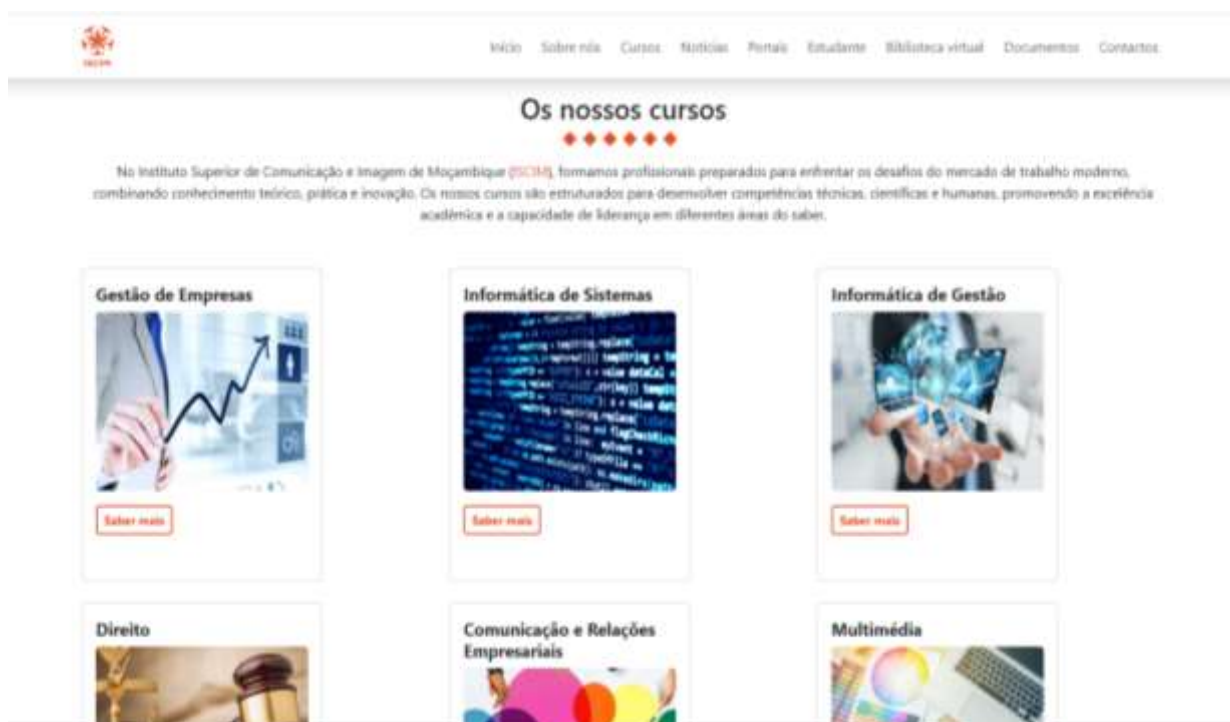


Figura 5: Página de cursos

Página de notícias: Além das três últimas notícias exibidas na página inicial, a página de notícias (Figura 6) permite ao utilizador aceder ao **arquivo completo de publicações**, incluindo notícias passadas, eventos e comunicados institucionais. Cada notícia apresenta título, imagem, resumo e data de publicação, possibilitando uma navegação mais detalhada e facilitando o acompanhamento da vida académica e institucional do ISCIM.

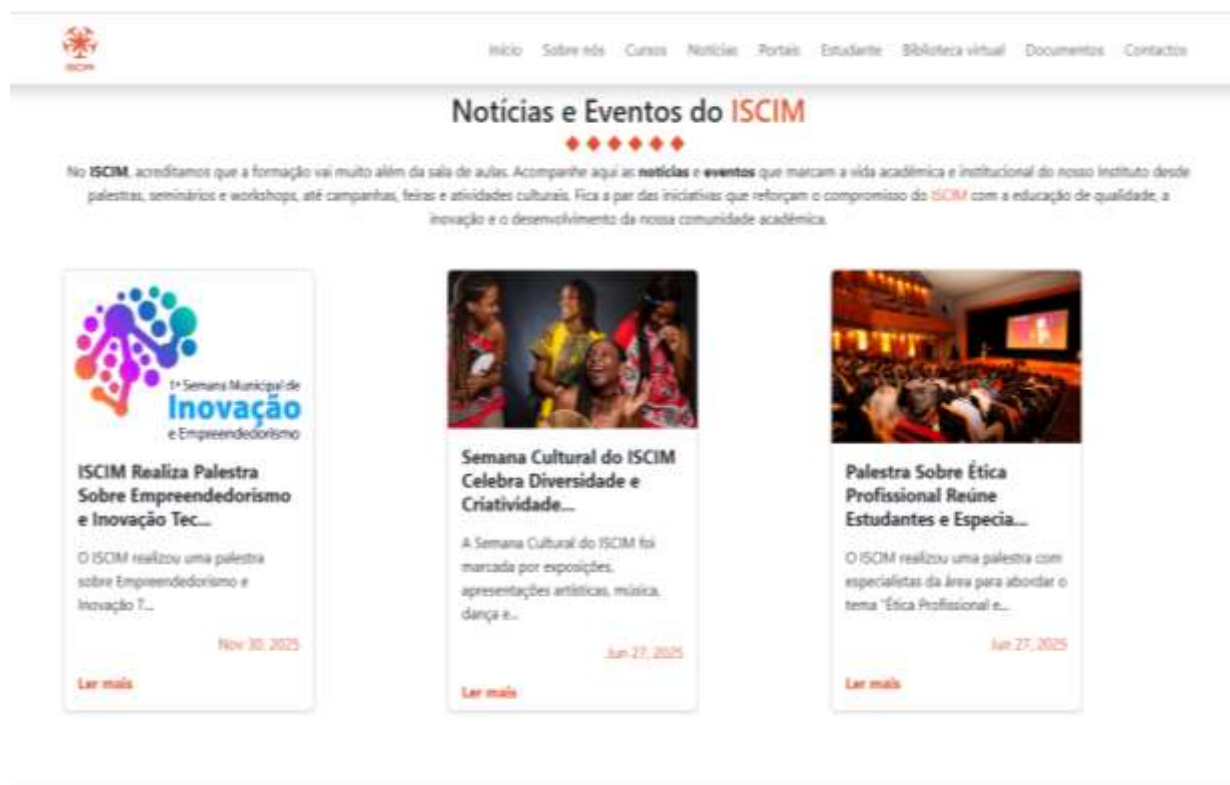


Figura 6: Página de notícias

4.6.2. Interfaces Privadas (dashboard)

No presente subcapítulo é mostrado de forma visual o sistema e suas principais funcionalidade de uso, que são as de actualizar e publicar conteúdo e gerir utilizadores e suas permissões. De início deve ter um administrador predefinido com a função e sua permissão para permitir o início de sessão no painel administrativo.

Tela de Login: Ao aceder ao sistema a partir do *link* de acesso, o utilizador será direccionado a página do *login*, que é ilustrado na figura 7, que é o ponto de partida do sistema. Com as credenciais facultadas, o utilizador pode preencher os campos, *email* e senha, em seguida, clicar no botão entrar.

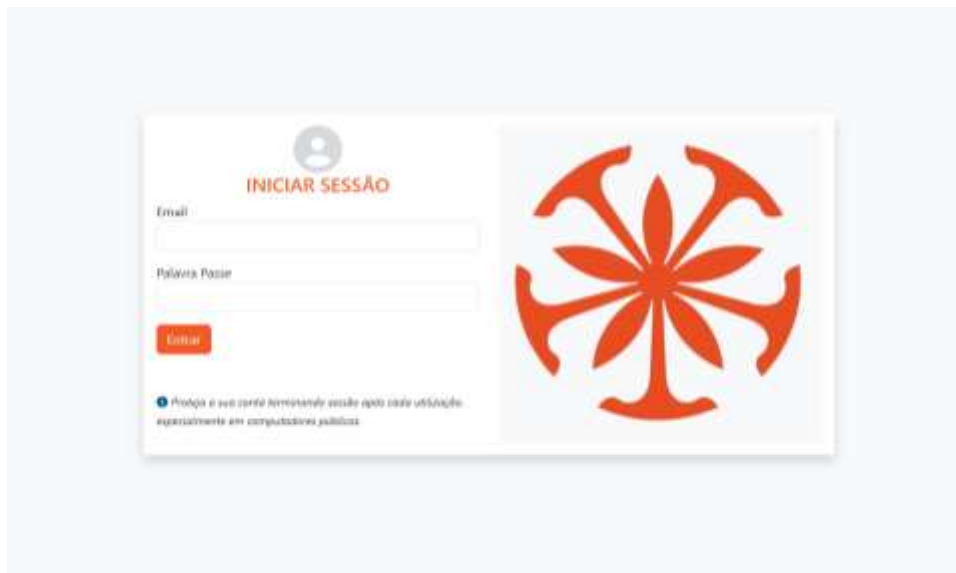


Figura 7: Tela de *Login* do painel administrativo

Tela inicial: Após o processo de autenticação, o utilizador é redireccionado a página principal do sistema, que é ilustrada na figura 8. O conteúdo visualizado nesta página, varia de acordo com o nível de acesso de cada utilizador, acedendo como administrador, o utilizador terá acesso a todas as abas, páginas e funcionalidades do sistema. O sistema é constituído por quatro áreas principais, nomeadamente: painel de navegação lateral (*Side bar*), painel de navegação superior (*Navbar*), o rodapé e o ambiente de trabalho. A *side bar* permite ao utilizador aceder ao painel de controle (*Dashboard*), gerir *carousels* e testemunhos da página *home* do *website*; gerir cursos, gerir notícias, gerir *chatbot*, gerir utilizadores, atribuir funções e permissões. O painel de navegação superior dá acesso ao perfil do utilizador e possibilita o *logout*, e o ambiente de trabalho é o local onde são renderizados os conteúdos dinâmicos do sistema.



Figura 8: Tela Inicial do Sistema (Painel de Controle)

Interfaces de CRUD de conteúdo (Cursos): Na *Side bar* ao clicarmos em “cursos”, somos redireccionados a página de visualização de todos os cursos criados, apresentada na figura 9, com a possibilidade de criar mais cursos, editar cursos previamente adicionados e deletar cursos, ao fazer operações de criação e actualização, aparece registrado o nome do utilizador que fez a última alteração a um determinado curso.

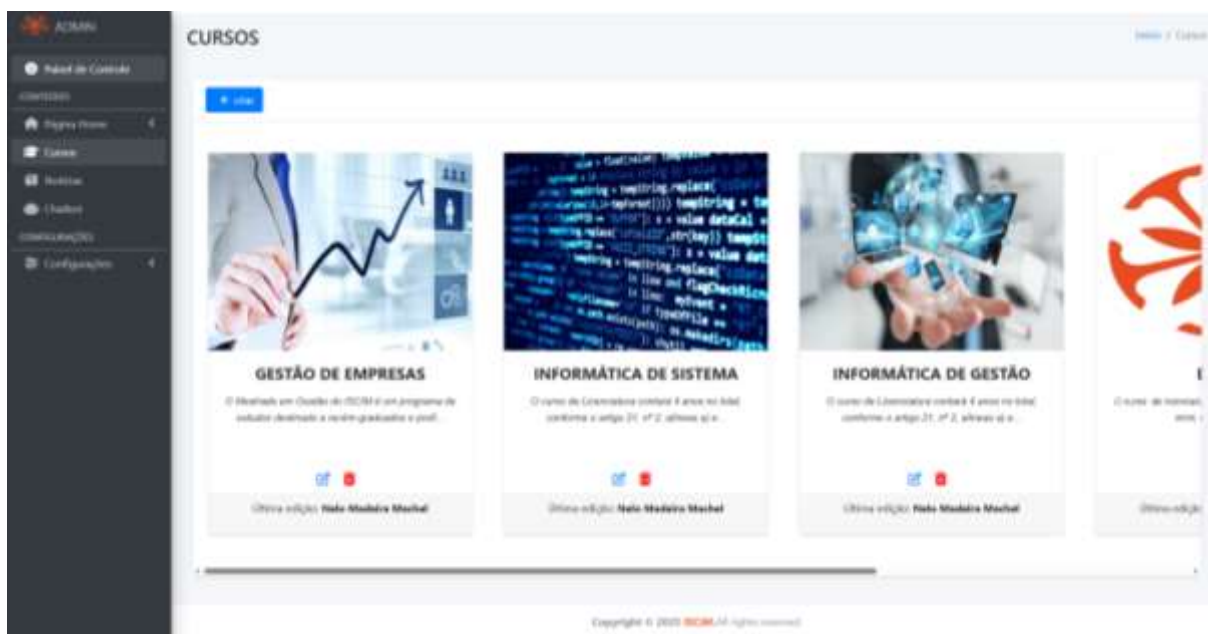
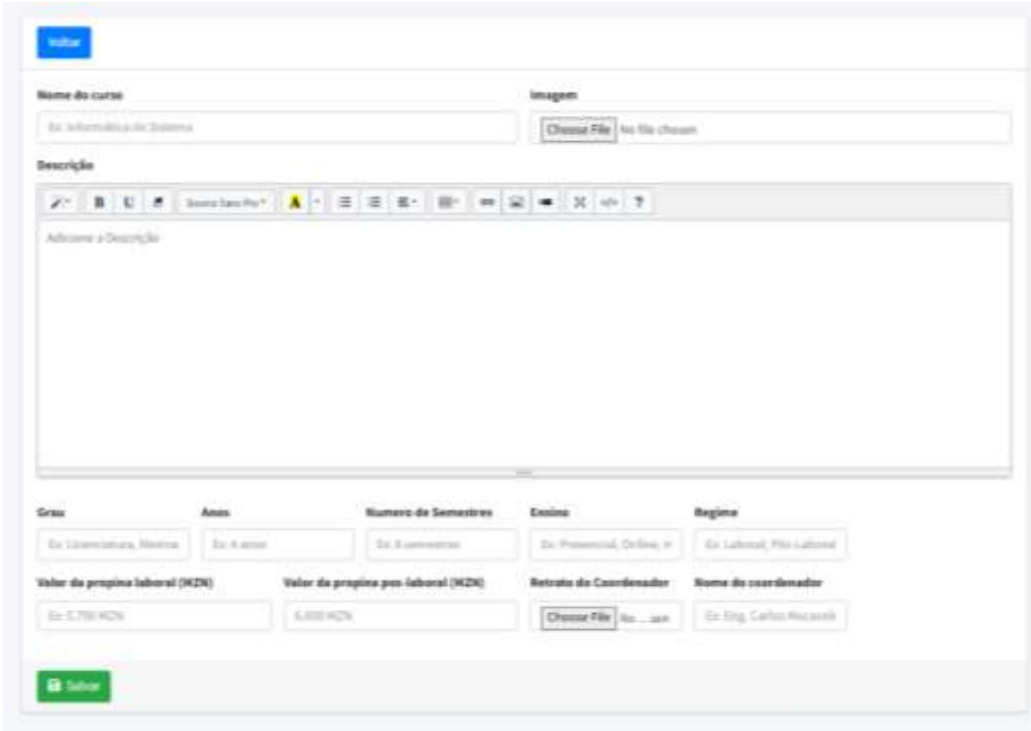


Figura 9: Tela de visualização de cursos

Tela de criação de curso: ao clicar o botão “criar” na página de listagem dos cursos, redireciona para a página de criação de cursos, apresentada na figura 10, onde se pode inserir: nome do curso, imagem, descrição, grau, anos, número de semestres, ensino, regime, valor da propina laboral, valor da propina pós-laboral, retrato do coordenador e nome do coordenador e de seguida clicar o botão “salvar”.



The image shows a web form for creating a course. At the top left, there is a blue button labeled "criar". The form contains several input fields and a rich text editor. The fields are: "Nome do curso" (with the example "Ex: Informática do Sistema"), "Imagem" (with a "Escolher Ficheiro" button and "No file chosen" text), "Descrição" (a rich text editor with a toolbar and the placeholder "Adicione a Descrição"), "Grau" (with the example "Ex: Licenciatura, Mestrado"), "Anos" (with the example "Ex: 4 anos"), "Número de Semestres" (with the example "Ex: 8 semestres"), "Ensino" (with the example "Ex: Presencial, Online, M"), "Regime" (with the example "Ex: Laboral, Não Laboral"), "Valor da propina laboral (KZ\$)" (with the example "Ex: 5.750 KZ\$"), "Valor da propina pós-laboral (KZ\$)" (with the example "6.000 KZ\$"), "Retrato do Coordenador" (with a "Escolher Ficheiro" button and "No file chosen" text), and "Nome do coordenador" (with the example "Ex: Eng. Carlos Mucari"). At the bottom left, there is a green button labeled "salvar".

Figura 10: Formulário para a criação de curso

Após a criação o sistema redirecciona para a página de listagem dos cursos, listando o curso criado e uma *modal* com uma mensagem de sucesso apresentada na figura 11.

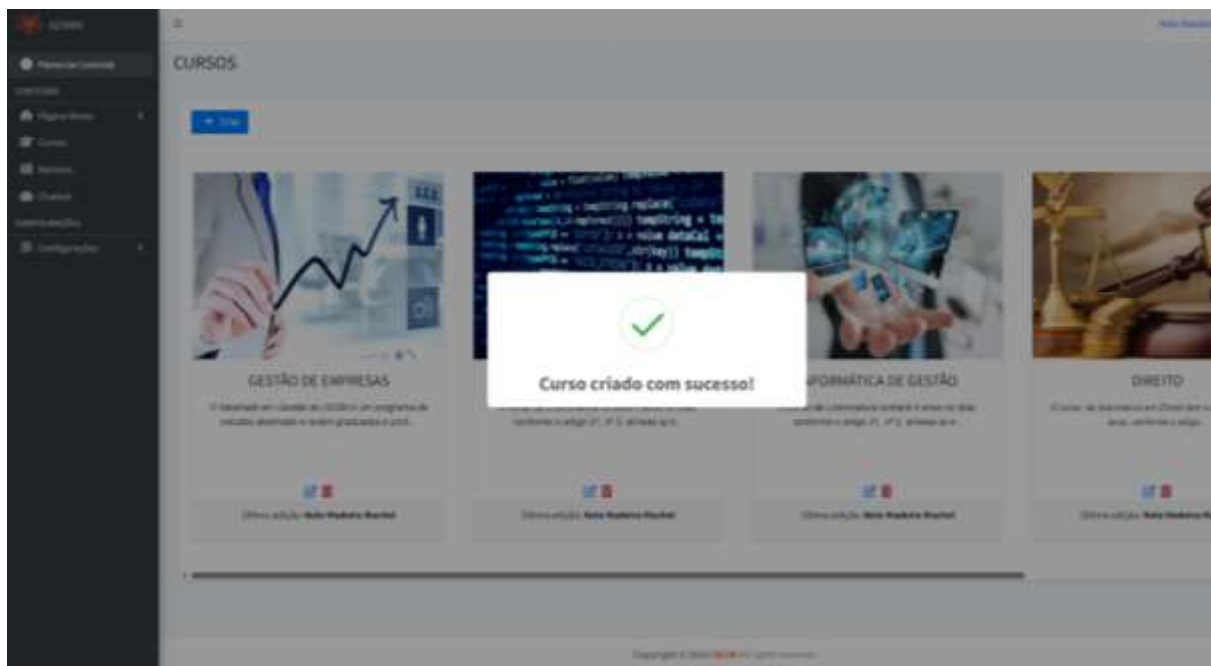


Figura 11: *Modal* com mensagem de sucesso após a criação de curso

Tela de edição de curso: ao clicar o ícone de edição (ao lado do ícone de exclusão) em um determinado curso na página de listagem dos cursos, redireciona para a página de edição de cursos, apresentada na figura 12, onde se pode editar os campos: nome do curso, imagem, descrição, grau, anos, número de semestres, ensino, regime, valor da propina laboral, valor da propina pós-laboral, retrato do coordenador e nome do coordenador e de seguida clicar o botão “actualizar”.

The screenshot shows a web interface for editing a course. At the top, there is a blue header with a green 'Início' button. Below it, the form is titled 'Nome do curso' and contains a text input field with the value 'COMUNICAÇÃO E RELAÇÕES EMPRESARIAIS'. The 'Descrição' section features a rich text editor with a toolbar and a text area containing two paragraphs of text. Below the description, there are two image upload sections: 'Imagem' and 'Retrato do Coordenador', each with a 'Choose File' button and a 'No file chosen' message. The form also includes several input fields for course details: 'Nome do coordenador' (Cláudio Faria), 'Grau' (Licenciatura), 'Anos' (4), 'Número de Semestres' (8), 'Ensino' (Presencial), and 'Regime' (Laboral e Pós-laboral). Additionally, there are input fields for 'Valor da propina laboral (MZN)' (4000) and 'Valor da propina pós-laboral (MZN)' (5000). At the bottom left, there is a blue 'Actualizar' button.

Figura 12: Formulário de edição de curso

Após a edição, o sistema redirecciona para a página de listagem dos cursos, listando o curso actualizado e uma mensagem de sucesso, apresentada na figura 13.

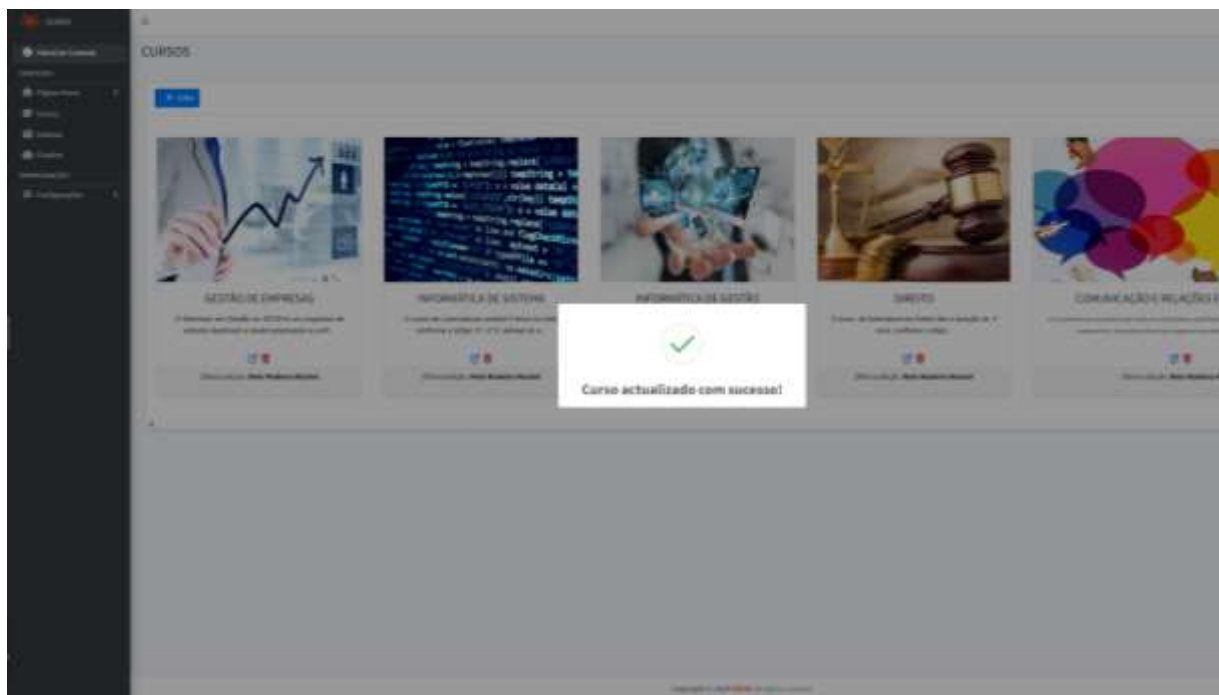


Figura 13: *Modal* com mensagem de sucesso após actualização do curso

Tela de exclusão de curso: ao clicar o ícone de exclusão (o ícone vermelho ao lado do ícone de edição) em um determinado curso na página de listagem dos cursos, apresentado na figura 14, primeiro apresenta uma *modal* de confirmação perguntando se realmente deseja eliminar o curso, ao clicar o botão “Sim, deletar!” prossegue com a exclusão do curso, e quando clica o botão “cancelar”, cancela a operação e fecha a *modal*.

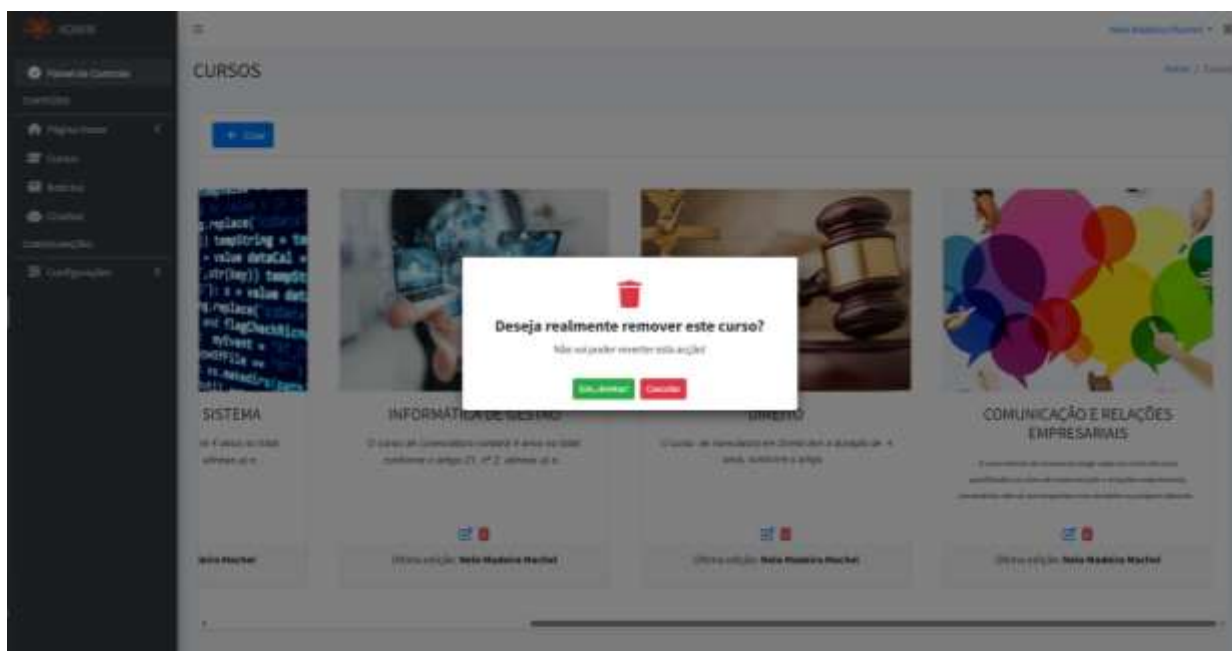


Figura 14: *Modal* para confirmação de exclusão de curso

Após a exclusão, o sistema redirecciona para a página de listagem dos cursos, e apresenta uma mensagem de sucesso conforme mostra a figura 15.

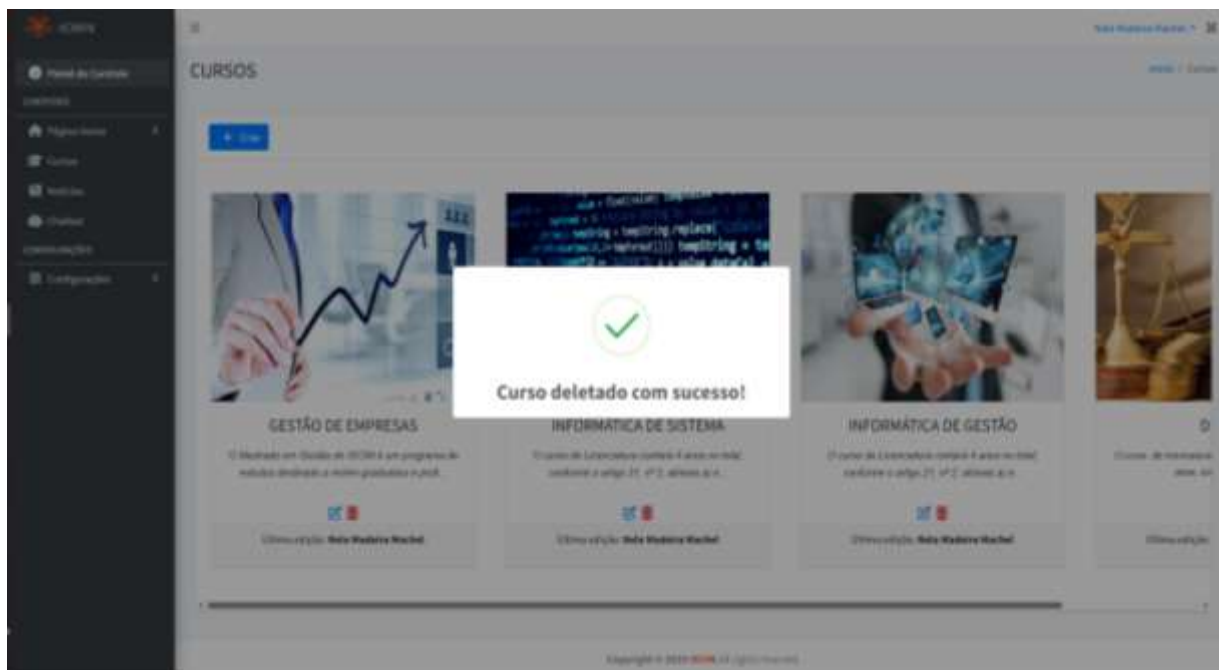


Figura 15: Modal com mensagem de sucesso após exclusão de curso

As demais páginas (“Carousel”, “Testemunhos” e “Notícias”), seguem o mesmo padrão, possuem uma página onde são listados todos os conteúdos criados, com opções de criar, editar e deletar.

E possui também, por fim, a página de gestão de todos os utilizadores, area que apenas o administrador tem permissões para a gestão dos utilizadores do sistema.

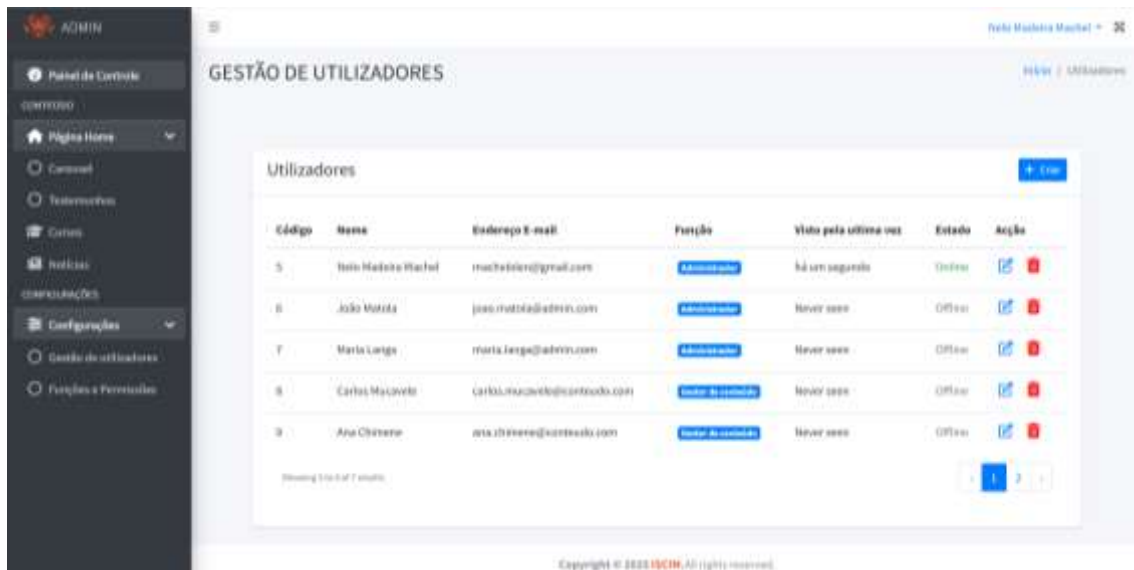


Figura 16: Tela de gestão de utilizadores do sistema

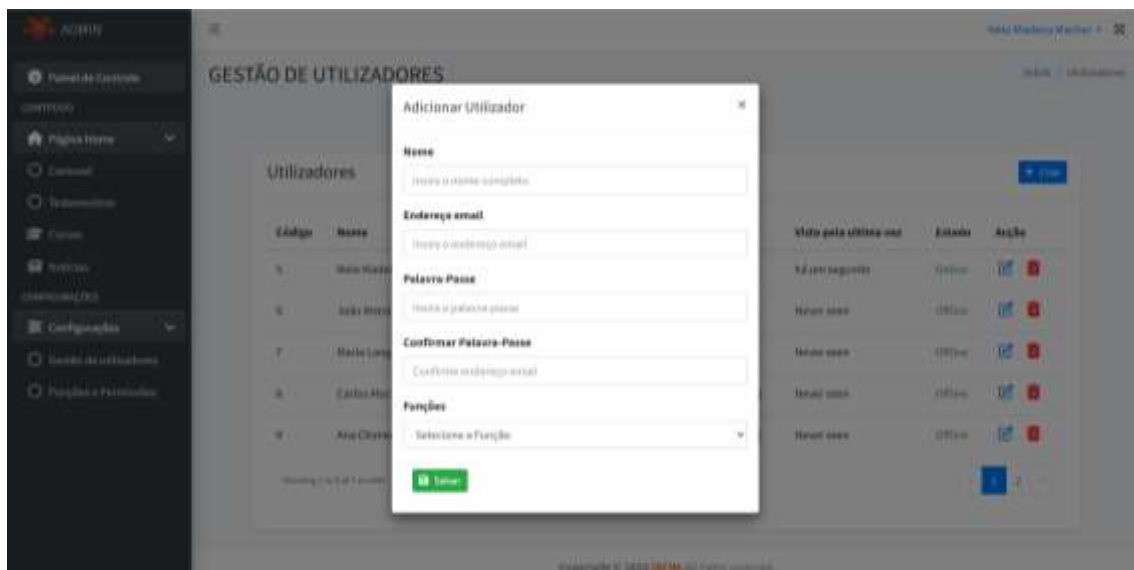


Figura 17: Modal de adição de utilizadores do sistema

5. Conclusão

O presente projecto teve como objectivo principal o desenvolvimento de um *website* dinâmico tendo como caso de estudo o Instituto Superior de Comunicação e Imagem de Moçambique (ISCIM). Com o fim de desenvolver uma solução moderna, responsiva e tecnologicamente avançada para acesso e disponibilização da informação em tempo real no *website*. Através da realização de uma análise detalhada de *snapshots* via *web archive*, foi possível descrever o actual *website* do ISCIM, essa análise permitiu constatar que o *website* existente foi desenvolvido com CMS *WordPress* e apresentava características de um *site* estático, com conteúdos desactualizados e limitações na gestão da informação. Com base nessas constatações, o desenvolvimento de uma solução tecnológica composta por um *website* dinâmico, um *dashboard* administrativo surge como uma solução para agilizar o processo de publicação e actualização da informação em tempo real.

Através dessas análises foi possível elaborar requisitos, e desses requisitos foi possível modelar o sistema através da modelação UML elaborando os diagramas de caso de uso, de classe, de actividade e de sequência, o que permitiu estruturar de forma clara e lógica os componentes e funcionalidades essenciais para a gestão de conteúdo e de utilizadores, mostrando as acções e processos que o *dashboard* administrativo disponibiliza. Com base nessa modelação, procedeu-se ao desenvolvimento do protótipo funcional, utilizando o *framework Laravel* para o *back-end* e o *Bootstrap* no *front-end*, garantindo uma interface moderna, interactiva e responsiva, adaptada a diferentes dispositivos (computador, *smartphone* e *tablet*). O sistema foi implementado em ambiente local com o *XAMPP* e recorreu ao *MySQL* via *PHPMysqlAdmin* para o armazenamento e gestão dos dados.

A proposta culminou na apresentação de uma solução integrada após a codificação, onde o *dashboard* administrativo actua como núcleo de gestão, permitindo controlar simultaneamente o conteúdo dinâmico do *website* através de CRUDs. Esta integração representa um avanço significativo na forma como a instituição comunica com os seu público em geral, promovendo agilidade, centralização e consistência na disponibilização de dados.

6. Recomendações

Com base nos resultados obtidos e na experiência desenvolvida ao longo do projecto, recomenda-se ao ISCIM as seguintes acções:

- ✓ **Realizar a integração em ambiente real**, com diferentes perfis de utilizadores, para validar a robustez da solução e identificar melhorias funcionais;
- ✓ **Expandir as funcionalidades do site**, incluindo formulários dinâmicos para pré-inscrições e matrículas *online*, com integração a sistemas internos de gestão académica;
- ✓ **Utilizar ferramentas de análise de dados no site**, como o *Google Analytics* para compreender o comportamento dos utilizadores e otimizar a navegação e o conteúdo;
- ✓ **Planear ciclos de manutenção e actualização contínua**, com base em *feedback* dos utilizadores e nas tendências tecnológicas do sector educacional.

Referências Bibliográficas

- Almeida, G. A. M. (2017). *Factores de escolha entre metodologias de desenvolvimento de software tradicionais e ágeis*. <https://teses.usp.br/teses/disponiveis/3/3136/tde-11042017-143311/pt-br.php>
- Assis, S. R. (2021). *Frameworks Conceitos e Aplicações*. https://www.researchgate.net/publication/356509021_Frameworks_Conceitos_e_Aplicacoes
- Elmasri, R. & Navathe, S. B. (2005). *Sistemas de Banco de Dados*. Editora Pearson. https://www.tonysoftwares.com.br/attachments/article/5297/Sistema_de_banco_de_dados_Navathe.pdf
- Gil, A. C. (2002). *Como Elaborar Projectos de Pesquisa*. (4a edição). Editora Atlas S.A. <https://archive.org/details/metodologia-da-pesquisa-antonio-carlos-gil-como-elaborar-projetos-de-pesquisa/page/n2/mode/1up?view=theater>
- Júnior, C. F. C. & Carvalho, K. R. S. A. (2018). *Chatbot: uma visão geral sobre aplicações inteligentes*. *Revista Sitio Novo*, 2(2), p.17. Disponível em: <https://core.ac.uk/download/pdf/268417746.pdf>
- Miguel, L. V. N. (2019). *Implementação de um Web Site para a Empresa Casa dos Moveis*. https://d1wqtxts1xzle7.cloudfront.net/61352286/Monografia_final.Leonel_Vander_IMPLEMENTACAO_DE_UM_WEB_SITE_PARA_EMPRESA_CASA_DOS_MOVEIS20191127-82684-fdr7tc-libre.pdf?1574848771=&response-content-disposition=inline%3B+filename%3DMonografia_final_Leonel_Vander_IMPLMENT.pdf&Expires=1744639739&Signature=CRCdYGCIn-kV26GktT0XVEZwRB2auZDz-IDUeuVO1fe6AQBGAAbWtRSPLEITKgs1kjuT7crdJJB2k-rZftl0gstKB~kEEF-LRgy2R1Q1AVbaLS~WPGBDFNncP0IIutiutOLNorri~8hiwA4QzBppp6PjR9bUWkrlJ27hzYGcR2PYbVG4eY9Lp9VV7vHY6rzB7lfSZCNvNWg3NU56-LGZslNPjy-vPhLg~UP2qpaZcagry6B00Eo81h0xkAHdbNDuX-1ueGR0-o25nEma3oLS~~pV6JMyGKqqpv7YrCpl3vf8TISOmRFRwasKGUXWys2017T~9ylqmeLwdKpRoPgHsyA &Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

Neves, B. D. P. (2010). *Os Sistemas de Gestão de Conteúdos aplicados à gestão da informação em bibliotecas universitárias*. <https://www.proquest.com/openview/c58bda0d64d1393904e1d50e45639d27/1?pq-origsite=gscholar&cbl=2026366&diss=y>

Prodanov, C. C. & Freitas, E. C. (2013). *Metodologia do Trabalho Científico: Métodos e Técnicas da Pesquisa e do Trabalho Acadêmico*. (2a edição). Editora Feevale. <https://archive.org/details/metodologia-do-trabalho-cientifico/page/n1/mode/1up?view=theater>