Uma imagem com texto

Descrição gerada automaticamente

**Licenciatura em Engenharia Informática**

**Relatório de Projeto**

**Gestor de Alojamentos**

**Muaiad Mhd Fahd Al Hadad**

Oliveira do Hospital, fevereiro de 2024

Honestidade Intelectual

*Eu, Muaiad Mhd Fahd Al Hadad, estudante n.º 2020130486 da Licenciatura em Engenharia informática, declaro que o relatório de estágio/trabalho de projeto intitulado gestão de alojamentos é original e que, ao longo da sua elaboração, não pratiquei plágio ou qualquer forma de falsificação de conteúdo. O trabalho de projeto resulta do meu próprio trabalho, sendo reconhecidas todas as fontes utilizadas por se encontrarem devidamente citadas no corpo do texto e identificadas na secção de referências bibliográficas. Assumo ter plena consciência de que a prática de plágio - utilização como sendo criação ou prestação sua de obras, ideias, afirmações, dados, imagens ou ilustrações de outra autoria, no todo em parte, sem o adequado reconhecimento explícito - constitui, no âmbito académico, grave falta ética e desonestidade intelectual, tendo como consequência a anulação do trabalho apresentado, para além de poder constituir crime de violação dos direitos de autor e infração disciplinar.*

*Mais declaro que tomei conhecimento integral do Código de Ética e Conduta do Instituto Politécnico de Coimbra e demais regulamentos aplicáveis e que foram respeitadas as orientações recebidas quanto à pseudonimização ou anonimização de dados pessoais ou organizacionais.*

*Oliveira do Hospital, 12 de fevereiro de 2024*

*Nome completo: Muaiad Mhd Fahd Al Hadad*

*Assinatura: (digital)*

Agradecimentos

Gostaríamos de expressar a nossa sincera gratidão a todos os professores da Escola Superior de Tecnologia e Gestão de Oliveira do Hospital (ESTGOH) que contribuíram de maneira significativa para o nosso percurso académico. O seu conhecimento, orientação e apoio foram fundamentais para o nosso desenvolvimento pessoal e profissional.

Em particular, queremos estender os nossos agradecimentos especiais ao Prof. Luis Veloso, ao Prof. Marco Veloso e ao Prof. Nuno Gil. As suas instruções, conselhos e incentivo foram essenciais para o sucesso deste projeto e para a nossa formação como engenheiros de informática.

Agradecemos por compartilharem connosco o seu profundo conhecimento, paixão pela área e dedicação ao ensino. As suas contribuições não apenas nos ajudaram a adquirir habilidades técnicas, mas também nos inspiraram a alcançar os nossos objetivos com determinação e excelência.

Que este simples gesto de gratidão expresse a nossa profunda admiração e respeito por vocês, que continuemos a manter uma relação de aprendizado mútuo e crescimento ao longo das nossas trajetórias profissionais.

Com os nossos mais sinceros agradecimentos,

Muaiad Mhd Fahd Al Hadad.

Resumo

Este projeto desenvolveu um sistema de gestão de alojamento para a Escola Superior de Tecnologia e Gestão de Oliveira do Hospital (ESTGOH), com o objetivo de facilitar a busca, reserva e comunicação entre alunos e senhorios. Utilizando HTML, JavaScript, PHP e uma base de dados, a plataforma permite aos alunos procurar quartos disponíveis, comunicar-se com os senhorios e efetuar reservas eficientemente. A metodologia envolveu o desenvolvimento iterativo do sistema, integrando “feedback” dos utilizadores para garantir usabilidade e funcionalidade adequadas. Os resultados incluem uma plataforma intuitiva e responsiva, melhorando significativamente a experiência dos utilizadores na busca por alojamento. Conclusões relevantes destacam a importância de soluções tecnológicas para otimizar processos académicos e a necessidade contínua de adaptação às demandas dos alunos.

**Palavras-chave**

Alojamento, Gestão, Escola, Estudantes, Plataforma.

Abstract

This project developed a housing management system for the Oliveira do Hospital School of Technology and Management (ESTGOH), aiming to facilitate the search, reservation, and communication between students and landlords. Using HTML, JavaScript, PHP, and a database, the platform allows students to search for available rooms, communicate with landlords, and make reservations efficiently. The methodology involved iterative development of the system, integrating user feedback to ensure adequate usability and functionality. Results include an intuitive and responsive platform, significantly improving users' experience in housing search. Relevant conclusions highlight the importance of technological solutions to streamline academic processes and the ongoing need to adapt to students' demands.

**Keywords**

Housing, Management, School, Students, Platform.

**Índice**

Índice

[Honestidade Intelectual i](#_Toc160987980)

[Agradecimentos ii](#_Toc160987981)

[Resumo iii](#_Toc160987982)

[Abstract iv](#_Toc160987983)

[Lista de Figuras vi](#_Toc160987984)

[Lista de Tabelas vii](#_Toc160987985)

[Lista de Acrónimos viii](#_Toc160987986)

[2. Introdução 1](#_Toc160987987)

[3. Estado da Arte 2](#_Toc160987988)

[4. Objectivos e Metodologias 3](#_Toc160987989)

[4.1. Ferramentas e Tecnologias 3](#_Toc160987990)

[4.2. Planeamento 4](#_Toc160987991)

[4.3. Diagramas 5](#_Toc160987992)

[4.3.1. Fluxogramas 5](#_Toc160987993)

[4.3.2. Caso de use 9](#_Toc160987994)

[4.3.3. Classes UML 10](#_Toc160987995)

[4.3.4. ER (Base Dados): 12](#_Toc160987996)

[5. Trabalho Desenvolvido 13](#_Toc160987997)

[5.1. Requisitos Implementados 23](#_Toc160987998)

[5.1.1. REQUISITOS 23](#_Toc160987999)

[5.1.2. Requisitos planeados 26](#_Toc160988000)

[5.1.3. Requisitos feitos 27](#_Toc160988001)

[5.1.4. Principais motivos para eventuais desvios (Sprint 1) 27](#_Toc160988002)

[5.1.5. Lista dos diagramas 27](#_Toc160988003)

[6. Conclusões 28](#_Toc160988004)

[6.1. Forças 29](#_Toc160988005)

[6.2. Limitações 29](#_Toc160988006)

[6.3. Trabalho Futuro 30](#_Toc160988007)

[7. Referências 32](#_Toc160988008)

[7.1. Lista de Referências 33](#_Toc160988009)

[7.2. Como Integrar Referências 34](#_Toc160988010)

[8. Anexos 35](#_Toc160988011)

# Lista de Figuras

[Figure 1-Fluxograma de Autenticação 5](#_Toc160991020)

[Figure 2- Fluxograma de Filtragem de anúncios 6](#_Toc160991021)

[Figure 3- Fluxograma de Gestor na aprovação/reprovação de anúncios 7](#_Toc160991022)

[Figure 4- Fluxograma de Registro de senhorios/alunos na plataforma 8](#_Toc160991023)

[Figure 5-diagrama caso de use 9](#_Toc160991024)

[Figure 6-Classes UML 10](#_Toc160991025)

[Figure 7-ER 12](#_Toc160991026)

[Figure 8-Cabeçalho de página inicial 13](#_Toc160991027)

[Figure 9- barra de pesquisa 14](#_Toc160991028)

[Figure 10-Cartas de anúncios 15](#_Toc160991029)

[Figure 11-codigo de JS de Cartas de anúncios 17](#_Toc160991030)

[Figure 12- Código JS de frente de cartão e atrás 18](#_Toc160991031)

[Figure 13- filtragem 19](#_Toc160991032)

[Figure 14- Menu inicial desta página 19](#_Toc160991033)

[Figure 15-tabela com filtro 20](#_Toc160991034)

[Figure 16-formulário de adicionar alojamento 20](#_Toc160991035)

[Figure 17-tabela de remover com filtro 21](#_Toc160991036)

[Figure 18-conversa Aluno e senhorio 21](#_Toc160991037)

[Figure 19-form login 22](#_Toc160991038)

[Figure 20-form registar 23](#_Toc160991039)

# Lista de Tabelas

[Tabela 1-Requsitos 26](#_Toc160988019)

[Tabela 2-requsitos planeados 27](#_Toc160988020)

[Tabela 3-requisitos feitos 27](#_Toc160988021)

[Tabela 4-Lista dos diagramas 28](#_Toc160988022)

Lista de Acrónimos

**ER** Modelo Entidade-Relacionamento

**JS** JavaScript

# Introdução

*A crescente demanda por soluções eficientes de gestão de alojamento, especialmente em ambientes académicos como a Escola Superior de Tecnologia e Gestão de Oliveira do Hospital (ESTGOH), tem sido evidente nos últimos anos. Nesse contexto, este projeto de final de curso em Engenharia de Informática surge para enfrentar esse desafio, desenvolvendo um sistema abrangente e intuitivo para gestão de alojamentos, direcionado especificamente para os alunos desta instituição.*

*O cerne deste projeto é facilitar o processo de busca, reserva e comunicação entre os estudantes e os senhorios, proporcionando uma solução centralizada e eficaz. Utilizando uma combinação de linguagens de programação “web”, como HTML, JavaScript e PHP, e integrando uma base de dados para armazenamento e recuperação de informações críticas, busca-se oferecer uma plataforma que atenda às necessidades específicas dos utilizadores.*

*A plataforma permitirá aos alunos buscar quartos disponíveis de acordo com critérios específicos, como localização, preço e comodidades, além de possibilitar a comunicação direta com os senhorios, facilitando negociações e esclarecimento de dúvidas antes da reserva. Garantindo a integridade dos dados e a disponibilidade dos quartos selecionados, evitados conflitos de reservas, o sistema visa proporcionar uma experiência fluida e transparente para os utilizadores.*

*Os senhorios terão acesso a um painel de controle dedicado para gerir as suas propriedades, atualizar informações e responder às consultas dos alunos. A segurança e a privacidade dos dados dos utilizadores serão prioridades essenciais durante todo o desenvolvimento do projeto, em conformidade com as regulamentações de proteção de dados.*

*Em resumo, este projeto oferecerá uma solução abrangente e eficiente para a gestão de alojamentos, promovendo uma experiência amigável e intuitiva por meio de uma “interface” “web” moderna e responsiva, atendendo às necessidades específicas dos alunos da ESTGOH.*

# Estado da Arte

*Neste estado da arte, examinamos duas plataformas líderes no mercado de acomodações para estudantes: Uniplaces e Student at Home. Ambas as plataformas pretendem facilitar a busca e reserva de acomodações para estudantes universitários, embora possam ter diferenças significativas em termos de recursos, alcance geográfico e modelo de negócios. Abaixo está uma análise detalhada de cada plataforma:*

*Uniplaces:*

*Descrição: Uniplaces é uma plataforma “online” estabelecida que conecta estudantes universitários a acomodações em várias cidades ao redor do mundo. a plataforma tem crescido significativamente, tornando-se uma escolha popular entre estudantes internacionais em busca de moradia durante os estudos.*

*Características:*

*“Interface” amigável e intuitiva que permite aos utilizadores pesquisar e filtrar acomodações com base nas preferências.*

*Opções de pagamento seguras e processamento de reservas “online”.*

*Avaliações e comentários de propriedades por outros utilizadores para auxiliar na tomada de decisão.*

*Suporte ao cliente dedicado para resolver quaisquer problemas relacionados à reserva ou à estadia.*

*Student at Home:*

*Descrição: Student at Home é outra plataforma “online” que oferece acomodações para estudantes universitários. a plataforma pode ter um foco regional específico ou oferecer recursos adicionais para estudantes em comparação com outras plataformas.*

*Características:*

*Oferece uma variedade de opções de moradia para estudantes, desde quartos individuais até apartamentos completos.*

*Pode incluir recursos adicionais específicos para estudantes, como guias de vida estudantil, informações sobre universidades locais e eventos estudantis.*

*Opções de reserva e pagamento “online”, embora difiram em termos de funcionalidades específicas em comparação com outras plataformas.*

*Comparação:*

*Ambas as plataformas pretendem principal fornecer uma experiência de busca e reserva de acomodações simplificada para estudantes universitários. No entanto, existem algumas diferenças notáveis entre elas. Enquanto Uniplaces pode ter um alcance global mais amplo e uma base de utilizadores mais estabelecida, Student at Home pode oferecer recursos adicionais específicos para estudantes, dependendo do mercado em que opera.*

*Conclusão:*

*Tanto Uniplaces quanto Student at Home desempenham um papel vital no mercado de acomodações para estudantes, oferecendo opções convenientes e confiáveis para estudantes universitários em busca de moradia durante os estudos. Com a contínua evolução do mercado e as demandas em constante mudança dos estudantes, essas plataformas continuarão a se adaptar e inovar para atender às necessidades em constante mudança dos utilizadores.*

*Este estado da arte fornece uma visão abrangente das duas plataformas, destacando características distintas e papel no mercado de acomodações para estudantes.*

*Ao examinar o estado da arte, podemos observar que há uma variedade de soluções disponíveis no mercado que abordam aspetos semelhantes aos propostos neste projeto. No entanto, a proposta deste projeto é oferecer uma solução específica e personalizada para atender às necessidades dos alunos da ESTGOH, integrando funcionalidades de busca, reserva e comunicação numa plataforma única e intuitiva.*

# Objectivos e Metodologias

*Objetivos:*

*Desenvolver uma plataforma “web” abrangente para gestão de alojamentos, direcionada especificamente para os alunos da Escola Superior de Tecnologia e Gestão de Oliveira do Hospital (ESTGOH).*

*Facilitar o processo de busca, reserva e comunicação entre os alunos e os senhorios, proporcionando uma solução centralizada e eficaz para as necessidades de alojamento dos estudantes.*

*Integrar funcionalidades de busca avançada, reserva de quartos e comunicação direta entre alunos e senhorios, garantindo uma experiência fluida e transparente para os utilizadores.*

*Metodologias:*

*Levantamento de Requisitos: Realizar entrevistas cliente para identificar as necessidades e preferências relativamente ao sistema de gestão de alojamentos.*

*Projeto e Desenvolvimento: Utilizar metodologias ágeis, como Scrum, para o desenvolvimento iterativo da plataforma, com entregas incrementais e ‘feedback’ contínuo dos utilizadores.*

*Implementação: Utilizar uma combinação de linguagens de programação “web”, incluindo HTML, JavaScript e PHP, para desenvolver a “interface” do utilizador e a lógica de negócios da plataforma.*

*Integração de Banco de Dados: Implementar um banco de dados para armazenar e recuperar informações críticas, como detalhes dos quartos, reservas e dados dos utilizadores.*

*Testes e Validação: Realizar testes de usabilidade, funcionais e de segurança para garantir a qualidade e a robustez da plataforma antes do lançamento.*

*Implantação e Manutenção: Implementar a plataforma num ambiente de produção e fornecer suporte contínuo, atualizações e manutenção conforme necessário.*

## Ferramentas e Tecnologias

*Para o desenvolvimento deste projeto, serão utilizadas as seguintes ferramentas e tecnologias:*

*Laravel: Um “framework” de desenvolvimento “web” em PHP, conhecido por sua elegância, eficiência e facilidade de uso. O Laravel oferece recursos poderosos para o desenvolvimento rápido de aplicações “web”, incluindo gestão de rotas, ORM (Object-Relational Mapping), autenticação de utilizadores e muito mais.*

*MySQL: Um sistema de gestão de banco de dados relacional amplamente utilizado, que oferece desempenho confiável, escalabilidade e suporte a uma variedade de recursos avançados. O MySQL será utilizado para armazenar e recuperar dados críticos do sistema, como detalhes dos quartos, informações de utilizadores e histórico de reservas.*

*HTML, CSS e JavaScript: as linguagens fundamentais para o desenvolvimento de “interfaces” “web”. O HTML será utilizado para estruturar o conteúdo da página, o CSS para estilização e o JavaScript para adicionar interatividade e dinamismo à “interface” do utilizador.*

*Git: Um sistema de controlo de versão distribuído, essencial para o trabalho colaborativo e a gestão de código-fonte durante o desenvolvimento do projeto. O Git permite rastrear alterações no código, colaboração entre membros da equipa e a reversão para versões anteriores, se necessário.*

*Essas ferramentas e tecnologias foram selecionadas com base na minha capacidade de atender aos requisitos do projeto e fornece uma base sólida para o desenvolvimento de uma plataforma robusta e eficiente de gestão de alojamentos. O uso do Laravel e MySQL, em particular, proporcionará um ambiente de desenvolvimento seguro e escalável, enquanto o uso de HTML, CSS, JavaScript garantirá uma experiência de utilizador moderna e intuitiva. O Git será utilizado para garantir o controlo de versão.*

## Planeamento

*O planeamento do projeto será realizado conforme as seguintes etapas:*

*Definição de Requisitos: Durante esta fase, serão identificados e documentados os requisitos funcionais e não funcionais da plataforma de gestão de alojamentos. Isso incluirá a determinação das funcionalidades necessárias, como busca de quartos, reserva, comunicação entre alunos e senhorios, além de requisitos de segurança e privacidade de dados.*

*Análise e “Design”: após a definição dos requisitos, será realizada uma análise detalhada para identificar as melhores práticas de “design” e arquitetura para o sistema. Isso incluirá a criação de diagramas de fluxo de dados, diagramas de classes e outros artefactos de “design” para orientar o desenvolvimento.*

*Desenvolvimento Iterativo: O desenvolvimento será realizado iterativamente, com entregas incrementais ao longo do tempo. Isso permitirá a rápida prototipagem e validação das funcionalidades, bem como a incorporação contínua do “feedback” dos utilizadores.*

*Testes e Depuração: Após cada iteração de desenvolvimento, serão realizados testes rigorosos para garantir a qualidade e a estabilidade do sistema. Isso incluirá testes de unidade, integração e aceitação, bem como testes de segurança e desempenho.*

*Implantação e Lançamento: Uma vez que o desenvolvimento e os testes estejam concluídos, o sistema será implantado num ambiente de produção e lançado para os utilizadores finais. Isso será acompanhado por atividades de monitora mento e suporte para garantir que o sistema funcione conforme o esperado.*

*O planeamento será conduzido de forma ágil e flexível, permitindo ajustes conforme necessário ao longo do ciclo de vida do projeto. Isso garantirá que a plataforma de gestão de alojamentos seja entregue no prazo e atenda às expectativas dos utilizadores finais.*

## Diagramas

### Fluxogramas

Gráfico, Diagrama

Descrição gerada automaticamente

Figure 1-Fluxograma de Autenticação

Este fluxograma representa o processo de autenticação do utilizador quando ele tenta fazer “login” no sistema utilizando o seu e-mail e senha.

Início: O processo começa quando o utilizador solicita a autenticação, geralmente clicando num botão de “login”.

Utilizador fornece e-mail e senha: O utilizador digita o e-mail e senha nos campos apropriados.

Verificar se e-mail e senha estão corretos: O sistema verifica se o e-mail e a senha fornecidos correspondem a uma conta válida no banco de dados.

Autenticação bem-sucedida ou falha: se as credenciais estiverem corretas, o utilizador é autenticado com sucesso e tem acesso ao sistema. Caso contrário, a autenticação falha e o utilizador é notificado do erro.

Opção para o utilizador encerrar a sessão: após autenticado, o utilizador pode escolher encerrar a sessão, o que o desconecta do sistema.

Fim: O fluxo termina aqui.

Gráfico

Descrição gerada automaticamente

Figure 2- Fluxograma de Filtragem de anúncios

Este fluxograma representa o processo de filtragem de anúncios por gestores e alunos.

Início: O processo começa quando um gestor ou aluno solicita a filtragem de anúncios.

Escolha dos critérios de filtragem: O utilizador escolhe os critérios pelos quais deseja filtrar os anúncios, como distância à ESTGOH, valor da renda, número de quartos etc.

Aplicar filtros aos anúncios: O sistema aplica os critérios de filtragem selecionados aos anúncios disponíveis.

Exibir anúncios filtrados: O sistema exibe os anúncios que correspondem aos critérios de filtragem selecionados pelo utilizador.

Fim: O fluxo termina aqui.

Diagrama

Descrição gerada automaticamente

Figure 3- Fluxograma de Gestor na aprovação/reprovação de anúncios

Este fluxograma representa o processo de avaliação de anúncios por um gestor, com a opção de aprovar ou reprovar os anúncios.

Início: O processo começa quando um novo anúncio é submetido à plataforma e um gestor recebe uma notificação sobre isso.

Gestor avalia o anúncio: O gestor revê o anúncio e decide se ele deve ser aprovado ou reprovado.

Anúncio aprovado ou reprovado: Se o gestor aprovar o anúncio, ele é publicado na plataforma. Se for reprovado, o anúncio não é publicado.

Notificar senhorio sobre aprovação/reprovação: O gestor notifica o senhorio sobre a decisão tomada em relação ao seu anúncio.

Fim: O fluxo termina aqui.

Diagrama

Descrição gerada automaticamente

Figure 4- Fluxograma de Registro de senhorios/alunos na plataforma

Este fluxograma representa o processo de registo de um novo senhorio na plataforma.

Início: O processo começa quando um senhorio decide se registar na plataforma.

Senhorio fornece nome, e-mail e senha: O senhorio preenche os campos necessários para o registo, incluindo nome, e-mail e senha.

Verificar se todos os campos foram preenchidos corretamente: O sistema verifica se todas as informações necessárias foram fornecidas adequadamente pelo senhorio.

Registo bem-sucedido ou falho: se todas as informações estiverem corretas, o registo é bem-sucedido e o senhorio pode a cessar a plataforma. Caso contrário, o registo falha e o senhorio é notificado do erro.

Enviar e-mail de aprovação ou reprovação: O sistema envia um e-mail ao senhorio informando se o registo foi aprovado ou reprovado.

Fim: O fluxo termina aqui.

### Caso de use

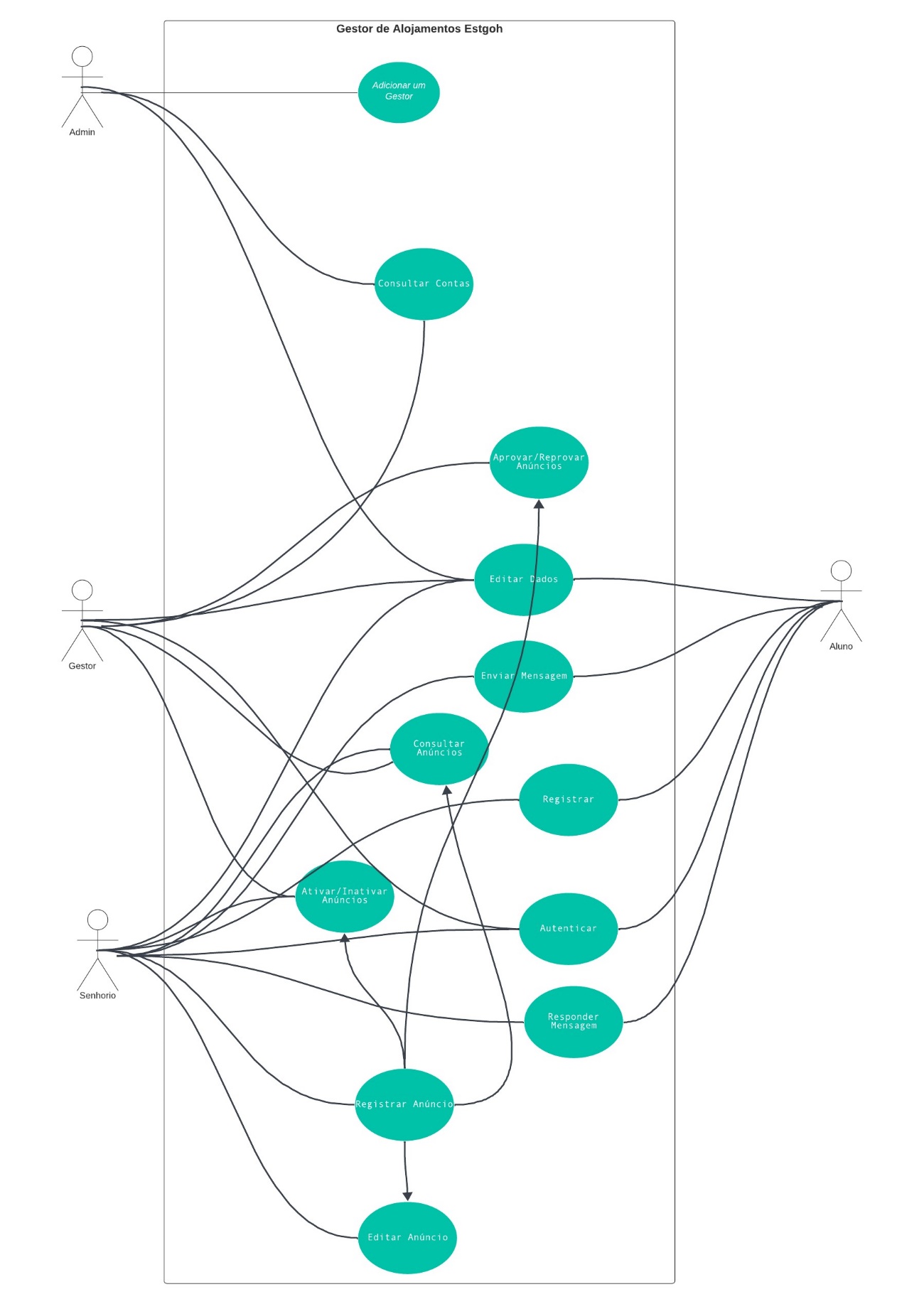


Figure 5-diagrama caso de use

Gestor: Este ator é responsável pela gestão dos anúncios dentro do sistema. Eles podem aprovar, reprovar, ativar, inativar anúncios, consultar anúncios, consultar contas, editar dados e responder mensagens.

Senhorio: Representa os proprietários de imóveis que desejam anunciar suas propriedades no sistema. Eles podem registar anúncios, editar anúncios existentes, inativar e ativar anúncios, consultar anúncios, editar seus dados e responder mensagens.

Aluno: Este ator representa os usuários que estão procurando por acomodações. Eles têm permissão para consultar anúncios, enviar mensagens e responder mensagens.

Administrador: Este é o ator responsável pela gestão geral do sistema. Eles têm autoridade para realizar tarefas administrativas, como aprovar anúncios, consultar anúncios, consultar contas, editar dados de usuários, além de enviar e responder mensagens, se necessário.

### Classes UML

Diagrama

Descrição gerada automaticamente

Figure 6-Classes UML

Classes:

Utilizador: Esta classe representa um utilizador genérico no sistema. Ela possui os atributos comuns a todos os tipos de utilizadores, como Nome, Email, Palavra-passe e Avatar.

Administrador: É uma subclasse de utilizador e representa um administrador do sistema. Herda os atributos de utilizador.

Gestor: Também é uma subclasse de utilizador e representa um gestor. Herda os atributos de utilizador.

Senhorio: Outra subclasse de utilizador, representa um senhorio. Além dos atributos herdados de utilizador, possui atributos adicionais relacionados aos detalhes da propriedade que ele oferece para aluguer.

Anúncio: Esta classe representa um anúncio de aluguer de propriedade. Possui atributos relacionados às características do anúncio, como Distância à ESTGOH, Valor da Renda, Número de Quartos e Sexo.

Aluno: Subclasse de Utilizador, representa um aluno. Herda os atributos de utilizador.

Associações:

Gestor gerencia Senhorio: Indica que um gestor pode gerir vários senhorios. Esta associação é representada por uma relação de composição ("1" para muitos).

Senhorio possui Anúncio: Indica que um senhorio pode possuir vários anúncios de propriedades para aluguer. Esta também é uma relação de composição.

Senhorio recebe Mensagem: Indica que um senhorio pode receber várias mensagens de alunos interessados nas propriedades que ele oferece para aluguer. Esta é uma associação simples ("1" para muitos).

Aluno envia Mensagem: Similar à relação anterior, indica que um aluno pode enviar várias mensagens para senhorios relativamente aos anúncios de propriedades. Também é uma associação simples.

Classes Adicionais:

Mensagem: Esta classe representa uma mensagem enviada entre um aluno e um senhorio relativamente a um anúncio de propriedade. Possui atributos como Conteúdo e Data para armazenar o conteúdo da mensagem e a data em que foi enviada.

Em resumo, o diagrama mostra a estrutura básica do sistema de aluguer de propriedades, incluindo os diferentes tipos de utilizadores, relacionamentos e as entidades relacionadas aos anúncios e mensagens.

### ER (Base Dados):

Diagrama

Descrição gerada automaticamente

Figure 7-ER

# Trabalho Desenvolvido

## Front-end

### Página inicial

Placa de letreiro luminoso

Descrição gerada automaticamente com confiança média

Texto

Descrição gerada automaticamente

Figure 8-Cabeçalho de página inicial

Em resumo, esta estrutura HTML representa um menu de navegação com ligações para diferentes secções do sítio web. A ligação "Alojamentos" está ativa, indicando que representa a página atual. A ligação "Filtrar" possui um “ID” exclusivo para manipulação futura, e as ligações "Login" e "Sobre" ainda não estão definidos para páginas específicas. Padrão do plano de fundo

Descrição gerada automaticamente com confiança baixa

Texto

Descrição gerada automaticamente

Figure 9- barra de pesquisa

Este é um trecho de código HTML que representa um campo de pesquisa em um contêiner:

<div class="container">: Define um contêiner para o campo de pesquisa e o botão de busca. A classe "container" é provavelmente usada para aplicar estilos ou para organizar elementos na página.

<div class="search-container">: define um contêiner específico para o campo de pesquisa e o botão de busca. A classe "search-container" pode ser usada para aplicar estilos específicos a esses elementos.

<input type="text" id="search-box" placeholder="Search...">: Define um campo de entrada de texto onde os utilizadores podem digitar o termo de busca. O “ID” "search-box" pode ser usado para selecionar e manipular esse elemento usando JavaScript. O atributo "placeholder" fornece um texto de orientação para os utilizadores, indicando o que eles devem digitar no campo de busca.

<button id="search-button">Search</button>: Define um botão de busca que os utilizadores podem clicar para iniciar a pesquisa. O “ID” "search-button" pode ser usado para selecionar e manipular esse botão usando JavaScript. O texto "Search" no botão indica sua função.

Interface gráfica do usuário

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Figure 10-Cartas de anúncios

além de manipulação de efeitos visuais nos cards quando o rato é movido sobre eles.

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Figure 11-codigo de JS de Cartas de anúncios

Explicação resumida do que cada parte do código faz:

Seleção de Elementos:

const cardS = document.querySelectorAll(".card");: Seleciona todos os elementos com a classe "card" e armazena-os em uma NodeList.

const modal = document.getElementById("myModal");: Seleciona o modal pelo seu ID.

const modalImg = document.getElementById("modalImage");: Seleciona a imagem dentro do modal pelo seu ID.

const closeBtn = document.querySelector(".close");: Seleciona o botão de fechar pelo seletor de classe.

Iteração sobre os Cards:

Itera sobre todos os cards selecionados.

Adiciona um ouvinte de eventos de clique em cada imagem num card.

Ao clicar em uma imagem, exibe o modal, define a imagem do modal como a imagem clicada e armazena o índice do card clicado.

Fechar o Modal:

Adiciona um ouvinte de eventos de clique no botão de fechar.

Ao clicar no botão de fechar, oculta o modal.

Fechar o Modal clicando fora:

Adiciona um ouvinte de eventos de clique na janela.

Se o clique for fora do modal, fecha o modal.

Controle de Navegação com Teclado:

Adiciona um ouvinte de eventos de teclado no documento.

Se a tecla pressionada for "Seta para a Esquerda", avança para a imagem anterior.

Se a tecla pressionada for "Seta para a Direita", avança para a próxima imagem.

Funções de Avanço/Retorno de Slides:

plusSlides(n): Adiciona ou subtrai um número de slides da posição atual.

showSlides(n): Exibe o slide na posição atual.

Inicialização do Carrossel e Efeito de Inclinação:

showSlides(slideIndex);: Inicializa a exibição dos slides no modal.

VanillaTilt.init(...): Inicializa o efeito de inclinação nos cards usando a biblioteca VanillaTilt.

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Figure 12- Código JS de frente de cartão e atrás

Em resumo, quando o botão no card é clicado, esse código alterna entre a exibição da frente e do verso do card. Se a frente estiver visível, o verso é exibido e vice-versa. Isso pode ser utilizado para criar um efeito de virada de cartas na “interface” do utilizador.

Interface gráfica do usuário

Descrição gerada automaticamente

Retângulo

Descrição gerada automaticamente com confiança baixa

Figure 13- filtragem

Esta página tem de controlar várias funcionalidades de uma página web, incluindo filtragem, controle de contadores, seleção de faixa de preço e distância.

### Página de Gestor/Admin/senhorio/aluno

Placa branca com letras pretas

Descrição gerada automaticamente com confiança média

Figure 14- Menu inicial desta página

Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

Figure 15-tabela com filtro

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Uma imagem contendo Interface gráfica do usuário

Descrição gerada automaticamente

Figure 16-formulário de adicionar alojamento

Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente com confiança média

Figure 17-tabela de remover com filtro

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto

Descrição gerada automaticamente com confiança média

Figure 18-conversa Aluno e senhorio

### Login/registar

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Figure 19-form login

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Figure 20-form registar

## Requisitos Implementados

### REQUISITOS

Aqui nesta ação disponibilizo os todos requisitos.

|  |  |  |
| --- | --- | --- |
| Nº Requisito | Descrição | Implementado |
| 1 | Autenticação do utilizador com e-mail e palavra-passe. | Não |
| 2 | Opção para o utilizador terminar a sessão. | Não |
| 3 | Recuperação da palavra-passe com envio de nova palavra-passe por e-mail. | Não |
| 4 | Acesso à aplicação sem autenticação. | Não |
| 5 | Possibilidade de alterar nome, palavra-passe e avatar por qualquer utilizador. | Não |
| 6 | Os gestores e alunos poderão filtrar os anúncios das casas por:  • Distância à ESTGOH  • Valor da renda  • Nº quartos  • Sexo | Não |
| 7 | Possibilidade de alterar nome, palavra-passe e avatar por qualquer utilizador. | Não |
| 8 | Existência de um administrador capaz de ativar/inativar gestores e administradores. | Não |
| 9 | Capacidade de o administrador criar contas de gestores sem necessidade de aprovação. | Não |
| 10 | Capacidade de o administrador ativar e desativar a aplicação | Não |
| 11 | Listagem de gestores pelo administrador. | Não |
| 12 | Responsabilidade do gestor na aprovação/reprovação de anúncios, com notificação por e-mail | Não |
| 13 | Possibilidade de o gestor inativar/reativar anúncios, com notificação por email ao senhorio. | Não |
| 14 | Consulta de todos os anúncios pelo gestor. | Não |
| 15 | Consulta de anúncios por estado pelo gestor. | Não |
| 16 | Capacidade de o gestor inativar/reativar contas de senhorios, com notificação por e-mail. | Não |
| 17 | Listagem de senhorios pelo gestor | Não |
| 18 | Edição de dados pessoais pelo gestor | Não |
| 19 | Responsabilidade do gestor na aprovação/reprovação de anúncios, com notificação por e-mail. | Não |
| 20 | Possibilidade de o gestor inativar/reativar anúncios, com notificação por email ao senhorio. | Não |
| 21 | Consulta de todos os anúncios pelo gestor. | Não |
| 22 | Consulta de anúncios por estado pelo gestor | Não |
| 23 | Registo de senhorios na plataforma com nome, e-mail e palavra-passe. | Não |
| 24 | Inserção de anúncios por senhorios. | Não |
| 25 | Listagem dos próprios anúncios pelo senhorio. | Não |
| 26 | Edição de anúncios próprios pelo senhorio. | Não |
| 27 | Opção de inativar/definir como alugados anúncios próprios pelo senhorio. | Não |
| 28 | Edição de dados pessoais pelo senhorio. | Não |
| 29 | Responder às mensagens dos alunos pelo senhorio. | Não |
| 30 | Pesquisa na lista dos próprios anúncios pelo senhorio. | Não |
| 31 | Mudança de ordenação da lista dos próprios anúncios pelo senhorio. | Não |
| 32 | Remoção de anúncios próprios pelo senhorio. | Não |
| 33 | Recebimento de e-mail sobre aproximação do fim de validade de anúncio pelo senhorio. | Não |
| 34 | Registo de aluno na plataforma com nome, e-mail e palavra-passe. | Não |
| 35 | O aluno poderá consultar anúncios. | Não |
| 36 | O aluno pode enviar uma mensagem com senhorio. | Não |
| 37 | Encriptação das palavras-passe. | Não |
| 38 | Recebimento de e-mail de aprovação de registo de conta pelos utilizadores. | Não |
| 39 | Recebimento de e-mail de aprovação ou reprovação de anúncio pelos utilizadores. | Não |
| 40 | Criação automática do administrador na primeira execução da aplicação. | Não |
| 41 | Utilização de PHP 7, HTML 5, Laravel, MySQL para o desenvolvimento da aplicação. | Não |
| 42 | Utilização da framework Laravel para facilidade de desenvolvimento e manutenção. | Não |
| 43 | Armazenamento dos dados em uma base de dados MySQL para escalabilidade e robustez. | Não |
| 45 | Indisponibilidade de casa quando o número de quartos é zero. | Não |

Tabela 1-Requsitos

### Requisitos planeados

|  |  |  |
| --- | --- | --- |
| Nº Requisito | Descrição | Implementado |
| 1 | Autenticação do utilizador com e-mail e palavra-passe. | Não |
| 2 | Opção para o utilizador terminar a sessão. | Não |
| 3 | Recuperação da palavra-passe com envio de nova palavra-passe por e-mail. | Não |
| 4 | Acesso à aplicação sem autenticação. | Não |
| 5 | Possibilidade de alterar nome, palavra-passe e avatar por qualquer utilizador. | Não |
| 6 | Os gestores e alunos poderão filtrar os anúncios das casas por:  • Distância à ESTGOH  • Valor da renda  • Nº quartos  • Sexo | Não |
| 7 | Possibilidade de alterar nome, palavra-passe e avatar por qualquer utilizador. | Não |
| 8 | Existência de um administrador capaz de ativar/inativar gestores e administradores. | Não |
| 9 | Capacidade de o administrador criar contas de gestores sem necessidade de aprovação. | Não |

Tabela 2-requsitos planeados

### Requisitos feitos

|  |  |  |
| --- | --- | --- |
| Nº Requisito | Descrição | Implementado |
|  |  |  |

Tabela 3-requisitos feitos

### Principais motivos para eventuais desvios (Sprint 1)

O principal motivo de não fazer os requisitos que foram planeados que caso de criar o fronte-end e falta em requisitos.

### Lista dos diagramas

|  |  |  |
| --- | --- | --- |
| Nome de diagrama | Descrição | Implementado |
| Fluxograma Autenticação | processo de login dos utilizadores na plataforma | Sim |
| Fluxograma Registro de senhorios na plataforma | Mostra como os proprietários se registam na plataforma. | Sim |
| Fluxograma Gestor na aprovação/reprovação de anúncios | Define como os gestores reveem os anúncios submetidos pelos senhorios. | Sim |
| Fluxograma Filtragem de anúncios por alunos | Descreve como os alunos podem filtrar anúncios de propriedades. | Sim |
| Diagrama de Classe UML | Representação visual das classes de objetos, os atributos e relacionamentos num sistema. | Sim |
| Diagrama de caso de uso | Ilustra as interações entre os utilizadores e o sistema em diferentes cenários. | Sim |
| ER Gestor De Alojamentos | Diagrama que mostra a estrutura do banco de dados do sistema de gestão de alojamentos. | Sim |

Tabela 4-Lista dos diagramas

# Conclusões

*O desenvolvimento do projeto de gestão de alojamentos para a Escola Superior de Tecnologia e Gestão de Oliveira do Hospital (ESTGOH) representou uma jornada significativa, culminando numa solução abrangente e eficiente para atender às necessidades dos alunos em busca de alojamento. Ao longo deste processo, várias conclusões importantes foram alcançadas.*

*Atendimento às Necessidades dos Utilizadores: A plataforma desenvolvida foi projetada para oferecer uma experiência amigável e intuitiva, facilitando a busca, reserva e comunicação entre alunos e senhorios. Com funcionalidades adaptadas às necessidades específicas dos utilizadores da ESTGOH, a solução proporciona uma resposta eficaz às demandas do público-alvo.*

*Utilização de Tecnologias Modernas: A escolha de tecnologias modernas, como o “framework” Laravel e o banco de dados MySQL, demonstrou ser fundamental para o desenvolvimento de uma plataforma robusta e escalável. Essas ferramentas possibilitaram a implementação de funcionalidades avançadas e a garantia da segurança e integridade dos dados dos utilizadores.*

*Metodologia Ágil e Iterativa: A adoção de uma metodologia de desenvolvimento ágil e iterativa permitiu uma abordagem flexível e adaptável ao longo do projeto. Isso possibilitou a entrega de incrementos funcionais em curtos períodos, possibilitando uma rápida validação das funcionalidades e a incorporação contínua do “feedback” dos utilizadores.*

*Importância da Avaliação Contínua: A realização de testes regulares e avaliações de usabilidade foi fundamental para garantir a qualidade e a eficácia da plataforma. Através dessas avaliações, foi possível identificar áreas de melhoria e implementar ajustes necessários para otimizar a experiência do utilizador.*

*Compromisso com a Privacidade e Segurança dos Dados: Num ambiente cada vez mais preocupado com questões de privacidade e segurança de dados, foi essencial garantir a conformidade com as regulamentações vigentes e adotar medidas robustas de proteção de informações sensíveis dos utilizadores.*

*Em suma, o projeto de gestão de alojamentos para a ESTGOH representa não apenas a conclusão bem-sucedida de um desafio académico, mas também uma contribuição tangível para a melhoria da experiência dos alunos em busca de alojamento. Ao oferecer uma plataforma eficiente e intuitiva, o projeto visa facilitar o processo de alojamento, contribuindo para a qualidade de vida e bem-estar dos estudantes.*

## Forças

*Durante o desenvolvimento do projeto de gestão de alojamentos para a Escola Superior de Tecnologia e Gestão de Oliveira do Hospital (ESTGOH), várias forças foram identificadas, contribuindo positivamente para o sucesso e eficácia da solução desenvolvida:*

*Usabilidade Aprimorada: A plataforma foi projetada com foco na usabilidade, oferecendo uma “interface” intuitiva e fácil de usar para os utilizadores. Isso aumenta a eficiência e a satisfação do utilizador ao navegar, buscar e reservar alojamentos.*

*Tecnologias Modernas: A utilização de tecnologias modernas, como Laravel e MySQL, permitiu o desenvolvimento de uma plataforma robusta, escalável e segura. Essas tecnologias oferecem recursos avançados que suportam as necessidades do sistema e garantem uma experiência de utilizador consistente e confiável.*

*Agilidade no Desenvolvimento: A adoção de uma metodologia ágil e iterativa possibilitou a entrega rápida de incrementos funcionais, permitindo uma rápida validação das funcionalidades e a adaptação contínua às necessidades dos utilizadores.*

*Adaptação às Necessidades dos Utilizadores: A plataforma foi desenvolvida com base nas necessidades específicas dos alunos da ESTGOH, garantindo que as funcionalidades e recursos atendam adequadamente às demandas do público-alvo.*

*Segurança e Privacidade dos Dados: Foram implementadas medidas robustas de segurança e privacidade dos dados dos utilizadores, garantindo a proteção e confidencialidade das informações pessoais e sensíveis.*

*Essas forças destacam os aspetos positivos e vantagens da solução desenvolvida, demonstrando o potencial para atender às necessidades dos utilizadores e proporcionar uma experiência de gestão de alojamentos eficaz e satisfatória para os alunos da ESTGOH.*

## Limitações

*Durante o desenvolvimento e implementação do projeto de gestão de alojamentos para a Escola Superior de Tecnologia e Gestão de Oliveira do Hospital (ESTGOH), algumas limitações foram identificadas, as quais são importantes reconhecer e considerar para futuras melhorias e aprimoramentos:*

*Escopo Limitado: Devido a restrições de tempo e recursos, algumas funcionalidades desejáveis podem não ter sido totalmente implementadas ou exploradas. Isso pode incluir recursos adicionais de comunicação entre alunos e senhorios, integração com sistemas de pagamento “online”, entre outros.*

*Testes Limitados: Apesar dos esforços para realizar testes rigorosos, incluindo testes de unidade, integração e aceitação, o escopo e a profundidade dos testes podem ter sido limitados. Isso pode resultar em possíveis falhas ou bugs não detetados, afetando a qualidade e estabilidade da plataforma.*

*Limitações de “Hardware” e Infraestrutura: Dependendo dos recursos disponíveis, como capacidade de armazenamento e poder de processamento do servidor, podem surgir limitações no desempenho e escalabilidade da plataforma, especialmente em períodos de alta demanda.*

*Adoção e Aceitação dos Utilizadores: A aceitação e adoção da plataforma pelos utilizadores pode ser uma limitação, especialmente se houver resistência à mudança ou falta de concretização sobre os benefícios da nova solução. Isso pode impactar a eficácia e a utilidade da plataforma no longo prazo.*

*Questões de Privacidade e Segurança: Apesar das medidas de segurança implementadas, podem surgir preocupações adicionais relacionadas à privacidade e segurança dos dados dos utilizadores. A conformidade com regulamentações de proteção de dados, como GDPR, deve ser constantemente monitorada e aprimorada.*

*Ao reconhecer essas limitações, é possível direcionar esforços para mitigar os seus impactos e buscar oportunidades para melhorias futuras, garantindo uma plataforma de gestão de alojamentos mais eficaz, segura e satisfatória para os utilizadores da ESTGOH.*

## Trabalho Futuro

*Com base nas experiências adquiridas durante o desenvolvimento do projeto de gestão de alojamentos para a Escola Superior de Tecnologia e Gestão de Oliveira do Hospital (ESTGOH), diversas oportunidades de trabalho futuro foram identificadas, visando aprimorar e expandir a plataforma para atender ainda melhor às necessidades dos utilizadores. Algumas dessas oportunidades incluem:*

*Implementação de Recursos Adicionais: Desenvolver e integrar novos recursos e funcionalidades à plataforma, como sistemas de pagamento “online”, calendário de disponibilidade de quartos em tempo real, avaliações e “feedback” dos utilizadores, entre outros.*

*Melhoria da “Interface” do Utilizador: Realizar análises de usabilidade e “design” para identificar áreas de melhoria na “interface” do utilizador e na experiência do utilizador, visando tornar a plataforma mais intuitiva, acessível e esteticamente agradável.*

*Expansão para Outros Dispositivos e Plataformas: Adaptar a plataforma para dispositivos móveis, como “smartphones” e “tablets”, garantindo uma experiência consistente e otimizada em diferentes dispositivos e sistemas operativos.*

*Integração com Outros Sistemas: Integrar a plataforma com outros sistemas e serviços utilizados pela ESTGOH, como sistemas de gestão académico e portal do aluno, para oferecer uma experiência integrada e coesa aos utilizadores.*

*Análise de Dados e Inteligência Artificial: Utilizar técnicas de análise de dados e inteligência artificial para obter intuições e prever tendências relacionadas à demanda por alojamentos, otimizando a alocação de recursos e melhorando a experiência dos utilizadores.*

*Ampliação da Base de Utilizadores: Expandir o alcance da plataforma para além dos alunos da ESTGOH, tornando-a disponível para outras instituições de ensino superior ou mesmo para o mercado de alojamentos em geral.*

*Essas são apenas algumas das oportunidades de trabalho futuro que podem ser exploradas para aprimorar e expandir a plataforma de gestão de alojamentos. Ao continuar a investir em inovação e desenvolvimento contínuo, é possível garantir que a plataforma permaneça relevante e eficaz, atendendo às necessidades em constante evolução dos utilizadores da ESTGOH e além.*

# Referências

*As referências procuram atingir três objectivos: documentar o texto, reconhecer o contributo de outros autores externos e disponibilizar apontadores para fontes externas. Os autores podem e devem citar fontes que permitam suportar ou reforçar os seus argumentos. A construção de conhecimento é baseada em conhecimento já existente, sendo assim necessário indicar que trabalho e fontes são a base desta solução. Devem ser indicados os livros, páginas da Internet, artigos, ou outro qualquer tipo de documento e fonte consultados e que apoiaram o desenvolvimento deste trabalho.*

*Para gerir as referências, é recomendável o uso de um gestor de referências (desde o Microsoft Source Manager, integrado na ferramenta Word – aba “Referencias”, secção “Citações e Bibliografia” –, a aplicativos mais capazes e independentes de plataforma, como o Mendeley, Zotero ou EndNote).*

*Existem vários formatos para a apresentação de referências, sendo os mais comuns a* ***APA*** *(American Psychological Association),* ***ACM*** *(Association for Computing Machinery), e* ***IEEE*** *(Institute of Electrical and Electronics Engineers). Na aba Referencias do MS Word, secção Citações e Bibliografia, é possível definir o formato de referências* *através da opção “Estilo”.*

*Alguns exemplos de referências, usando as normas da ACM:*

*Artigo numa revista:*

ABDELBAR, A.M., AND HEDETNIEMI, S.M. 1998. Approximating MAPs for belief networks in NP-hard and other theorems. Artificial Intelligence 102, 21-38.

*Um livro:*

GINSBERG, M. 1987. Readings in Nonmonotonic Reasoning. Morgan Kaufmann, Los Altos, CA.

*Capítulo de um livro:*

GREINER, R. 1999. Explanation-based learning. In The Encyclopedia of Cognitive Science, R. WILSON AND F. KEIL, Eds. MIT Press, Cambridge, MA, 301-303.

*Artigo publicado numa conferência:*

MAREK, W., AND TRUSZCZYNSKI, M. 1989. Relating autoepistemic and default logics. In Proceedings of the 1st International Conference on Principles of Knowledge Representation and Reasoning, Toronto, Canada, May 1989, H. BRACHMAN AND R. REITER, Eds. Morgan Kaufmann, San Mateo, CA, 276-288.

*Página da Internet:*

POLITÉCNICO DE COIMBRA, 2021. Plataforma de Inquéritos do Sistema Integrado de Garantia de Qualidade do IPC. Disponível em: <https://sigq.ipc.pt/inqueritos/>, acedido em: Janeiro de 2021, última actualização em: 2018.

## Lista de Referências

*Deve assim surgir uma lista das fontes externas utilizadas. As referências devem ser ordenadas alfabeticamente (formatos APA ou ACM) ou pela ordem em que surgem no texto (formato IEEE, neste caso cada referência tem um número se sequência associado):*

Jet Propulsion Laboratory. (2020). *Solar System Sizes and Distances, Reference Guide.* California Institute of Technology.

Mathworks. (2020). *Creating 3-D Plots*. Obtido em Janeiro de 2021, de Matlab Documentation: https://www.mathworks.com/help/matlab/visualize/creating-3-d-plots.html

## Como Integrar Referências

*Enquanto esta secção lista todas as fontes, no texto do documento estas fontes devem ser referenciadas sempre que algum elemento seja baseado ou replicado de uma fonte externa, de acordo com o formato de referência adoptado (e.g. APA: (Author et al., 2020), IEEE: [1] ).*

*Por exemplo, considere-se a seguinte passagem, onde é realizada uma citação explicita (cópia):*

a abstracção de controlo “*implementa uma forma simples de reutilização de código, ao permitir que um bloco de instruções seja escrito independentemente do programa principal*” **(Martins, 2010)**.

*É realizada uma citação explicita, ou seja, parte do texto é integralmente copiado da fonte, sendo formatado em itálico e referenciado.*

*No exemplo seguinte não é realizada uma citação explícita. O texto é escrito pelo autor mas baseado numa fonte externa:*

Em programação orientada a objectos, um objecto caracteriza-se por uma identidade única, um conjunto de atributos que o caracterizam e um conjunto de acções ou métodos **(Martins, 2010)**.

*O autor usou uma fonte, interpretou e apresentou um texto próprio baseado nessa fonte. Não realizou cópia integral, pelo que não é necessário a sua formatação em itálico, mas uma vez que o conhecimento é baseado num outro autor, continua a ser necessário citar a fonte.*

*Na secção final do documento, Referências, deve então surgir a lista de fontes que suportaram o texto. No exemplo seguinte é usado o formato APA:*

Referências

Martins, F., 2010. “Java 5 e Programação por Objectos”, Editora Lidel Edições Técnicas.

Oracle, 2016. “Java Collections”, disponível em <https://docs.oracle.com/javase/tutorial/collections/index.html>, acedido em: Fevereiro 2016, última actualização: 2015.

# Anexos

*Elementos técnicos devem ser colocados como anexo. A especificação técnica (diagramas conceptual e físico, código de criação da base de dados, diagramas de classe), protótipos, especificação de testes, manuais de utilizadores e outros documentos técnicos podem ser disponibilizados como anexo ao documento.*

*O documento principal descreve os objectivos, metodologia e resultados do trabalho. As componentes técnicas não devem ser abordadas no documento principal, mas disponibilizadas como anexo.*