

**HappyGuest**

Licenciatura em Engenharia Informática

Diogo Santos Mendes

Tomás José Carraco Neves

Leiria, julho de 2023



**HappyGuest**

Licenciatura em Engenharia Informática

Diogo Santos Mendes

Tomás José Carraco Neves

Trabalho da unidade curricular Projeto Informático realizado sob a orientação da Professora Doutora Maria Micaela Gonçalves Dinis Esteves e da Professora Doutora Ângela Margarida de Sousa Pereira.

Leiria, julho de 2023

# Agradecimentos

Na realização deste projeto final de curso, foram muitas as pessoas que, direta ou indiretamente, contribuíram para que conseguíssemos concluir esta etapa da nossa formação académica.

Às nossas orientadoras, Professora Doutora Maria Micaela Gonçalves Dinis Esteves e Professora Doutora Ângela Margarida de Sousa Pereira, pelo esforço que fizeram em estarem connosco semana após semana, motivando-nos e ajudando-nos.

Aos nossos familiares e amigos, que sempre nos apoiaram e acreditaram nas nossas capacidades.

A todos aqueles não mencionados, que, direta ou indiretamente, contribuíram para a concretização deste projeto.

A todos dedicamos este projeto e um grande obrigado.

# Resumo

Com a proliferação das redes sociais, os clientes insatisfeitos com o serviço de uma unidade hoteleira têm tendência a expor o seu descontentamento nessas plataformas. Por esse motivo, as unidades hoteleiras estão cada vez mais preocupadas em acompanhar os clientes ao longo da sua estadia, a fim de evitar a insatisfação.

Desta forma, surge a necessidade de facilitar a comunicação entre o cliente e a sua unidade hoteleira de maneira prática (através do telefone), proporcionando-lhe acesso aos serviços do hotel, possibilitando-lhe entrar em contacto para expressar a sua insatisfação ou até mesmo avaliar a sua estadia. Por outro lado, é crucial que haja um painel de controlo para os gestores hoteleiros, que permita visualizar, controlar, resolver e editar os serviços do hotel, de modo a atender às solicitações dos clientes.

O presente documento descreve os trabalhos desenvolvidos no âmbito da unidade curricular Projeto Informático, que faz parte do currículo da licenciatura em Engenharia Informática. Ele aborda o desenvolvimento do sistema *HappyGuest*, que consiste em uma aplicação *web*, uma aplicação móvel para *Android* e, por fim, uma Interface de Programação de Aplicações (API) para permitir a comunicação entre ambos os componentes.

**Palavras-chave:** *Hotel, Satisfação, Clientes, Serviços, Aplicação Móvel, Aplicação Web*

# Abstract

With the proliferation of social media, dissatisfied customers of a hotel tend to express their discontent on these platforms. Therefore, hotel establishments are increasingly concerned about monitoring customers throughout their stay to prevent dissatisfaction.

Thus, there is a need to facilitate practical communication between the customer and the hotel unit, particularly using a telephone. This would provide customers with access to hotel services and enable them to contact the hotel to express their dissatisfaction or even evaluate their stay. On the other hand, it is crucial to have a control panel for hotel managers that allows them to view, control, resolve, and edit hotel services to meet customer requests.

This document describes the work carried out within the scope of the course unit "Informatics Project", which is part of the curriculum of the bachelor’s degree in computer engineering. It focuses on the development of the HappyGuest system, which comprises a web application, a mobile application for Android, and, finally, an Application Programming Interface (API) to enable communication between both components.

**Keywords:** *Hotel, Satisfaction, Customers, Services, Mobile Application, Web Application.*

Índice

[Agradecimentos ii](#_Toc144420464)

[Resumo iii](#_Toc144420465)

[Abstract iv](#_Toc144420466)

[Lista de Figuras ix](#_Toc144420467)

[Lista de Tabelas x](#_Toc144420468)

[Lista de Siglas e Acrónimos xi](#_Toc144420469)

[1. Introdução 1](#_Toc144420470)

[1.1. Enquadramento e Definição do Problema 1](#_Toc144420471)

[1.2. Objetivos 1](#_Toc144420472)

[1.3. Estrutura do Relatório 2](#_Toc144420473)

[2. Estudo Teórico 3](#_Toc144420474)

[2.1. Debate 3](#_Toc144420475)

[2.2. Pesquisa 3](#_Toc144420476)

[2.2.1. Estudo do Mercado 3](#_Toc144420477)

[2.2.2. Público-alvo 4](#_Toc144420478)

[2.2.3. Aplicações Concorrentes e Funcionalidades 5](#_Toc144420479)

[2.2.3.1 Reservas por Código 6](#_Toc144420480)

[2.2.3.2 Interface de Serviços 7](#_Toc144420481)

[2.2.3.3 Pedidos de Quarto 7](#_Toc144420482)

[2.2.3.4 Atividades / Sessões 8](#_Toc144420483)

[2.2.3.5 Reservas de Restaurante 9](#_Toc144420484)

[2.2.3.6 Informações da Região 9](#_Toc144420485)

[2.2.3.7 Informações do Hotel 10](#_Toc144420486)

[2.2.3.8 Perfil de Utilizador 10](#_Toc144420487)

[2.2.4. Estudo de Sistema Operativo 11](#_Toc144420488)

[3. Decisão 12](#_Toc144420489)

[3.1. Mercado 12](#_Toc144420490)

[3.2. Público-alvo 12](#_Toc144420491)

[3.3. Escolha de Metodologia 12](#_Toc144420492)

[3.4. Sistema Operativo Mobile 14](#_Toc144420493)

[3.5. Framework e Linguagens (*API* e *Dashboard)* 14](#_Toc144420494)

[3.6. Planeamento 15](#_Toc144420495)

[4. Estrutura 17](#_Toc144420496)

[4.1. Definição do Produto 17](#_Toc144420497)

[4.2. Logotipo 17](#_Toc144420498)

[4.3. Lista de Requisitos 18](#_Toc144420499)

[4.4. Protótipo 21](#_Toc144420500)

[5. Desenvolvimento 26](#_Toc144420501)

[5.1. Modelo Geral 26](#_Toc144420502)

[5.2. Arquitetura Lógica do Sistema 27](#_Toc144420503)

[5.3. Ferramentas e Linguagens Utilizadas 27](#_Toc144420504)

[5.4. Modelo de Dados 28](#_Toc144420505)

[5.5. Testes 30](#_Toc144420506)

[5.5.1. Teste Unitários 30](#_Toc144420507)

[5.5.2. Testes de Usabilidade 30](#_Toc144420508)

[5.5.3. Conclusões 31](#_Toc144420509)

[6. Conclusões e Trabalho futuro 32](#_Toc144420510)

[6.1. Conclusão 32](#_Toc144420511)

[6.2. Trabalho Futuro 32](#_Toc144420512)

[Bibliografia ou Referências Bibliográficas 34](#_Toc144420513)

[Anexos 35](#_Toc144420514)

[Anexo I - Planeamento Geral 35](#_Toc144420515)

[Anexo II - Diagrama UML 36](#_Toc144420516)

# Lista de Figuras

[Figura 1 – Gráfico de utilização de computador, telefone e internet por faixa etária - ALEA 4](#_Toc144395059)

[Figura 2 – Funcionalidade código de acesso 6](#_Toc144395060)

[Figura 3 - Funcionalidade interface de serviços 7](#_Toc144395061)

[Figura 4 - Funcionalidade pedidos de quartos 8](#_Toc144395062)

[Figura 5 - Funcionalidade atividades/ sessões 8](#_Toc144395063)

[Figura 6 - Funcionalidade reservas de restaurante 9](#_Toc144395064)

[Figura 7 - Funcionalidade informações da região 9](#_Toc144395065)

[Figura 8 - Funcionalidade informações do hotel 10](#_Toc144395066)

[Figura 9 - Funcionalidade perfil de utilizador 10](#_Toc144395067)

[Figura 10 - Gráfico comparativo da utilização de dispositivos Android vs IOS Global - Statcounter 11](#_Toc144395068)

[Figura 11 - Gráfico comparativo da utilização de dispositivos Android vs IOS em Portugal - Statcounter 11](#_Toc144395069)

[Figura 12 - Plataforma GitHub Projects 13](file:///C:\GIT\Happy_Guest\HG-Notes\Relatórios\Relatório_HappyGuest.docx#_Toc144395070)

[Figura 13 - User Stories 14](file:///C:\GIT\Happy_Guest\HG-Notes\Relatórios\Relatório_HappyGuest.docx#_Toc144395071)

[Figura 14 - Versão Final do logotipo 17](file:///C:\GIT\Happy_Guest\HG-Notes\Relatórios\Relatório_HappyGuest.docx#_Toc144395072)

[Figura 15 - Versão Inicial do logotipo 17](#_Toc144395073)

[Figura 16 - Diagrama de aplicações 17](#_Toc144395074)

[Figura 17 - Diagrama UML da base de dados 23](file:///C:\GIT\Happy_Guest\HG-Notes\Relatórios\Relatório_HappyGuest.docx#_Toc144395075)

# Lista de Tabelas

[Tabela 1 - Tabela comparativa de funcionalidades 6](#_Toc144391642)

[Tabela 2 - Nível de importância dos requisitos da aplicação mobile 19](#_Toc144391643)

[Tabela 3 - Nível de importância dos requisitos do dashboard web 21](#_Toc144391644)

# Lista de Siglas e Acrónimos

|  |  |
| --- | --- |
| ESTG | Escola Superior de Tecnologia e Gestão |
| API | Interface de Programação de Aplicações |
| US | *User Stories* (História de Utilizador) |
| UML | Linguagem de Modelagem Unificada |
| CRUD | *Create, Read, Update e Delete* (Criar, Ler, Atualizar e Apagar) |
|  |  |
|  |  |

Cuidados na elaboração da lista de siglas e acrónimos:

* Ordenação alfabética;
* Apenas as que sejam relevantes para a leitura do texto.

# Introdução

Neste capítulo introdutório, apresenta-se o enquadramento e a definição do problema, os objetivos gerais e específicos do projeto, e, por fim, a estrutura do relatório.

## Enquadramento e Definição do Problema

Com a proliferação das redes sociais, é comum que os clientes insatisfeitos com o serviço de um hotel optem por expressar publicamente as suas queixas nessas plataformas online. Esta exposição pública das insatisfações pode rapidamente prejudicar a reputação de um estabelecimento hoteleiro, exercendo uma influência negativa na decisão de potenciais hóspedes. Este fenómeno coloca diante da indústria hoteleira um desafio significativo, uma vez que a imagem e a reputação desempenham papéis cruciais no sucesso de qualquer empreendimento.

Neste contexto, os hotéis estão adotando uma postura cada vez mais vigilante ao longo de toda a jornada do cliente, desde o momento da reserva até o *check-out*. O objetivo é evitar a ocorrência de experiências negativas. A administração eficaz das expectativas dos clientes e a pronta resolução de eventuais problemas se transformaram em prioridades primordiais. Essas medidas visam garantir a satisfação dos clientes e, por consequência, manter a competitividade no setor hoteleiro.

Os dispositivos móveis, como *smartphones* e *tablets*, surgem como uma valiosa oportunidade para o desenvolvimento de serviços inovadores na área da hotelaria. Isso é especialmente verdadeiro no que diz respeito à gestão das reclamações apresentadas pelos clientes.

## Objetivos

Neste projeto, pretendemos desenvolver uma aplicação que esteja disponível tanto para smartphones *Android* quanto para uso através da *web*. O objetivo primordial é aprimorar os serviços oferecidos e simplificar a comunicação entre os clientes e os estabelecimentos hoteleiros. A conceção desta aplicação busca a praticidade e a eficácia no seu uso por parte dos hotéis situados na região de Leiria. Isso visa proporcionar uma experiência mais gratificante aos hóspedes, enquanto otimiza a gestão interna dos hotéis.

Por meio dessa aplicação, os hóspedes poderão expressar suas necessidades e preocupações diretamente, enquanto os hotéis terão a capacidade de reagir de imediato e solucionar eventuais problemas. Esse processo contribui para assegurar uma estadia mais agradável para todas as partes envolvidas.

## Estrutura do Relatório

Este relatório está organizado em quatro capítulos que abrangem todo o percurso da pesquisa, planeamento, construção e desenvolvimento do projeto em questão.

No primeiro capítulo (**Estudo Teórico**), concentramo-nos na exploração teórica do contexto em que a aplicação está inserida. Isso compreende a investigação e discussão sobre hotéis, a população envolvida e os sistemas relevantes para a concretização deste projeto.

No segundo capítulo (**Decisão**), detalhamos as metodologias de desenvolvimento de *software*, especificando a abordagem escolhida para este projeto específico. Revelamos todas as decisões relativas à seleção do sistema, incluindo as linguagens de programação adotadas, entre outros aspetos. Além disso, delineamos o planeamento necessário para a execução do projeto.

O terceiro capítulo (**Estrutura**) oferece uma análise detalhada da definição do produto, logotipo, o protótipo da aplicação móvel e a lista de requisitos. Neste segmento, descrevemos todos os componentes que constituem a aplicação.

No quarto capítulo (**Desenvolvimento**), exploramos a totalidade do desenvolvimento do sistema *HappyGuest*. Isso abrange o progresso da API, a criação de ambas as aplicações (*web* e móvel), os testes realizados na aplicação móvel, bem como outros aspetos técnicos relevantes. Destacamos os desafios técnicos predominantes enfrentados durante o processo de desenvolvimento e as respetivas soluções concebidas.

Para concluir, apresentamos as considerações finais do projeto, incluindo uma avaliação holística dos resultados alcançados. Também delineamos as perspetivas futuras, as quais podem abranger melhorias na aplicação e a incorporação de novas funcionalidades, destinadas a atender às necessidades sempre mutáveis da indústria hoteleira e dos clientes.

# Estudo Teórico

## Debate

Nesta primeira fase, reunimos todas as ideias sem qualquer compromisso firme em relação à direção ou resultado. Começamos por interpretar o objetivo deste trabalho, ou seja, entender profundamente as necessidades do cliente e o propósito da aplicação que estamos a desenvolver. É crucial, nesta fase inicial, pensar nos recursos necessários para satisfazer completamente o cliente. Exploramos minuciosamente tudo o que o cliente poderia querer ou não querer ver e fazer na aplicação, buscando uma compreensão abrangente das suas expectativas e requisitos. Esta etapa é fundamental para estabelecer as bases sólidas do projeto, antes de avançarmos para a definição mais concreta de metas e estratégias.

## Pesquisa

Uma das fases fundamentais deste projeto é a fase de pesquisa. Nesta etapa, começamos por analisar o que já existe no mercado em termos de produtos concorrentes. Isso envolve um estudo minucioso de aplicações e soluções semelhantes, a fim de compreender o que funciona bem e identificar eventuais lacunas a serem preenchidas.

Além disso, durante esta fase, realizamos investigações abrangentes para identificar as melhores práticas e oportunidades que possam contribuir para o sucesso deste projeto. Essa pesquisa não se restringe apenas ao mercado, mas também abrange questões tecnológicas, considerando as melhores ferramentas e tecnologias disponíveis para alcançar os nossos objetivos.

### Estudo do Mercado

Após a análise do mercado, verificamos que existem poucos hotéis que utilizam este tipo de *software*. No entanto, é interessante notar que a maioria dos hotéis que adotam esse tipo de *software* são estabelecimentos de cinco estrelas. Esta descoberta revela uma oportunidade significativa, pois sugere que a adoção de tal tecnologia pode ser um fator distintivo para hotéis de luxo, proporcionando um serviço ainda mais personalizado e eficiente aos hóspedes.

Esta informação é valiosa para a nossa estratégia, pois indica que a procura por este *software* não está completamente saturada no mercado hoteleiro. Portanto, a nossa aplicação tem a oportunidade de servir não apenas hotéis de alto padrão, mas também um segmento mais amplo de estabelecimentos que desejam melhorar a sua eficiência operacional e a experiência do cliente.

### Público-alvo

Este tipo de aplicações pode abranger diversos tipos de público, desde jovens adultos com idades compreendidas entre os 20 e 39 anos, que são frequentes viajantes, tanto a negócios como a lazer, e estão familiarizados com a utilização de aplicações móveis para facilitar as suas estadias em hotéis. Esta faixa etária procura conveniência e eficiência nas suas experiências de hospedagem.

Até adultos de meia-idade, na faixa dos 40 aos 59 anos, muitos deles envolvidos na gestão ou propriedade de hotéis, utilizam estas aplicações para monitorizar o desempenho do estabelecimento, tomar decisões relacionadas com reservas e operações, e melhorar a eficiência global.

Para além destas faixas etárias, gestores e profissionais da indústria hoteleira de diversas idades também fazem parte deste público. Dependem destas aplicações para gerir eficazmente as operações hoteleiras, independentemente da sua idade.

Uma imagem com texto, file, captura de ecrã, Gráfico

Descrição gerada automaticamente

Figura – Gráfico de utilização de computador, telefone e internet por faixa etária

*[Fonte: ALEA]*

### Aplicações Concorrentes e Funcionalidades

No que diz respeito aos produtos concorrentes, deparamo-nos com várias aplicações, tais como "*Happy Stays X*", "*Palladium HG*", "*Vincci Hoteles*" e "*RIU Hotels*", nas quais identificamos diversas funcionalidades que pretendíamos incorporar na nossa própria aplicação.

**Termos de Pesquisa:** *Happy Guest, Hotel Group, Hotel, Reserves*

Quanto às funcionalidades, estas desempenham um papel fundamental na melhoria da experiência global dos utilizadores. A possibilidade de efetuar reservas através de códigos específicos, por exemplo, facilita o processo de reserva e pode ser uma característica distintiva que nos diferencia dos concorrentes. Além disso, uma interface de serviços intuitiva torna mais fácil para os hóspedes aceder a uma variedade de comodidades e informações relacionadas com o hotel, garantindo que aproveitem ao máximo a sua estadia.

Os pedidos de quarto são uma funcionalidade valiosa, permitindo aos hóspedes solicitar serviços adicionais diretamente através da aplicação, tornando o processo mais eficiente e conveniente. As atividades e sessões oferecem uma forma de os hóspedes participarem em eventos e programas organizados pelo hotel, enriquecendo a sua experiência durante a estadia.

A capacidade de efetuar reservas no restaurante é especialmente vantajosa, proporcionando aos hóspedes a oportunidade de desfrutar de refeições deliciosas sem complicações. Informações detalhadas sobre a região local e o próprio hotel também podem ser um diferencial, uma vez que fornecem uma visão abrangente do ambiente envolvente e dos serviços disponíveis.

Por último, mas não menos importante, o perfil de utilizador personalizado desempenha um papel vital na adaptação da experiência às preferências individuais. Isso permite que os hóspedes personalizem a sua estadia de acordo com as suas necessidades e gostos específicos, aumentando a satisfação global.

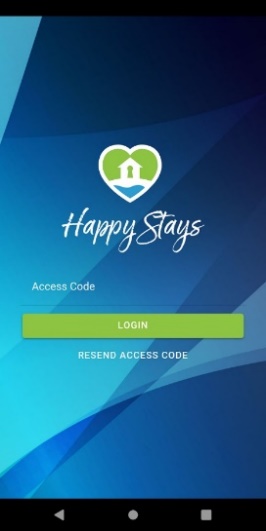
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Funcionalidades | Happy  Stays x | RIU Hotels | Palladium  HG | Vincci Hoteles | *Happy Guest* |
| Reservas por Código | ✔️ | ✔️ |  |  | ✔️ |
| Interface de Serviços |  |  | ✔️ | ✔️ | ✔️ |
| Pedidos de Quarto |  | ✔️ | ✔️ | ✔️ | ✔️ |
| Atividades / Sessões |  |  | ✔️ | ✔️ | ✔️ |
| Reservas de Restaurante |  | ✔️ | ✔️ | ✔️ | ✔️ |
| Informações da Região |  |  |  | ✔️ | ✔️ |
| Informações do Hotel | ✔️ | ✔️ |  | ✔️ | ✔️ |
| Perfil de Utilizador |  | ✔️ |  |  | ✔️ |

Tabela - Tabela comparativa de funcionalidades

### Reservas por Código

Com esta funcionalidade, cada reserva terá um código de acesso único, o que permite aos utilizadores acesso conveniente aos serviços e atividades do hotel durante a sua estadia. Este código de acesso desempenha um papel crucial como forma de identificação da estadia do cliente, fornecendo informações essenciais, como o número do quarto, a data de entrada e a data de saída, entre outras informações relevantes.

Além disso, este sistema de códigos de acesso contribui significativamente para garantir a segurança e a experiência do cliente, ao simplificar o processo de *check-out* e o uso dos serviços disponíveis no hotel.

Uma imagem com texto, Telemóvel, captura de ecrã, Dispositivo móvel

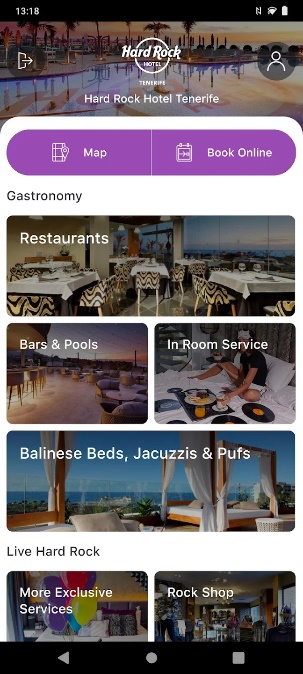
Descrição gerada automaticamente

Figura e 3 – Funcionalidade código de acesso

### Interface de Serviços

Nesta funcionalidade, o principal objetivo será listar todos os serviços disponibilizados pelo hotel, os quais os clientes podem requisitar para aprimorar sua experiência durante a estadia. A interface de serviços funcionará como um catálogo interativo, fornecendo aos hóspedes informações detalhadas sobre cada serviço oferecido, incluindo descrições, preços, horários de disponibilidade e qualquer outra informação relevante.

Além da simples listagem, a interface de serviços será projetada para facilitar a solicitação e reserva de serviços. Os clientes poderão selecionar os serviços desejados diretamente na plataforma, especificar datas e horários preferenciais, e até mesmo personalizar certos aspetos dos serviços de acordo com suas preferências individuais.

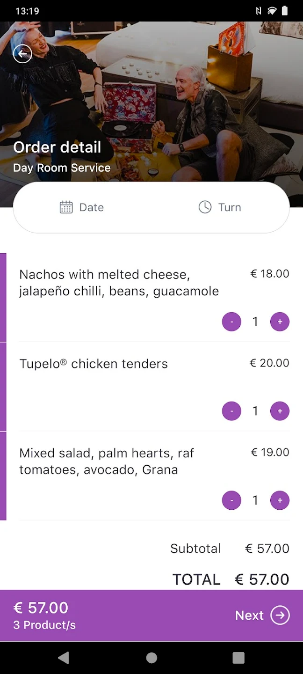
Uma imagem com texto, captura de ecrã, colagem

Descrição gerada automaticamente

Figura 4 e 5 - Funcionalidade interface de serviços

### Pedidos de Quarto

Com esta funcionalidade, os hóspedes podem fazer pedidos relacionados com os quartos onde estão alojados. Através desta interface, os clientes podem solicitar serviços específicos, como a entrega de alimentos, a requisição de objetos adicionais e até mesmo pedir a limpeza do quarto, com o objetivo de garantir uma estadia confortável e personalizada. Esta funcionalidade tem como objetivo proporcionar comodidade aos hóspedes, permitindo-lhes personalizar a sua estadia de acordo com as suas necessidades e preferências individuais.

Uma imagem com texto, captura de ecrã, Telemóvel, Dispositivo móvel

Descrição gerada automaticamente

Figura 6 e 7 - Funcionalidade pedidos de quartos

### Atividades / Sessões

Nesta funcionalidade, o objetivo é mostrar e permitir que o utilizador aceda a todas as atividades e sessões disponíveis no hotel. Isto abrange diversas opções para melhorar a experiência do hóspede, como o acesso ao SPA e ao ginásio, proporcionando assim uma experiência completa de relaxamento e bem-estar durante a estadia. Adicionalmente, nesta secção, os hóspedes podem encontrar informações sobre eventos especiais, workshops, aulas de fitness, sessões de ioga ou outros programas que possam estar disponíveis durante a sua estadia.

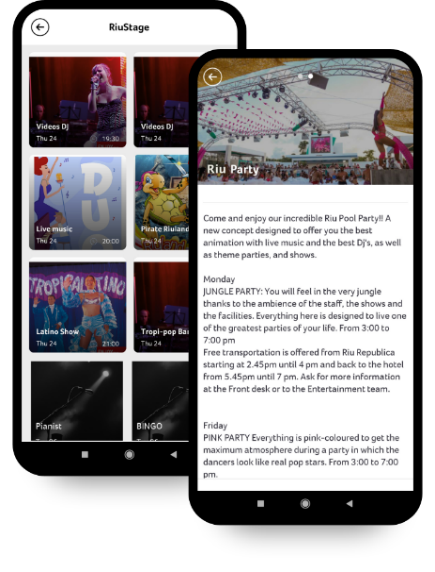


Figura 8 e 9 - Funcionalidade atividades / sessões

### Reservas de Restaurante

O foco principal desta funcionalidade é garantir que o cliente possa fazer reservas em nosso restaurante sem precisar sair do conforto do seu quarto. Esta conveniência visa melhorar a experiência do cliente, tornando o processo de reserva de mesa no restaurante do hotel o mais simples e acessível possível.

Uma imagem com texto, captura de ecrã

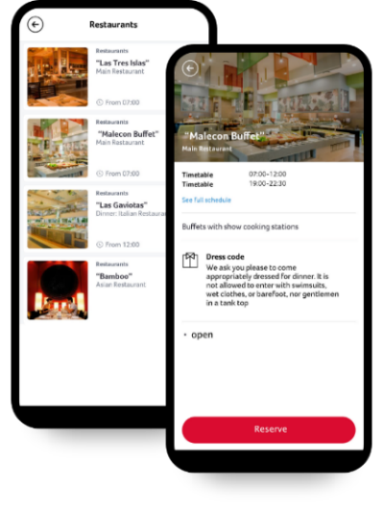
Descrição gerada automaticamente

Figura 10 e 11 - Funcionalidade reservas de restaurante

### Informações da Região

Nesta funcionalidade, o foco principal é disponibilizar aos hóspedes informações detalhadas e úteis sobre a região circundante ao hotel. Através desta funcionalidade, os hóspedes terão acesso a dados relevantes sobre atrações turísticas, pontos de interesse e outras informações que possam enriquecer a sua estadia e experiência na área. Querendo assim que os hóspedes aproveitem ao máximo a sua visita, explorando as atrações locais, desfrutando de restaurantes populares e descobrindo a cultura única da região.

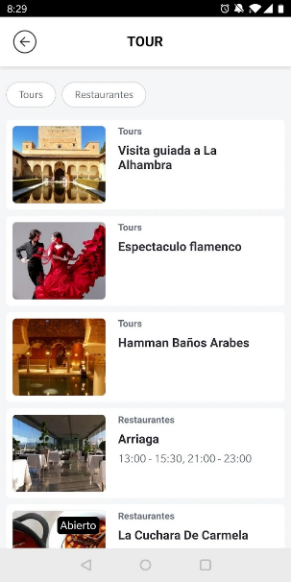


Figura 12 - Funcionalidade informações da região

### Informações do Hotel

O foco principal desta funcionalidade é disponibilizar aos hóspedes informações completas e relevantes sobre as instalações e serviços oferecidos pelo próprio hotel.

Através desta funcionalidade, os visitantes têm acesso a detalhes importantes que lhes permitirão aproveitar ao máximo as comodidades do hotel durante a sua estadia.



Figura 13 - Funcionalidade informações do hotel

### Perfil de Utilizador

Nesta funcionalidade, o foco principal é permitir que os hóspedes criem e gerenciem os seus perfis pessoais na plataforma do hotel. Através desta interface, os utilizadores podem fornecer informações importantes e detalhes de contacto, garantindo uma estadia altamente personalizada e conveniente.

Uma imagem com texto, captura de ecrã, Tipo de letra, número

Descrição gerada automaticamente

Figura 14 - Funcionalidade perfil de utilizador

### Estudo de Sistema Operativo

Após a realização de uma análise aprofundada dos sistemas operativos mais adotados em Portugal, verificamos que a preferência da maioria recai sobre o sistema *Android*. Este sistema operativo, desenvolvido pela *Google*, demonstrou uma presença robusta no mercado português, sendo escolhido por uma significativa parcela de utilizadores de dispositivos móveis. A sua natureza de código aberto, a variedade de fabricantes que o suportam e a ampla gama de dispositivos que o utilizam contribuíram para a sua popularidade.

Por outro lado, o sistema operativo *iOS*, desenvolvido pela *Apple*, também possui uma base sólida de utilizadores em Portugal. Reconhecido pela sua ênfase na segurança, design elegante e integração com o ecossistema da *Apple*, o *iOS* atrai uma base leal de utilizadores que apreciam a experiência coesa entre dispositivos da marca.

Uma imagem com gráfico

Descrição gerada automaticamente

Figura 15 - Gráfico comparativo da utilização de dispositivos Android vs IOS Global

Uma imagem com texto, captura de ecrã, diagrama, Gráfico

Descrição gerada automaticamente

Figura 16 - Gráfico comparativo da utilização de dispositivos Android vs IOS em Portugal

*[Fonte: StatCounter]*

# Decisão

Nesta etapa, tomamos decisões cruciais que, em conjunto com a fase anterior de pesquisa, exercerão influência no desenvolvimento e construção da aplicação.

## Mercado

Para a nossa aplicação, decidimos direcioná-la principalmente para os hotéis classificados **entre 3 e 4 estrelas**. Esta escolha visa facilitar o trabalho desses estabelecimentos, uma vez que muitos hotéis de 5 estrelas já possuem algum tipo de software de gestão e atendimento aos clientes.

Ao concentrarmo-nos nos hotéis de 3 e 4 estrelas, a nossa aplicação pretende preencher uma lacuna no mercado, oferecendo uma solução acessível e eficiente para melhorar a experiência dos hóspedes e otimizar a gestão dos estabelecimentos. Acreditamos que esta faixa de hotéis beneficiará significativamente da nossa aplicação, permitindo-lhes competir de forma mais eficaz no mercado e elevar a qualidade dos seus serviços.

## Público-alvo

O público-alvo que pretendemos abranger situa-se entre os **20 e os 50 anos** de idade. Geralmente, estes clientes possuem um nível cultural e económico médio a elevado, uma vez que são os tipos de pessoas que mais frequentam hotéis. Além disso, esta faixa etária é conhecida por adotar a tecnologia com maior frequência nas suas vidas diárias.

Com base nisso, o nosso objetivo é desenvolver uma aplicação simples e de fácil utilização, projetada para atender às necessidades e expectativas de qualquer pessoa dentro desta faixa etária. Queremos garantir que a nossa aplicação seja acessível e intuitiva, mesmo para aqueles que não são especialistas em tecnologia.

## Escolha de Metodologia

No desenvolvimento de *software*, é essencial adotar uma metodologia que simplifique o processo complexo de criação de um programa. Uma metodologia de desenvolvimento ou processo de *software* consiste em um conjunto estruturado de atividades vitais para a conceção de um sistema de software. Embora existam várias abordagens de processos de software, todas partilham princípios fundamentais: especificação, desenho, implementação, validação e evolução.

As metodologias de desenvolvimento podem ser categorizadas em dois grupos principais: metodologias de desenvolvimento com plano e metodologias de desenvolvimento ágil. A escolha da metodologia a aplicar depende do contexto específico em análise.

Para este projeto, optamos por seguir os princípios da **metodologia *Scrum***, uma abordagem ágil, visto acreditarmos que esta nos permitirá um melhor controlo da divisão de tarefas, otimização do tempo e integração eficiente dos testes. Ao adotar esta abordagem, estabelecemos ciclos de trabalho semanais, conhecidos como sprints, cujos objetivos são definidos em reuniões para garantir uma organização sólida.

Uma imagem com texto, captura de ecrã, número, Tipo de letra

Descrição gerada automaticamentePara gerir este processo de forma eficaz, utilizamos a plataforma *GitHub Projects*, que permite a integração do nosso repositório com um painel personalizável. Esta ferramenta auxilia-nos na gestão das tarefas, na criação de *User Stories* (US), na definição dos sprints semanais e no acompanhamento da realização dos testes associados a cada funcionalidade. Isso promove a transparência, a colaboração e a visibilidade durante todo o ciclo de desenvolvimento do projeto.

Figura 17 - Plataforma GitHub Projects

Uma imagem com texto, captura de ecrã, software, número

Descrição gerada automaticamente

Figura 18 - User Stories - GitHub Projects

## Sistema Operativo Mobile

Após conduzirmos uma análise abrangente dos sistemas operativos mais adotados em Portugal, constatamos que a preferência da maioria dos utilizadores recai sobre o *Android*. Com base nesse cenário, decidimos iniciar o desenvolvimento da aplicação móvel, dando prioridade à **plataforma *Android***. Esta escolha estratégica se fundamenta na popularidade sólida e generalizada desse sistema operativo no mercado português.

No entanto, não deixamos de considerar a perspetiva de uma expansão futura. Mantemos em mente a possibilidade de desenvolver a mesma aplicação para a plataforma *iOS* em um projeto subsequente. Essa decisão visa potencializar a abrangência da nossa aplicação e alcançar um público ainda mais vasto e diversificado.

## Framework e Linguagens (*API* e *Dashboard)*

Para o desenvolvimento deste projeto, é necessário criar uma API que permita a comunicação entre as diversas "plataformas", como a aplicação móvel e o *Dashboard* *Web*. A escolha recai sobre o ***Laravel*** (PHP) para a implementação da API, dada a sua notável facilidade e simplicidade, aliadas a recursos robustos que permitem a gestão e o processamento eficaz de dados.

Na parte do *Dashboard* (*Front-End*), a seleção recai sobre o ***Vue.js*** (*JavaScript*) para assegurar páginas altamente otimizadas e proporcionar uma experiência de utilização intuitiva. O *Vue.js* destaca-se pela sua capacidade de criar interfaces de utilizador interativas e responsivas, fator crucial para a gestão eficiente de dados e operações no *Dashboard*. Esta escolha visa garantir a agilidade, a usabilidade e a eficácia do mesmo, essenciais para a boa execução do projeto.

## Planeamento

Nesta fase, descreveremos todo o processo de planeamento definido no início do projeto. O planeamento é uma parte crítica para garantir que o projeto seja bem-sucedido desde o seu início até à conclusão. É aqui que estabelecemos as bases e diretrizes que guiarão a nossa equipa durante todo o processo.

Após a tomada de decisão das peças fundamentais para o desenvolvimento deste projeto, chegamos a uma das fases mais importantes, que é a fase de planeamento. Nesta fase, iremos considerar todas as etapas necessárias para a sua conclusão e definir datas e objetivos a cumprir ao longo das semanas. É fundamental estabelecer um cronograma realista e metas claras para manter o projeto no caminho certo.

Na primeira semana, planeamos reunir todas as informações disponíveis para iniciar o projeto o mais rapidamente possível. Começaremos com uma reunião de debate com as nossas orientadoras, onde recolheremos todas as ideias possíveis para a sua realização. Esta fase de *brainstorming* é crucial para capturar uma variedade de perspetivas e conceitos que podem enriquecer o projeto. Após o debate, iniciaremos a fase de pesquisa, onde testaremos as ideias discutidas anteriormente. Realizar pesquisas sólidas ajudar-nos-á a fundamentar as nossas decisões e a fornecer *insights* valiosos. Após a pesquisa, definiremos os objetivos e requisitos do nosso produto, estabelecendo as bases para o desenvolvimento subsequente.

A partir da segunda semana e nas semanas seguintes, planeamos começar definindo alguns protótipos para que possamos ter uma visão precisa do aspeto do produto, com o máximo de detalhes possível. A criação de protótipos permitir-nos-á visualizar o *design* e a funcionalidade do produto antes de gastar recursos significativos em desenvolvimento. Após a definição dos protótipos, iniciaremos o desenvolvimento das três aplicações: *API*, *WEB* e *MOBILE*. É importante notar que conduziremos testes e melhorias simultaneamente, não necessariamente em sequência. Isso permitirá uma abordagem mais ágil e eficiente, onde podemos identificar e corrigir problemas à medida que surgem, garantindo um produto final de alta qualidade.

Por fim, planeamos a entrega, com o objetivo de a concluir em agosto. Esta é a data-alvo que estabelecemos para a conclusão bem-sucedida do projeto. O planeamento está presente no *Anexo I*.

## Interface Aplicação Mobile

Dado que o nosso público-alvo abrange faixas etárias entre os 25 e 50 anos, optamos por uma interface simples e moderna. Inspiramo-nos em algumas aplicações semelhantes, como o "*Happy Stays*" e o "*Palladium*", uma vez que o nosso objetivo central é proporcionar uma experiência de interação fácil com a aplicação.

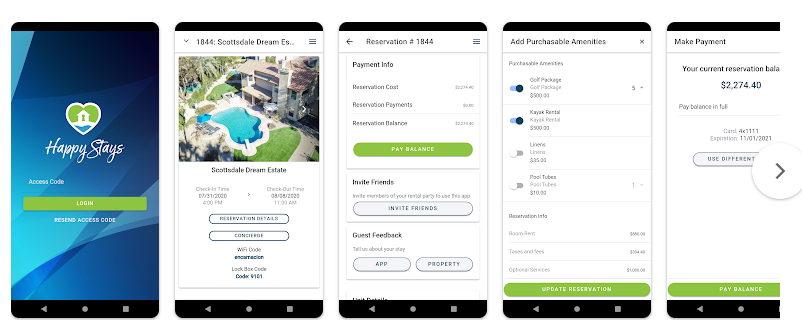


Figura - Aplicação Happy Stays

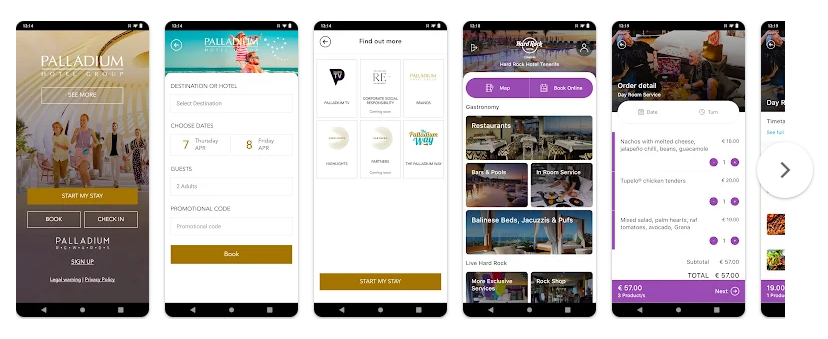


Figura - Aplicação Palladium

# Estrutura

## Definição do Produto

De modo a compreender quais as funcionalidades a considerar no sistema *HappyGuest* foi necessário realizar a análise de sistemas semelhantes, como resultado obtivemos uma pequena lista de funcionalidades que poderíamos vir a implementar na nossa aplicação final. Para além disso foram realizadas várias reuniões de acompanhamento (para além das funcionalidades descritas no enunciado) para obter uma lista final de funcionalidades a implementar tanto na aplicação móvel como na *web.*

Além do desenvolvimento das aplicações móvel e *web*, é essencial estabelecer uma forma de manter uma conexão contínua, 24 horas por dia, 7 dias por semana. Dado que a aplicação pode ser utilizada a qualquer momento e em qualquer lugar, criamos uma API hospedada num servidor privado, o qual também foi configurado por nós, contando com a colaboração de um colega do curso.

## Logotipo

Após a definição do produto, surgiu a necessidade de visualizar como este poderia efetivamente tomar forma no término do seu desenvolvimento. Antes mesmo de iniciarmos a criação das interfaces, reconhecemos a importância de dotar a nossa aplicação com um logotipo. Para tal, recorremos à aplicação "Photoshop".

Uma imagem com símbolo, design

Descrição gerada automaticamenteUma imagem com Gráficos, Tipo de letra, logótipo, design gráfico

Descrição gerada automaticamenteNo processo de conceção do logotipo, consideramos cuidadosamente diversos elementos cruciais que captariam a identidade e o propósito subjacente à nossa aplicação. Ao combinar elementos visuais e design, procuramos transmitir a essência da aplicação de maneira distintiva e coerente com a sua finalidade.

Figura - Versão inicial do logotipo

Figura - Versão final do logotipo

## Lista de Requisitos

Com todos os pontos essenciais e funcionalidades identificados, tornou-se necessário listar todos os requisitos necessários para ambas as plataformas. Para isso, registamos os requisitos específicos de cada plataforma e atribuímos a cada um deles um nível de importância.

No que diz respeito à aplicação mobile, dividimos os requisitos em três categorias:

* **Não Autenticado:** Nesta categoria, consideramos o cenário em que o utilizador entra na aplicação tendo todas as funcionalidades de autenticação disponíveis.
  + **Funcionalidades permitidas:**
    - **Autenticação:**
      * Login
      * Lembrar Sessão
      * Criar Conta
* **Autenticado Sem Código:** Aqui, consideramos o cenário em que o utilizador já está autenticado, mas não possui nenhum código de reserva. Isso permite ao utilizador ter acesso a algumas funcionalidades restritas.
  + **Funcionalidades permitidas:**
    - **Autenticação:**
      * Alterar Password
      * Terminar Sessão
    - **Perfil de Utilizador:**
      * Ver Perfil
      * Alterar dados do Perfil
      * Apagar Perfil
    - **Códigos de Acesso:**
      * Inserção de Código / QR
      * Listar Códigos
    - **Reclamações:**
      * Listar Reclamações
      * Ver Reclamação
    - **Avaliações:**
      * Listar Avaliações
      * Ver Avaliações
* **Autenticado e Com Código:** Nesta categoria, consideramos o cenário em que o utilizador está autenticado e possui um código de reserva válido. Isso permite ao utilizador ter acesso a todas as funcionalidades disponíveis na aplicação.
  + **Funcionalidades permitidas:**
* **Autenticação:**
  + - * Alterar Password
      * Terminar Sessão
    - **Perfil de Utilizador:**
      * Ver Perfil
      * Alterar dados do Perfil
      * Apagar Perfil
    - **Códigos de Acesso:**
      * Inserção de Código / QR
      * Listar Códigos
      * Desassociar Código
    - **Reclamações:**
      * Registar nova Reclamação
      * Listar Reclamações
      * Ver Reclamação
    - **Avaliações:**
      * Registar nova Avaliação
      * Listar Avaliações
      * Ver Avaliação
    - **Serviços do Hotel:**
      * Pedir Limpeza de Quarto
      * Pedir Comida no Quarto
      * Pedir Objeto no Quarto
      * Listar Serviços
      * Cancelar Serviço
    - **Reservas do Hotel:**
      * Reservar Mesa em Restaurante
      * Reservar Atividades
      * Listar Reservas
      * Editar Reserva
      * Cancelar Reserva
    - **Efetuar Check-Out**
    - **Informações:**
      * Informações do Hotel
      * Informações de Atividades
      * Informações da Região

|  |  |
| --- | --- |
| Requisitos – Aplicação *Mobile* | Importância |
| Autenticação | **1** |
| Perfil de Utilizador | **2** |
| Códigos de Acesso | **3** |
| Reclamações | **4** |
| Serviços do Hotel | **5** |
| Reservas do Hotel | **6** |
| Informações | **7** |
| Efetuar Check-Out | **8** |
| Avaliações | **9** |

Tabela - Nível de importância dos requisitos da aplicação mobile

**Nota:** Nível de importância 1-9, sendo 1 muito importante e 9 menos importante.

No que diz respeito à aplicação *web*, dividimos os requisitos em duas categorias:

* **Não Autentificado:** Nesta categoria, consideramos o cenário em que o utilizador pode efetuar o login e manter a sua sessão ativa.
  + **Funcionalidades permitidas:**
    - **Autenticação:**
      * Login
      * Lembrar Sessão
* **Autentificado:** Aqui consideramos que, o utilizador tem acesso a todas as funcionalidades disponíveis na aplicação após efetuar o login com sucesso.
  + **Funcionalidades permitidas:**
    - **Autenticação:**
      * Login
      * Lembrar Sessão
      * Alterar Password
      * Terminar Sessão
    - **Perfil de Utilizador:**
      * Ver Perfil
      * Alterar dados do Perfil
      * Apagar Perfil
    - **Gerir Contas de Acesso:**
      * Registo de novas Contas
      * Listar Contas
      * Ver Conta
      * Remover Conta
    - **Códigos de Acesso:**
      * Criação de Código / QR
      * Listar Códigos
      * Editar Código
      * Remover Código
    - **Reclamações:**
      * Listar Reclamações
      * Ver Reclamação
      * Resolver Reclamação
      * Remover Reclamação
    - **Avaliações:**
      * Listar Avaliações
      * Ver Avaliação
    - **Serviços do Hotel:**
      * Gerir Menu de Comida
      * Gerir Menu de Objetos
      * Listar Serviços
      * Ver Serviço
      * Gerir Serviço
    - **Reservas do Hotel:**
      * Gerir Atividades
      * Listar Reservas
        + Restaurante
        + Atividades
        + Sessões
      * Gerir Reserva
    - **Listar Check-Outs**
    - **Atualizar Informações:**
      * Hotel
      * Serviços
      * Atividades

|  |  |
| --- | --- |
| Requisitos – Dashboard *Web* | Importância |
| Autenticação | **1** |
| Perfil de Utilizador | **2** |
| Gerir Contas de Acesso | **3** |
| Códigos de Acesso | **4** |
| Reclamações | **5** |
| Serviços do Hotel | **6** |
| Reservas do Hotel | **7** |
| Atualizar Informações | **8** |
| Listar Check-Outs | **9** |
| Avaliações | **10** |

Tabela - Nível de importância dos requisitos do dashboard web

**Nota:** Nível de importância 1-10, sendo 1 muito importante e 10 menos importante.

## Protótipo

Para a conceção do protótipo da aplicação móvel, a opção recaiu no *Figma*, uma plataforma *web* de *design* de aplicações. Esse recurso foi escolhido com o intuito de criar uma representação minuciosa do aspeto que a nossa aplicação final irá apresentar. Conforme mencionado previamente, delineamos a interface com base nos requisitos identificados e em aplicações similares.

A escolha do *Figma* deve-se à sua capacidade de oferecer uma abordagem colaborativa, permitir visualização em tempo real e ser de utilização simples. Essas características essenciais contribuíram para a nossa compreensão acerca da aparência e do funcionamento da aplicação final, de modo abrangente.

Na próxima secção deste relatório, todas as janelas criadas no protótipo serão descritas em detalhe.

Na *Figura* seguinte, é possível observar a página inicial, que funciona como uma tela de carregamento da aplicação.

Uma imagem com texto, póster, edifício, céu

Descrição gerada automaticamente

Figura 17 - Página de entrada

Nas *Figuras* seguintes, apresentamos as duas páginas relacionadas com a autenticação: a janela de início de sessão, que possibilita ao cliente aceder à aplicação, e a janela de criação de conta, na qual o cliente pode efetuar o seu registo na aplicação.

Uma imagem com texto, captura de ecrã, Tipo de letra, design

Descrição gerada automaticamente

Figura 18 - Página de login

Uma imagem com texto, captura de ecrã, Tipo de letra, design

Descrição gerada automaticamente

Figura 9 - Página de criação de conta

No que diz respeito à página inicial, existem duas variantes possíveis. A primeira é apresentada quando o cliente não possui um código de reserva atualmente válido (*Figura 23*). A segunda corresponde a uma página inicial que inclui um código de reserva válido (*Figura 24*).

Uma imagem com texto, captura de ecrã, Website, Página web

Descrição gerada automaticamente

Figura 20 - Página *home* sem código de reserva

Uma imagem com captura de ecrã, colagem

Descrição gerada automaticamente

Figura 21 - Página *home* com código de reserva

Na *Figura* seguinte podemos verificar o menu lateral da aplicação com o objetivo de simplificar a navegação do utilizador, permitindo aceder a páginas com facilidade.

Uma imagem com texto, captura de ecrã

Descrição gerada automaticamente

Figura 22 - Menu lateral

Nas cinco figuras subsequentes, estão representadas todas as páginas relativas ao perfil. Na *Figura 26*, é possível observar o perfil do utilizador. Na *Figura 27*, é apresentado o perfil do utilizador com a capacidade de atualizar informações. A *Figura 28* mostra a página onde o utilizador pode modificar a sua palavra-passe, por fim, na *Figura 29*, está retratada a página de confirmação, onde o utilizador pode proceder à eliminação da sua conta.

Uma imagem com texto, captura de ecrã, Cara humana, Página web

Descrição gerada automaticamente

Figura 23 - Perfil de utilizador

Uma imagem com texto, captura de ecrã, Cara humana, Website

Descrição gerada automaticamente

Figura 24 - Atualizar perfil

Uma imagem com texto, captura de ecrã, Cara humana

Descrição gerada automaticamente

Figura 25 - Atualizar palavra-passe

Uma imagem com texto, Cara humana, captura de ecrã

Descrição gerada automaticamente

Figura 26 - Eliminar conta

# Desenvolvimento

Neste capítulo, serão expostos o modelo geral e a arquitetura lógica do sistema *HappyGuest*, detalhando os diversos componentes que o constituem. Serão ainda exploradas as ferramentas e linguagens empregues ao longo do desenvolvimento do projeto, juntamente com o modelo de dados da aplicação, representado pelo Diagrama UML. Além disso, será descrito o processo de desenvolvimento das aplicações, abrangendo as várias funcionalidades incorporadas, os testes de usabilidade e unitários realizados e, por último, as melhorias e recursos adicionais implementados.

## Modelo Geral

Com o propósito de atender aos requisitos identificados no capítulo anterior e oferecer uma solução abrangente, concluiu-se que a criação de uma aplicação *web* para gestão hoteleira e de uma aplicação móvel para clientes era essencial.

A aplicação *web* destina-se a todos os membros da equipa hoteleira, com especial foco nos gestores. Desenvolvida como um painel de controlo com várias funcionalidades, optou-se por uma interface *web*, visto que a sua utilização primária ocorrerá em computadores e tablets. A aplicação móvel, por outro lado, direciona-se aos clientes do hotel. Neste caso, a prioridade era garantir usabilidade, praticidade e rapidez, razão pela qual se optou por desenvolver a aplicação para a plataforma *Android*.

|  |  |
| --- | --- |
| Cargos | Aplicação |
| Administradores | **Aplicação *web*** |
| Gestores Hoteleiros | **Aplicação *web*** |
| Funcionários | **Aplicação *web*** |
| Clientes | **Aplicação *móvel*** |

Apesar de serem duas aplicações distintas, operam de forma interligada. Toda a informação é processada através de uma API, permitindo sincronização em tempo real e facilitando o tratamento de dados. Este design de sistema, composto por aplicações *web* e móvel, resultou na criação de diferentes tipos de utilizadores, cada um associado a um tipo de dispositivo e função, como é detalhado na tabela seguinte.

Tabela - Tipos de utilizador por aplicação

Importa salientar que, na aplicação *web*, os administradores detêm acesso a todos os recursos e utilizadores, conferindo-lhes a capacidade de efetuar modificações e remoções conforme necessário. Por outro lado, os gestores possuem a capacidade de alterar informações de outros clientes, embora não disponham sempre da autorização para efetuar remoções indiscriminadas.

## Arquitetura Lógica do Sistema

A arquitetura lógica do sistema repousa no modelo cliente-servidor, em que os clientes emitem solicitações de serviços ao servidor e este disponibiliza os recursos/serviços requeridos. As aplicações móvel e *web* exibem a totalidade da informação aos utilizadores, sendo para tal necessário que efetuem pedidos ao servidor através de uma API. Além disso, na base de dados, as operações CRUD (*Create*, *Read*, *Update* e *Delete*) são realizadas. Na Figura subsequente, é possível visualizar um esquema que ilustra todo este contexto.

Uma imagem com diagrama

Descrição gerada automaticamente

Figura - Arquitetura lógica do sistema

## Ferramentas e Linguagens Utilizadas

Para concretizar o desenvolvimento da aplicação *HappyGuest*, foi indispensável empregar diversas linguagens e técnicas de programação. No âmbito da aplicação móvel, concebida para sistemas operativos *Android*, utilizou-se a linguagem de programação *Java*, desenvolvendo-a no ambiente de desenvolvimento *Android Studio*. No que diz respeito à aplicação *web*, adotou-se uma variedade de linguagens de programação, nomeadamente *HTML*, *CSS* e *JavaScript*, com destaque para a *framework* *Vue.js*.

No segmento do servidor, a escolha recaiu sobre a *framework* de desenvolvimento *Laravel* 10.0, que também viabiliza a criação de uma API externa. Essa API desempenha um papel essencial na segurança da aplicação, atuando como intermediário entre o cliente e o servidor. Recorre ao sistema REST (*Representational State Transfer*) para a transmissão de dados entre a aplicação cliente e o servidor. A transferência de dados ocorre por meio do formato JSON (*JavaScript Object Notation*), devido à sua natureza leve, baseada em texto e facilmente. O *Laravel*, uma *framework* de desenvolvimento *web*, baseia-se na linguagem de programação *PHP*.

No que concerne à gestão do armazenamento de informação, criou-se uma base de dados *MySQL* que possibilita a consulta de dados por intermédio de um conjunto de consultas *SQL*.

## Modelo de Dados

Durante o decorrer do projeto, elaboramos um diagrama UML da base de dados com o intuito de compreender os atributos essenciais e para nos apoiar em caso de necessidade de alguma alteração.

Este diagrama UML da base de dados desempenhou um papel crucial na definição da estrutura subjacente ao projeto. À medida que o projeto progredia e novos requisitos emergiam, o diagrama permitia-nos ter uma visão clara das interligações entre atributos e tabelas. Isso facilitava a identificação dos atributos essenciais e possibilitava a realização de ajustes na estrutura da base de dados, quando necessário.

Em suma, o diagrama UML da base de dados revelou-se uma ferramenta inestimável ao garantir que a arquitetura da base de dados estivesse perfeitamente alinhada com as exigências do projeto.

O diagrama UML está apresentado no *Anexo II*, descrevendo de seguida as suas tabelas:

**Utilizadores**: Armazena várias informações de utilizadores, como email, nome, número de telefone, tipo de utilizador, entre outros.

**Códigos**: Mantém dados sobre códigos, incluindo entrada, saída, quartos, entre outros detalhes.

**Códigos do utilizador**: Tabela de associação que relaciona utilizadores e códigos, registando os códigos associados a cada utilizador.

***Check-outs***: Regista as saídas de códigos (reservas) e respetivas informações, como data, etc.

**Avaliações**: Armazena avaliações do hotel realizadas pelos utilizadores, incluindo estrelas, data, comentário, etc.

**Reclamações**: Guarda informações sobre as reclamações submetidas pelos utilizadores, como título, comentário, local, data, etc.

**Ficheiros de reclamações**: Tabela associada a reclamações, onde são mantidos os ficheiros submetidos junto com as reclamações.

**Hotéis**: Contém dados do hotel, como nome, contactos, emails, capacidade, políticas, etc.

**Regiões**: Guarda informações sobre a região, como atividades e monumentos próximos.

**Serviços**: Descreve os serviços do hotel, incluindo dados de contacto, localização, horário, etc.

**Itens**: Mantém os menus de objetos e alimentação utilizados pelos serviços, com informações como nome, preço, stock, etc.

**Itens de serviços**: Tabela de associação entre itens e serviços, indicando quais itens estão ativos em cada serviço.

**Reservas**: Regista todas as reservas de serviços feitas pelos clientes, incluindo data, número de pessoas, comentário, etc.

**Pedidos**: Armazena pedidos feitos pelos clientes para determinados serviços, contendo comentário, data, serviço, etc.

**Itens de pedidos**: Tabela de associação entre pedidos e itens, determinando quais itens estão incluídos em cada pedido.

## Desenvolvimento da API

Após a definição dos requisitos da aplicação e a elaboração do modelo de dados, procedemos ao desenvolvimento da nossa API, uma vez que este constituía um dos primeiros passos na concretização das nossas aplicações.

### Instalação e Configuração

Durante a instalação e configuração do Laravel, também incorporamos o *Laravel Passport*, um componente que proporcionará autenticação através de *tokens*. Além disso, incluímos o *Laravel Lang*, que nos possibilitará a configuração da nossa API em diversas línguas, incluindo o português e o inglês, uma vez que estamos a tratar de assuntos hoteleiros, abrangendo potencialmente clientes de várias nacionalidades. Adicionalmente, implementamos o *Laravel Cascade Soft Deletes*, uma ferramenta que aprimora a gestão da remoção de dados em tabelas onde a utilização de soft deletes seja aplicável.

Sempre que é efetuado um *request* à API ela retorna as mensagens em português caso o dispositivo móvel esteja configurado em PT ou caso seja o caso de ser outra língua retorna em inglês.

Uma imagem com texto, captura de ecrã, Tipo de letra, design

Descrição gerada automaticamente

Figura - Línguas da API

### Middlewares

Com o intuito de salvaguardar as rotas e, por conseguinte, os dados na nossa API, desenvolvemos quatro *middlewares* distintos: o "*Authenticate*", o "*Authorize*" e, por último, "*Role*" e por último o “*ValidCode*”. Estes mecanismos possibilitam a proteção de quaisquer rotas que requeiram tal segurança.

### Authenticate

O propósito fundamental deste *middleware* é garantir um nível de autenticação, concedendo o acesso exclusivamente àqueles que já concluíram com sucesso o processo de autenticação. O mesmo desempenha um papel crucial na segurança da aplicação, assegurando que apenas utilizadores autenticados tenham permissão para interagir com as diferentes funcionalidades e recursos disponibilizados pela API.

### Authorize

O objetivo deste *middleware* é restringir o acesso dos utilizadores a visualizar ou atualizar informações de terceiros, com foco nos clientes. Por outro lado, os gestores e administradores têm a capacidade de superar esta camada de proteção. No caso dos gestores, têm a permissão para visualizar outros gestores, embora não lhes seja permitido efetuar edições ou eliminações. No entanto, os administradores possuem pleno acesso e controlo sobre todos os recursos e funcionalidades. Esse mecanismo assegura uma segregação adequada de funções, garantindo a privacidade dos utilizadores enquanto permite que os gestores e administradores cumpram as suas responsabilidades específicas.

### Role

De acordo com a sequência dos *middlewares* anteriores, este *middleware* em específico tem a finalidade de controlar o acesso a rotas com base nos tipos de utilizadores. Se uma rota for restrita a gestores, os administradores também terão permissão para acedê-la. No entanto, caso a rota esteja exclusivamente destinada aos administradores, nenhum outro tipo de utilizador terá acesso. Estabelece-se, assim, uma hierarquia na qual os clientes detêm o nível de acesso mais baixo, seguidos pelos gestores, funcionários e, por fim, os administradores. Essa estrutura hierárquica de acesso ajuda a manter a segurança, as permissões e os privilégios bem definidos, contribuindo para um ambiente controlado e adequado às funções específicas de cada utilizador.

### ValidCode

Concluindo, a função principal deste *middleware* é autorizar exclusivamente os clientes que possuam, no momento, um código válido. Em casos contrários, as operações não são autorizadas. Esta medida visa garantir que apenas clientes com códigos válidos possam aceder e interagir com as funcionalidades da aplicação. Isto contribui para manter a integridade das operações e assegurar que apenas aqueles que possuem os devidos códigos válidos possam realizar ações autorizadas.

### Modelos

Seguindo o diagrama UML desenvolvido ao longo do projeto foram implementados todos os modelos e o relacionamento dos mesmos, permitindo assim criar os controladores dos mesmos, onde vão estar presentes todas as operações realizadas.

### Rotas

Após a criação dos modelos e controladores, avançamos para a implementação das rotas do projeto, considerando as devidas proteções por meio dos middlewares. Essas rotas foram desenvolvidas para abranger as operações CRUD para praticamente todos os modelos envolvidos no sistema. Esse processo garante uma estrutura completa de gestão de dados, permitindo que as operações essenciais sejam executadas com segurança e eficiência.

Uma imagem com texto, captura de ecrã, software

Descrição gerada automaticamente

Figura - Rotas da API

### Autenticação

Para realizar a autenticação, conforme mencionado anteriormente, optamos por utilizar o *Laravel Passport*. Configuramos rotas específicas para os processos de login e registo, com a funcionalidade adicional de lembrar sessão. Nesse sentido, implementamos um sistema de autenticação com base em endereço de email e palavra-passe, que, quando bem-sucedido, resulta na geração de um *token*. Esse *token* torna-se fundamental para todas os *requests* subsequentes, garantindo que as interações da aplicação sejam validadas e autorizadas.

### Filtros, Ordenação e Pesquisa

Após completarmos a implementação das operações CRUD nos controladores, seguindo a lógica definida pelos requisitos do projeto, avançamos para a incorporação de funcionalidades adicionais nos pedidos (*requests*) das tabelas. Isso inclui a adição de filtros, recursos de ordenação e capacidades de pesquisa. Através da aplicação desses elementos, permitimos que os utilizadores possam refinar e personalizar as suas consultas, tornando a interação com os dados mais versátil e adaptada às suas necessidades específicas.

### Notificações

Optamos pela utilização do serviço *Firebase Messaging* para a implementação das notificações no nosso sistema. Este serviço viabiliza a criação de um servidor que possibilita o envio de notificações direcionadas para utilizadores específicos. A escolha desta abordagem deve-se à sua praticidade tanto para dispositivos *web* como móveis, facilitando a implementação de notificações bidirecionais entre clientes e gestores do hotel. Esta solução proporciona uma experiência mais interativa e eficaz de comunicação entre as partes envolvidas no sistema.

Uma imagem com texto, captura de ecrã, software, Ícone de computador

Descrição gerada automaticamente

Figura - Notificações Firebase

### Estatísticas

Adicionalmente, integramos várias funcionalidades estatísticas na nossa API, permitindo assim que os gestores possam obter informações detalhadas sobre vários conjuntos de dados ao longo dos meses. Isso proporciona uma análise profunda dos padrões e tendências, fornecendo aos gestores uma visão mais informada e orientada para a tomada de decisões estratégicas.

### Serviço de Email

Uma vez que não criámos um servidor de email próprio, incorporámos na API uma funcionalidade para simular o envio de emails, incluindo casos como a partilha de códigos ou avaliações. Para atingir esse objetivo, utilizamos o serviço *MailTrap*, que permite enviar emails para uma caixa de correio privada. Isso proporciona uma abordagem prática para testar e verificar o processo de envio de emails dentro da aplicação.

Uma imagem com texto, captura de ecrã, software, Tipo de letra

Descrição gerada automaticamente

Figura - Serviço de emails - MailTrap

## Desenvolvimento da *WEB*

### Instalação e Configuração

No processo de desenvolvimento *web*, optamos por utilizar um *template* gratuito denominado "*Tailwind Vue 3*," desenvolvido por *JustBoil* e baseado em *Vue.js*. Esta escolha possibilitou uma economia significativa de tempo, em comparação com a criação completa desde o início. No entanto, realizamos ajustes personalizados para atender às nossas necessidades específicas. Uma vez concluída esta etapa, procedemos à configuração do nosso website, marcando o início da fase de desenvolvimento propriamente dita. É importante também referir que o website tem 2 temas, o modo claro e o modo escuro, sendo possível alterar o mesmo estando autenticado.

Uma imagem com texto, captura de ecrã, software, Tipo de letra

Descrição gerada automaticamente

Figura - Website do template

Uma imagem com texto, captura de ecrã, software, Software de multimédia

Descrição gerada automaticamente

Figura - Website inicial

### *Login* e Lembrar Sessão

Após a instalação e configuração, iniciámos a implementação do processo de *login* no *website*, o qual requer o uso de um endereço de email e uma palavra-passe. A página inicial do *website* foi projetada de forma simples, sendo que o acesso não é permitido sem autenticação. Nesses casos, os utilizadores são sempre redirecionados para a página inicial ou para a página de *login*. Caso o utilizador selecione a opção de lembrar sessão os seus dados vão ser guardado em memória e a sessão vai ser sempre restaurada até terminar a mesma. A *Figura* seguinte exibe a página inicial, enquanto a seguinte apresenta a página de login.

Uma imagem com texto, design, Tipo de letra, captura de ecrã

Descrição gerada automaticamente

Figura - Página inicial

Uma imagem com texto, captura de ecrã, software, Sistema operativo

Descrição gerada automaticamente

Figura - Página de login

Após o utilizador (neste caso funcionários) insira os dados e clique login, é efetuado um pedido à API para tentar realizar a autenticação, é importante referir que os clientes não têm acesso a esta aplicação. Sempre que existe um erro é apresentado no ecrã como pode ser visto na próxima *Figura*.

Uma imagem com texto, captura de ecrã, software, Sistema operativo

Descrição gerada automaticamente

Figura - Tentativa de login

Uma imagem com texto, captura de ecrã, software, Sistema operativo

Descrição gerada automaticamente

Figura - Tentativa 2 de login

### *Home*

Após a autenticação bem-sucedida, o próximo ecrã apresentado é a página principal do *website*. Neste ecrã, são exibidas diversas estatísticas relacionadas com vários dados registados nos últimos meses. A barra de navegação varia de acordo com o tipo de utilizador. Existem dois tipos distintos: uma para os funcionários e outra para os gestores e administradores. A primeira imagem retrata a página principal com a barra de navegação para os gestores e administradores e a segunda a barra de navegação destinada aos funcionários.

Uma imagem com texto, captura de ecrã, Software de multimédia, Software gráfico

Descrição gerada automaticamente

Figura - Página inicial dashboard

Uma imagem com texto, captura de ecrã, diagrama, Software gráfico

Descrição gerada automaticamente

Figura - Página inicial funcionários

Não é permitido ver as avaliações e reclamações, eliminar dados e apenas podem editar alguns dados simples, o tipo de utilizador funcionário tem as permissões muito restritas.

### Perfil

Na página de perfil do utilizador, é possível efetuar a atualização dos seus dados pessoais, como o nome, email, morada, entre outros. Além disso, proporcionamos a opção de atualizar a palavra-passe e, por último, permitimos a remoção da conta do *website*.

Uma imagem com texto, captura de ecrã, software, Software de multimédia

Descrição gerada automaticamente

Figura - Página perfil de utilizador

Caso deseje proceder à remoção da conta, será apresentada uma verificação na qual terá de inserir a palavra-passe para confirmar a ação, conforme ilustrado na *Figura* seguinte.

Uma imagem com texto, captura de ecrã, software, Tipo de letra

Descrição gerada automaticamente

Figura - Confirmação de eliminar conta

Se o utilizador for um administrador ou gestor, terá a capacidade de aceder aos perfis de outros utilizadores. Quando o utilizador selecionado tem um nível hierárquico inferior ao do administrador ou gestor (Cliente < Funcionário < Gestor < Administrador), é possível realizar diversas ações, tais como: atualizar os dados do utilizador, redefinir a palavra-passe, eliminar a conta, bloquear ou desbloquear a conta. Em caso de bloqueio, o acesso do utilizador será negado, sendo recusado o login. Sendo as opções de remoção e de reposição de palavra-passe com confirmação de palavra-passe do utilizado atual.

Uma imagem com texto, captura de ecrã, software

Descrição gerada automaticamente

Figura - Perfil de outro utilizador (visto por admin)

### Utilizadores

Como gestor ou administrador, terá a possibilidade de aceder à página que lista os utilizadores registados. Nesta página, será exibida uma tabela que pode ser ordenada tanto de forma crescente como decrescente. Além disso, é possível aplicar filtros, como tipo de utilizador ou utilizadores bloqueados, entre outros. A funcionalidade de pesquisa também está disponível, permitindo pesquisar utilizadores por nome ou email. Isso proporciona uma gestão eficaz dos utilizadores, permitindo identificar, organizar e aceder rapidamente às informações necessárias. Permite também alguma ações rápidas como: bloquear / desbloquear utilizadores, abrir o perfil do mesmo e até remover a conta.

Uma imagem com captura de ecrã, texto, software, Ícone de computador

Descrição gerada automaticamente

Figura - Tabela de utilizadores

### Registo de Utilizadores

A única forma de registar utilizadores é através de um processo interno, o registo não está acessível publicamente. Apenas os administradores têm a permissão para registar clientes, funcionários, gestores ou outros administradores. No entanto, os gestores têm a capacidade de registar exclusivamente clientes. Ao aceder à página de registo de utilizadores, é suficiente preencher as informações necessárias e a conta será criada.

Uma imagem com texto, captura de ecrã, software, Software de multimédia

Descrição gerada automaticamente

Figura - Página registo de utilizador

### Códigos

Uma vez mais, na qualidade de gestor ou administrador, terá a capacidade de visualizar todos os códigos criados (reservas). Cada código inclui informações sobre a data de entrada e saída, podendo estar associado a vários quartos. Cada código destina-se a permitir que os clientes se autentiquem na aplicação móvel.

Na tabela de códigos, é possível verificar se os códigos ainda estão válidos, se já foram utilizados ou se estão em processo de check-out. Além disso, existem opções para filtrar, ordenar e pesquisar por código. Também estão disponíveis ações rápidas, tais como visualizar os clientes associados a um código, editar um código, mostrar o código QR (para mostrar a um cliente no hotel) ou até mesmo eliminar um código.

Uma imagem com captura de ecrã, texto, software, Sistema operativo

Descrição gerada automaticamente

Figura - Tabela de códigos

Uma imagem com texto, captura de ecrã, quadrado, Retângulo

Descrição gerada automaticamente

Figura - Código QR de código

Uma imagem com texto, captura de ecrã, multimédia, Tipo de letra

Descrição gerada automaticamente

Figura - Clientes associados a código

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Figura - Remover código

### Criar / Editar Códigos

Ao clicar no botão para criar um código, surge um *pop-up* no qual é introduzido o código, as datas e os quartos correspondentes. Além disso, há um campo destinado a inserir um endereço de email. Este endereço de email é utilizado para enviar o código ao cliente. Se desejar, pode partilhar imediatamente o código com o cliente. O código será enviado para a plataforma *MailTrap* (uma vez que não possuímos um servidor de emails próprio), e o cliente receberá a notificação correspondente. Isso permite uma comunicação eficaz com os clientes, simplificando a partilha de códigos de reserva.

Adicionalmente, está disponível um botão para criar um código aleatório, o que pode ser útil para agilizar o processo, simplificando a criação de códigos de reserva. Isso oferece uma opção conveniente para casos em que é necessário criar rapidamente um código.

Uma imagem com texto, captura de ecrã, software, Tipo de letra

Descrição gerada automaticamente

Figura - Pop-up criação de código

A opção de edição de códigos é bastante simples e semelhante. Basta selecionar o código que deseja editar e realizar as alterações nos dados desejados.

Uma imagem com texto, captura de ecrã, Tipo de letra, software

Descrição gerada automaticamente

Figura - Pop-up editar código

## Testes

Nesta secção, serão descritos todos os testes realizados durante e após a realização do projeto. Esta etapa desempenha um papel crucial no assegurar da qualidade e funcionamento adequado das aplicações desenvolvidas, bem como na identificação e correção de eventuais problemas ou falhas.

### Teste Unitários

Testes unitários são uma prática de teste de *software* em que pequenas unidades individuais de código são testadas de forma isolada para garantir que funcionem corretamente. Essas unidades de código podem ser funções, métodos de classes ou mesmo partes específicas de uma função maior. A ideia fundamental dos testes unitários é verificar se cada unidade de código produz os resultados esperados quando é chamada ou executada.

Inicialmente, após a conclusão de uma US, realizávamos testes relacionados com a mesma utilizando uma aplicação de testes autónomos chamada *Katalon*. Nesta fase, escrevíamos os casos de teste e a aplicação *Katalon* executava-os de forma autónoma. Esta abordagem inicial proporcionou-nos uma forma eficaz de validar as funcionalidades à medida que eram desenvolvidas.

No entanto, à medida que o desenvolvimento da aplicação progredia e a complexidade aumentava, começaram a surgir desafios na realização destes testes automatizados. Problemas de integração e de adaptação a mudanças frequentes na aplicação levaram-nos a reavaliar a nossa abordagem.

Foi então que optámos por realizar os testes unitários manualmente, sem recorrer a aplicações externas. Esta mudança permitiu-nos ter um controlo mais direto sobre os testes e adaptar-nos facilmente às mudanças na aplicação à medida que esta evoluía.

### Testes de Usabilidade

Os testes de usabilidade desempenham um papel crítico no processo de desenvolvimento de *software*, centrando-se na avaliação da experiência do utilizador ao interagir com a aplicação. O seu principal objetivo é assegurar que a aplicação seja intuitiva, de fácil utilização e que corresponda às necessidades dos utilizadores finais.

Numa fase final do desenvolvimento, elaboramos testes de usabilidade para a aplicação *mobile*. Para a realização destes testes, criamos um formulário no *Google Forms* com várias tarefas que o utilizador teria de realizar. Após a conclusão de cada tarefa, o utilizador registava o tempo gasto, a quantidade de erros cometidos e o nível de dificuldade. Após completar todas as tarefas, o utilizador era convidado a avaliar a aplicação, respondendo a algumas perguntas no formulário.

Após cada teste, foi criado um registo de avaliação para cada utilizador. Este registo incluía informações sobre o tempo previsto e o tempo efetivamente gasto, qualquer erro cometido pelo utilizador e eventuais desafios encontrados durante as tarefas.

### Conclusões

Os testes de usabilidade eram compostos por 10 tarefas. Como ilustrado nas Figuras x, x e x, os utilizadores, em geral, não enfrentaram dificuldades significativas ao realizar essas tarefas como podemos verificar comparando os temos previstos e os tempos realmente gastos.

Relativamente aos questionários finais compostos por X perguntas…..

# Conclusões e Trabalho futuro

Neste capítulo, procede-se a uma análise de todo o trabalho desenvolvido e dos objetivos alcançados. São também mencionados alguns aspetos relevantes num contexto de trabalho futuro.

## Conclusão

A aplicação implementada tem como objetivo reduzir o número de reclamações de uma unidade hoteleira, facilitando a comunicação entre o cliente e o gestor hoteleiro. Mais concretamente, agilizar a forma como o cliente faz pedidos ou reservas no hotel durante a sua estadia.

De todas as funcionalidades descritas, destacamos as principais como o pedido de objetos e alimentos no quarto em que o cliente pede o(s) objeto(s) ou alimentos consoante o stock existente no hotel, desta forma permite ao hotel ter uma gestão de stock e facilitar tanto a parte do cliente como a parte da gestão hoteleira. A destacar também as reservas de mesa no restaurante em que o cliente simplesmente indica o número de pessoas e o horário que pretende, no entanto a parte hotelaria tem um limite por hora de reservas que pode ser alterado a qualquer momento, permitindo uma melhor gestão do restaurante.

Por fim, considera-se que os objetivos inicialmente propostos para este projeto foram cumpridos na íntegra com êxito.

## Trabalho Futuro

Numa perspetiva futura, a aplicação *HappyGuest* poderia expandir-se para outras plataformas móveis, como o *iOS*, por exemplo. Além disso, seria benéfico facilitar ao gestor a partilha das avaliações dos clientes com outros websites, como o *TripAdvisor*, entre outros serviços similares. Para melhorar a experiência do utilizador, consideraríamos a integração com um servidor de e-mails que permitisse a confirmação após a criação de uma conta, bem como a recuperação da palavra-passe, caso fosse necessário.

Em unidades hoteleiras que oferecem um leque mais amplo de serviços, poderá tornar-se essencial criar uma forma de automatizar a inserção e remoção de serviços na aplicação móvel. Para atrair clientes que valorizam a privacidade dos seus dados, poderia ser vantajoso permitir o uso da aplicação sem autenticação, simplificando assim o processo de entrada e uso da plataforma.

Além disso, para facilitar a gestão dos pagamentos dos serviços, poderia ser implementada a opção de pagamento digital quando um cliente solicita um serviço. Em relação à administração do hotel, seria útil permitir que o gestor crie tipos de utilizadores com base nas suas funções específicas, melhorando assim a eficiência da gestão do estabelecimento hoteleiro.

# Bibliografia ou Referências Bibliográficas

Sistema de Gestão Hoteleira | RoomRaccoon :

Create Presentations, Infographics, Design & Video | Visme : <https://www.visme.co/?vc=Top-Bar-CTA>

Figma – Protótipo : <https://www.figma.com/>

Mobile Operating System Market Share Portugal | Statcounter Global Stats : <https://gs.statcounter.com/os-market-share/mobile/portugal/#monthly-202202-202302-bar>

Icons : <https://www.iconfinder.com/>

Underline Design And Simple Design Hand Drawn Elements - Fancy Underline Clipart, HD Png Download , Transparent Png Image – PNGitem : <https://www.pngitem.com/middle/JxiooJ_underline-design-and-simple-design-hand-drawn-elements/>

# Anexos

## Anexo I - Planeamento Geral

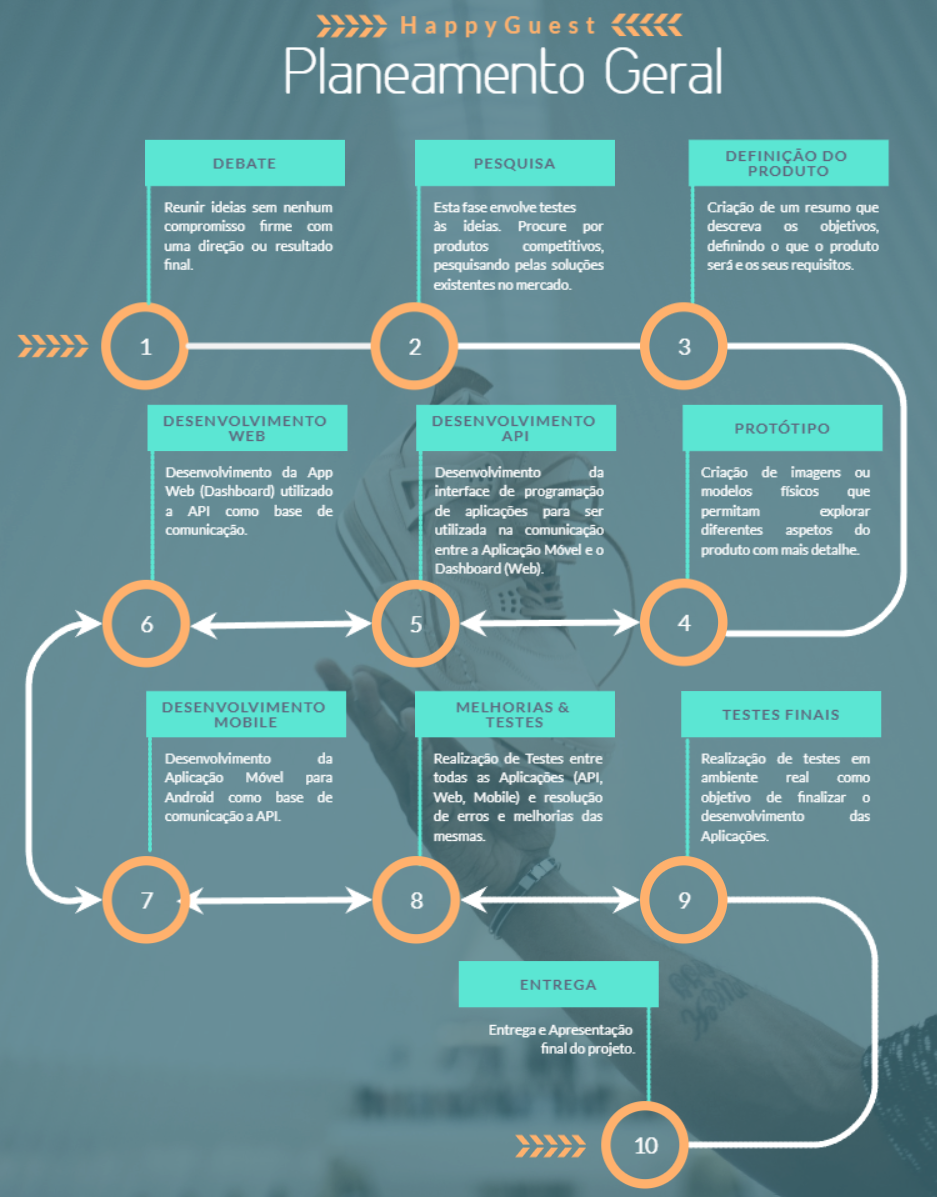


Figura - Planeamento geral do projeto

## Anexo II - Diagrama UML

Uma imagem com captura de ecrã, escuridão, preto

Descrição gerada automaticamente

Figura - Diagrama UML