



UNIVERSIDADE
LUSÓFONA

Sistema de Gestão de uma Escola de Atividades Náuticas

Trabalho Final de curso

Relatório Final 2º Semestre

Nome do Aluno: João Martins Nº 21701951

Nome do Aluno: Rui Prata Nº 21703781

DEISI173

Orientador: Dr. Pedro Arroz Serra

Trabalho Final de Curso | LEI | 27/11/2020

www.lusofona.pt

Direitos de cópia

(Sistema de Gestão de uma Escola de Atividades Náuticas), Copyright de (João Pedro Afonso Martins, Rui Carlos Serafim Prata), ULHT.

A Escola de Comunicação, Arquitetura, Artes e Tecnologias da Informação (ECATI) e a Universidade Lusófona de Humanidades e Tecnologias (ULHT) têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Resumo

A “Wakevilla LDA”, uma empresa de atividades náuticas, solicitou que fosse desenvolvido uma WebApp que correspondesse às solicitações diárias por parte dos clientes, de uma forma mais rápida, prática e funcional. A empresa identificou algumas necessidades que podem ser automatizadas através do desenvolvimento de uma aplicação web.

O principal objetivo da solução proposta é fazer a gestão dos serviços prestados pela empresa, tais como: passeios de barco, paddle board, aulas de wakeboard, entre outros.

Neste trabalho propõe-se a continuação de uma WebApp já existente com algumas funcionalidades implementadas, tais como, a possibilidade das pessoas se registarem na empresa, visualizando o seu histórico de atividades, tal como o horário em que estas se encontram disponíveis.

Desta forma, a empresa conseguirá oferecer um maior leque de ofertas aos clientes, responder às suas expectativas, havendo maior proximidade, rapidez e obtenção de informação, sendo mais vantajoso e lucrativo para a empresa, conseguindo uma maior adesão por parte do seu público-alvo.

Abstract

“Wakevilla LDA”, it’s a nautical company, that provides water activities to their customers, the company has identified some weaknesses that can be improved by developing a web application that will respond to customers' daily requests, in a faster, more practical and functional way.

The main objective of the proposed solution is to manage the services provided by the company, such as: boat trips, paddle board, wakeboard classes, among others.

In this work, we propose the continuation of an existing WebApp with some features implemented, such as the possibility for people to register with the company, viewing their activity history, and the time when they are available.

In this way, the company will be able to obtain a greater range of offers to customers, respond to their expectations, with greater proximity, speed and obtaining information, being more advantageous and profitable for the company, thus achieving greater adhesion on the part of society.

Índice

Resumo.....	3
Abstract	4
Índice.....	5
Lista de Figuras	6
Lista de Tabelas.....	7
1 Identificação do Problema.....	8
2 Levantamento de Requisitos.....	9
3 Viabilidade e Pertinência.....	14
4 Solução Proposta	15
4.1. Python.....	15
4.1.1 Django.....	16
4.2. SQLite.....	17
4.3. JavaScript	17
5 Benchmarking.....	18
6 Proposta de roadmap para deployment	20
7 Booking	21
8 API Interna	23
9 Calendário	25
Bibliografia.....	26
Anexo 1 – Questionário.....	27
Glossário.....	28

Lista de Figuras

Figura 1- Arquitetura Django	16
Figura 2- Lista dos quartos disponiveis para reservar	21
Figura 3- Reserva de um quarto.....	21
Figura 4- Calendário Booking	22
Figura 5- Ecrã com informação completa e opção para cancelar reserva	22
Figura 6- Arquitetura Django + API.....	23

Lista de Tabelas

Tabela 1- Levantamento de requisitos	10
Tabela 2- Test Case	13
Tabela 3- Comparação entre soluções	18
Tabela 4- Roadmap para deployment	20
Tabela 5- Calendário.....	25

1 Identificação do Problema

“Wakevilla LDA” é uma empresa de atividades náuticas, que se situa no distrito de Santarém, Vale da Menina, Serra, Tomar. Esta pequena empresa foi criada por Nuno Eça, em 2009. Oferece aos seus clientes experiências como: sessões de *wakeboard*, *wakesurf*, *ski*, *waterfoil*, boia, passeios de barco, *padle*, *kayak* e alojamento, um dos principais focos desta empresa.

Este projeto vem colmatar uma necessidade urgente da Wakevilla. Na reunião inicial o responsável da empresa explicou que o método atualmente utilizado não permite uma gestão eficiente do negócio. É pouco eficiente, o que impossibilita a análise dos resultados da empresa. Para poder crescer, são necessárias ferramentas que permitam, não só fazer a gestão eficiente do negócio da empresa, mas também tirar conclusões sobre os resultados obtidos através da geração de relatórios.

Um dos objetivos da empresa é através do próprio site, conseguir reservar atividades náuticas, gerir os seus recursos e fazer uma gestão do seu alojamento. Criando assim uma alternativa à atual forma de reservar e pedir informações, por via telefónica.

O trabalho atual consiste na continuação do desenvolvimento da WebApp já criada (Tootsie). Esta continuação tem como objetivo simplificar e automatizar a gestão de alojamento e realizar a sincronização com alguns sites de alojamento utilizados manualmente pelo proprietário da empresa, poupando tempo e trabalho ao mesm

Este projeto assenta sobre o Django, uma framework web escrita em Python. Esta framework resolve muitos problemas em relação à segurança. Possui módulos para autenticação e registo de utilizadores, recorrendo a tecnologias atuais e com *updates* constantes.

2 Levantamento de Requisitos

Neste segmento estão indicados todos os requisitos a serem implementados e os que já foram feitos. Primeiro é indicado o requisito, posteriormente é feita a sua descrição e por fim o estado do mesmo: realizado, em progresso e não realizado. É importante salientar que alguns pontos referidos são requisitos realizados no trabalho final de curso do ano passado, sendo que os restantes pontos estão a ser desenvolvidos, ou já foram concluídos, este ano (consultar última coluna). À exceção do último ponto, sendo este uma sugestão da continuação para este projeto no futuro.

1.1 Gestão de Recursos

Pretende-se que a aplicação permita registar membros do staff e clientes novos, barcos, material para alugar, quartos disponíveis etc.

1.2 Registo de Serviços Prestados

Pretende-se registar os serviços prestados, como: atividades náuticas – *wakeboard*, *ski*, boia, etc. (cada um com o seu preço), aluguer de barcos ou material, aluguer de quartos.

Cada sessão é efetuada com um instrutor e um driver, sendo que o driver pode ser um instrutor também. Cada registo de sessão deverá guardar os dados da atividade, a duração, o nome do instrutor, o condutor e o cálculo do preço.

1.3 Gestão de Alojamento

Pretende-se criar um calendário com ocupações por quarto para que seja possível sincronizar com o booking utilizando uma API.

Tabela 1- Levantamento de requisitos

<u>Tarefa</u>	<u>Descrição</u>	<u>Realizado?</u> (Sim/Em progresso/Não)
Login	Deverá ser disponibilizado ao administrador um ecrã com as opções para a autenticação através de <i>username</i> e <i>password</i>	Realizado TFC anterior
Criação de Perfil (1.1)	<p>Deverá ser possível ao administrador criar perfis, de staff ou de cliente. O perfil staff tem a opção de o mesmo ser instrutor ou um <i>Driver</i>, tendo a opção de ser os dois ao mesmo tempo. Enquanto o perfil cliente tem o dinheiro que este irá depositar na sua conta e a lista com todas as sessões que o cliente irá realizar ou já realizou, tanto como o seu pagamento.</p> <p>Ambos os perfis possuem informação pessoal relativa a cada individuo. Esta informação só pode ser vista pelo staff da empresa, não sendo partilhada nenhuma informação com terceiros.</p>	Realizado TFC anterior

<p>Nova Sessão (1.2)</p>	<p>Deverá ser possível ao administrador adicionar uma nova sessão de atividades náuticas. Esta sessão irá ser associada ao cliente que usufruiu dela, tal como ao staff (Driver ou Instrutor) que a realizou. Esta sessão dá a opção de ser escolhido um barco, a modalidade que o cliente pretende realizar, o seu preço e a duração.</p>	<p>Realizado TFC anterior</p>
<p>Booking (1.3)</p>	<p>Deverá ser possível ao administrador adicionar novos bookings para reservar os quartos existentes. Ao reservar um quarto, num certo intervalo de tempo, o software verifica se é possível reservar o quarto, não sendo possível sobrepor a uma reserva já existente. Depois de realizar o booking, é mostrada a informação de todos os bookings efetuados. A informação de cada booking é mostrada por:</p> <ul style="list-style-type: none"> • Nome do quarto • Nome do cliente que fez a reserva • Data check-in • Data check-out • Número de camas a serem utilizadas • Preço total a pagar <p>Deverá ainda ser possível cancelar bookings.</p>	<p>Sim</p>

<p>Calendário</p>	<p>Deverá existir um calendário onde estarão disponíveis todos os bookings efetuados e também existir uma opção para adicionar novos bookings</p>	<p>Sim</p>
<p>Criação de uma API compatível para uma integração futura com sites de reservas existentes.</p>	<p>Criação de uma API interna, o que facilita a gestão da base de dados em relação aos bookings efetuados. Esta API tem como objetivo ajudar na integração de <i>Third Partys APIs</i>, tais como Booking.com, Airbnb etc.</p>	<p>Sim</p>
<p>Integração da API com terceiros.</p>	<p>Depois de criada a API interna, deverá ser possível integrar com API de terceiros (Ex: Booking.com, Airbnb etc). Esta integração irá gerir os bookings em simultâneo, tanto nos servidores de sites famosos de alojamento como na própria base de dados do tootsie</p>	<p>Sugerido para futuro</p>
<p>Relatório/Estatísticas</p>	<p>O administrador deve conseguir aceder a um relatório que lhe diz se está a gastar mais gasóleo no fim do mês do que no mês anterior, ou ainda a percentagem de alojamento</p>	<p>Sugerido para futuro</p>

1.4 Test Case

Tabela 2- Test Case

Test Case ID	Test Case Descrição	Dependências	Resultado Esperado
1	Login no site por parte do proprietário ou membro do staff.	N/A.	O utilizador consegue aceder ao site e as suas funcionalidades.
2	Criar um cliente novo.	Test Case 1	O utilizador consegue inscrever com sucesso um novo cliente no site e guardar as suas informações.
2.1	Acéder à lista com as informações de todos os clientes.	Test Case 2	O utilizador consegue ver todos as informações ao detalhe de cada cliente, bem como editar ou remover informações.
3	Criar uma nova sessão.	N/A.	O utilizador consegue criar uma nova sessão.
3.1	Acéder à lista com as informações de todas as sessões.	Test Case 3	O utilizador consegue consultar as sessões que já se realizaram, bem como editar o

			remover sessões.
4	Aceder ao calendário	N/A.	O utilizador consulta no calendário os quartos que tem reservados.
5	Aceder ao Booking	N/A.	O cliente deve conseguir reservar um quarto

3 Viabilidade e Pertinência

O negócio e o método de trabalho da Wakevilla, são muito específicos e como tal é muito difícil encontrar no mercado um *software* adequado. As soluções existentes são demasiado gerais e dispendiosas para serem suportadas por uma empresa pequena.

Este trabalho tem uma grande importância pois vai permitir ao dono da empresa automatizar o negócio de modo a conseguir fazer uma melhor gestão da empresa, que permitirá o seu crescimento e analisar os seus lucros e despesas.

Este software implementa todas as funcionalidades essenciais para o cliente desempenhar o negócio da empresa e que necessita no seu dia a dia. Depois de realizadas várias reuniões com o dono da empresa, foi apresentado a ideia dos alunos para a resolução do problema.

4 Solução Proposta

Esta aplicação web onde o administrador e o seu staff podem fazer login e todos os registos de serviços prestados, reserva de quartos. Esta aplicação será desenvolvida utilizando a *framework* Django, que engloba Python, MySQL, JavaScript, HTML e CSS.

Relativamente ao trabalho desenvolvido anteriormente existem algumas melhorias que achamos que podem ser feitas tais como:

- Limpeza e organização de código;
- Melhorar o aspeto visual dos ecrãs;
- Melhorar alguns bugs já encontrados e outros que se possam vir a encontrar.

4.1. Python

A linguagem utilizada neste projeto irá ser *Python*. Esta é uma linguagem de programação dinâmica de uso geral, popular, que permite múltiplos paradigmas.

Python suporta o paradigma de orientação a objetos, funcional e uma das maiores características do mesmo é a permissão de uma leitura fácil do código, usando menos linhas comparando com outras linguagens.

Python possui um leque de características muito particulares, usadas principalmente para processamento de textos, dados científicos e criação de páginas dinâmicas para a *web*.

Uma *framework web* oferece um conjunto de componentes que ajuda a desenvolver aplicações *web* de forma mais rápida, fácil e segura.

Este tipo de framework serve para evitar que quando se está a construir uma aplicação nova não seja necessário reinventar a “roda” de novo e, com isto, ajudar a aliviar uma parte do trabalho extra.

4.1.1 Django

Django é uma framework segue o padrão de arquitetura model-view-template. É uma *framework* que tem certas características considerada principais, sendo estas:

- ✦ Mapeamento Objeto-Relacional (ORM), com isto é possível definir a modelagem de dados através de classes em *Python*, ou seja, é possível gerar as tabelas da base de dados e manipulá-las sem recorrer ao SQL.
- ✦ *Interface Admin*, com isto é possível gerar automaticamente uma interface para a administração.
- ✦ Sistema de Templates, o *Django* tem uma linguagem de templates poderosa, extensível e amigável, com ela podemos separar conteúdo, design e o código em *Python*.
- ✦ Sistema de Cache, o *Django* possui um sistema de cache que se integra ao *memcached* ou em outros *frameworks* de cache.

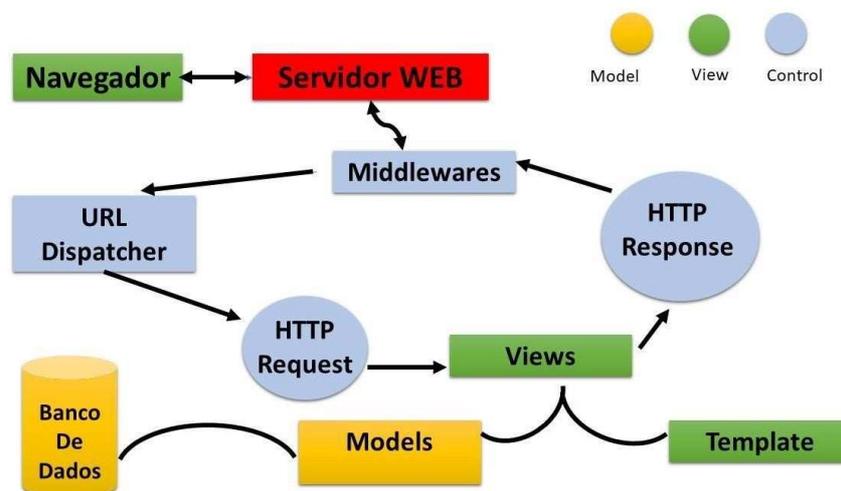


Figura 1- Arquitetura Django

4.2. SQLite

SQLite é o sistema, default de django, gerenciador de base de dados, de *open source* usado na maioria das aplicações gratuitas para gerir as suas bases de dados.

O serviço utiliza a linguagem SQL (*Structured Query Language*), sendo esta a mais popular para inserir, aceder e gerir o conteúdo armazenado numa base de dados.

4.3. JavaScript

O *JavaScript* é uma linguagem de programação do lado do cliente, ou seja, é processada pelo navegador que está a correr na máquina do cliente. Esta linguagem permite a utilização de múltiplos paradigmas conhecidos como POO, Funcional, Imperativo, entre outras.

Com o *JavaScript*, podemos criar efeitos especiais para nossas páginas na *Web*, além de podermos proporcionar uma maior interatividade com nossos utilizadores.

O *JavaScript* é uma linguagem orientada a objetos, ou seja, ela trata todos os elementos da página como objetos distintos, facilitando a tarefa da programação.

Esta linguagem é uma poderosa linguagem, quando os programadores *Web* desejam criar páginas *Web* dinâmicas e interativas, devendo assim ser uma preocupação, destes mesmos, a dominar.

5 Benchmarking

Neste capítulo é feita uma avaliação de benchmarking, onde comparamos a nossa solução (Tootsie) com outras existentes no mercado, tais como a SagePt, MSEXcel (versão utilizada anteriormente pelo proprietário) e InVoiceExpress.

Tabela 3- Comparação entre soluções

	Tootsie	<u>SagePt</u>	<u>MsExcel</u>	<u>InVoiceExpress</u>	<u>CodiSurf</u>
Faturação Eletrónica	Sim	Sim	Não	Sim	Sim
Grátis	Sim	Não	Sim	Não	Não
Gestão de Stock	Sim	Sim	Sim	Não	Sim
Envio de Documentos por Email	Sim a implementar	Não	Não	Sim	Não
Base Dados	Sim		Sim	Sim	Sim
Backups Automáticos	Não	Sim	Sim	Sim	Sim
Complexidade do Software	Intuitivo	Intuitivo	Complexo	Intuitivo	Intuitivo
Pesquisa	Sim	Sim	Não	Sim	Sim
Suporte Técnico	Não	Sim	Não	Sim	Sim
Reserva de Atividades	Sim	Não	Não	Não	Sim
Alojamento	Sim	Não	Sim	Não	Não
Gestão de Equipamentos	Sim	Não	Não	Não	Sim

Em comparação com outras ferramentas de gestão de negócio o Tootsie (nome do software desenvolvido), é a melhor opção para a WakeVilla por ser uma aplicação desenvolvida apenas para satisfazer as necessidades desta empresa. Assim não é necessário de gastar dinheiro em funcionalidades extras que não irão ser utilizadas.

No que diz respeito à solução Codisurf esta apresenta uma grande semelhança com a nossa, é uma solução que ajuda na gestão de uma escolha de surf, que faz também gestão de equipamentos, financeira e relatório de vendas. Esta solução difere da nossa pois é uma subscrição paga e não abrange uma área importantíssima do negócio da Wakevilla que é o alojamento.

Podemos concluir que a oferta mais comum existente é focada em programas de faturação predefinidos para uso geral, dessa forma diferenciamos-nos do mercado existente. A aplicação que mais se assemelha com a nossa, em termos de funcionalidades, é a CodiSurf, no entanto tem um custo de subscrição, ao contrário do nosso serviço que é totalmente gratuito e não contempla uma área essencial do negócio.

6 Proposta de roadmap para deployment

A tabela seguinte mostra a proposta de *roadmap* para o *deployment* da WebApp que estamos a desenvolver.

Tabela 4- Roadmap para deployment

<u>Descrição</u>	<u>Data</u>
Início do projeto	29/10/2020
Construção do código	15/11/2020 – 21/04/2021
<i>Test Case</i> para validação da solução	23/04/2021
Implementação e correção de falhas	1/05/2021 – 20/06/2021
<i>Release</i> do projeto	25/06/2021

Começámos a construir a WebApp em setembro de 2020 e desenvolvemos até estar capaz de ser testada pelo dono da empresa através do link: <http://tootsie.pythonanywhere.com/>

Uma vez testada será nos dado um feedback onde saberemos o que está bem feito e aquilo que temos que melhorar para que no fim do prazo a solução final seja aquela que o proprietário pretende.

7 Booking

Uma das principais funcionalidades do site e capacidade para fazer o *booking* dos quartos.

Quando entra na secção do *booking* escolhe um dos cinco quartos.

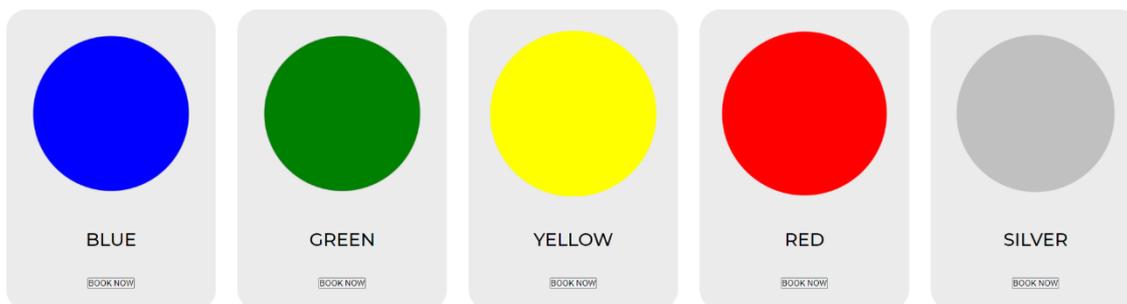


Figura 2- Lista dos quartos disponíveis para reservar.

Depois de escolhido o quarto a semelhança do que acontece noutros sites de reserva são preenchidas informações acerca da data de *check in* e *check out* bem como o nome do cliente.

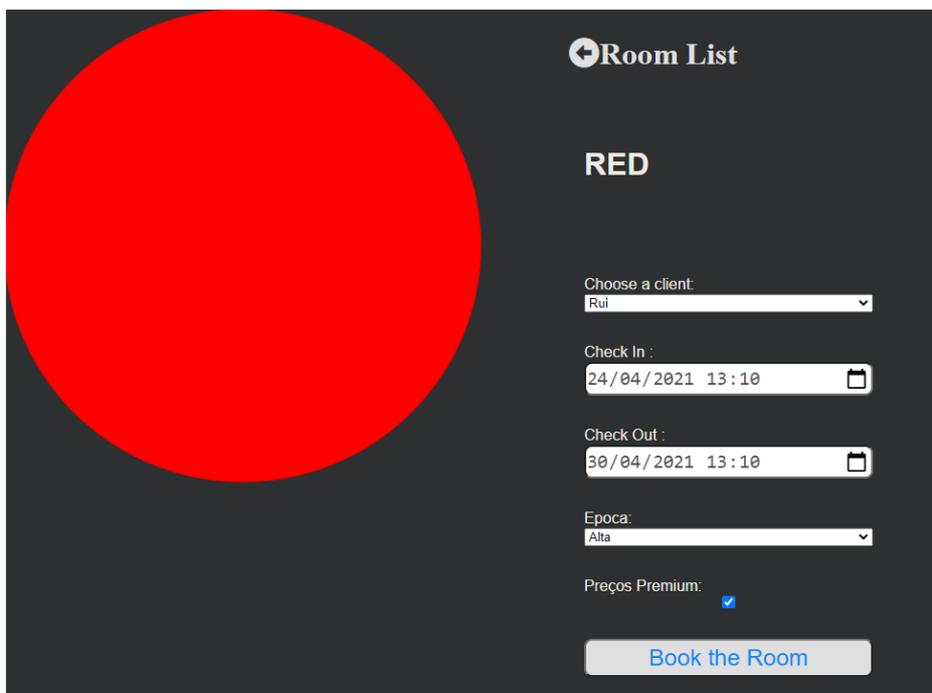
A screenshot of a web form titled 'Room List' for booking a 'RED' room. On the left is a large red circle. On the right, the form includes: a dropdown menu for 'Choose a client' with 'Rui' selected; 'Check In' and 'Check Out' date and time pickers set to '24/04/2021 13:10' and '30/04/2021 13:10' respectively; a dropdown menu for 'Epoca' with 'Alta' selected; a checked checkbox for 'Preços Premium'; and a 'Book the Room' button.

Figura 3- Reserva de um quarto

Uma vez finalizada a reserva é possível consultar o calendário completo com todas as reservas feitas.

April 2021					
Wed	Thu	Fri	Sat	Sun	
	1	2	3	4	
7	8	9	10	11	
14 ID: 97 Cliente_name booked GREEN 11:00 60 €	15 ID: 98 Cliente_name booked BLUE 18:15 175 €	16	17	18 ID: 99 Rui booked BLUE 12:09 80 €	
21 ID: 101 Rui booked BLUE 11:21 255 € ID: 102 Rui booked YELLOW 11:27 160 €	22	23	24 ID: 104 Rui booked RED 12:10 400 €	25	
28	29	30			

Figura 4- Calendário Booking

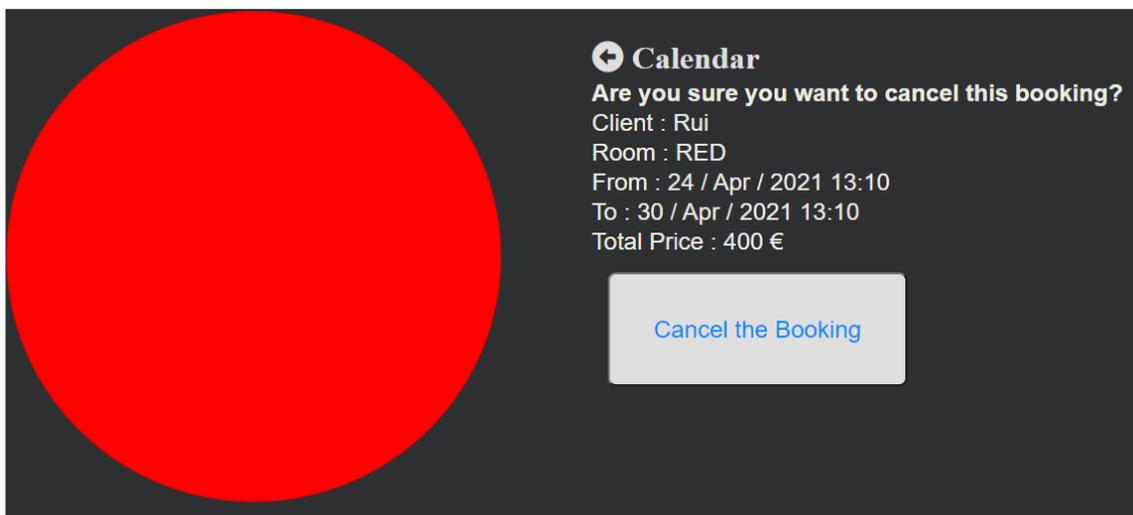


Figura 5- Ecrã com informação completa e opção para cancelar reserva

8 API Interna

O objetivo principal da criação desta API, é simplificar a integração com outras bookings APIs. Para criarmos esta API, utilizámos uma *framework* disponível pela tecnologia utilizada no desenvolvimento do projeto. *Django Rest Framework* permite criar *Restful APIs*, o que permite fazer uma ponte entre para transferir informação entre a base de dados e o servidor de terceiros. Esta *framework* vai transformar a informação de cada booking guardado, em formato JSON. Tem ainda acesso CRUD à base de dados, podendo alterar um booking, adicionar um novo booking, ver informação específica sobre um booking e apagar um booking. Para este processo ser seguro, apenas utilizadores credenciados podem aceder a informação da API

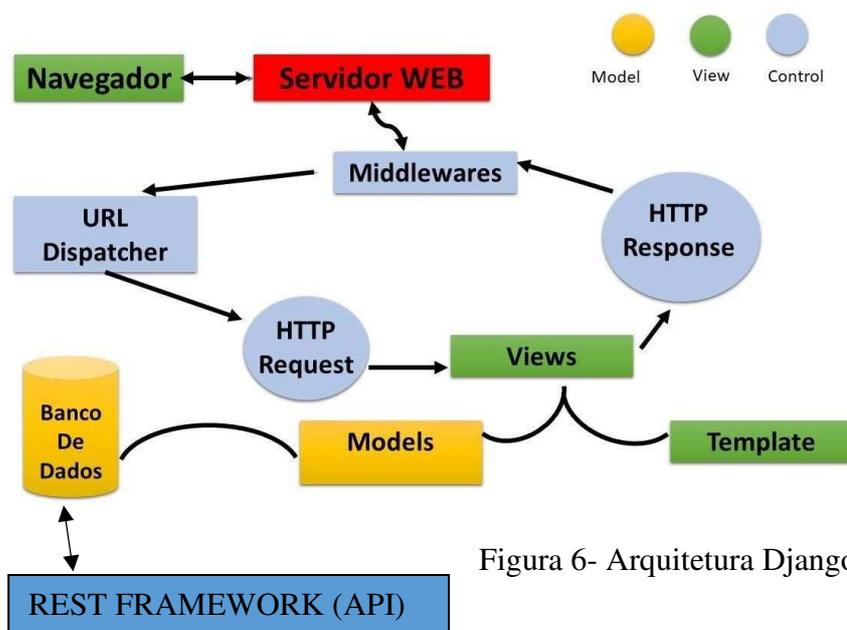


Figura 6- Arquitetura Django + API

Para facilitar a ligação entre a API interna e outras APIs, optámos por dar uma informação o mais geral possível sobre o booking.

Isto é:

- Nome do cliente que reservou o quarto;
- Qual o quarto reservado;
- Data do Check-in e data do Check-out;
- Preço total;
- Se a reserva é feita em época alta ou época baixa (influencia o preço total);
- Se o preço é premium ou não (influencia o preço total);

Com uma breve pesquisa, concluímos que a API a que pretendíamos conectar é paga (booking.com). Para minimizar custos, preferimos não avançar com esta conexão enquanto o software que se encontra a ser desenvolvido não for concluído.

9 Calendário

Neste capítulo é descrito o calendário da solução proposta, o intervalo de tempo do início e fim de cada tarefa ao longo do tempo.

Tabela 5- Calendário

<u>Tarefa</u>	<u>Descrição</u>	<u>Data</u>
T.1	Início do projeto	29/10/2020
T.1.1	Leitura sobre a empresa	2/10/2020
T.1.2	Leitura sobre <i>Django</i>	4/10/2020
T.1.3	Revisão de leitura	5/10/2020
T.1.4	Realização do relatório intercalar	24/10/2020
T.1.5	Entrega de relatórios intercalar	27/11/2020
T.2	Levantamento de Requisitos	7//12/2020
T.2.1	Análise e Compreensão de Código	14/12/2020
T.2.2	<i>Deploy</i> da WebApp	4/01/2021
T.2.3	Limpeza e Organização de código	13/10/2021
T.2.4	Realização do Relatório Intermédio	19/10/2021
T.3	Entrega do Relatório Intermédio	22/01/2021
T.3.1	Implementação <i>Booking</i>	21/02/2021
T.3.2	Criação da API	4/03/2021
T.3.3	Booking formato calendário	25/03/2021
T.3.4	Realização do Relatório Intercalar	19/04/2021
T.3.5	Entrega do Relatório Intercalar	23/04/2021
T.4	Testes de aceitação	3/06/2021
T.4.1	Melhorias na WebApp	15/06/2021
T.4.2	Realização Relatório Final	23/06/2021
T.5	Entrega Final	25/06/2021

Bibliografia

- [1] DEISI, Regulamento de Trabalho Final de Curso, setembro 2020.
- [2] Udemy Online Courses, (Python and Django Full Stack Developer), novembro 2020.
- [3] Django, Site Oficial <https://www.djangoproject.com/>, dezembro 2020.
- [4] Coding Entrepreneurs
https://www.youtube.com/watch?v=oQvMHPkKms&ab_channel=CodingEntrepreneurs,
novembro 2020.
- [5] Bootstrap <https://getbootstrap.com/>, janeiro 2021.
- [6] Django Rest Framework <https://www.django-rest-framework.org/>, março 2021.

Anexo 1 – Questionário

Glossário

LEI	Licenciatura em Engenharia Informática
LIG	Licenciatura em Informática de Gestão
TFC	Trabalho Final de Curso