



UNIVERSIDADE  
LUSÓFONA

# Plataforma para Gestão de Agrupamento Musical

## Trabalho Final de Curso

Relatório Final

Gonçalo Matos a21600767

André Rego a21604285

Orientador: Lúcio Studer Ferreira

Trabalho Final de Curso | LIG | 10-09-2021

[www.ulusofona.pt](http://www.ulusofona.pt)

## **Direitos de cópia**

*(Plataforma para gestão de agrupamento musical)*, Copyright de *(Gonçalo Matos, André Rego)*, 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

Desde sempre, a música foi algo muito presente na vida do Homem, é vista, não só, como uma forma de arte, mas também como uma modalidade capaz de promover o desenvolvimento e o equilíbrio da mente humana, bem como o bem-estar e a concentração, sendo estes essenciais para resolver questões ligadas ao pensamento lógico em geral, estando o mesmo tão presente âmbito da nossa área de estudo. O grupo Quarteto Arabesco, do qual faz parte o nosso orientador, o professor Lúcio Ferreira, é um grupo musical de música clássica composto por quatro membros.

Este trabalho tem como objetivo desenvolver uma plataforma de gestão de grupos musicais que permita criar uma relação de maior proximidade entre os fãs de música em geral e os seus grupos musicais favoritos. Esta plataforma, desenvolvida em Django, deverá ser capaz de fazer a gestão de conteúdos do grupo, tais como a organização da lista de eventos, criação de *posts*, de forma automática e nos dias que antecedem os eventos do grupo fazer a ligação com as Redes Sociais de forma a gerar *posts* automáticos. Pretendemos ainda que a mesma seja customizável a cada agrupamento musical, de forma a facilitar a manipulação de conteúdos.

## **Abstract**

Music has always been very present in man's life, it is seen not only as an art form, but also as a modality capable of promoting the development and balance of the human mind, as well as well-being and concentration, being essential to resolve issues related to logical thinking in general, the same has been present in our area of study. The Quarteto Arabesco group, which includes our advisor, professor Lúcio Ferreira, is a musical group of classical music composed of four members.

This work aims to develop a management platform for musical groups that allows creating a closer relationship between music fans in general and their favorite musical groups. This platform should be able to manage the group's content, such as organizing the list of events, creating posts automatically in the days leading up to the group's events, among other events. This platform, developed in django, must be able to manage the group's content, such as organizing the list of events, creating posts, automatically, in the days before the group's events and making the connection with Social Networks, in order to generate automatic posts. It is also intended to be customizable for each musical group, to facilitate the management of content.

# Índice

Resumo.....	iii
Abstract .....	iv
Índice.....	v
Lista de Figuras .....	vii
Lista de Tabelas.....	x
1 Identificação do Problema.....	1
Enquadramento .....	1
Motivação e Objetivo .....	1
2 Levantamento e análise dos Requisitos .....	3
Análise de Requisitos .....	3
Casos de Uso.....	6
Diagramas de Atividades .....	6
Diagramas de Entidade Relação .....	8
Estrutura da Aplicação .....	9
3 Viabilidade e Pertinência .....	12
Análise dos Inquéritos .....	12
Testes de Validação e Questionários de Satisfação .....	12
4 Solução Proposta .....	13
Arquitetura .....	13
Componentes.....	14
Model .....	14
Criação de Conteúdos com a Aplicação Admin .....	15
Views .....	17
Consulta de Dados.....	19
Consulta de Dados por Data .....	23
Criação e transferência de Canvas .....	24
Zonas restritas do site .....	26
Envio automático e calendarizado de posts de eventos ao Facebook.....	28
Envio automático calendarizado de anúncio de Evento para mailing list.....	30
Hosting do site .....	31

Armazenamento na Cloud.....	31
Visualizações da Aplicação Desenvolvida.....	31
Ambientes disponíveis .....	35
5 Benchmarking .....	36
Plataformas.....	36
Aplicações.....	36
6 Método e Planeamento.....	40
7 Resultados .....	42
8 Conclusões e trabalhos futuros .....	47
Conclusão .....	47
Trabalhos Futuros .....	47
Bibliografia .....	48
Anexo A – Questionários de Viabilidade .....	49
A.1 Questionário a Membros de Grupos.....	49
A.2 Questionário a Ouvintes.....	57
Anexo B – Análise das Respostas aos Questionários de Viabilidade .....	61
B.1 Questionário a membros de grupos musicais .....	61
B.2 Questionário aos ouvintes.....	67
Anexo C – Mockups.....	72
Anexo D - Benchmarking de Plataformas.....	76
D.1 React .....	77
D.2 Flask .....	77
D.3 Express .....	78
Anexo E - Questionário de Satisfação .....	79
Anexo F - Respostas ao questionário de satisfação .....	82
Anexo G - Manual de Utilizador.....	86
Anexo H - Guião de Testes .....	94
Glossário.....	98

## Lista de Figuras

Figura 1- Página de agenda de eventos do Quarteto Arabesco .....	2
Figura 2- Página de propostas de concerto do Quarteto Arabesco.....	2
Figura 3-Casos de uso .....	6
<b>Figura 4- Diagrama de atividade de visualização de dados.....</b>	<b>7</b>
<b>Figura 5- Diagrama de atividade de atualização de dados.....</b>	<b>7</b>
<b>Figura 6- Diagrama de atividade de inserção de dados .....</b>	<b>8</b>
<b>Figura 7- Diagrama de atividade de eliminação de dados .....</b>	<b>8</b>
Figura 8- Diagrama de entidade relação.....	10
Figura 9- Estrutura das páginas da aplicação .....	11
Figura 10- Arquitetura da aplicação web com identificação das tecnologias utilizadas. ....	13
Figura 11- Classe projeto e evento do ficheiro models.....	15
Figura 12- Criar um novo projeto ao criar um novo evento.....	16
Figura 13- Classe Imagens dentro da classe Evento .....	17
Figura 14 - Urls.py.....	18
Figura 15 - <i>View</i> Membros.....	18
Figura 16 - Página de membros.....	19
Figura 17- Ficheiro <i>JavaScript</i> responsável pela pesquisa por texto .....	20
Figura 18- Resultados da pesquisa de teste.....	21
Figura 19- <i>View</i> responsável pela pesquisa de dados.....	22
Figura 20- Página eventos.html responsável pela apresentação dos dados.....	23
Figura 21-Exemplo de pesquisa por intervalo de datas .....	23
Figura 22- Resultados da pesquisa por intervalo de datas.....	24
<b>Figura 23- Função de download e de nomeação do <i>Canvas</i> .....</b>	<b>25</b>
Figura 24- Exemplo do nome do <i>Canvas</i> no download.....	25
Figura 25 - Exemplo de <i>Canvas</i> .....	26
Figura 26 - Ficheiro de restrição de permissões.....	27
Figura 27 - <i>View</i> de Login e <i>Canvas</i> com as restrições de acesso.....	28
Figura 28- Função de <i>Post</i> no Facebook .....	29
Figura 29- Heroku Scheduler.....	30
Figura 30- Navigation Bar.....	31
Figura 31-Home.....	32
Figura 32-História .....	32
Figura 33 - História .....	33
Figura 34- Quem somos.....	34
Figura 35 - Pesquisa de Eventos .....	35
Figura 36- Secção da página do Facebook do grupo Metropolitana .....	37
Figura 37- Secção da página do Facebook do grupo Metropolitana .....	38
Figura 38- Secção do web site do grupo Artway .....	38
Figura 39-Aspetos negativos.....	43
Figura 40- Aspetos positivos .....	43
Figura 41- Descrição do questionário a membros dos grupos. ....	49
Figura 42- Pergunta 1 do questionário a membros dos grupos. ....	49
Figura 43- Pergunta 2 do questionário a membros dos grupos. ....	50
Figura 44- Pergunta 3 do questionário a membros dos grupos. ....	50
Figura 45- Pergunta 4 do questionário a membros dos grupos. ....	51
Figura 46- Pergunta 5 do questionário a membros dos grupos. ....	51
Figura 47- Pergunta 6 do questionário a membros dos grupos. ....	52

Figura 48- Pergunta 7 do questionário a membros dos grupos. ....	52
Figura 49- Pergunta 8 do questionário a membros dos grupos. ....	53
Figura 50- Pergunta 9 do questionário a membros dos grupos. ....	53
Figura 51- Pergunta 10 do questionário a membros dos grupos. ....	54
Figura 52- Pergunta 11 do questionário a membros dos grupos. ....	54
Figura 53- Pergunta 12 do questionário a membros dos grupos. ....	55
Figura 54- Pergunta 13 do questionário a membros dos grupos. ....	55
Figura 55- Pergunta 14 do questionário a membros dos grupos. ....	55
Figura 56- Pergunta 15 do questionário a membros dos grupos. ....	56
Figura 57- Pergunta 16 do questionário a membros dos grupos. ....	56
<b>Figura 58- Descrição do questionário a visitantes de websites de grupos musicais. ....</b>	<b>57</b>
<b>Figura 59- Pergunta1 do questionário a visitantes de websites de grupos musicais. ....</b>	<b>57</b>
<b>Figura 60- Pergunta2 do questionário a visitantes de websites de grupos musicais. ....</b>	<b>58</b>
<b>Figura 61- Pergunta3 do questionário a visitantes de websites de grupos musicais. ....</b>	<b>58</b>
<b>Figura 62- Pergunta4 do questionário a visitantes de websites de grupos musicais. ....</b>	<b>58</b>
<b>Figura 63- Pergunta5 do questionário a visitantes de websites de grupos musicais. ....</b>	<b>59</b>
<b>Figura 64- Pergunta6 do questionário a visitantes de websites de grupos musicais. ....</b>	<b>59</b>
<b>Figura 65- Pergunta7 do questionário a visitantes de websites de grupos musicais. ....</b>	<b>60</b>
Figura 66- Resposta à questão 1 do questionário dos membros de grupos musicais .....	61
Figura 67- Resposta à questão 2 do questionário dos membros de grupos musicais .....	61
Figura 68- Resposta à questão 3 do questionário dos membros de grupos musicais .....	62
Figura 69- Resposta à questão 4 do questionário dos membros de grupos musicais .....	62
Figura 70- Resposta à questão 5 do questionário dos membros de grupos musicais .....	63
Figura 71- Resposta à questão 6 do questionário dos membros de grupos musicais .....	63
Figura 72- Resposta à questão 7 do questionário dos membros de grupos musicais .....	64
Figura 73- Resposta à questão 8 do questionário dos membros de grupos musicais .....	64
Figura 74- Resposta à questão 9 do questionário dos membros de grupos musicais .....	65
Figura 75- Resposta à questão 10 do questionário dos membros de grupos musicais .....	65
Figura 76- Resposta à questão 11 do questionário dos membros de grupos musicais .....	66
Figura 77- Resposta à questão 12 do questionário dos membros de grupos musicais .....	66
Figura 78- Resposta à questão 13 do questionário dos membros de grupos musicais .....	67
Figura 79- Resposta à questão 14 do questionário dos membros de grupos musicais .....	67
Figura 80- Resposta à questão 1 do questionário dos ouvintes .....	68
Figura 81- Resposta à questão 2 do questionário dos ouvintes .....	68
Figura 82- Resposta à questão 3 do questionário dos ouvintes .....	69
Figura 83- Resposta à questão 4 do questionário dos ouvintes .....	69
Figura 84- Resposta à questão 5 do questionário dos ouvintes .....	70
Figura 85- Resposta à questão 6 do questionário dos ouvintes .....	70
Figura 86- Resposta à questão 7 do questionário dos ouvintes .....	71
Figura 87- Página inicial .....	72
Figura 88- Página inicial na área de registo.....	73
Figura 89- Página de registo.....	74
Figura 90- Página secundária .....	75
Figura 91- Gráfico de popularidade de Web frameworks.....	76
Figura 92-Descrição do questionário de satisfação .....	79
Figura 93- Pergunta 1 do questionário de satisfação .....	79
Figura 94- Pergunta 2 do questionário de satisfação .....	79
Figura 95- Pergunta 3 do questionário de satisfação .....	80
Figura 96- Pergunta 4 do questionário de satisfação .....	80
Figura 97- Pergunta 5 do questionário de satisfação .....	80

Figura 98-Pergunta 6 do questionário de satisfação .....	81
Figura 99- Pergunta 7 do questionário de satisfação .....	81
Figura 100- Resposta à pergunta 1 do questionário de satisfação .....	82
Figura 101- Resposta à pergunta 2 do questionário de satisfação .....	82
Figura 102- Resposta à pergunta 3 do questionário de satisfação .....	83
Figura 103- Resposta à pergunta 4 do questionário de satisfação .....	83
Figura 104- Resposta à pergunta 5 do questionário de satisfação .....	84
Figura 105- Resposta à pergunta 6 do questionário de satisfação .....	84
Figura 106- Resposta à pergunta 7 do questionário de satisfação .....	85
Figura 107 - Caminho para a pasta .....	86
Figura 108- Menu cmd.....	86
Figura 109 - Instalação de Pipenv shell .....	87
Figura 110 - Instalação do ficheiro requirements.txt .....	88
Figura 111 - Aplicação a correr localmente.....	89
Figura 112 - Página de login .....	89
Figura 113 - Criação de um user com premissões de admin.....	90
Figura 114 - Login na página de admin .....	91
Figura 115 - Menu da página admin .....	91
Figura 116 - Staff status .....	92
Figura 117 - Evento.....	93

## **Lista de Tabelas**

Tabela 1 - Requisitos funcionais .....	3
Tabela 2 - Requisitos Não Funcionais .....	5
Tabela 3- Avaliação das diferentes frameworks.....	36
Tabela 4 - Cronograma das atividades realizadas (verde) e por realizar (laranja).....	41
Tabela 5- Resultados dos testes de usabilidade.....	44

# 1 Identificação do Problema

## Enquadramento

Atualmente, as redes sociais têm um papel importante na vida de grande parte de cada um de nós, passamos muito do nosso tempo em interação com as mesmas, como tal, é fundamental manter uma presença assídua e apelativa para qualquer grupo musical. No entanto, grande parte dos grupos musicais não têm capacidade de ter esta função gestão de conteúdos online atribuída a alguém dedicado exclusivamente a estas tarefas, como tal, é essencial que existam mecanismos automatizados que permitam realizar estas funções de forma mais fácil e eficiente. As redes sociais têm também uma grande relevância quando falamos dos agentes organizadores de eventos, uma vez que estes utilizam as redes sociais como uma plataforma de observação de conteúdos de grupos musicais por forma a recrutá-los para os seus eventos

Um dos grandes problemas encontrados pelos membros de grupos musicais reside no tempo despendido na gestão dos conteúdos presentes nas redes sociais e ou no web site. É necessário organizar toda a informação relativa a eventos, desde a divulgação, até ao pós evento, que pode conter excertos do concerto em formato áudio ou vídeo, notícias e outros artigos relacionados com o evento.

Este trabalho tem como objetivo desenvolver uma plataforma de gestão de agrupamentos musicais escalável a qualquer agrupamento musical que permita criar uma relação de maior proximidade entre os fãs de música em geral e os seus grupos musicais favoritos. Esta plataforma deverá ser capaz de fazer a gestão de conteúdos do grupo, tais como a organização da lista de eventos, criação de postagens de forma automática nos dias que antecedem os eventos do grupo entre outras funcionalidades, tudo isto interligado com as redes sociais, fazendo com que a plataforma de gestão e as redes sociais funcionem como um só, na tentativa de manter a notoriedade do grupo musical.

## Motivação e Objetivo

Como caso de estudo, é tomada como referência a plataforma do Quarteto Arabesco [Arab21]. Esta plataforma estrutura-se da seguinte forma. É uma plataforma, com um design simples, estática, sem base de dados, difícil de gerir e que, atualmente, não apresenta nenhuma componente de *back-end*. Na Figura 1 apresenta-se a página de agenda de eventos do Quarteto. Nela conseguimos encontrar toda a informação relativa a concertos. De referir, que no seu estado atual, a plataforma encontra-se bastante desatualizada. Devido ao facto da mesma ter sido desenvolvida há alguns anos atrás faz com que as tecnologias da altura sejam muito inferiores às disponíveis atualmente, o que torna a plataforma difícil de gerir e como tal pode beneficiar de um tratamento de dados em *back-end*, para que seja possível manipular informação e gerir conteúdos de uma forma mais fácil e produtiva.



Figura 1- Página de agenda de eventos do Quarteto Arabesco



Figura 2- Página de propostas de concerto do Quarteto Arabesco

## 2 Levantamento e análise dos Requisitos

### Análise de Requisitos

A análise de requisitos consiste no levantamento de todas as funcionalidades que o sistema deve ser capaz proporcionar, tanto do ponto de vista do utilizador, como do gestor do sistema.

Na Tabela 1 apresentam-se os requisitos funcionais agrupados em diferentes categorias. Alguns dos requisitos foram deprecados pelo facto de, depois de varias reuniões para discussão sobre que requisitos seriam mais importantes, chegamos à conclusão que alguns deles não seriam tão relevantes como inicialmente pensamos.

Tabela 1 - Requisitos funcionais

	#	Descrição	Estado
Autenticação	RF1	A plataforma possui um sistema de registo e autenticação através de email password	Implementado
	RF2	A plataforma deve ter uma base de dados que guarda os dados dos utilizadores após o seu registo	Implementado
	RF3	A plataforma permite que utilizadores (administradores) autenticados editem conteúdos (inserir, editar, remover). Por exemplo, deve permitir fazer upload de uma fotografia, e associar a um determinado evento.	Implementado
	RF4	A plataforma deve ter uma base de dados que guarda conteúdos vários	Implementado
	RF5	A plataforma deve ser capaz de congrega conteúdos para a newsletter, com base em eventos passados e futuros	Implementado
	RF6	A plataforma deve permitir que um utilizador se registe para receber newsletters e noticias	Implementado
Usabilidade	RF7	A plataforma deve ser capaz de enviar por email uma newsletter aos utilizadores que já se registaram na plataforma sobre noticias e eventos do grupo (cria uma página web, e envia uma imagem da página com um link)	Implementado
	RF8	A plataforma deve ter uma secção/página que permita inserir um comentário ou fazer uma encomenda / pedido de serviço / uma proposta/ pedido de um concerto, sugestões. Deve também permitir anexar ficheiros, fotos, vídeos. O gestor depois pode decidir o que fazer com estes, se os descarta, ou armazena na base de dados, associado a determinado evento. O gestor da plataforma deverá ser notificado quando existir um novo comentário.	Deprecado

	#	Descrição	Estado
	RF9	A plataforma deve permitir a um utilizador registado marcar determinados comentários para ficarem visíveis numa secção	Deprecado
	RF10	A plataforma deve ter uma página que faz <i>webscrapping</i> de redes sociais, pesquisando conteúdos que possam interessar integrar na plataforma. Deverá apresentar os conteúdos disponíveis em publicações no Facebook do agrupamento, permitindo ao gestor decidir integrar na plataforma alguns destes conteúdos/fotos que existam lá, podendo decidir onde colocar. Também deve ser capaz de identificar onde o agrupamento foi identificado ( <i>hashtags</i> ).	Não Implementado
	RF11	Se um utilizador inserir conteúdos relativos a um evento, a plataforma deverá automaticamente notificar as redes sociais destes.	Deprecado
	RF12	A plataforma permite calendarizar a publicação automática, nas redes sociais, de informações sobre eventos ou notícias (dias antes de um evento acontecer, assim como depois).	Implementado
	RF13	A plataforma deve ser capaz de gerar uma página que congrega todos os conteúdos relacionados com um programa (programa, agrupamento, notas, locais onde foi apresentado, fotos, vídeos)	Implementado
	RF14	A plataforma deve ser capaz de gerar uma página que congrega todos os conteúdos relacionados com um evento (local, data, hora, programa, agrupamento, link para comercializador de venda de bilhetes, notas, comentários, fotos, vídeos)	Implementado
Requisitos de Conteúdo	RF15	A plataforma deve ter uma página de notícias que apresenta conteúdos que foram recentemente adicionados ou gerados	Deprecado
	RF16	A plataforma deve apresentar informação visual do número visitantes da plataforma.	Deprecado
	RF17	A plataforma deve ter um calendário atualizado de eventos futuros e passados, com links para páginas com mais detalhe sobre cada um destes (caso existam).	Implementado
	RF18	A plataforma deve ter uma secção com loja online onde é possível comprar produtos.	Deprecado
	RF19	A plataforma deve ter uma secção com entrevistas aos membros dos grupos e ou fãs	Deprecado
	RF20	A plataforma deve ter uma biografia/Historia sobre o grupo/membros	Implementado

#	Descrição	Estado
RF21	A plataforma deve ter uma secção que mostra comentários escolhidos	Deprecado
RF22	A plataforma deve permitir que novos talentos possam postar/mostrar os seus projetos, conteúdos. Seriam revistos pelos gestores da plataforma, e podem ser integrados numa secção de "novos talentos / outros projetos"	Deprecado
RF23	A plataforma deve ter uma secção com áudio e vídeo do agrupamento que pode ser visionado.	Deprecado
RF24	A plataforma deve ter um sistema de pesquisa de eventos e projetos através de palavras escritas pelo utilizador.	Implementado
RF25	A plataforma deve permitir ouvir música enquanto se visita.	Deprecado
RF26	A plataforma deve ter um sistema de pesquisa através das datas dos eventos.	Implementado
RF27	A plataforma deve possuir um sistema que permita gerar conteúdos publicáveis no Facebook e mailing list.	Implementado

Em baixo temos a Tabela 2 com os requisitos não-funcionais:

Tabela 2 - Requisitos Não Funcionais

#	Descrição	Estado
RNF1	A plataforma deve ser fácil de utilizar e intuitiva para os utilizadores	Atingido
RNF2	O utilizador deve ser capaz de atingir os seus objetivos na página de forma rápida.	Atingido
RNF3	A plataforma deve ter a opção de tradução automática para vários idiomas	Não Atingido
RNF4	A plataforma deve ser fácil de manter em online	Atingido
RNF5	A plataforma deve ser fácil alterar, retirar e acrescentar funcionalidades.	Atingido
RNF6	A plataforma deve ter um tempo de restabelecimento curto, no caso da ocorrência de falhas no sistema	Atingido
RNF7	A probabilidade de falhas no sistema deve ser baixa	Atingido
RNF8	O sistema deve ter mecanismos de segurança tal como o protocolo <i>HTTPS</i> e o sistema de <i>recaptcha</i> no login.	Parcialmente atingido

Casos de Uso

Os diagramas de casos de uso são responsáveis pela descrição do que o sistema faz do ponto de vista do utilizador. Na Figura 3 estão apresentados os diferentes casos de uso identificados. Os respetivos diagramas de atividades são descritos nas secções seguintes.

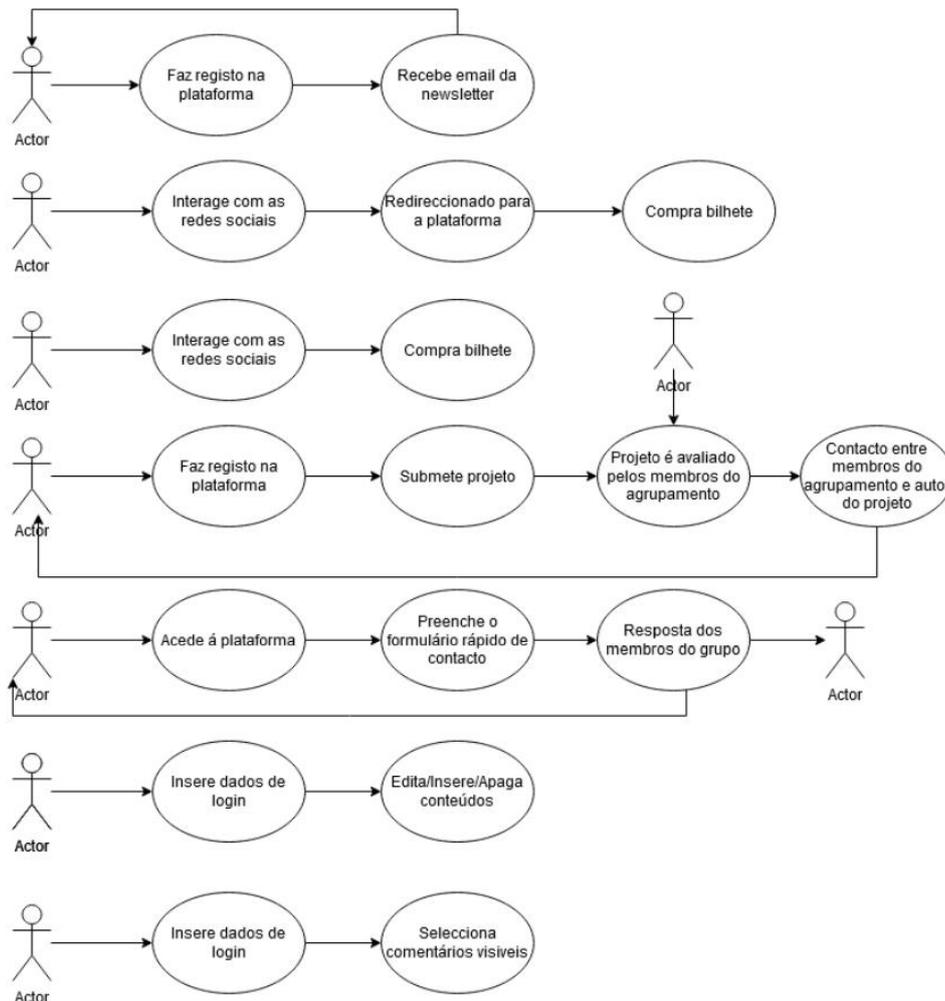


Figura 3-Casos de uso

Diagramas de Atividades

Na Figura 4, Figura 5, Figura 6 e Figura 7 apresentam-se os diagramas de atividades da visualização, atualização, inserção e eliminação de dados, respetivamente. O utilizador inicia a aplicação Web no ponto de entrada que é um círculo branco e termina num círculo preenchido de preto. As caixas são as atividades, enquanto que as setas representam o fluxo entre as atividades. Os losangos são os diamantes de decisão, onde pode ocorrer uma alteração do fluxo consoante cada situação específica.

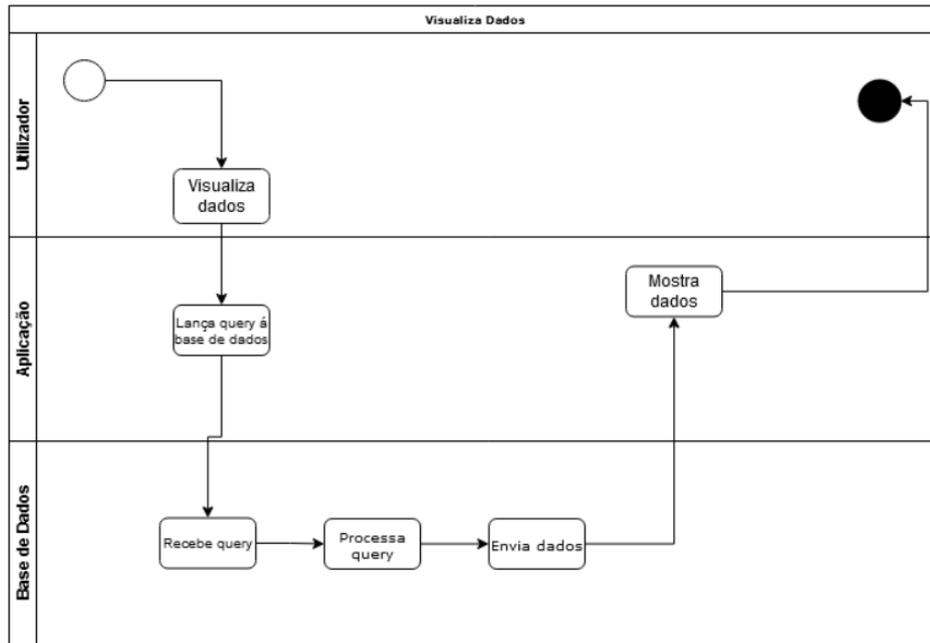


Figura 4- Diagrama de atividade de visualização de dados

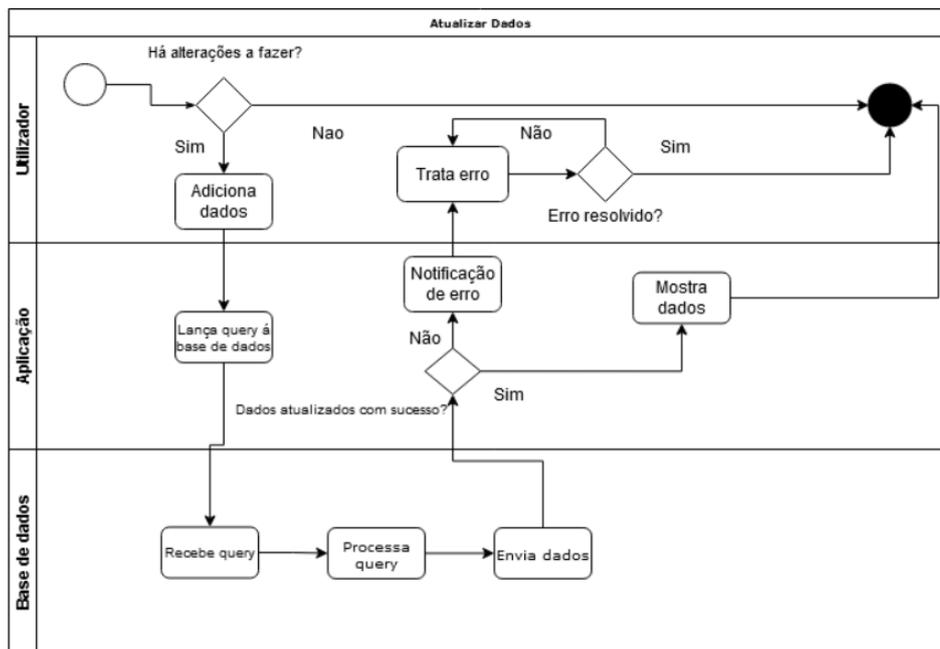


Figura 5- Diagrama de atividade de atualização de dados

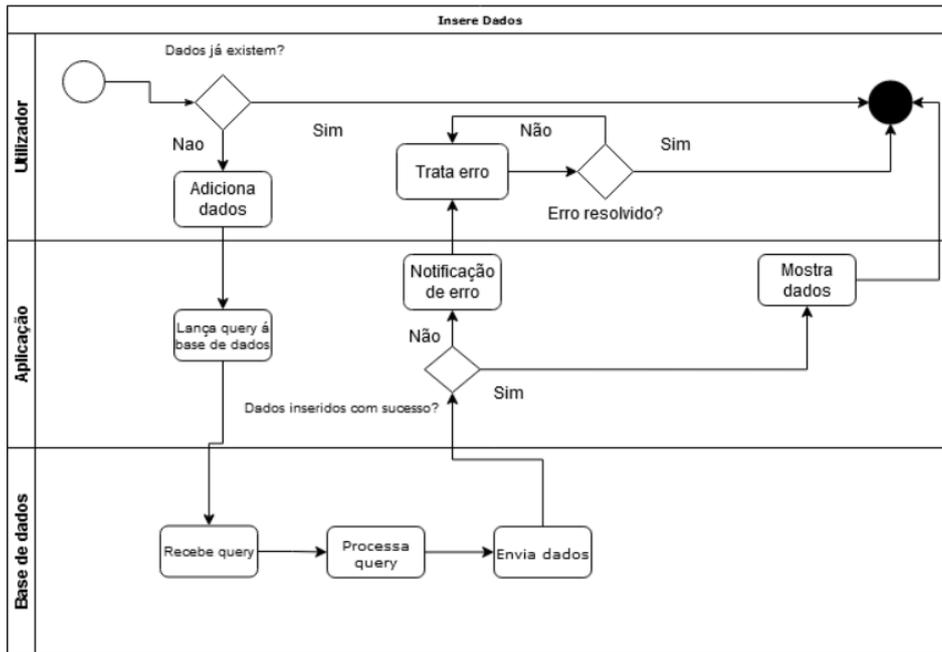


Figura 6- Diagrama de atividade de inserção de dados

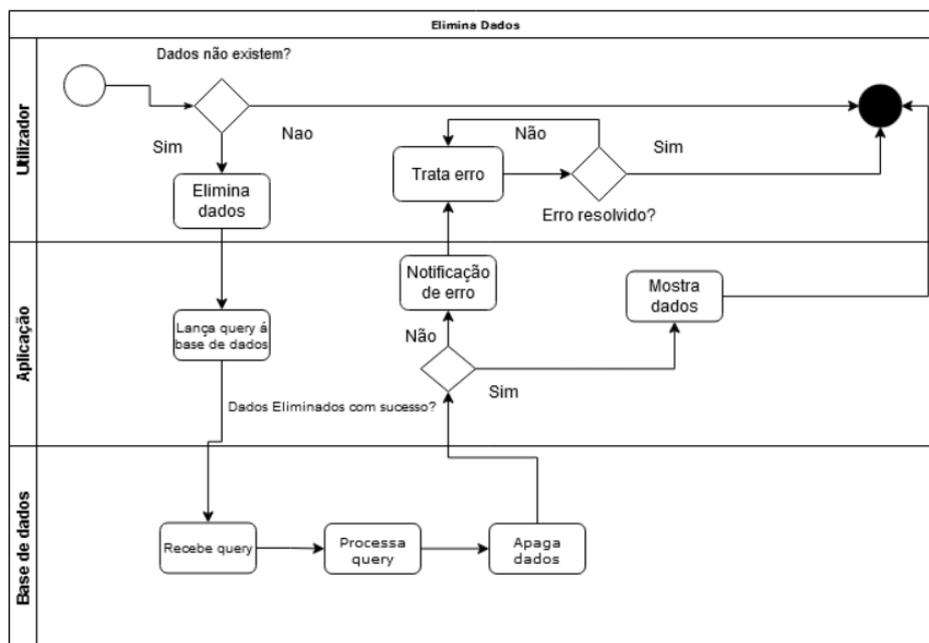


Figura 7- Diagrama de atividade de eliminação de dados

### Diagramas de Entidade Relação

Na Figura 8 podemos observar o diagrama de entidade relação, no qual constam todas as classes necessárias para o bom funcionamento da plataforma. Este vai ditar como será o modelo de dados a implementar.

### Estrutura da Aplicação

Na Figura 9 está representada a estrutura das páginas da aplicação Web desenvolvida. Temos uma página inicial em que temos apresentados os vários conteúdos que por sua vez se vão dividindo em subcategorias referentes á categoria principal. No **Erro! A origem da referência não foi encontrada.** estão disponíveis os *mockups* que desenvolvemos antes da construção da aplicação.

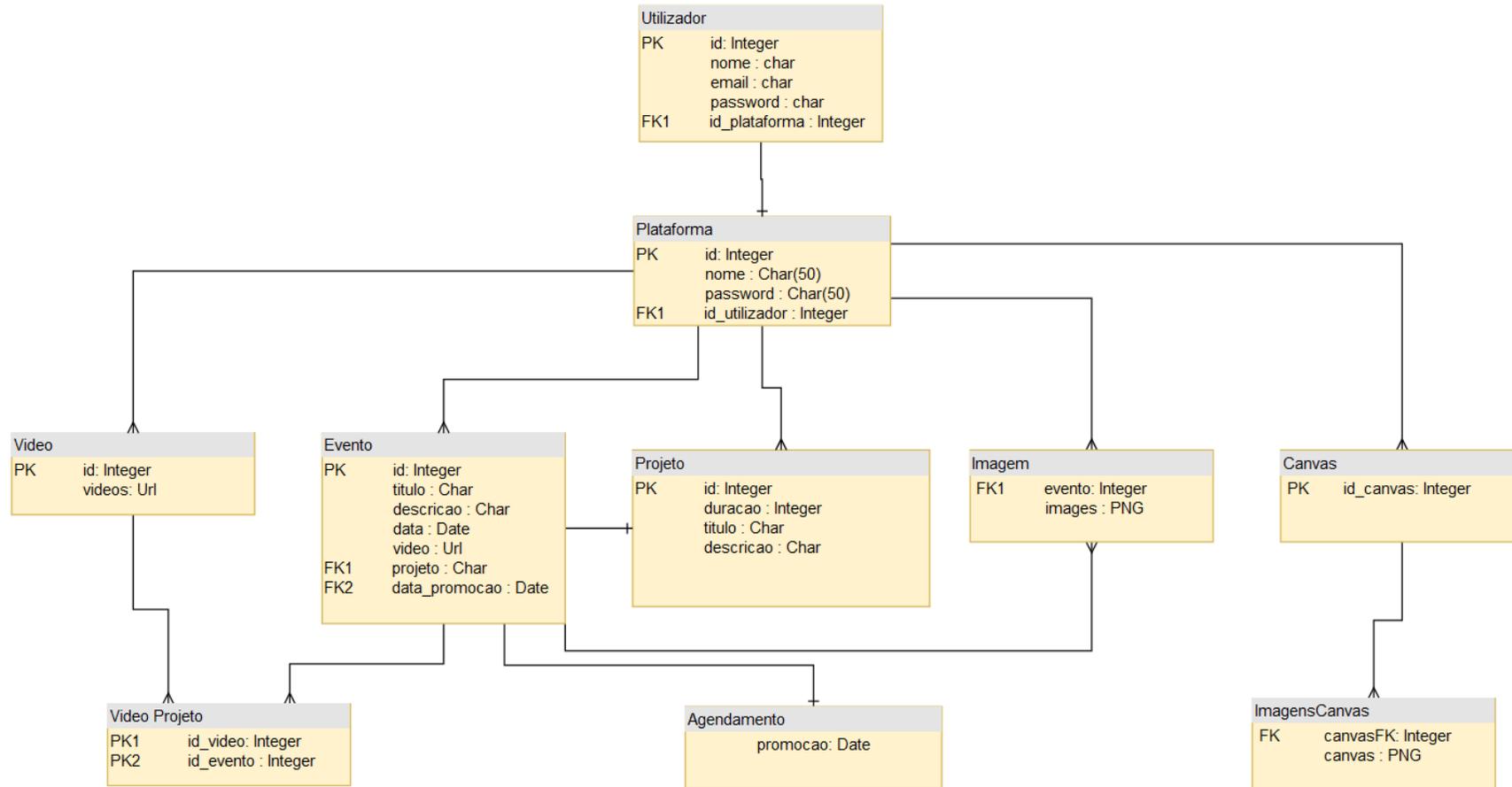


Figura 8- Diagrama de entidade relação

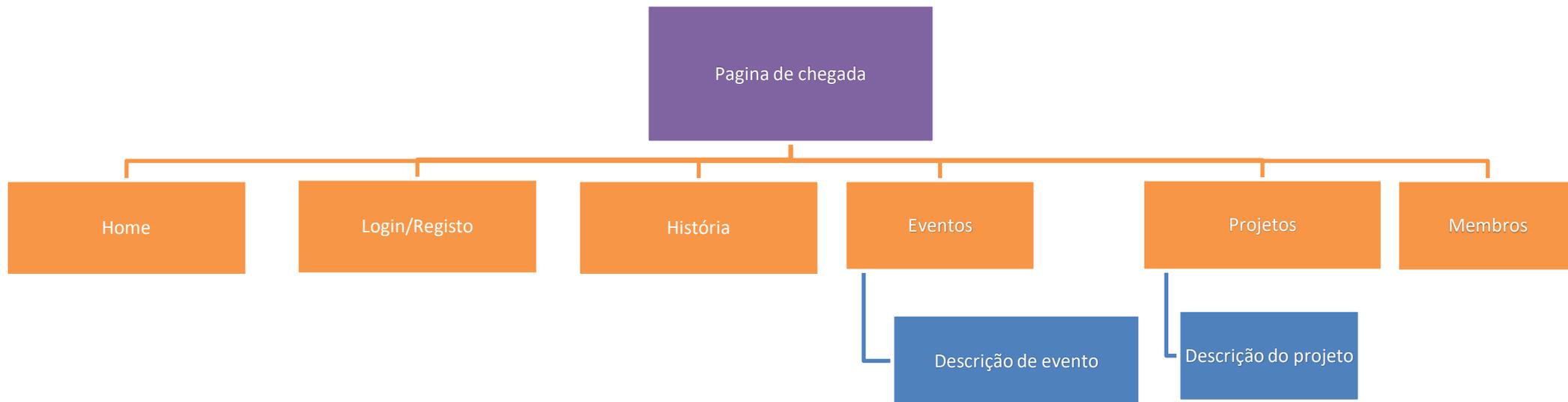


Figura 9- Estrutura das páginas da aplicação

### **3 Viabilidade e Pertinência**

Para avaliar a pertinência deste problema bem como a exequibilidade da solução proposta realizamos inquéritos a membros de grupos musicais e aos seus ouvintes, com o objetivo de compreender a pertinência das ideias inicialmente definidas por nós, bem como o grau importância das mesmas quando colocadas lado a lado com outras ideias. Perguntamos também a ambos os grupos que outras funcionalidades para além daquelas que enumeramos gostariam de ver implementadas. No grupo dos membros as propostas que achamos mais interessantes foram a de criação de um formulário de contacto rápido forma a receberem propostas de concertos ou colaborações mais diretamente.

#### **Análise dos Inquéritos**

Após a análise das respostas dadas aos inquéritos, presentes no A.1 Questionário a Membros de Grupos e A.2 Questionário a Ouvintes, concluímos que grande parte funcionalidades por nós propostas, são vistas como bastante uteis para os utilizadores responsáveis por gerir plataformas de agrupamentos musicais.

#### **Testes de Validação e Questionários de Satisfação**

Com a análise realizada na presente secção e depois da realização dos testes de validação, é possível concluir que a proposta para este TFC é viável e pertinente num ambiente real, apresentando a aprovação e interesse de terceiros, e das variadas respostas obtidas de uma amostra populacional variada, através do Inquérito e do Questionário de satisfação. Assim, a viabilidade inicial encontra-se atual e fundamentada através da realização de testes num contexto real com a participação de terceiros, cujo resultados se encontram na Secção 8.

## 4 Solução Proposta

### Arquitetura

A solução que propomos, para resolver o problema apresentado, é a criação de uma aplicação web que utilizará a *framework Model View Template (MVT)* semelhante ao tradicional MVC [MVC20], mas customizada pelo *Django*. Como lecionado ao longo do nosso curso, a MVT (*Model View Template*) encontra-se dividida em 3 camadas:

- **Model** – *Model*, ou Modelo, é a parte onde são criadas as classes que irão constar na nossa base de dados.
- **View** – *View*, ou Visão, é onde se encontra a parte logica da aplicação, é onde criamos funções (python) para manipular os dados e realizar tarefas.
- **Template**– *Template* é a camada responsável por renderizar a informação contida no models gerar o resultado.

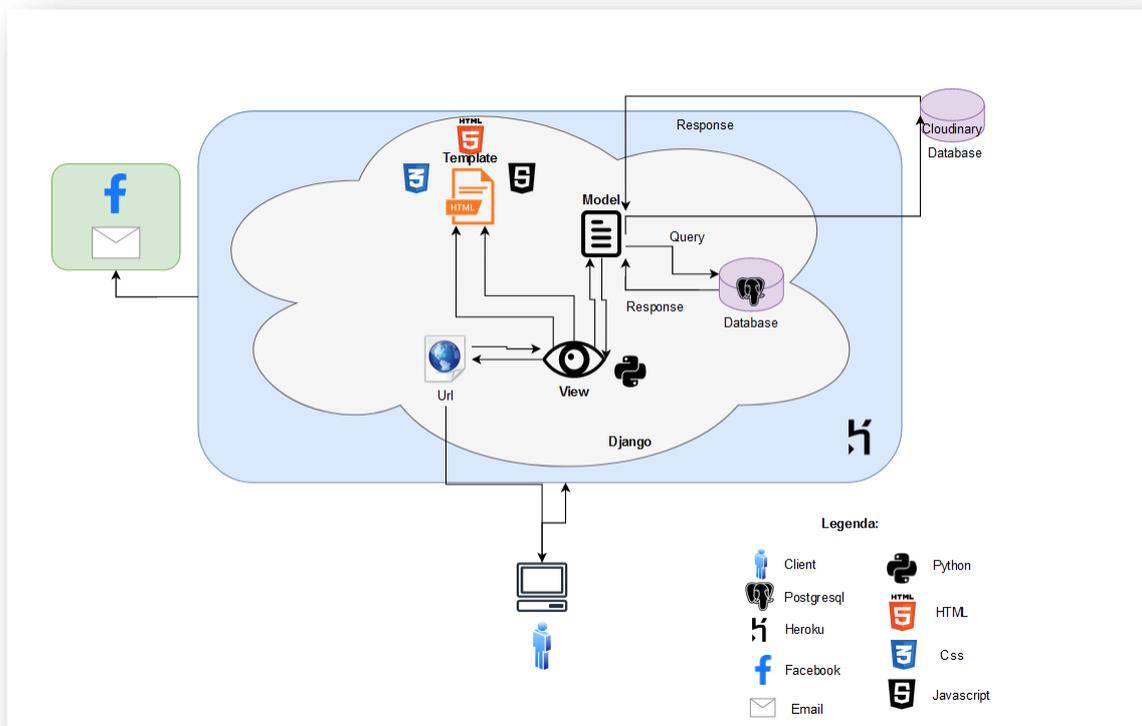


Figura 10- Arquitetura da aplicação web com identificação das tecnologias utilizadas.

Esta aplicação usa várias tecnologias, representadas na figura:

- **Python**, linguagem usada no desenvolvimento de backend;
- **Django**, web framework
- **Heroku**, no alojamento da aplicação
- **HTML e CSS**, na criação de páginas renderizadas

- **JavaScript**, na implementação de funcionalidades no front-end
- **Cloudinary** para alojamento de imagens
- **PostgreSQL** como base de dados.

Componentes

### **Model**

No ficheiro *Models* temos presente todas as classes criadas no projeto com os respetivos campos. É responsável pela criação de tabelas integradas na base de dados. Para consultar o conteúdo destas tabelas podemos recorrer á ferramenta “admin” do Django, através de uma conta de administrador e mapear as respetivas tabelas. Para este efeito o Django utiliza uma base de dados local em SQLite. Esta base de dados é de muito fácil criação, temos apenas de correr dois comandos, nomeadamente “python manage.py makemigrations” e “python manage.py migrate”. Nesta componente temos várias classes:

- **Agendamento**: utilizado para definir a data de publicitação do evento
- **Vídeo**: Recebe um Url correspondente a um vídeo de um evento
- **Projeto**: É utilizado para a criação do projeto
- **Evento**: É utilizado para a criação do projeto
- **Canvas**: É utilizado para carregar Canvas na cloud
- **ImagemCanvas**: Funciona como um inline, ou seja, será inserido dentro da classe *Canvas*. Quantas vezes seja necessário, desta forma podemos inserir quantos *Canvas* quisermos de uma só vez.
- **Imagem**: É utilizado para inserir imagens relativas aos eventos.

```

class Projeto(models.Model):
    id_projeto=models.AutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')
    titulo_projeto= models.CharField(max_length=200, default='')
    autores = models.CharField(max_length=200, default='')
    programa= models.TextField(max_length=20000, default='')
    notas = models.CharField(max_length=200, default='')

    def get_absolute_url(self):
        return reverse('projetos', kwargs={'pk': self.id_projeto})

    def __str__(self):
        return f"{self.titulo_projeto}"

class Evento2(models.Model):
    id_evento =models.AutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')
    titulo_evento = models.CharField(max_length=200, default='')
    descricao_evento = models.TextField(max_length=20000, default='')
    data_evento = models.DateField(default=datetime.now, blank=True)
    local=models.CharField(max_length=200, default='')
    artistas = models.CharField(max_length=200, default='')
    projeto =models.ForeignKey(Projeto, on_delete=models.CASCADE, null=True)
    video_evento=models.ManyToManyField(Videos, blank=True)
    data_promocao = models.ForeignKey(agendamento, on_delete=models.CASCADE, null=True)

    def __str__(self):
        return f"{self.titulo_evento} {self.data_promocao}"

    def get_absolute_url(self):
        return reverse('eventos', kwargs={'pk': self.id_evento, 'pk2': self.data_evento})

```

Figura 11- Classe projeto e evento do ficheiro models

Como podemos observar na Figura 11, as classes Projeto e Evento são as mais complexas. A classe evento possui muitos atributos e relações, entre as quais temos **foreign key** e de **many-to-many**. Nas relações de **foreign key** temos o **projeto** e o **vídeo\_evento**, isto permite-nos adicionar projetos e vídeos novos aquando da criação de um evento ou atribuir vídeos ou projetos já existentes a um novo evento, tal como ilustrado na Figura 12. Para além destas classes temos também duas funções chamadas “get\_absolute\_url” que nos irão permitir gerar uma página de detalhe de evento e projeto sempre que um evento ou projeto é criado.

### Criação de Conteúdos com a Aplicação Admin

A aplicação admin do Django é responsável pelo mapeamento dos atributos das classes criadas anteriormente no models. Desta forma conseguimos adicionar todo o tipo de conteúdos de forma simples, sem necessitarmos de criar *forms* no website. Esta aplicação permite também eliminar e editar conteúdos já existentes. Podemos ver um exemplo da criação de um novo projeto quando criamos um novo evento na aplicação Admin na Figura 12, isto graças à relação de **foreign key** existente entre estas classes. Esta é uma das versáteis ferramentas ao dispor do Django e também uma das principais razões pela escolha desta framework.

Esta aplicação apenas esta acessível a utilizadores que tenham sido registados através da plataforma, desta forma os restantes utilizadores registados pela plataforma via front-end, não terão acesso a esta ferramenta, permitindo assim manter um nível maior de abstração.

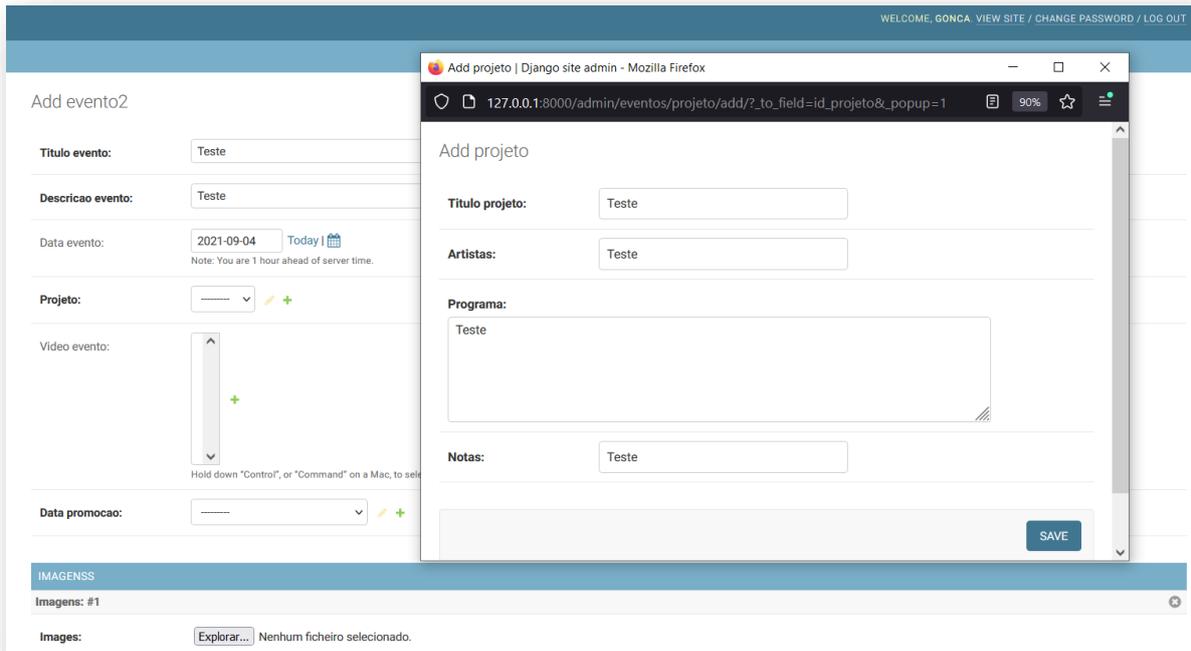


Figura 12- Criar um novo projeto ao criar um novo evento

Uma das funcionalidades por nos utilizadas consiste na criação de *inlines* nas classes **Imagens** e **ImagensCanvas**. Isto permite adicionar esta classe diretamente dentro das classes que possuam uma relação de *foreign key* com esta. Esta funcionalidade é nos muito útil porque desta forma conseguimos adicionar quantas imagens quantas quisermos de uma só vez tanto na *cloud* como em eventos diretamente. Na Figura 13 podemos observar a classe Imagem dentro da classe Evento.

The screenshot shows a web form titled "Add evento2". It contains several input fields: "Titulo evento:", "Descricao evento:", "Data evento:" (with a date picker set to 2021-09-04 and a note "Note: You are 1 hour ahead of server time."), "Projeto:" (a dropdown menu), "Video evento:" (a vertical scrollbar with a plus sign), and "Data promocao:" (another dropdown menu). Below these fields is a section titled "IMAGENS" with three rows, each labeled "Imagens: #1", "#2", and "#3". Each row has an "Imagens:" label, an "Explorar..." button, and the text "Nenhum ficheiro selecionado.". At the bottom of the image section is a link "+ Add another Imagens". At the very bottom of the form are three buttons: "Save and add another", "Save and continue editing", and "SAVE".

Figura 13- Classe Imagens dentro da classe Evento

### Views

No ficheiro `views.py` definimos as funções para renderizar cada Template. Cada URL está associado a uma *View*, que, por sua vez invoca classes do ficheiro `models.py`, podendo atribuir o retorno dessas funções a variáveis a enviar para o Template. O funcionamento da aplicação começa com um pedido URL ao Django, com a finalidade de obter uma resposta da sua parte. O pedido URL é processado e associado a uma *View*. Esta, estabelece a ligação com o Models, preparando o conteúdo para enviar para o Template.

O nosso projeto tem várias páginas existentes criadas através de URLs (**Figura 14**) e *views*. Quando um utilizador acede a um link através da URL, é feito um pedido HTTP que é mapeado através da *view*, como podemos observar na **Figura 15**, e que por sua vez irá retornar um HTML renderizado, representado na **Figura 16**.

```
from django.urls import path, include
from django.contrib import admin
from . import views

urlpatterns = [
    path("home", views.home, name='home'),
    path("", views.home, name='home'),
    path("membros", views.Membros, name='membros'),
    path("historia", views.Historia, name='historia'),
    path("index/livrohonra", views.LivroHonra, name='livrohonra'),
    path("eventos_new/", views.Eventos_new, name="eventos_new"),
    path("quem_somos_new", views.Quem_somos_new, name="quem_somos_new"),
    path("Projetos", views.projetos, name="Projetos"),
    path('projetos/<id>/', views.detail_view_projetos, name="projetos"),
    path('eventos/<id>/<data>/', views.detail_view_eventos, name='eventos'),
    path("registo/", views.UserRegisterView2, name="registo"),
    path("login/", views.loginPage, name="login"),
    path("canvas/<id>/<data>/", views.detail_view_canvas, name="canvas"),
]
```

Figura 14 - Urls.py

```
def Membros(request):
    return render(request, 'Membros.html')
```

Figura 15 - View Membros



Figura 16 - Página de membros

mostrar imagem do urls.py com todos os paths

mostrar image de views.py com um ou duas funções views

### Consulta de Dados

Na secção do site denominada eventos podemos filtrar os dados das tabelas de eventos e projetos. É possível pesquisar por palavras que estejam contidas no título e/ou descrição do evento/projeto. Esta pesquisa é realizada por um ficheiro *javascript* com uma função de *ajax search* em que sempre que o utilizador insere uma letra a função entra em execução e verifica em tempo real os dados que fazem match com o que está a ser inserido. Na **Figura 17** podemos observar a função responsável pela pesquisa. Esta função de front-end estabelece uma ponte com o back-end através da *view* representada na Figura 19.

```
const user_input = $("#user-input")
const search_icon = $('#search-icon')
const artists_div = $('#replaceable-content')
const endpoint2 = '/eventos_new/'
const delay_by_in_ms = 700
let scheduled_function = false

let ajax_call = function (endpoint2, request_parameters) {
  $.getJSON( endpoint2, request_parameters)
    .done(response => {
      // fade out the artists_div, then:
      artists_div.fadeTo('slow', 0).promise().then(() => {
        // replace the HTML contents
        artists_div.html(response['html_from_view'])
        // fade-in the div with new contents
        artists_div.fadeTo('slow', 1)
        // stop animating search icon
        search_icon.removeClass('blink')
      })
    })
}

user_input.on('keyup', function () {

  const request_parameters = {
    a: $(this).val(),
  }
  // value of user_input: the HTML element with ID user-input

  // start animating the search icon with the CSS class
  search_icon.addClass('blink')

  // if scheduled_function is NOT false, cancel the execution of the function
  if (scheduled_function) {
    clearTimeout(scheduled_function)
  }

  // setTimeout returns the ID of the function to be executed
  scheduled_function = setTimeout(ajax_call, delay_by_in_ms, endpoint2, request_parameters)
})
```

Figura 17- Ficheiro JavaScript responsável pela pesquisa por texto

### Pesquisa de Eventos

Preencha os campos seguintes para filtrar os resultados

Evento

Data mínima do evento

Data máxima do evento

#### Resultados da pesquisa

Título do Evento	Projeto Associado	Artistas	Local	Data
teste	teste	teste	teste	Sept. 10, 2021

Figura 18- Resultados da pesquisa de teste

Na *view* "Eventos\_new" é onde vamos recolher todos os dados, fazendo "Eventos.objects.all()" e "Projetos.objects.all()". Se existirem parâmetros a serem inseridos, entramos dentro do *if url parameter* em que iremos comparar se em cada objeto do tipo eventos e projetos existe algum título ou descrição de ambos, que corresponda ao parâmetro que esta a ser inserido. De seguida é a função de pesquisa *ajax* em que iremos renderizar a página "eventos.html" que é onde estão presentes os dados para apresentar os resultados da pesquisa.

```
def Eventos_new(request):
    eventos = Evento.objects.all()
    projetos = Projeto.objects.all()
    date_min_eventos = request.GET.get('date_min_eventos')
    date_max_eventos = request.GET.get('date_max_eventos')
    date_min_projetos = request.GET.get('date_min_projetos')
    date_max_projetos = request.GET.get('date_max_projetos')

    ctx = {}
    xpto={}
    url_parameter = request.GET.get("a")
    if date_min_eventos != '' and date_min_eventos is not None:
        eventos = eventos.filter(data_evento__gte=date_min_eventos)
    if date_min_projetos != '' and date_min_projetos is not None:
        projetos = projetos.filter(data_projeto__gte=date_min_projetos)

    if date_max_eventos != '' and date_max_eventos is not None:
        eventos = eventos.filter(data_evento__lt=date_max_eventos)
    if date_max_projetos != '' and date_max_projetos is not None:
        projetos = projetos.filter(data_projeto__lt=date_max_projetos)

    if url_parameter:
        eventos = Evento.objects.filter(Q(titulo_evento__icontains=url_parameter) | Q(descricao_evento__icontains=url_parameter))
        projetos = Projeto.objects.filter(Q(titulo_projeto__icontains=url_parameter) | Q(descricao_projeto__icontains=url_parameter))

    ctx={"eventos":eventos, "projetos":projetos}
    if request.is_ajax():

        html = render_to_string(
            template_name="eventos.html", context={"eventos":eventos, "projetos":projetos}
        )
        data_dict = {"html_from_view": html}
        return JsonResponse(data=data_dict, safe=False)
    ctx={"eventos":eventos, "projetos":projetos}
    return render(request, "Eventos_new.html", context=ctx)
```

Figura 19- View responsável pela pesquisa de dados

```

<h3>Resultados da pesquisa </h3>
</div>
<div class="card card-body">
  <table class="table">
    <tr>
      <th>Titulo do Evento</th>
      <th>Projeto Associado</th>
      <th>Artistas</th>
      <th>Local</th>
      <th>Data</th>
    </tr>

    {% for evento in eventos %}
    <tr>
      <td>
        <ul><a href="{% url 'eventos' evento.id_evento evento.data_evento %}">{{ evento.titulo_evento }}</a></ul>
      </td>
      <td>{{evento.projeto}}</td>
      <td>{{evento.artistas}}</td>
      <td>{{evento.local}}</td>
      <td>{{evento.data_evento}}</td>
    </tr>
    {% endfor %}
  </div>

{% endblock content %}

```

Figura 20- Página eventos.html responsável pela apresentação dos dados

### Consulta de Dados por Data

Na pesquisa de dados temos também a opção de consulta por data, em que podemos selecionar um intervalo de datas em que será exibida a lista de eventos ou projetos que se incluam nesse intervalo. Esta pesquisa é realizada nesta mesma *view* onde criamos os campos de pesquisa com as datas mínimas e máximas com a notação *GET* que nos permite fazer a consulta. Em seguida verificamos se os *inputs* dados pelo utilizador são validos, por fim retornamos os eventos ou projetos em que a data presente na base de dados é maior ou igual à data selecionada, no caso de ter selecionado a data mínima e menor no caso de ter selecionado a data máxima.

### Pesquisa de Eventos

Preencha os campos seguintes para filtrar os resultados

Evento

Data minima do evento

Data maxima do evento

Figura 21-Exemplo de pesquisa por intervalo de datas

Resultados da pesquisa				
Título do Evento	Projeto Associado	Artistas	Local	Data
teste	teste	teste	teste	Sept. 10, 2021

Figura 22- Resultados da pesquisa por intervalo de datas

### **Criação e transferência de Canvas**

A página do evento terá como URL o id do evento e a data da sua ocorrência. Escolhemos utilizar este sistema uma vez que será necessário guardar a data do evento na URL para posteriormente ser utilizada na função JavaScript que é responsável pela criação do nome do ficheiro **Canvas** (print da página do evento em formato PNG) que é criado por uma outra função, também ela escrita em JavaScript. Na Figura 23 podemos observar a função `saveCanvas`. Esta função vai transformar a URL completa do evento em apenas uma *String* com a data do evento. Depois, a função `html2canvas` utiliza um seletor para determinar qual área da página nós queremos transformar em PNG. De seguida definimos qual o formato da imagem, e por fim vamos buscar a função `saveAs`, que é responsável pelo download propriamente dito, e acrescentamos a *tag* “evento-canvas” à *String* com a data do evento que tínhamos guardado anteriormente. Decidimos utilizar esta *tag* para facilitar a identificação dos *Canvas* pelo utilizador no momento do carregamento dos mesmos na *cloud*. Na **Figura 24** podemos ver um exemplo de um nome de um *Canvas* e na **Figura 25** podemos ver um *Canvas* já criado.

```
function saveCanvas(){
  var str= String(window.location.href)
  var id= str[str.length- 2];
  var splits = str.split(/[0-9]+/);
  var id = splits[13] + splits[14] + splits[15] + splits[16] + splits[17]
  //var id = splits[3] + splits[4] + splits[5] + splits[6] + splits[7] // se estiver a correr em heroku

  html2canvas(document.querySelector("#xpto")).then(canvas => {
    var image = canvas.toDataURL("image/png");
    var image_png = saveAs(canvas.toDataURL(), 'evento-canvas'+id);
  });
}

function saveAs(uri, filename) {
  var link = document.createElement('a');
  if (typeof link.download === 'string') {
    link.href = uri;
    link.download = filename;

    //Firefox requires the link to be in the body
    document.body.appendChild(link);
    //simulate click
    link.click();
    //remove the link when done
    document.body.removeChild(link);
  } else {
    window.open(uri);
  }
}

saveAs()
saveCanvas()
</script>
```

Figura 23- Função de download e de nomeação do *Canvas*

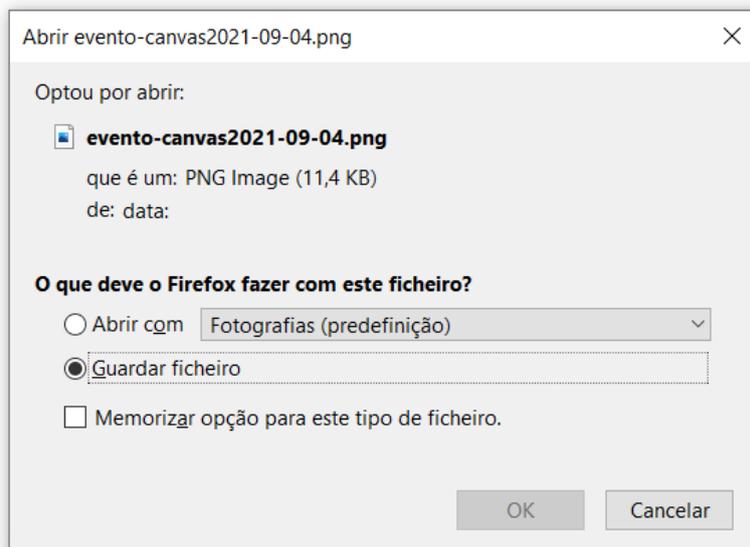


Figura 24- Exemplo do nome do *Canvas* no download



Figura 25 - Exemplo de *Canvas*

### ***Zonas restritas do site***

Existem zonas no site que só estão visíveis se o utilizador tiver permissões de administrador. No caso da secção *Canvas*, e como se trata de uma ferramenta que só será utilizada por administradores quando querem publicitar um determinado evento, resolvemos deixar esta secção apenas visível para os mesmos. Para além desta restrição existe também uma outra, em que quando um utilizador já efetuou o login, já não lhe é permitido voltar a essa mesma página enquanto estiver o login ativo, se tentar, é redirecionado para a página principal.

Para isto utilizamos um ficheiro python (**Figura 26**) com funções que utilizam grupos de utilizadores da aplicação *admin* do Django. Criamos dois grupos, um chamado “admin”, no qual estão presentes todas as permissões, e outro grupo chamado “Customer” que apenas possui as permissões de um utilizador dito normal. Este ficheiro é depois chamado no ficheiro *views.py* e as funções são chamadas através do símbolo “@” para “envolver” as funções presentes na view, como podemos ver na **Figura 27**.

```
from django.http import HttpResponseRedirect
from django.shortcuts import redirect

def unauthenticated_user(view_func):
    def wrapper_func(request, *args, **kwargs):
        if request.user.is_authenticated:
            return redirect('home')
        else:
            return view_func(request, *args, **kwargs)

    return wrapper_func

def allowed_users(allowed_roles=[]):
    def decorators(view_func):
        def wrapper_func(request, *args, **kwargs):
            group= None
            if request.user.groups.exists():
                group = request.user.groups.all()[0].name

            if group in allowed_roles:
                return view_func(request, *args, **kwargs)
            else:
                return HttpResponseRedirect('Não está autorizado a aceder a esta página')

        return wrapper_func
    return decorators
```

Figura 26 - Ficheiro de restrição de permissões

```
@unauthenticated_user
def loginPage(request):

    if request.method == 'POST':
        username = request.POST.get('username')
        password = request.POST.get('password')

        user = authenticate(request, username=username, password=password)

        if user is not None :
            login(request, user)
            return redirect('home')
        else:
            messages.info(request, 'Username OU Password incorreta ')

    context = {}
    return render(request, 'login.html', context)

@login_required(login_url='login')
@allowed_users(allowed_roles=['admin'])
def detail_view_canvas(request, id, data):
    evento= get_object_or_404(Evento2, id_evento=id, data_evento=data)
    eventos = Evento2.objects.filter(id_evento= id, data_evento=data)
    return render(request, 'Canvas.html', {'evento': evento, 'eventos': eventos})
```

Figura 27 - View de Login e Canvas com as restrições de acesso

### ***Envio automático e calendarizado de posts de eventos ao Facebook***

Para o envio automático de *posts* desenvolvemos um ficheiro Python, presente na Figura 28 que utiliza uma API do Cloudinary que nos permite pesquisar ao longo dos diretórios, retornando uma *JSON response* com todos os dados relativos a cada uma das imagens nele contidas. Desta resposta vamos extrair a url do *Canvas* que nos interessa postar no Facebook. Como anteriormente gravamos todos os *Canvas* com uma *tag* e a data do evento, agora torna se simples a nossa pesquisa. Temos apenas de limpar a informação que não nos interessa, deixando apenas a URL. Como este *script* irá ser executado diariamente podemos criar uma variável em que definimos o tempo atual e acrescentamos-lhe o número de dias de intervalo entre a data de ocorrência do evento e data da sua publicitação. Assim o ficheiro vai procurar por um *Canvas* que tenha uma data de ocorrência para daqui a 7 dias por exemplo, se for esse o intervalo de

tempo definido por nós, e irá postar essa URL no Facebook, a 7 dias da data de ocorrência do evento. Para postar no Facebook utilizamos uma API do Facebook que necessita de um *access token*. É possível obter este *token* registrando-se na página de “Facebook for developers”, *token* esse que tem uma duração máxima de 60 dias, o que deve ser renovado ao fim dessa data. Utilizamos ainda uma biblioteca do python chamada “urllib” uma vez que a API do Facebook recebe um ficheiro e não um URL, só desta forma foi possível postar o *Canvas*.

```
def post():

    cloundinary.config(
        cloud_name = "da7qp20ja",
        api_key = "835922641349356",
        api_secret = "qt3MG0N2omjL_7ks_khMi7k5_1c"
    )
    tag = 'evento-canvas'
    time_today = datetime.now()
    time_x = time_today + timedelta(days=-21)
    time2 = time_x.strftime("%Y-%m-%d")
    result = cloundinary.Search()\
        .expression(tag)\
        .execute()

    cloud = 'http://res.cloundinary.com/da7qp20ja/image/upload/'

    json_response = json.dumps(result)
    json_response_to_string = ''.join(json_response)
    stripped_string = json_response_to_string.replace('\"', '')
    json_response_to_list = list(json_response.split(' '))

    urls = [[a] for a in json_response_to_list if cloud in a]
    urls2 = [''.join(ele) for ele in urls]
    urls3 = ''.join(urls2)
    urls4 = urls3.replace('\"', '')
    urls_list = list(urls4.split(","))
    url=""
    link=""
    for url in urls_list:
        if time2 in url:
            link = url
    access_token = "EAAEZAZAyZC4tuYBAAocZAbS2Vh5yWR50y6rkBXfrZAU5ffZB"
    facebook = fb.GraphAPI(access_token)
    facebook.put_photo(urllib.request.urlopen(link))

post()
```

Figura 28- Função de *Post* no Facebook

Para executar este ficheiro diariamente utilizamos o add-on do Heroku chamado Heroku Scheduler, que nos permite executar ficheiros de forma automática com a frequência por nós escolhida, como podemos observar na **Figura 29**.

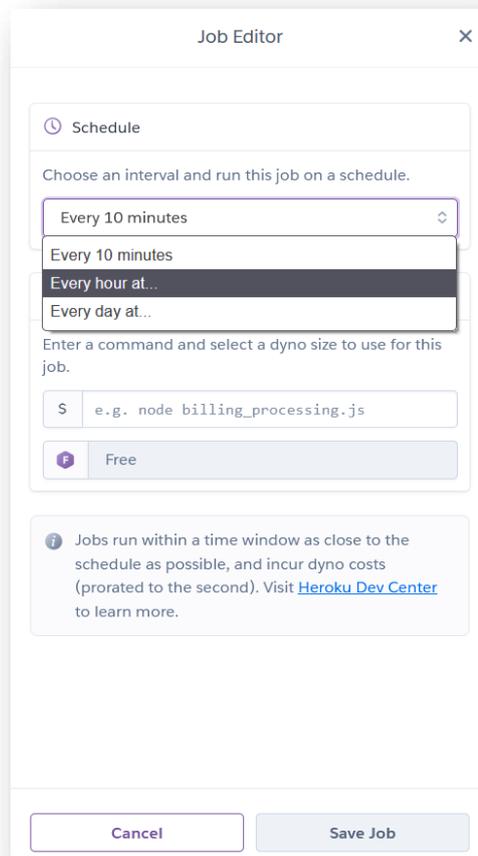


Figura 29- Heroku Scheduler

### Envio automático calendarizado de anúncio de Evento para *mailing list*

Para o envio de emails a utilizadores registados utilizamos a ferramenta de envio de emails do Django. Utilizamos o mesmo sistema de pesquisa de imagens utilizado na publicitação de eventos no Facebook explicado anteriormente, de seguida é feito uma *query* á base de dados para obter a lista completa de emails dos utilizadores registados na plataforma. Por fim substituímos o campo de corpo do email por uma pequena página de HTML com a *URL* do *Canvas* do evento.

### Hosting do site

A nível do deploy do website utilizamos uma integração em Heroku com uma base de dados em PostgreSQL através de um add-on do Heroku. Escolhemos o Heroku pelo facto da sua utilização ser gratuita e também pela vasta lista de add-ons. Outro dos add-ons que também incluímos no nosso projeto é o Heroku Scheduler. Este add-on permite-nos escolher a data exata e a frequência com que vamos executar um ficheiro. Esta aplicação está disponível online no *URI*: <https://quarteto-arabesco.herokuapp.com>

### Armazenamento na Cloud

Para o armazenamento de imagens da plataforma utilizamos o Cloudinary, que é um serviço de armazenamento em nuvem, permite armazenar tanto imagens como vídeos. Escolhemos este serviço pelo facto de ser gratuito nos primeiros 25GB de armazenamento e também pela sua facilidade de utilização.

### Visualizações da Aplicação Desenvolvida

Em todas as páginas da Plataforma Web apresentamos uma *Navigation Bar* que segue a página, permitindo ao utilizador navegar por diferentes separadores sempre que assim pretender.

Esta barra é utilizada como layout, no Django, o que nos permite alterar sempre que pretendermos e a alteração será efetuada em todos os separadores.



**Figura 30- Navigation Bar**

A página *Home*, (**Figura 31**) que é sempre a primeira página a abrir, apresenta uma imagem, do grupo, com uma breve apresentação do grupo em questão. A página tem presente o efeito de *parallax* que faz com que seja sobreposto à imagem de fundo os conteúdos pretendidos.

A página *História* (**Figura 32**) apresenta dois blocos distintos de apresentação.

O primeiro Bloco é constituído pela história do grupo e uma imagem do mesmo. O Segundo Bloco apresenta o percurso, por ano, do grupo Quarteto Arabesco. As imagens ainda não foram alteradas, mas no futuro irá apresentar fotos, do grupo, dos eventos nos anos apresentados.

### Quem somos?

De forma pioneira em Portugal, o Quarteto Arabesco tem-se revelado um agrupamento versátil e multifacetado. Procura uma interpretação autêntica do mais variado repertório, tendo disponível um alargado leque de propostas de concerto. É composto por Denys Stetsenko e Raquel Cravino (violino), Lúcio Studer (viola) e Ana Raquel Pinheiro (violoncelo).

Em instrumentos da época, o Quarteto Arabesco dedica-se a interpretações historicamente informadas de música dos períodos Barroco e Clássico. Em instrumentos modernos, o Quarteto Arabesco aborda o mais variado repertório de música portuguesa dos séculos XX e XXI, desde o fado e o jazz à música contemporânea. Colabora regularmente com solistas e agrupamentos de destaque. O seu trabalho tem recebido reconhecimento dos mais variados quadrantes. Tem-se igualmente apresentado em formações mais alargadas como Ensemble Arabesco.

### O que fazemos?

Desde a sua estreia em 2006, tem desenvolvido uma carreira regular e consistente. Realizou mais de 180 concertos nos principais festivais e salas de Portugal, assim como para a rádio e em numerosas gravações. Destaca-se em baixo, do seu histórico de concertos, alguns concertos.

Em 2020 realizou uma série de concertos de apresentação do disco Reflexos gravado com o Cordis Ensemble e tem agendada uma série de apresentações pelo país da versão para quarteto de cordas da Criação de Haydn.

Em 2019 apresentou-se no Grande Auditório do CCB com Pedro Jóia, e fez a estreia moderna em Portugal da versão para quarteto de cordas da Criação de Haydn.

Em 2017 realizou um ciclo de concertos com o pianista Adriano Jordão com obras de Schumann e Webern. Apresentou nos Dias da Música, CCB, o "Requiem" de Mozart numa versão para quarteto de cordas.

Gravou o instrumental do tema "Amar pelos dois" de Luisa Sobral, interpretado por Salvador Sobral, música vencedora do Festival Eurovisão da canção.

Figura 31-Home



### A nossa história

O Quarteto Arabesco é um agrupamento que se tem dedicado, de forma pioneira em Portugal a interpretações historicamente informadas, em instrumentos da época, de música dos períodos barroco e clássico. Desde a sua estreia em 2006, com uma interpretação do Requiem de Mozart transmitida em directo pela Antena 2, realizou mais de 180 concertos nos principais festivais e salas de Portugal (Fundação Gulbenkian, Casa da Música, CCB, entre outros), assim como na rádio e em numerosas gravações. Colabora regularmente com solistas e agrupamentos de destaque, apresentando-se igualmente como Ensemble Arabesco em formações mais alargadas. O seu trabalho tem recebido reconhecimento dos mais variados quadrantes. Está actualmente a preparar a gravação de um conjunto de quartetos de Almeida Mota e um disco com o guitarrista Pedro Joia.

Figura 32-História

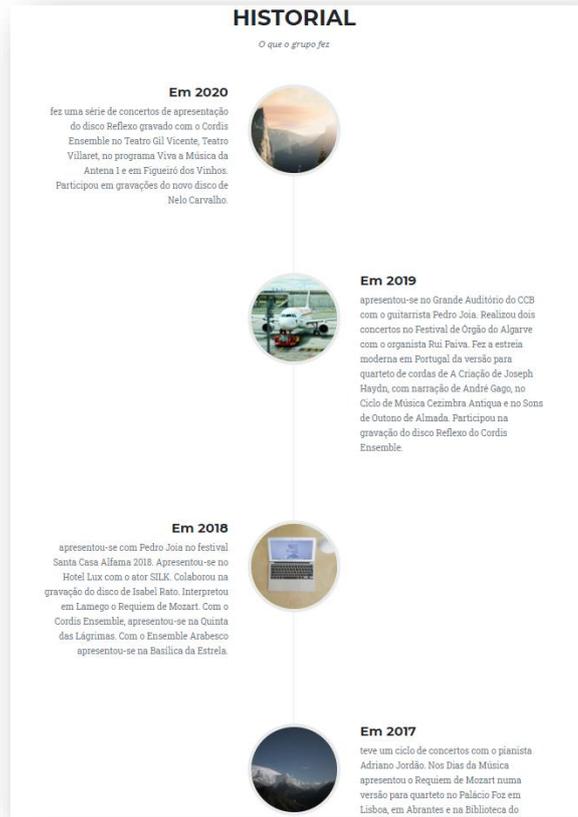


Figura 33 - História

A página “Quem Somos” representada na **Figura 34** apresenta os artistas explicando o seu percurso musical e a sua história individual.

## O NOSSO GRUPO



**Denys Stetsenko**  
Violino

Natural da Ucrânia onde cresceu numa família de tradição musical. Iniciou os estudos de violino aos 6 anos com o seu tio. Reside em Portugal desde 1995. Estudou na Academia Nacional Superior de Orquestra e na Escola Superior de Música de Lisboa onde concluiu o Curso de Licenciatura em Violino. Participou em vários Master Classes com os professores Felix Andrievsky, Zakhar Bron, Gerardo Ribeiro, Richard Gwilt, Chiara Banchini, Enrico Onofri e Vittorio Ghielmi, entre outros.

Concluiu o Mestrado em Música Antiga – Violino Barroco na Escola Superior de Música e Artes do Espectáculo do Porto, sob a orientação de professora Amandine Beyer. Foi Concertino da Orquestra Académica Metropolitana e da Orquestra Sinfónica Juvenil, integrou a Orquestra de Câmara de Coimbra. Colabora com as orquestras barrocas nacionais Divino Sospiro, Capela Real, Músicos do Tejo e agrupamentos de música antiga Sete Lágrimas, Concerto Campestre, La Nave Va. Faz parte de projectos de música e dança tradicional e lecciona Violino e Música de Câmara na Academia de Música de Santa Cecília. E também professor na Escola Superior de Música e Artes do Espectáculo do Porto, assistente de Amandine Beyer.

[Twitter](#) [Facebook](#) [LinkedIn](#)



**Raquel Cravino**  
Violino

Nasceu na cidade da Covilhã. Iniciou-se na música aos 5 anos de idade no Conservatório Regional de Música da Covilhã em piano e ballet. Sob a orientação do Professor António Oliveira e Silva, estudou violino na Escola Profissional de Artes da Beira Interior durante seis anos. Em 2002 obteve o diploma de Licenciatura na Academia Nacional Superior de Orquestra (ANSO) de Instrumentista de Orquestra - Violino, na classe da Professora Agnes Sárosi. É mestre em Pedagogia do Instrumento, na ANSO /Universidade Lusíada, sob a orientação do Professor Aníbal Lima. Realizou diversos cursos de aperfeiçoamento no âmbito do violino e da música de câmara com Joyce Tan, Sergey Kravchencko, Angelique Loyer, Gerardo Ribeiro, Augustin Dumay, Boris Kuniev, Jan Dobrelevky, Vladimir Ovcharech, James Dahlgren, Gilles Apap e Alexandre da Costa. O seu interesse pela música antiga levou-a a aperfeiçoar os conhecimentos no violino barroco frequentando master classes com Enrico Onofri, Richard Gwilt, Vittorio Ghielmi, Francesca Viccari e Chiara Banchini. Como freelancer, tem realizado concertos com a Orquestra Metropolitana de Lisboa, Orquestra de Câmara de Cascais e Oeiras; e tem uma participação activa em variados projectos artísticos de música antiga – Orquestra Barroca da Casa da Música, Flores da Música, Músicos do Tejo, La Nave Va, Capela Real. Dedicou-se ao ensino desde 2002 como docente de violino no Conservatório Metropolitano de Música de Lisboa e na Escola de Música do Conservatório Nacional de Lisboa. É membro fundador do Quarteto Arabesco.

[Twitter](#) [Facebook](#) [LinkedIn](#)



**Lúcio Studer**  
Viola

Nasceu em Lisboa. Iniciou os seus estudos musicais de viola na Escola de Música Nossa Senhora do Cabo, tendo prosseguido na Escola de Música do Conservatório Nacional com o Professor Fernando Afonso. Como bolsista da Fundação Calouste Gulbenkian, estudou com a Professora Barbara Friedhoff. Concluiu em 1999 o Curso Superior de Viola pela ANSO com o professor Paul Wakabayashi. Continuou os seus estudos em Espanha com Igor Soulyga (1999-2000), prosseguindo na Holanda no Conservatório Real de Haia e no Conservatório de Roterdão (2000-2002), graças a uma bolsa do Instituto Português das Artes do Espectáculo. Participou em Master Classes com Ana Bela Chaves, Gérard Caussé, Igor Soulyga, Jean Sulem, Richard Gwilt, Enrico Onofri e Alfredo Bernardini. Foi premiado no concurso da Juventude Musical Portuguesa. É igualmente doutorado pelo Instituto Superior Técnico, sendo professor universitário, investigador e gestor de projetos. Colabora regularmente com varias orquestras e ensembles, tais como a Orquestra Gulbenkian, Orquestra Metropolitana de Lisboa, Orquestra de Camara de Cascais e Oeiras, Orquestra Sinfonietta de Lisboa e Orquestra do Algarve. Colabora igualmente com os ensembles com os ensembles barrocos La Nave Va, Concerto Campestre, Ensemble Melleo Harmonia e Flores de Música, assim com as orquestras barrocas Capela Real, Orquestra del Montsalvat, Orquestra Barroca Casa da Música, Os Músicos do Tejo e Divino Sospiro.

[Twitter](#) [Facebook](#) [LinkedIn](#)



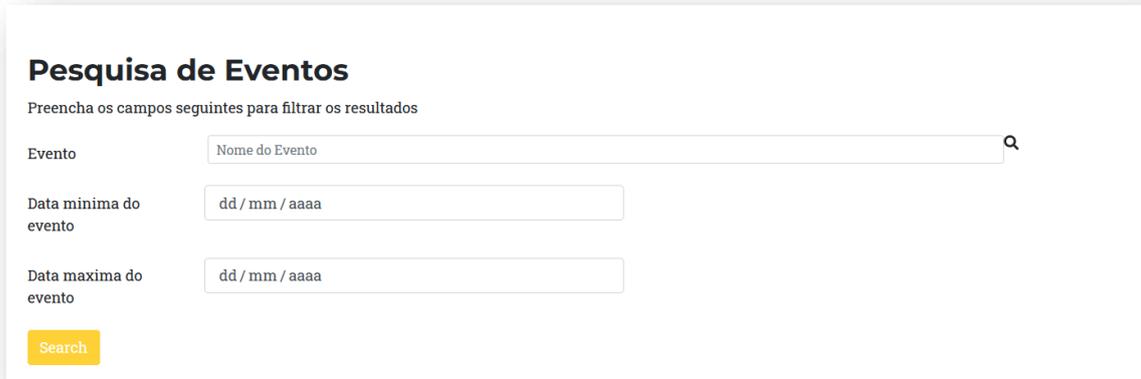
**Ana Raquel Pinheiro**  
violoncelo

Nasceu na Covilhã. Iniciou os estudos de violoncelo na Escola Profissional de Artes da Beira Interior na classe do prof. Rogério Peixinho. Concluiu o curso superior, Licenciatura em Violoncelo na Escola Superior

Figura 34- Quem somos

A página “Eventos” (Figura 35) apresenta um formulário de pesquisa e todos os eventos

A página atualiza automaticamente os eventos e projetos sempre que são adicionados caracteres nos campos de pesquisa.



**Pesquisa de Eventos**

Preencha os campos seguintes para filtrar os resultados

Evento  🔍

Data minima do evento

Data maxima do evento

**Figura 35 - Pesquisa de Eventos**

### Ambientes disponíveis

Para esta aplicação existe o ambiente de desenvolvimento local e o ambiente de produção.

A aplicação esta pronta a correr em ambos, requerendo apenas uma pequena alteração no ficheiro “settings.py”. Estas alterações encontram se indicadas através de comentários nesse mesmo ficheiro. Atualmente a aplicação esta pronta a correr localmente. No ficheiro README do github esta presente um link para um vídeo com a demo da plataforma.

- Link para repositório GitHub: <https://github.com/a21600767-goncalo/New>
- Link para Heroku: <https://quarteto-arabesco.herokuapp.com/>

## 5 Benchmarking

### Plataformas

Foi feito um estudo de mercado de outras plataformas existentes no mercado que possam ser usadas para a aplicação que se pretende desenvolver. Identificaram-se React, Laravel, Flask, Express e Django no Anexo D - Benchmarking de Plataformas está descrito cada uma destas. Na Tabela 3 mostra-se as características chaves, tendo sido avaliadas cada uma delas entre “bom”, “medio” e “mau”.

Escolheu-se Django visto ter sido a que apresenta um melhor desempenho tendo em conta as nossas necessidades.

	Flask	Laravel	React	Django	Express
Popularidade	😐	😐	😊	😊	😊
Curva de aprendizagem	😊	😐	😐	😐	😊
Alcance	😐	😐	😊	😊	😊
Adequado para aplicações complexas	😞	😊	😐	😊	😞
Escalabilidade	😊	😞	😊	😊	😐
Competência para desenvolver um CMS (content management system)	😐	😊	😊	😊	😞

Tabela 3- Avaliação das diferentes frameworks

### Aplicações

Depois de realizarmos uma pesquisa encontramos alguns websites com funcionalidades interessantes, alguns destes sites foram recomendados pelos nossos inquiridos como sendo sites de referência no âmbito de grupos musicais e com um modelo que gostariam de ver implementado nos seus grupos e porventura mais desenvolvido. A nosso ver, as soluções que achamos mais interessantes foram os sites da Orquestra Metropolitana de Lisboa [OML20], Mahler Chamber Orchestra [Mahl20], Academia de Música de Lisboa [Arab21] e Artway Culture and Arts [Artw20].

Todas elas apresentam soluções com funcionalidades que pretendemos desenvolver, no entanto, a nosso ver, todas elas possuem algumas lacunas, ou estão incompletas de alguma forma, como tal, a nossa proposta passa por agregar as melhores funcionalidades de cada uma delas, e construir o nosso modelo, tentando também apostar em vertentes pouco desenvolvidas por estes modelos e implementar novas funcionalidades. Em todos os sites que procuramos, nenhum deles possuía um dos atributos que achamos mais interessantes desenvolver, que é a interação com as redes sociais de forma automática, sempre que se aproximar a data de um concerto e informar onde pode comprar os bilhetes para esse mesmo espetáculo.

Uma das funcionalidades nas quais vimos bastante potencial é apresentada abaixo no Facebook da orquestra Metropolitana de Lisboa, Figura 36 e Figura 37 , uma vez que nos dá a opção de marcação de presença nos próximos eventos, quando clicamos em “marcar agora” é nos dada uma lista dos eventos futuros. No entanto, esta marcação é feita apenas para o evento existente no Facebook, não tendo uma ligação direta com um site de venda de bilhetes. Esse seria um atributo a acrescentar ao nosso modelo.

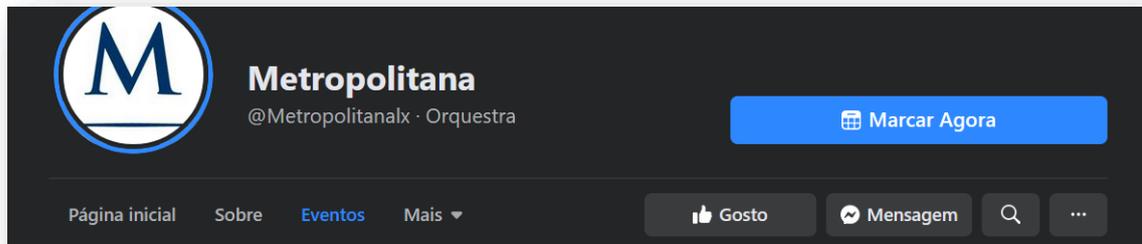


Figura 36- Secção da página do Facebook do grupo Metropolitana

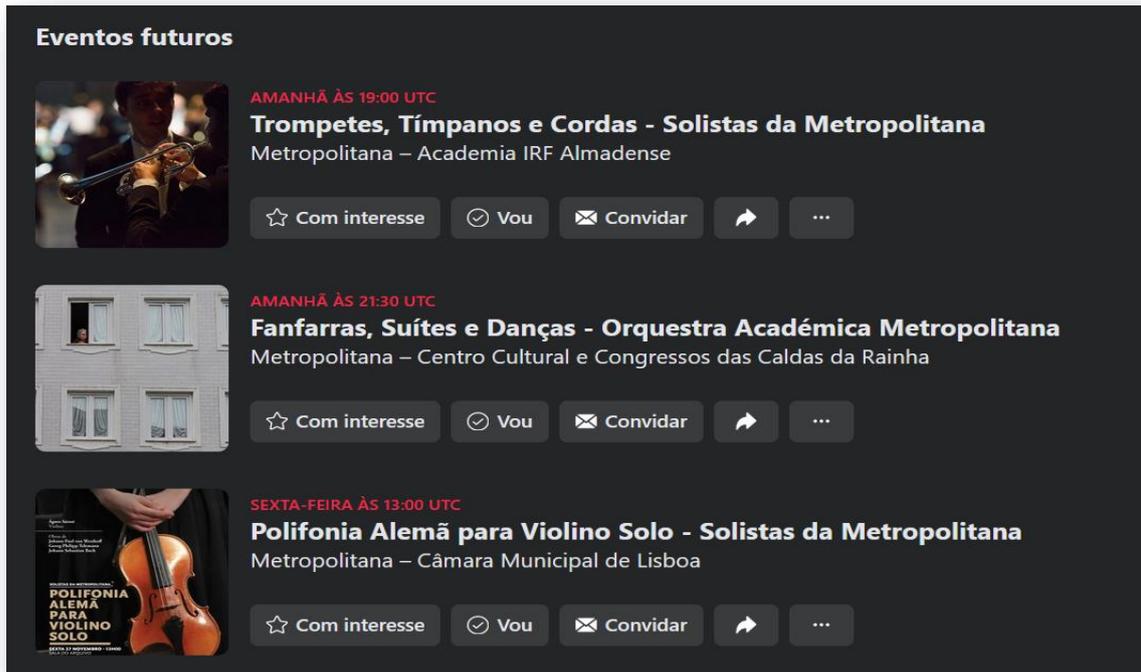


Figura 37- Secção da página do Facebook do grupo Metropolitana

Outra ideia que consideramos interessante é nos apresentada pela Artway sob a forma de um espaço no seu web site, Figura 38, dedicado ao apoio a novos talentos que queiram apresentar os seus projetos musicais de forma a proporcionar-lhes um suporte a nível estrutural para que possam desenvolver as suas capacidades e a sua criatividade.



Figura 38- Secção do web site do grupo Artway

Em suma, os inquéritos realizados demonstraram ser uma ferramenta muito útil na recolha de ideias e na compreensão das necessidades dos utilizadores. Todas as propostas apresentadas pelos inquiridos, de forma geral, continham soluções interessantes e às quais iremos ter em consideração ao longo do desenvolvimento deste projeto.

## 6 Método e Planeamento

No decorrer das semanas após a entrega do relatório intercalar, os membros do grupo tiveram *Sprints* semanais com o professor Lúcio Studer Ferreira de forma a poder acompanhar a execução do trabalho efetuado e delinear os próximos passos para a conclusão da entrega do primeiro relatório do trabalho em mãos, no dia 22/01/2021.

Na Tabela 4 apresenta-se o cronograma das atividades realizadas (a verde) e por realizar (a laranja). O trabalho realizado está dentro do prazo, tendo sido realizadas todas as tarefas propostas dentro do cronograma previsto. Foram adicionadas mais algumas tarefas, para detalhar o trabalho feito.

As Tarefas realizadas foram as seguintes:

- T1- Estudo do problema e análise de requisitos;
- T2- Levantamento da opinião pública sobre o tema tratado;
- T4– Estudo das ferramentas/tecnologias de desenvolvimento adotadas;
- T5– Levantamento e análise de requisitos
- T6– Identificação de Casos de uso e diagramas de atividade
- T7– Realização de Diagramas Entidade Relação
- T8– Estrutura da aplicação e *mockups*
- T9– Protótipo Funcional;
- T10– Validação da arquitetura;
- T13– Desenvolvimento da base de dados;

Tarefas por realizar:

- T11– *Test case* para validação da solução;
- T12–Desenvolvimento do website;
- T14– Testes e avaliação;
- T15– Escrita do relatório;
- T16– Demonstração de resultados;

No decorrer do projeto teremos sempre presente as entregas dos relatórios intercalares, obrigatórios, indicados, pelo corpo docente responsável pelo Trabalho Final de Curso.

Entregáveis:

- R1– Relatório Intercalar de 1º Semestre (27 de novembro de 2020)
- R2– Relatório intermedio (22 de janeiro de 2021)
- R3– Relatório intercalar de 2º Semestre (23 de abril de 2021)
- R4– Relatório final (25 de junho de 2021)

Tabela 4 - Cronograma das atividades realizadas (verde) e por realizar (laranja).

Tarefa	Nov	Dez	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set
T1	Verde	Verde	Verde								
T2	Verde	Verde	Verde								
T3		Verde	Verde								
T4		Verde	Verde								
T5		Verde	Verde								
T6		Verde	Verde								
T7		Verde	Verde								
T8		Verde	Verde								
T9				Verde	Verde	Verde					
T10				Verde	Verde	Verde					
T11							Verde	Verde	Verde	Verde	Verde
T12							Verde	Verde	Verde	Verde	Verde
T13					Verde						
T14							Verde	Verde	Verde	Verde	Verde
T15							Verde	Verde	Verde	Verde	Verde
T16							Verde	Verde	Verde	Verde	Verde

## 7 Resultados

De acordo com o analisado nos capítulos 2, 3 e 4, esta secção inclui resultados e validações da implementação das funcionalidades, especificadas nos diagramas de atividade presentes na Figura 4, Figura 5, Figura 6 e Figura 7, resultados obtidos da realização dos testes de usabilidade.

O objetivo principal dos testes foi avaliar e medir o nível de satisfação dos utilizadores na interação com a aplicação. Estes avaliaram os âmbitos da navegação, usabilidade, autenticação, fluidez, interação entre plataformas, interação entre front-end e backend e conteúdo. Outro dos objetivos é avaliar e aplicar melhorias identificando problemas de funcionalidade com base no feedback dos participantes. Nos testes estiveram envolvidos 12 participantes, 10 nas respostas ao questionário de satisfação e 2 em resposta aos testes de funcionalidade, tendo idades compreendidas entre os 23-56 anos. 58% da amostra foi masculina e os outros 42% feminina. Parte dos participantes eram estudantes universitários e outros trabalhadores, com diferentes graus de instrução e familiaridade com o mundo digital, representado assim uma amostra heterogenia e por isso bastante adequada para a realização dos testes. A nível de nacionalidades contámos com Portugueses e Brasileiros.

O material utilizado foi:

- **Anexo E - Questionário de Satisfação:** Um questionário com o intuito de perceber o grau de satisfação dos utilizadores.
- **Anexo C – Mockups:** Um manual com configurações necessárias para realizar os testes de usabilidade.
- **Anexo H - Guião de Testes:** Guião com passos a desempenhar pelo utilizador para testar as funcionalidades da plataforma.
- **1 Computador:** Computador com as configurações necessárias

Começando pelo questionário de satisfação, podemos observar os seus resultados no Anexo E - Questionário de Satisfação. Com os resultados obtidos podemos concluir que:

- A **navegação** pelo nosso site é feita de forma **intuitiva**. 80% afirmaram que sim e 20% foram neutros
- A **nomenclatura** utilizada encontra se **adequada**. 50% concordaram, 40% concordaram totalmente e 10% foram neutros
- As **cores** utilizadas foram **adequadas**. 50% concordaram, 40% concordaram totalmente e 10% foram neutros
- A **navegação** foi considerada **fácil** pela maioria. 30% foram neutros 40% concordaram e 30% concordaram totalmente.
- A **organização** do site foi bastante valorizada. 50% concordaram, 40% discordaram e 10% foram neutros.
- A **pesquisa** foi considerada **fácil** tendo tido 60% dos utilizadores a concordarem 30% a concordarem completamente e 10% foram neutros
- Na última questão perguntamos se com base nesta experiência acredita que as plataformas de agrupamentos musicais poderiam melhorar se seguissem este modelo. **80% estiveram de acordo** e 20 foram neutros.

Pedimos ainda aos utilizadores que escrevessem 1 aspeto positivo e negativo em relação á plataforma. Estes foram os resultados. Aspetos negativos na **Figura 39** e positivos na **Figura 40**.



**Figura 39-Aspetos negativos**



**Figura 40- Aspetos positivos**

Analizam-se de seguida os resultados aos testes de usabilidade, apresentados na Tabela 5. Ficamos bastante satisfeitos com o desempenho da nossa plataforma uma vez que os utilizadores conseguiram concluir todas as tarefas propostas com os resultados de acordo com o esperado. Contudo podemos concluir que os testes por nos realizados acabaram por se tornar demasiado exaustivos para quem os fazia, devido ao tempo despendido nos mesmos.

Tabela 5- Resultados dos testes de usabilidade

	Teste	Requisito testado	Tarefa Associada	Descrição	Resultado Esperado	Resultado Obtido
Autenticação	Registo com credenciais validas	RF1	Tarefa 3	A plataforma possui um sistema de registo e autenticação através de email password	Utilizador registado com sucesso	
	Configurações iniciais	RF2	Tarefa 1	A plataforma deve ter uma base de dados que guarda os dados dos utilizadores após o seu registo	Credenciais guardadas na base de dados	
	Configurações iniciais	RF4	Tarefa 1	A plataforma deve ter uma base de dados que guarda conteúdos vários	Dados guardados na base de dados	
Usabilidade	Inserir um Canvas pelo Admin	RF5	Tarefa 13	A plataforma deve ser capaz de congrega conteúdos para a newsletter, com base em eventos passados e futuros	Canvas inserido na Cloud	
	Executar o ficheiro "mail-sender2.py"	RF7	Tarefa 15	A plataforma deve ser capaz de enviar por email uma newsletter aos utilizadores que já se registaram na plataforma sobre noticias e eventos do grupo (cria uma página web, e envia uma imagem da página com um link)	Email enviado	

Tabela 5- Resultados dos testes de usabilidade

	Teste	Requisito testado	Tarefa Associada	Descrição	Resultado Esperado	Resultado Obtido
	Utilizar o Heroku Scheduler	RF12	Tarefa 16	A plataforma permite calendarizar a publicação automática, nas redes sociais, de informações sobre eventos ou notícias (dias antes de um evento acontecer).	Publicação feita de forma automática no Facebook através do Heroku Scheduler	
	Adicionar um evento preenchendo os campos Adicionar um evento preenchendo os campos	RF14	Tarefa 7 Tarefa 8	A plataforma deve ser capaz de gerar uma página que congrega todos os conteúdos relacionados com um evento (local, data, hora, programa, agrupamento, notas, fotos, vídeos)	Criar um evento pelo admin	
Conteúdo	Configurações iniciais	RF17	Tarefa 1	A plataforma deve ter um calendário atualizado de eventos futuros e passados, com links para páginas com mais detalhe sobre cada um destes (caso existam).	Página de eventos do site	

Tabela 5- Resultados dos testes de usabilidade

	Teste	Requisito testado	Tarefa Associada	Descrição	Resultado Esperado	Resultado Obtido
	Configurações iniciais	RF20	Tarefa 1	A plataforma deve ter uma biografia/Historia sobre o grupo/membros	Página de membros do site	
	Pesquisar com palavra existente	RF24	Tarefa 12	A plataforma deve ter um sistema de pesquisa de eventos e projetos através de palavras escritas pelo utilizador.	A lista de eventos que contem a palavra escrita	
	Pesquisar por data de eventos valida	RF26	Tarefa 10	A plataforma deve ter um sistema de pesquisa através das datas dos eventos.	A lista de eventos com data compreendida entre as datas selecionadas.	
	Inserir um Canvas pelo Admin Executar o ficheiro "post.py"	RF27	Tarefa 13 Tarefa 14	A plataforma deve possuir um sistema que permita gerar conteúdos publicáveis no Facebook e mailing list.	Canvas inserido e postado no Facebook	

## 8 Conclusões e trabalhos futuros

### Conclusão

Inicialmente as nossas motivações para este projeto prenderam-se na vontade de aprender novas tecnologias e no desafio de melhorar o website do agrupamento musical Quarteto Arabesco e também tornar a plataforma de gestão de conteúdos escalável a outros agrupamentos musicais. Acreditamos profundamente que esse objetivo foi alcançado muito devido aos resultados dos nossos testes de usabilidade e inquérito de satisfação.

Para o desenvolvimento deste projeto foram aprofundados os conceitos das unidades curriculares:

- **Sistemas de Informação Multimédia:** Utilizado no uso da linguagem Python e no tratamento de listas de texto com múltiplos elementos.
- **Base de Dados:** Utilizado no desenvolvimento da base de dados e nas suas interações.
- **Programação Web:** No desenvolvimento do front-end da plataforma
- **Interação Humano-Máquina:** Na realização dos testes, questionários e inquéritos.
- **Análise e Concessão de Sistemas, Engenharia de Software e Auditoria de Sistemas de Informação:** No levantamento de requisitos.

### Trabalhos Futuros

Em termos de trabalhos futuros, existem novas funcionalidades que podem ser implementadas neste projeto. Vamos começar por enumerar as melhorias:

- **Design de front-end:** Um dos aspetos que acreditamos que pode ser melhorado é o front-end, visto que foi um dos aspetos referidos pelos utilizadores. Graças à modularidade do Django será uma tarefa bastante fácil e independente
- **Publicitação no Instagram:** Acreditamos que seria interessante aplicar o nosso sistema de partilha de *canvas* de eventos também no Instagram.
- **Otimizar o sistema de partilha de *canvas*:** Tornar o sistema de partilha ainda mais automatizado. Por exemplo de cada vez que um evento é criado, ser também gerado um ficheiro PNG desse mesmo evento e enviado para a *cloud* automaticamente.
- **Melhorar o sistema de login:** Aumentar o nível de segurança e criar sistema de recuperação de password.
- **Melhorar o HTML utilizado na mailing list:** Tornar o HTML utilizado mais apelativo.

Em termos de novas funcionalidades:

- **Adicionar notícias ao website:** Adicionar uma secção de notícias sobre o agrupamento no website.
- **Integração de vídeos no website:** Poderia ser interessante integrar os vídeos ao próprio site.

## Bibliografia

- [AML20] Academia de Música de Lisboa, [www.academiamusicalisboa.com](http://www.academiamusicalisboa.com), acedido em Nov 2020
- [Arab21] Quarteto Arabesco, <http://www.quartetoarabesco.com/index.htm?quarteto>, acedido em Jan. 2021
- [Artw20] Artway, [www.artway.pt](http://www.artway.pt), acedido em Nov 2020
- [Bot21] Facebook bot, <https://dev.to/hhsm95/python-facebook-posts-scraper-with-requests-and-beautifulsoup4-273a>, acedido em Jan 2021
- [Djan21] Django, <https://medium.com/horadecodar/como-o-django-funciona-459184571a72>, acedido em Jan 2021
- [Expr21] Express, <https://www.treinaweb.com.br/blog/o-que-e-o-express-js/>, acedido em Jan 2021
- [Flas21] Flask Micro Web framework, <https://flask.palletsprojects.com/en/1.1.x/> acedido em Jan. 2021
- [Lara21] Laravel, <https://www.treinaweb.com.br/blog/o-que-e-laravel/>, acedido em Jan 2021
- [Mahl20] Mahler Chamber Orchestra, [www.mahlerchamber.com](http://www.mahlerchamber.com), acedido em Nov 2020
- [MVC20] MVC arquitetura, <https://www.portalgsti.com.br/2017/08/padrao-mvc-arquitetura-model-view-controller.html>, acedido em Nov 2020
- [OML20] Orquestra Metropolitana de Lisboa, [www.metropolitana.pt](http://www.metropolitana.pt), acedido em Nov 2020.
- [Reac21] React, <https://tableless.com.br/react-o-que-e-e-como-funciona-essa-ferramenta/>, acedido em Jan 2021
- [StOv21] Stackoverflow, *Technology web frameworks* survey, <https://insights.stackoverflow.com/survey/2020#technology-web-frameworks-all-respondents2>, acedido em Jan. 2021
- [UnLL20] Universidade Lusófona, [www.ulusofona.pt](http://www.ulusofona.pt), acedido em Nov. 2020.

i

## Anexo A – Questionários de Viabilidade

### A.1 Questionário a Membros de Grupos

# Plataforma de gestão de conteúdos musicais (Membros do grupo)

Este questionário enquadra-se no âmbito do trabalho final de curso da licenciatura em informática de gestão da universidade Lusófona intitulado: "Plataforma para gestão de agrupamento musical".

Destina-se a membros de grupos musicais, e ou gestores dos mesmos.

Tem como objectivo compreender a utilidade de uma plataforma de gestão de conteúdos e quais as razões para a sua utilização, bem como quais funcionalidades seriam interessantes implementar.

O problema proposto para este trabalho resume-se ao facto de que atualmente dispende-se muito tempo e esforço na tentativa de manter a notoriedade de um grupo musical nas redes sociais.

Como tal, a nossa motivação é fazer com que todos os grupos musicais estejam mais facilmente ligados aos seus fãs e que isso represente um esforço muito menor para os seus membros, tanto a nível de tempo, como de recursos.

O objetivo final passa por ter uma plataforma que seja capaz de agilizar todo o processo de publicação de conteúdos nas varias redes sociais (Facebook, Instagram, Youtube, Email) sendo este realizado de forma automática e periódica.

**Figura 41- Descrição do questionário a membros dos grupos.**

Possui um web site para o seu grupo musical?

Sim

Não

**Figura 42- Pergunta 1 do questionário a membros dos grupos.**

Se respondeu "Não" à questão anterior, indique-nos porque razão não possui uma plataforma de gestão de conteúdos (pode seleccionar varias opções)?

- Não sei como o fazer
- Não tenho tempo para o fazer
- Acho muito dispendioso
- Não acho que tenha utilidade
- Outro: \_\_\_\_\_

Figura 43- Pergunta 2 do questionário a membros dos grupos.

Classifique os seguintes conteúdos tendo em conta o grau de relevância, e que gostaria de ver destacados.

	Nada relevante	Pouco relevante	Relevante	Bastante relevante	Muito relevante
Agenda de eventos	<input type="radio"/>				
Conteúdos do grupo (concertos, fotos, etc)	<input type="radio"/>				
Notícias sobre o genero musical do grupo	<input type="radio"/>				
Notícias sobre o grupo	<input type="radio"/>				
História do grupo	<input type="radio"/>				
Contactos	<input type="radio"/>				

Figura 44- Pergunta 3 do questionário a membros dos grupos.

Consideraria a hipótese de adoptar uma plataforma de gestão de conteúdos se esta lhe permitisse fazer partilhas de conteúdos e postagens nas redes sociais de forma quase automática e que desta forma não necessitasse de muita manutenção nem tempo despendido?

- Sim
- Não
- Talvez

**Figura 45- Pergunta 4 do questionário a membros dos grupos.**

Acha interessante que a plataforma tenha a opção de criação de publicações automáticas de anuncio aos eventos do grupo, dias antes de eles acontecerem?

- Sim
- Não
- Indiferente

**Figura 46- Pergunta 5 do questionário a membros dos grupos.**

Classifique tendo em conta o grau de relevancia para o seu agrupamento as seguintes opções

	Nada relevante	Pouco relevante	Relevante	Bastante relevante	Muito relevante
Facebook	<input type="radio"/>				
Instagram	<input type="radio"/>				
Youtube	<input type="radio"/>				
Website	<input type="radio"/>				

Figura 47- Pergunta 6 do questionário a membros dos grupos.

Acha interessante que a plataforma tenha opção de envio de emails automáticos de anuncio aos eventos do grupo, dias antes de eles acontecerem?

Sim

Não

Indiferente

Figura 48- Pergunta 7 do questionário a membros dos grupos.

Acha interessante que a plataforma tenha uma newsletter, em que, de forma periodica, faça uma síntese das atividades passadas e planeadas num determinado período de tempo?

- Sim
- Não
- Indiferente

**Figura 49- Pergunta 8 do questionário a membros dos grupos.**

Qual acha que deveria ser a periodicidade dessa newsletter?

- Mensal
- Bimensal
- Trimestral
- Semestral

**Figura 50- Pergunta 9 do questionário a membros dos grupos.**

Quanto tempo gasta em media para tratar de todo o processo de manutenção/divulgação do seu grupo nas redes sociais e ou web site?

- Menos de 1 hora por mês
- 1 a 2 horas por mês
- 3 a 4 horas por mês
- mais de 4 horas por mês

**Figura 51- Pergunta 10 do questionário a membros dos grupos.**

Quanto tempo acha que seria ideal despende para a manutenção/divulgação do seu grupo nas redes sociais e ou web site?

- Menos de 1 hora por mês
- 1 a 2 horas por mês
- 3 a 4 horas por mês
- Mais de 4 horas por mês

**Figura 52- Pergunta 11 do questionário a membros dos grupos.**

Acha interessante a ideia de agregar todos os conteúdos presentes nas redes sociais no website?

- Sim
- Não
- Indiferente

**Figura 53- Pergunta 12 do questionário a membros dos grupos.**

Acha interessante a ideia de disponibilizar o calendário de eventos organizado no website?

- Sim
- Não
- Indiferente

**Figura 54- Pergunta 13 do questionário a membros dos grupos.**

Acha interessante a ideia de existir informação detalhada e organizada com todos os conteúdos envolventes de eventos passados no website( fotos, videos, artigos de jornal, notas de concerto)?

- Sim
- Não
- Indiferente

**Figura 55- Pergunta 14 do questionário a membros dos grupos.**

Sugestão de websites que conheça que considera bem feitos e eficientes (basta indicar nome do agrupamento, não precisa url)

Sua resposta

---

**Figura 56- Pergunta 15 do questionário a membros dos grupos.**

Indique-nos outros conteúdos ou funcionalidades que gostaria de ver implementados em web pages de grupos musicais?

Sua resposta

---

**Figura 57- Pergunta 16 do questionário a membros dos grupos.**

A.2 Questionário a Ouvintes

## Plataforma de gestão de conteúdos musicais

Este questionário enquadra-se no âmbito do trabalho final de curso da licenciatura em informática de gestão da universidade Lusófona intitulado: "Plataforma para gestão de agrupamento musical".

Destina-se a visitantes de web sites de grupos musicais.

Tem como objectivo compreender a utilidade de uma plataforma de gestão de conteúdos e quais as razões para a sua utilização, bem como quais funcionalidades seriam interessantes implementar.

O problema proposto para este trabalho resume-se ao facto de que atualmente dispende-se muito tempo e esforço na tentativa de manter a notoriedade de um grupo musical nas redes sociais.

Como tal, o nossa motivação é fazer com que todos os grupos musicais estejam mais facilmente ligados aos seus fãs e que isso represente um esforço muito menor para os seus membros, tanto a nível de tempo, como de recursos.

O objetivo final passa por ter uma plataforma que seja capaz de agilizar todo o processo de publicação de conteúdos nas varias redes sociais (Facebook, Instagram, Youtube, Email) sendo este realizado de forma automática e periódica.

Figura 58- Descrição do questionário a visitantes de websites de grupos musicais.

Com que frequência visita web sites das suas bandas ou grupos musicais favoritos (redes sociais não incluídas) ? \*

	1	2	3	4	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sempre que possivel

Figura 59- Pergunta1 do questionário a visitantes de websites de grupos musicais.

Quando visita páginas de grupos musicais, que tipo de conteúdos procura? Pode seleccionar varias opções. \*

- Datas dos proximos eventos
- Conteudos do grupo (musicas, concertos, fotografias, etc)
- Historia do grupo
- Contactos do grupo
- Noticias sobre o grupo
- Outro: \_\_\_\_\_

Figura 60- Pergunta2 do questionário a visitantes de websites de grupos musicais.

Acha relevante o web site conter os seguintes conteúdos ? \*

- Imagens
- Videos
- Audios (excertos de concertos, repertorio, etc)

Figura 61- Pergunta3 do questionário a visitantes de websites de grupos musicais.

Com que frequência utiliza as redes sociais, para ver os seus grupos favoritos?

- |       |                       |                       |                       |                       |        |
|-------|-----------------------|-----------------------|-----------------------|-----------------------|--------|
|       | 1                     | 2                     | 3                     | 4                     |        |
| Nunca | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Sempre |

Figura 62- Pergunta4 do questionário a visitantes de websites de grupos musicais.

Acha que seria relevante receber notificações da aproximação de eventos relacionados com um grupo musical À sua escolha? \*

- Nada relevante
- Relevante
- Muito relevante

Figura 63- Pergunta5 do questionário a visitantes de websites de grupos musicais.

Classifique os seguintes conteúdos tendo em conta o grau de relevância, e que gostaria de ver destacados. \*

	Nada relevante	Pouco relevante	Relevante	Bastante relevante	Muito relevante
Agenda de eventos	<input type="radio"/>				
Conteúdos do grupo (concertos, fotos, etc)	<input type="radio"/>				
Notícias sobre o genero musical do grupo	<input type="radio"/>				
Notícias sobre o grupo	<input type="radio"/>				
História do grupo	<input type="radio"/>				
Contactos	<input type="radio"/>				

Figura 64- Pergunta6 do questionário a visitantes de websites de grupos musicais.

Indique-nos outros conteúdos ou funcionalidades que gostaria de ver implementados em web sites de grupos musicais?

Sua resposta

---

**Figura 65- Pergunta7 do questionário a visitantes de websites de grupos musicais.**

## Anexo B – Análise das Respostas aos Questionários de Viabilidade

### B.1 Questionário a membros de grupos musicais

A primeira e segunda pergunta permitiram avaliar qual a percentagem de membros que possuem um web site próprio e descobrir as razões pelas quais os restantes membros não optam por essa solução, e como podemos constatar, a principal razão reside na falta de tempo para o seu desenvolvimento, logo seguida de não terem conhecimentos para o fazer. Tudo isto encontra se representado nas Figura 66 e Figura 67

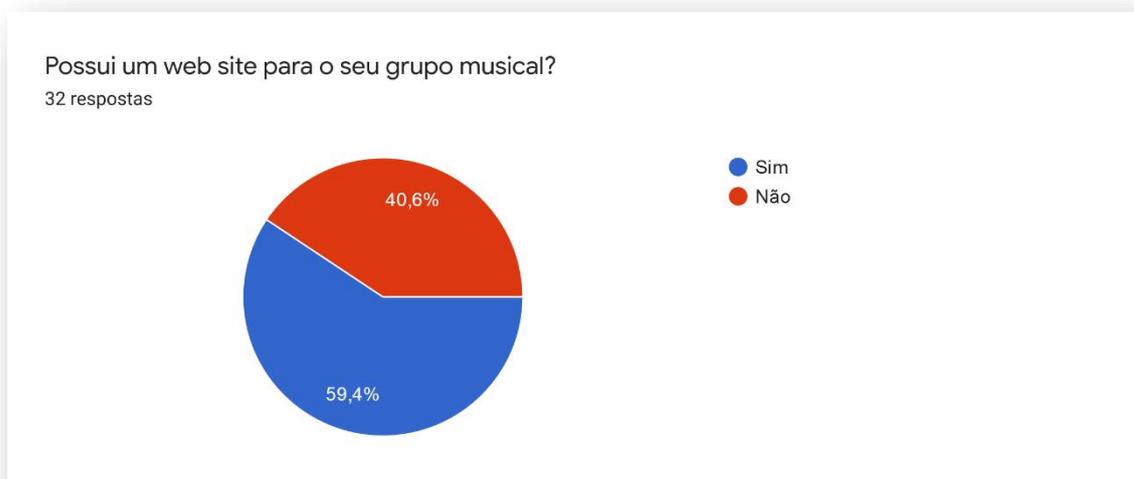


Figura 66- Resposta à questão 1 do questionário dos membros de grupos musicais



Figura 67- Resposta à questão 2 do questionário dos membros de grupos musicais

Na Figura 68 e Figura 69 foi perguntado qual será a importância de determinados conteúdos, redes sociais e web site, respetivamente. No que diz respeito aos conteúdos, os mais destacados foram os contactos, com 20 respostas de “muito relevante”, seguida dos conteúdos do grupo e agenda de eventos, com 15 e 14 respostas classificadas como “muito relevante”,

respetivamente. No âmbito das redes sociais e do web site, todas são consideradas muito relevantes, sendo o Instagram a menos relevante das inquiridas. Como isto podemos concluir a importância para os membros de estar presente no maior número de plataformas possíveis com uma oferta de conteúdos mais centrada no que é essencial, tal como são os contactos do grupo, a sua agenda de eventos e os seus conteúdos.

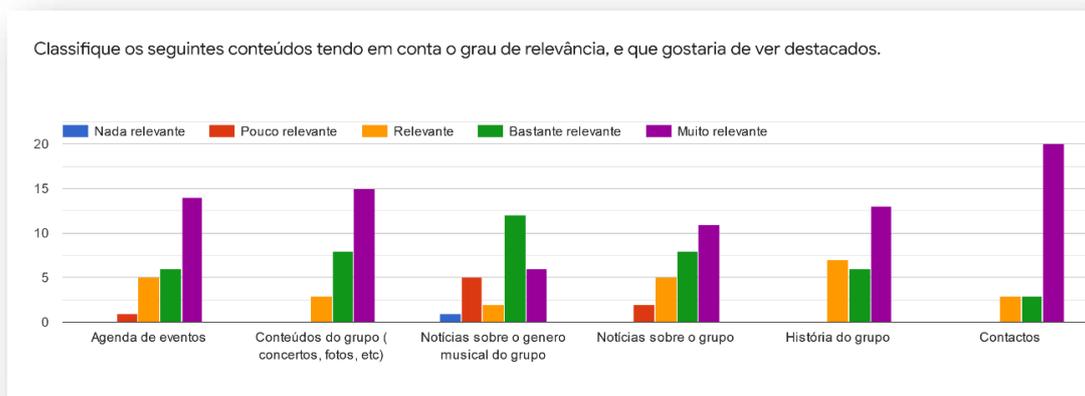


Figura 68- Resposta à questão 3 do questionário dos membros de grupos musicais

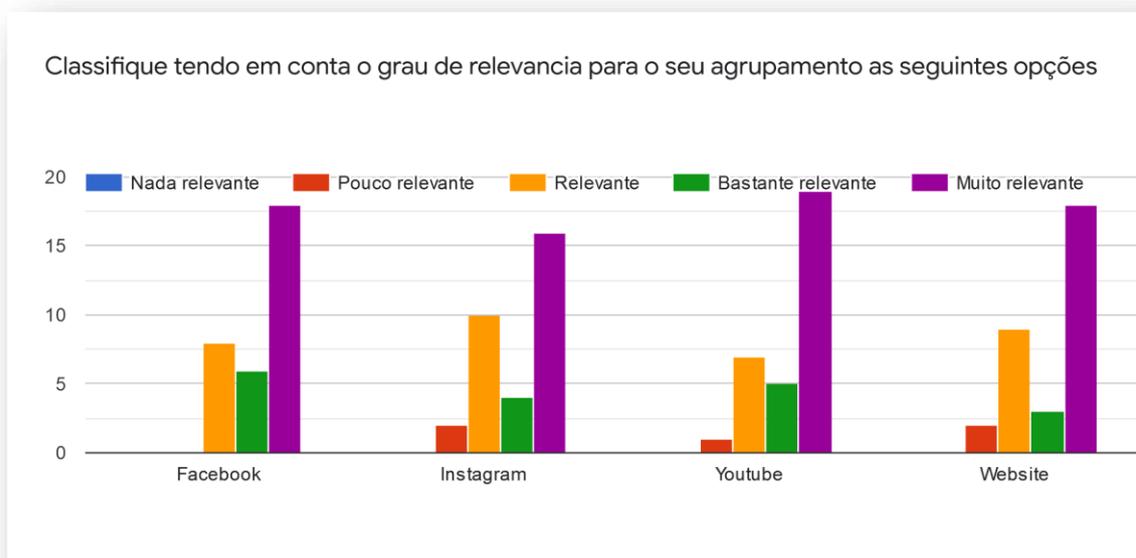


Figura 69- Resposta à questão 4 do questionário dos membros de grupos musicais

Analisando os resultados podemos concluir que grande parte das ideias inicialmente propostas por nós tiveram uma enorme taxa de aceitação como constatamos nas Figura 70 até à Figura 76. Como podemos observar em todas elas, a resposta “sim” tem uma percentagem perto dos 100%, chegando mesmo a este valor em algumas delas.

Consideraria a hipótese de adoptar uma plataforma de gestão de conteúdos se esta lhe permitisse fazer partilhas de conteúdos e postagens nas redes sociais de forma quase automática e que desta forma não necessitasse de muita manutenção nem tempo despendido?

32 respostas

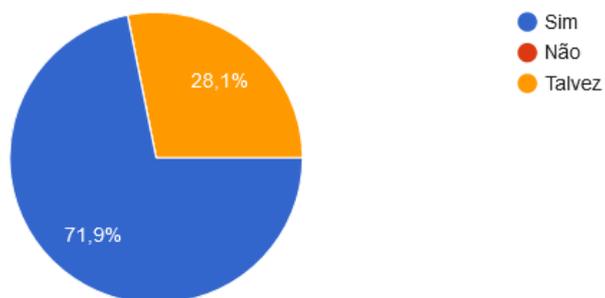


Figura 70- Resposta à questão 5 do questionário dos membros de grupos musicais

Acha interessante que a plataforma tenha a opção de criação de publicações automáticas de anuncio aos eventos do grupo, dias antes de eles acontecerem?

31 respostas

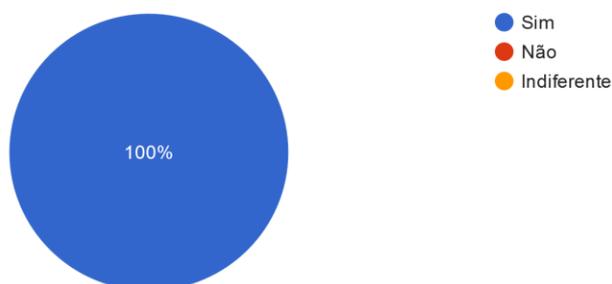


Figura 71- Resposta à questão 6 do questionário dos membros de grupos musicais

Acha interessante que a plataforma tenha opção de envio de emails automáticos de anuncio aos eventos do grupo, dias antes de eles acontecerem?

32 respostas

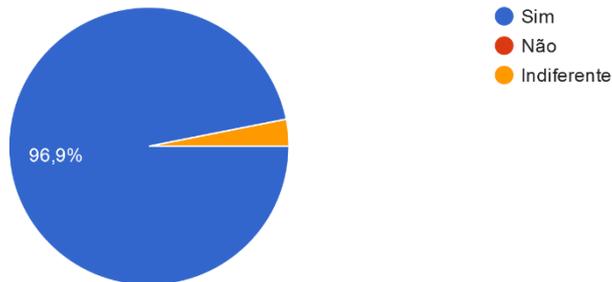


Figura 72- Resposta à questão 7 do questionário dos membros de grupos musicais

Acha interessante que a plataforma tenha uma newsletter, em que, de forma periodica, faça uma síntese das atividades passadas e planeadas num determinado periodo de tempo?

30 respostas

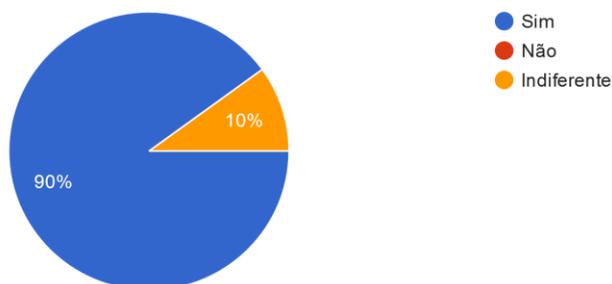


Figura 73- Resposta à questão 8 do questionário dos membros de grupos musicais

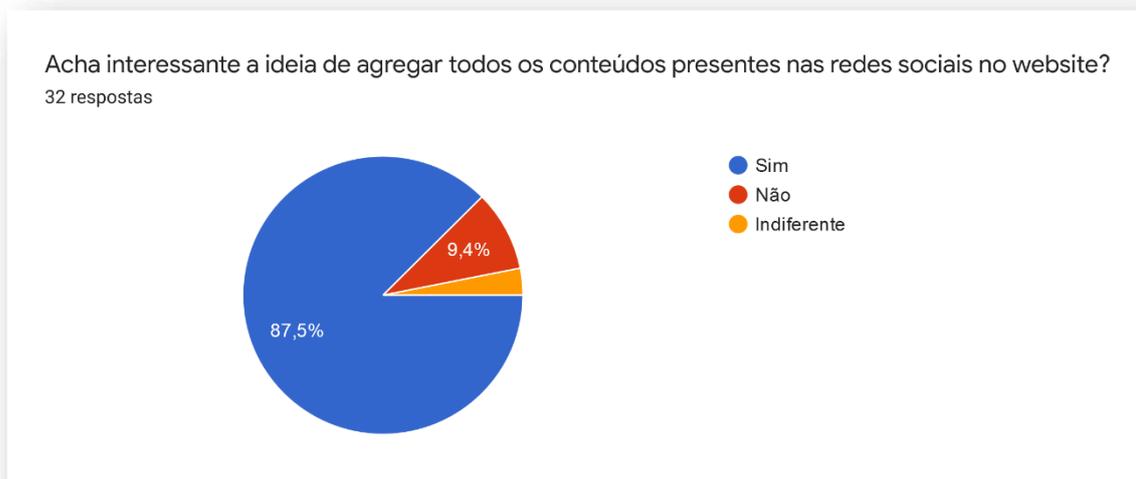


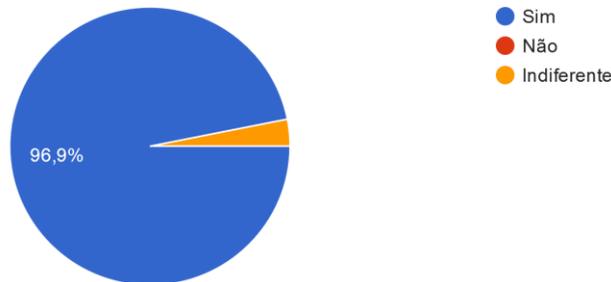
Figura 74- Resposta à questão 9 do questionário dos membros de grupos musicais



Figura 75- Resposta à questão 10 do questionário dos membros de grupos musicais

Acha interessante a ideia de existir informação detalhada e organizada com todos os conteúdos envolventes de eventos passados no website( fotos, videos, artigos de jornal, notas de concerto)?

32 respostas

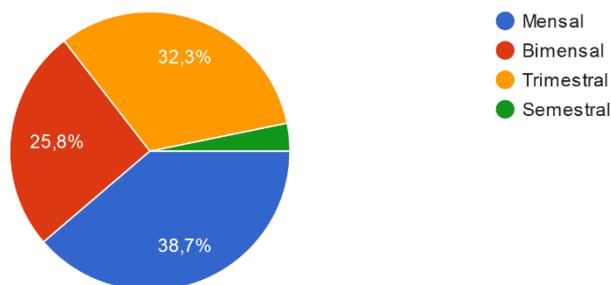


**Figura 76- Resposta à questão 11 do questionário dos membros de grupos musicais**

Na Figura 77 perguntamos sobre qual seria a periodicidade ideal para uma newsletter sintetizada com informações sobre um determinado período, enviada por email. As respostas a esta pergunta foram bastante dispare, ainda assim podemos concluir que a periodicidade da mesma nunca deverá ultrapassar um trimestre.

Qual acha que deveria ser a periodicidade dessa newsletter?

31 respostas



**Figura 77- Resposta à questão 12 do questionário dos membros de grupos musicais**

Na Figura 78 e Figura 79 foi perguntado quanto tempo gasta em média mensalmente na manutenção e divulgação do grupo nas redes sociais e ou web site e qual acha que seria o tempo ideal para realizar estas tarefas, respetivamente. Na pergunta 10 as respostas foram distribuídas em dois grandes grupos opostos, um deles respondeu que gasta menos de 1 hora, outro gasta mais de 4 horas.

Quanto tempo gasta em media para tratar de todo o processo de manutenção/divulgação do seu grupo nas redes sociais e ou web site?

31 respostas

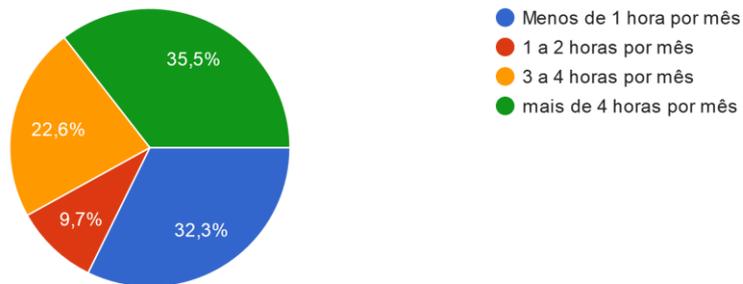


Figura 78- Resposta à questão 13 do questionário dos membros de grupos musicais

Quanto tempo acha que seria ideal despendere para a manutenção/divulgação do seu grupo nas redes sociais e ou web site?

32 respostas

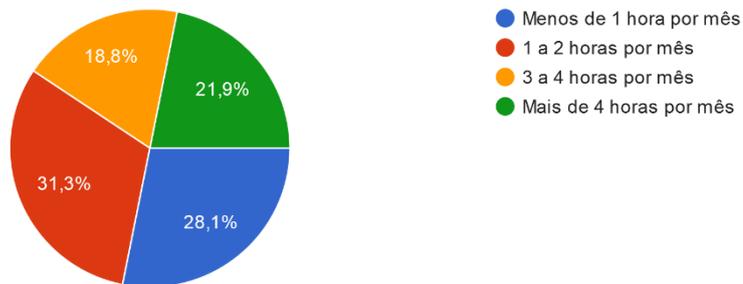


Figura 79- Resposta à questão 14 do questionário dos membros de grupos musicais

## B.2 Questionário aos ouvintes

De seguida iremos analisar as respostas dos ouvintes. Nas respostas dos ouvintes podemos concluir que a maioria não visita com frequência web sites de grupos que ouve frequentemente, mas quando perguntamos em relação as redes sociais a percentagem já aumenta substancialmente, como podemos observar na Figura 80 e Figura 81.

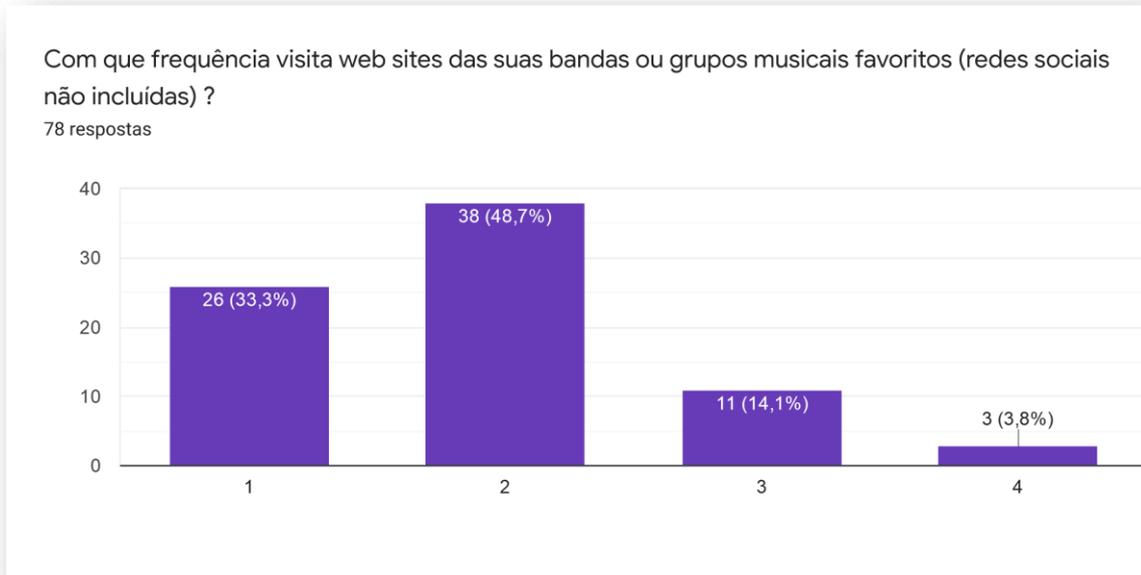


Figura 80- Resposta à questão 1 do questionário dos ouvintes



Figura 81- Resposta à questão 2 do questionário dos ouvintes

Na Figura 82, quando perguntámos que conteúdos procura mais quando visita um web site, as respostas com maior percentagem foram as datas dos eventos, conteúdos do grupo, história do grupo e notícias, por esta mesma ordem. Aqui encontramos semelhanças com as respostas dadas pelos membros do grupo quando perguntados a cerca de que funcionalidades acham mais relevantes, como tal estes são componentes que iremos atribuir especial atenção.

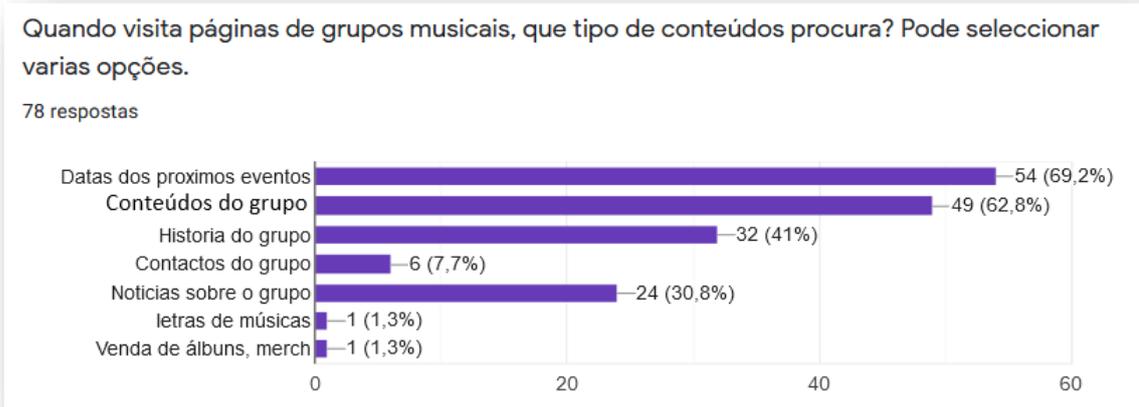


Figura 82- Resposta à questão 3 do questionário dos ouvintes

Podemos concluir pela Figura 83 que todos os conteúdos assinalados têm uma grande importância para ouvintes visto que todas elas tiveram uma percentagem bastante alta, no entanto a mais importante foram os áudios nos quais incluem excertos de concertos e exemplos de repertório.

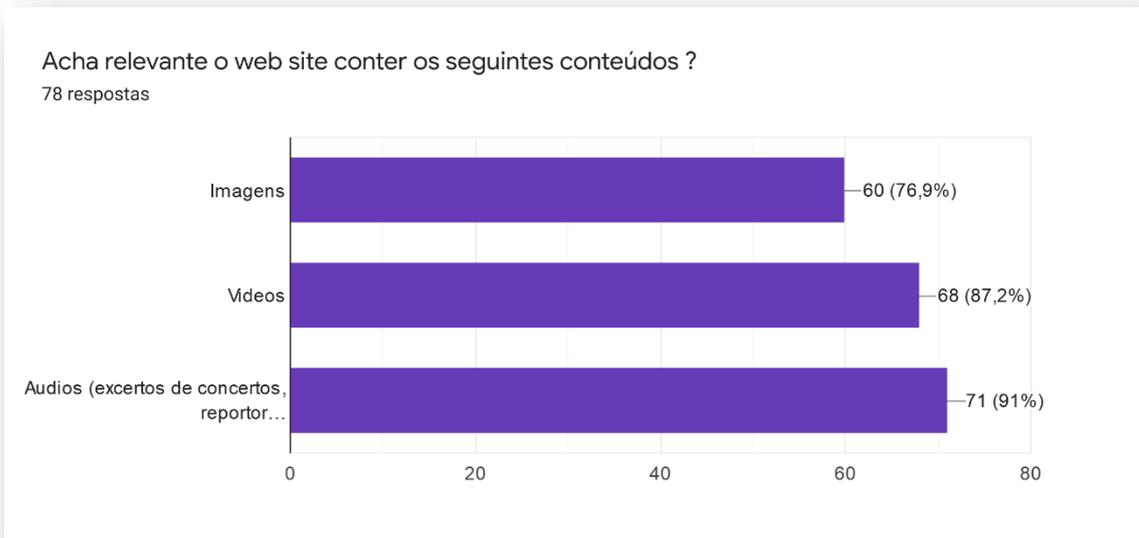


Figura 83- Resposta à questão 4 do questionário dos ouvintes

Na Figura 84 verificamos que o publico em geral acede com alguma frequência às redes sociais dos seus grupos favoritos.



Figura 84- Resposta à questão 5 do questionário dos ouvintes

Como podemos observar na Figura 85 a esmagadora maioria dos ouvintes acha relevante poder receber notificações sempre que um evento de um grupo musical da sua preferência se aproxima. Está é mais uma das funcionalidades que queremos fortemente incluir no nosso modelo, visto que é de grande importância para todos os intervenientes.

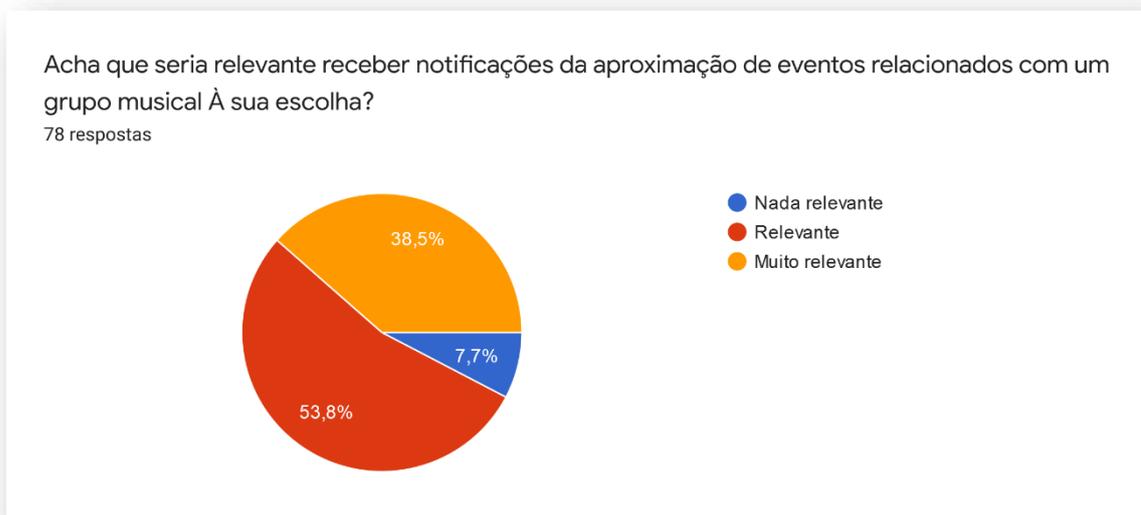


Figura 85- Resposta à questão 6 do questionário dos ouvintes

Na Figura 86 todos os conteúdos foram considerados como relevantes pela maioria, á exceção dos contactos, onde a distribuição foi muito idêntica, tendo ganho o “pouco relevante” por 1 ponto.

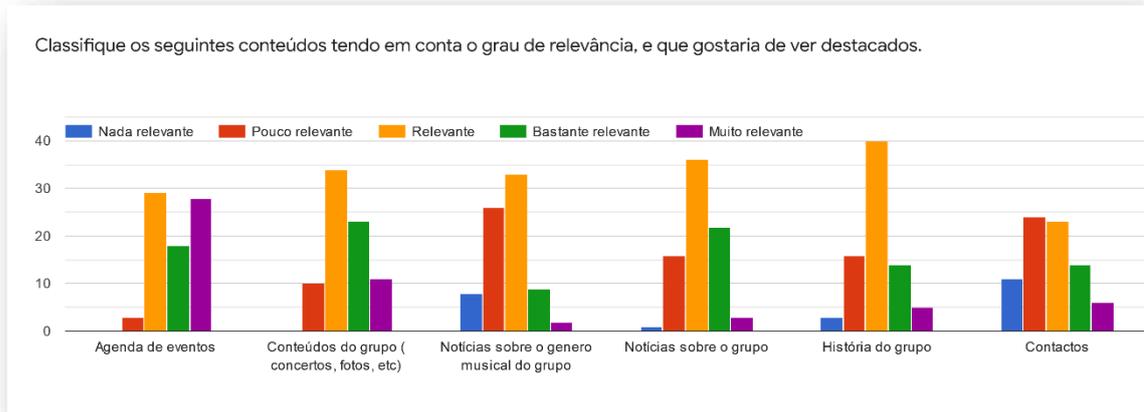


Figura 86- Resposta à questão 7 do questionário dos ouvintes

## Anexo C – Mockups

Apresentamos, de seguida, os *Mockups* da proposta de *Front-End* da Plataforma *WEB*.

Na Figura 87 podemos verificar a página inicial com uma barra de navegação, no topo da página, que contém as várias páginas da Plataforma (História, Grupo, etc.). A barra tem também acesso ao Perfil do *User* e uma barra de pesquisa. Propomos também no canto inferior direito a apresentação de um leitor de música que reproduz música do grupo durante a visita da Plataforma. No centro podemos apresentar imagens e pequenos textos de introdução ao grupo.

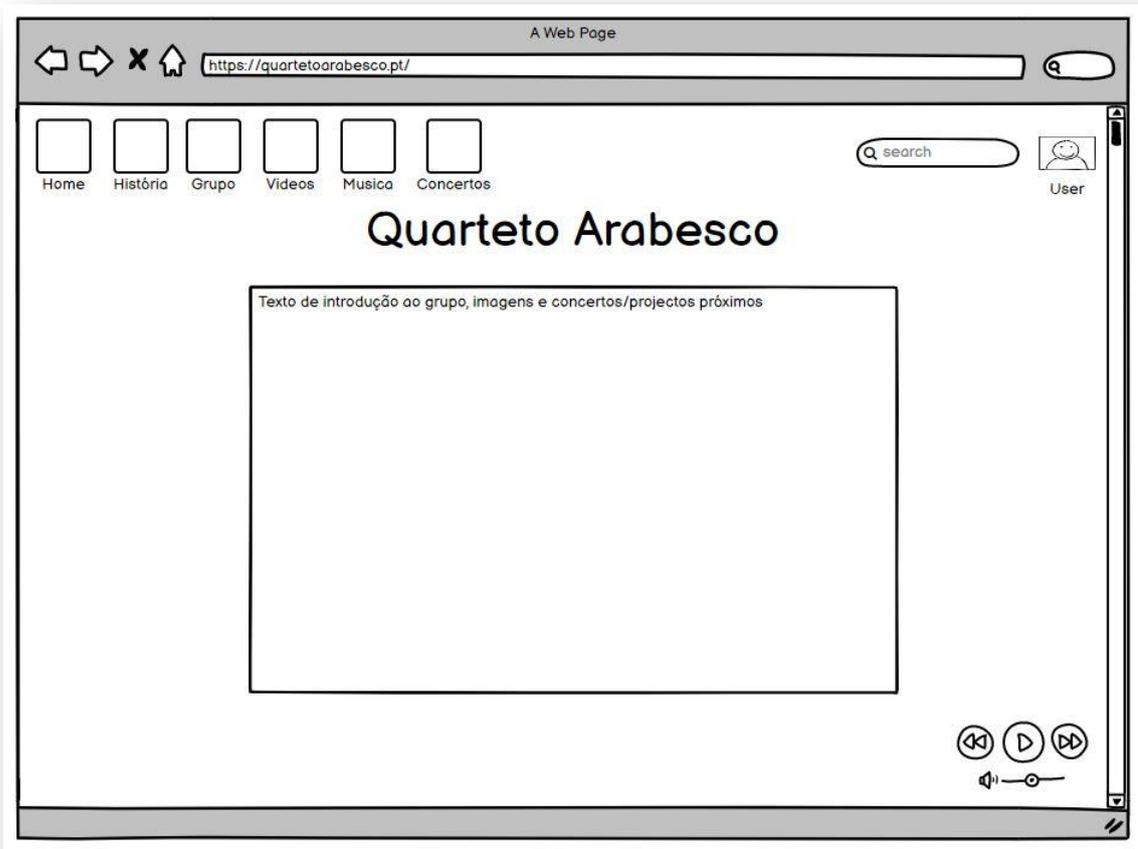


Figura 87- Página inicial

No caso do *User* ter iniciado a página, mas ainda não ter feito o *Log In*. Pode sempre o fazer, clicando no botão do *User*, onde, de seguida aparece um *pop-up* que demonstra tanto o *Log In* como Registo. Este exemplo encontra-se apresentado na Figura 88 .

No caso do *User* não estiver registado na plataforma pode clicar na área de registo onde é encaminhado para a página de registo, onde terá de preencher alguns campos e escolher as opções que pretende ter acesso. Este exemplo encontra-se apresentado na Figura 89

Na Figura 90 apresentamos a proposta de todas as páginas, exceto a *Home*(Principal). As páginas terão uma *Navigation Bar*, apenas com uma opção de pesquisa, um de acesso ao perfil do utilizador e um ícone que apresenta uma lista das páginas da Plataforma.

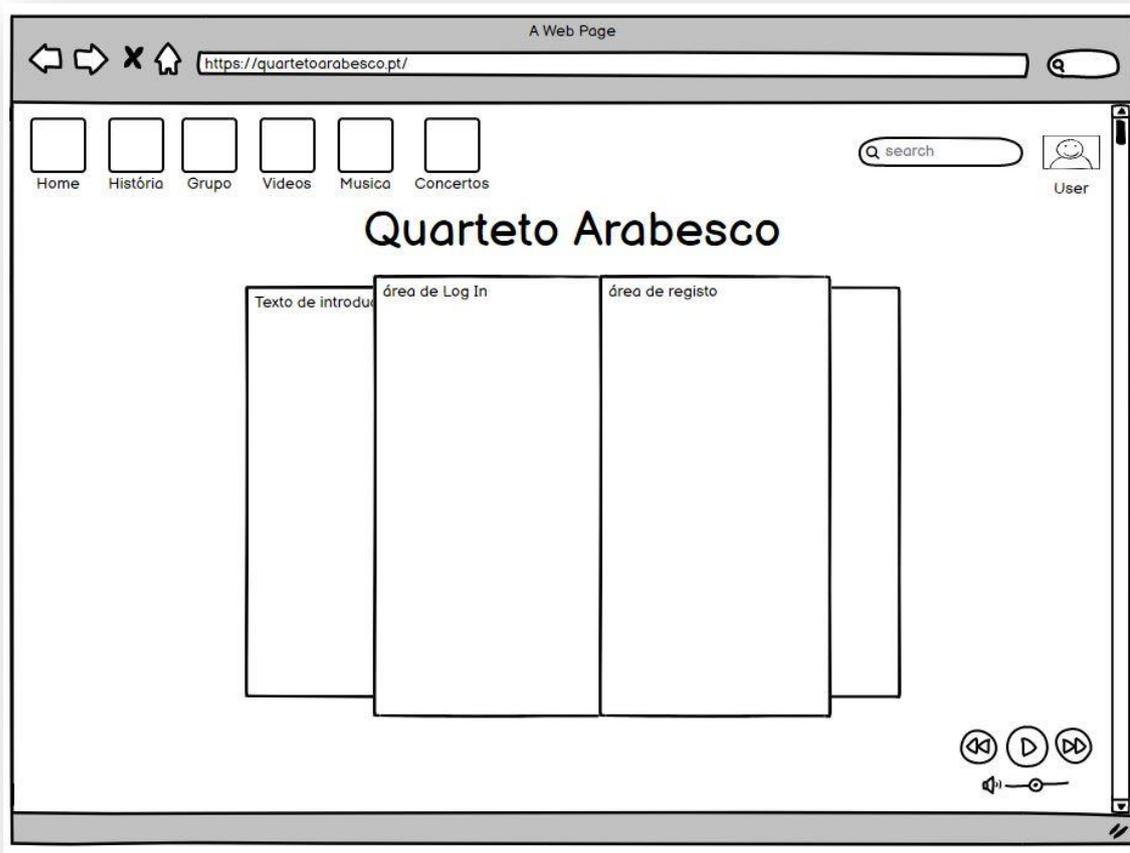


Figura 88- Página inicial na área de registo



Figura 89- Página de registo



Figura 90- Página secundária

## Anexo D - Benchmarking de Plataformas

Existem um conjunto vasto de *frameworks* e de abordagens possíveis para o desenvolvimento da nossa plataforma web, todas elas com vantagens e desvantagens entre si. Abaixo realizamos um comparativo entre a *framework* Django e as restantes *frameworks* presentes no mercado.

No ranking da *frameworks* mais populares do famoso *Stack Overflow* [StOv21], representado na Figura 91, podemos constatar que o Django tem uma classificação semelhante às outras *frameworks* de *back-end* (ASP.NET, Express, Spring, Flask, Laravel). Podemos observar atualmente uma maior tendência para as *frameworks* em *front-end*, muito devido ao facto de utilizarem o JavaScript como linguagem e também pelo facto de que este tipo de abordagem não sobrecarrega o servidor.

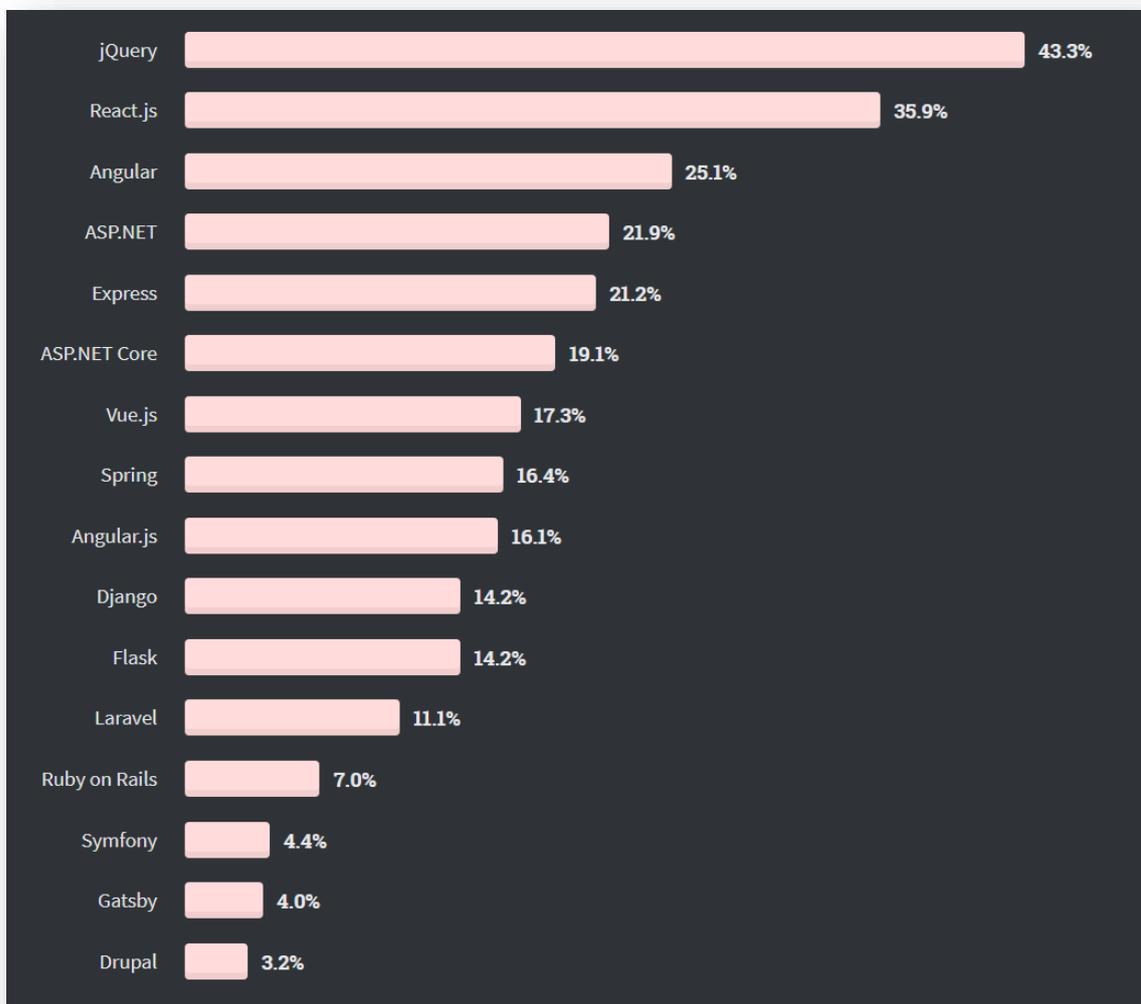


Figura 91- Gráfico de popularidade de Web frameworks

Nas subsecções seguintes, discute-se comparativamente algumas das principais *frameworks*.

## D.1 React

React [Reac21] é uma biblioteca de JavaScript, *open source*, Solução de *front-end* usada para construir *user interfaces* nomeadamente para aplicações de página única. O seu principal objetivo é ser rápida, escalável e simples, podendo ainda ser usada em combinação com outras bibliotecas ou *frameworks* de JavaScript, como o Angular js. Deste modo, a biblioteca React.js é usada para lidar com a camada de visualização para aplicações web e móveis, e permite também criar componentes de *user interface* reutilizáveis. Como tal, podemos afirmar que React é mais vantajoso quando se trata de aplicações mais leves e com um grau de complexidade menor, uma vez que se trata de uma biblioteca de javascript, não se tratando de uma *framework* como o Django que possui um conjunto de recursos que o permitem ser considerado *full stack*.

## D.2 Flask

*Flask* [Flas21] é uma *framework open-source* que foi lançada em 2010 que utiliza a linguagem *python* para o seu desenvolvimento. Devido á sua abordagem de design minimalista, faz com que seja mais rápida em termos de desempenho quando comparada com o Django. Pode suportar centenas de consultas por segundo sem sofrer um impacto negativo no seu desempenho.

Apesar do Flask e do Django possuírem características semelhantes, a sua aplicabilidade é um pouco distinta. Podemos dizer que o Django é mais indicado para aplicações mais complexas, uma vez que já inclui várias aplicações incluídas de raiz para além de possuir uma comunidade gigantesca. Por outro lado, o Flask é mais utilizado para pequenas aplicações e microserviços com design menos elaborado e com menores tempos de resposta.

Posto isto, consideramos o Django como sendo a *framework* mais indicada para o desenvolvimento da nossa plataforma, visto que esta apresenta um grau de complexidade elevado.

F.1.3-LaravelÉ uma *framework open source* que utiliza a linguagem PHP. O Laravel [Lara21] possui um sistema de templates (Blade) que facilita a criação das componentes visuais da plataforma (páginas HTML). Os recursos que mais se destacam no Blade são a herança de layouts e o sistema de *tags*. O Laravel, á semelhança do Django, utiliza o mesmo sistema de *MVC*.

Django e Laravel são *frameworks* distintas uma da outra, como principais diferenças podemos apontar o facto de o Laravel utilizar uma linguagem *PHP* enquanto que o Django utiliza *python*. Ao nível da escalabilidade o Laravel possui mais limitações face ao Django, sendo esta uma vantagem importante na hora de escolher entre as duas, uma vez que o nosso objetivo é desenvolver uma plataforma de gestão de um grupo musical escalável a qualquer grupo musical.

Em termos de performance e segurança existem também diferenças, ao nível da performance, o Django oferece mais rapidez de execução de processos, muito devido ao facto de o Laravel utilizar uma linguagem *PHP*, que por si só, é mais lenta que o *python*, mas também devido ao número elevado de *features* que possui. Ao nível da segurança, ambas apresentam soluções bastante viáveis, não existindo diferenças significativas no seu desempenho a esse nível.

### D.3 Express

Express [Expr21] define se como sendo uma *microframework* rápida e minimalista, que oferece soluções robustas para servidores *HTTP*, tornando-a uma solução viável para aplicações individuais, websites e *API's*. Em termos de comparação com o Django, podemos classificar o Express como sendo uma *microframework* voltada para a componente de *back-end*, como tal achamos mais viável a alternativa *fullstack* do Django

## Anexo E - Questionário de Satisfação

### Questionário de Satisfação (Grupo Musical Quarteto Arabesco)

Este questionário procura medir o grau de satisfação na interação com a plataforma gestão de conteúdos do grupo musical Quarteto Arabesco.

Este questionário é composto por 7 questões cotadas de 1 a 5 pontos de acordo com a sua concordância ou discordância.

Figura 92-Descrição do questionário de satisfação

Nesta plataforma a navegação é intuitiva? \*

1 2 3 4 5

Discordo totalmente      Concordo totalmente

Figura 93- Pergunta 1 do questionário de satisfação

Os nomes dos campos do site são adequados \*

1 2 3 4 5

Discordo totalmente      Concordo totalmente

Figura 94- Pergunta 2 do questionário de satisfação

As cores são adequadas? \*

1 2 3 4 5

Discordo totalmente      Concordo totalmente

Figura 95- Pergunta 3 do questionário de satisfação

A navegação é fácil?

1 2 3 4 5

Discordo totalmente      Concordo totalmente

Figura 96- Pergunta 4 do questionário de satisfação

A forma como a informação está organizada na página de eventos e projetos está adequada? \*

1 2 3 4 5

Discordo totalmente      Concordo totalmente

Figura 97- Pergunta 5 do questionário de satisfação

A pesquisa de eventos e projetos é de fácil utilização?

1      2      3      4      5

Discordo totalmente                                    Concordo totalmente

Figura 98-Pergunta 6 do questionário de satisfação

Com base na sua experiência, acredita que os web sites agrupamento musicais serão mais apelativos? \*

1      2      3      4      5

Discordo totalmente                                    Concordo totalmente

Figura 99- Pergunta 7 do questionário de satisfação

## Anexo F - Respostas ao questionário de satisfação

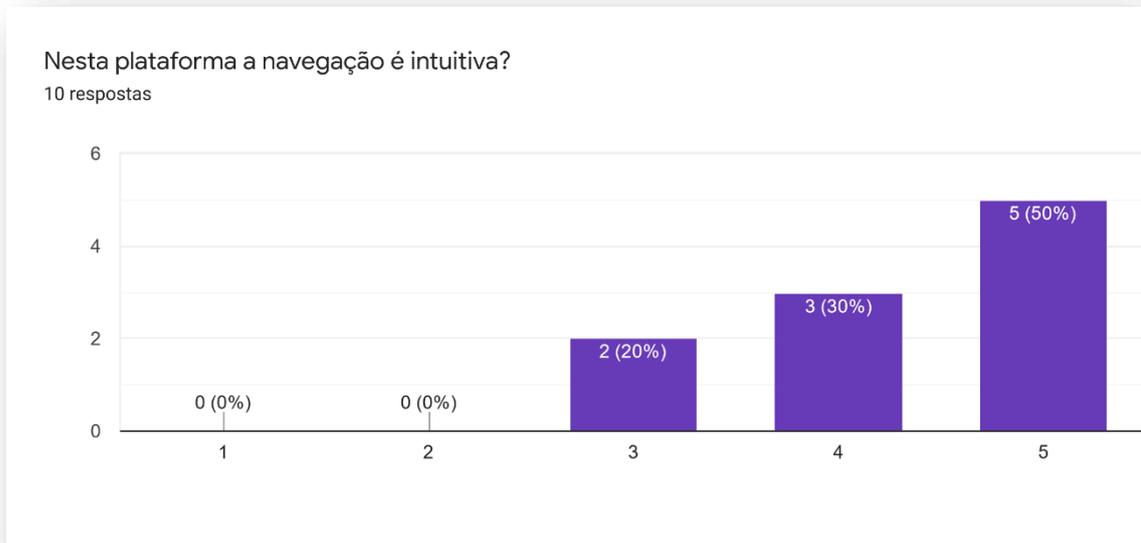


Figura 100- Resposta à pergunta 1 do questionário de satisfação

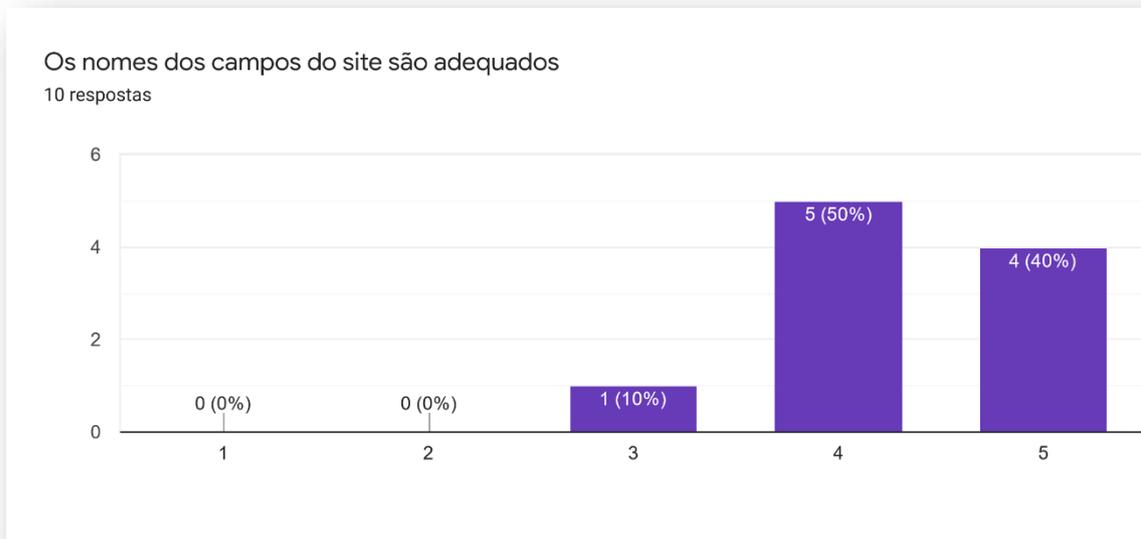


Figura 101- Resposta à pergunta 2 do questionário de satisfação

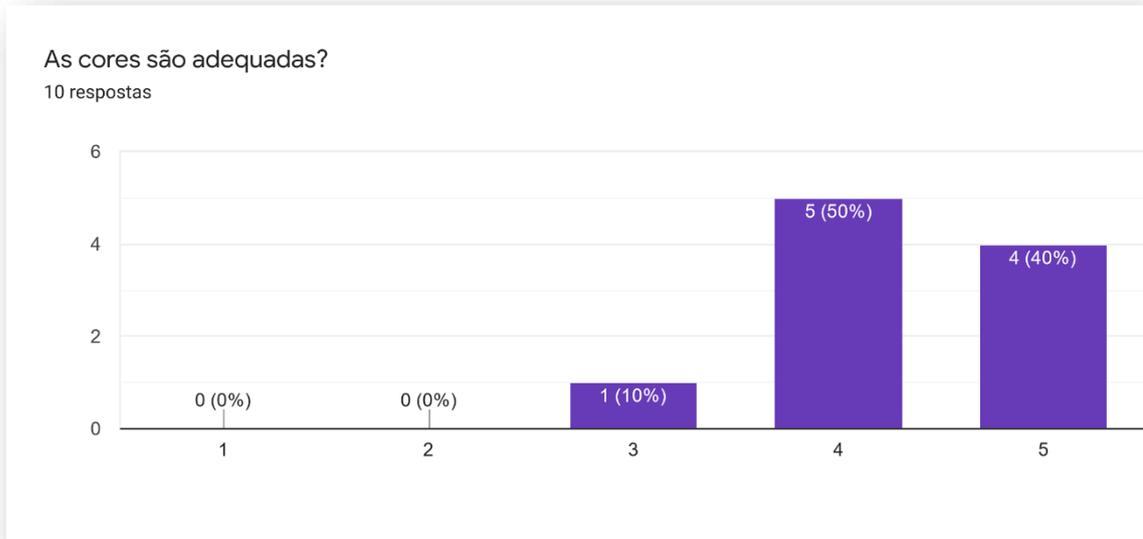


Figura 102- Resposta à pergunta 3 do questionário de satisfação

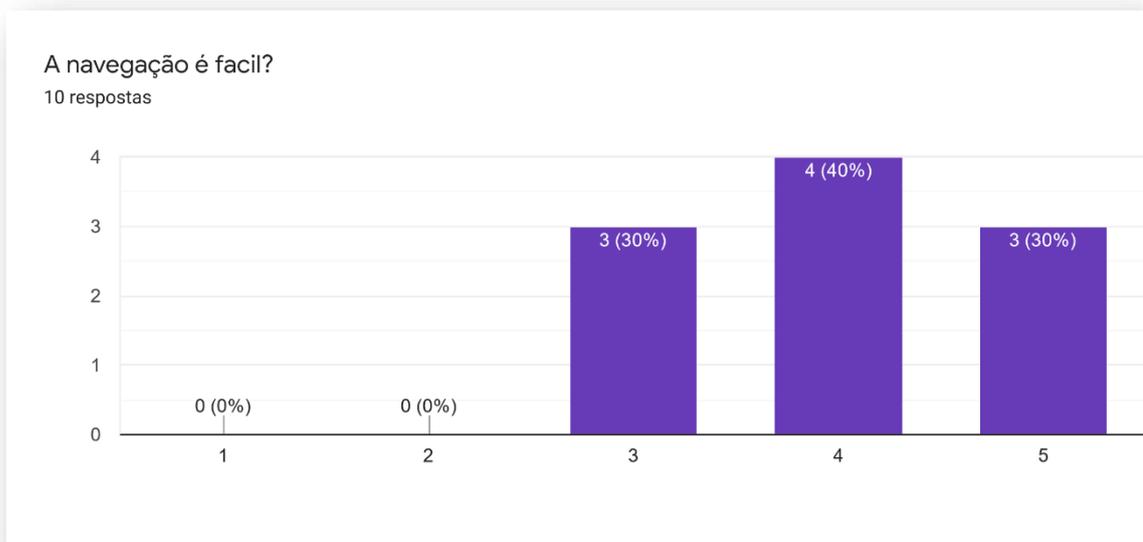


Figura 103- Resposta à pergunta 4 do questionário de satisfação

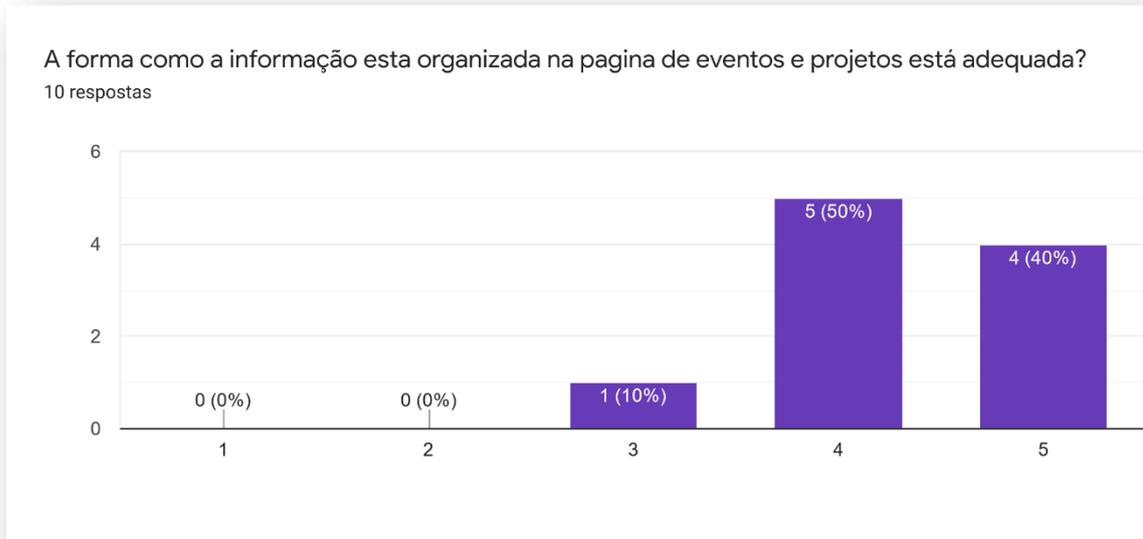


Figura 104- Resposta à pergunta 5 do questionário de satisfação

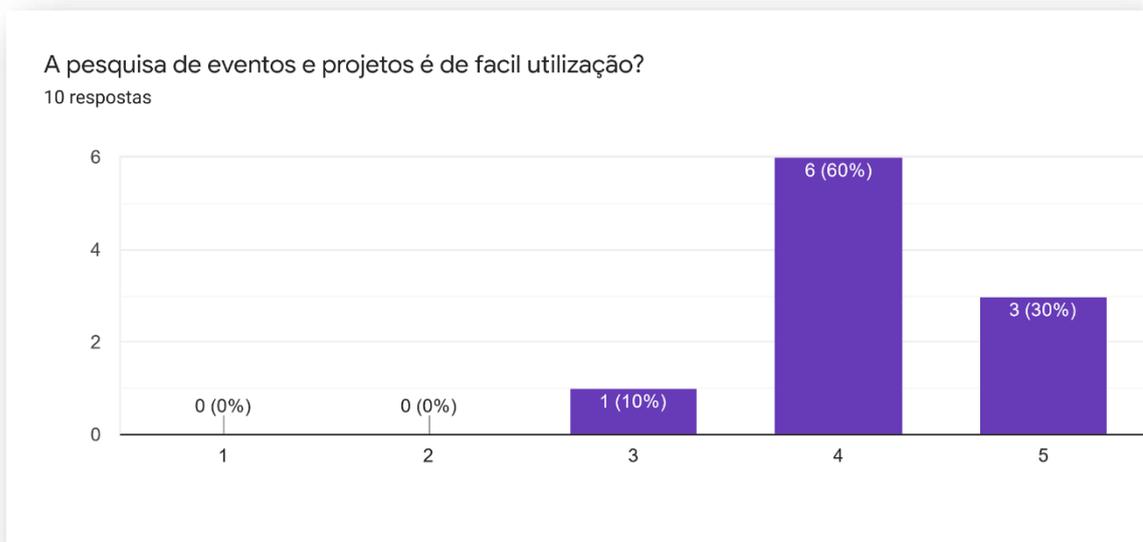


Figura 105- Resposta à pergunta 6 do questionário de satisfação

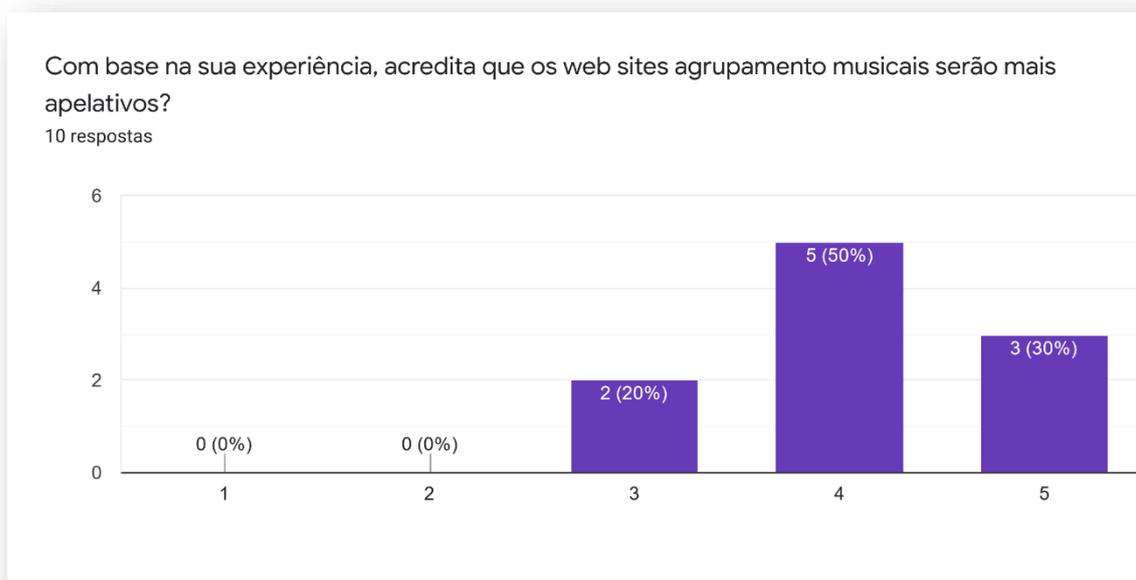


Figura 106- Resposta à pergunta 7 do questionário de satisfação

## Anexo G - Manual de Utilizador

1. Fazer o download do projeto
2. Extrair o projeto
3. Entrar no diretório do projeto e fazer um click na barra do caminho para a pasta

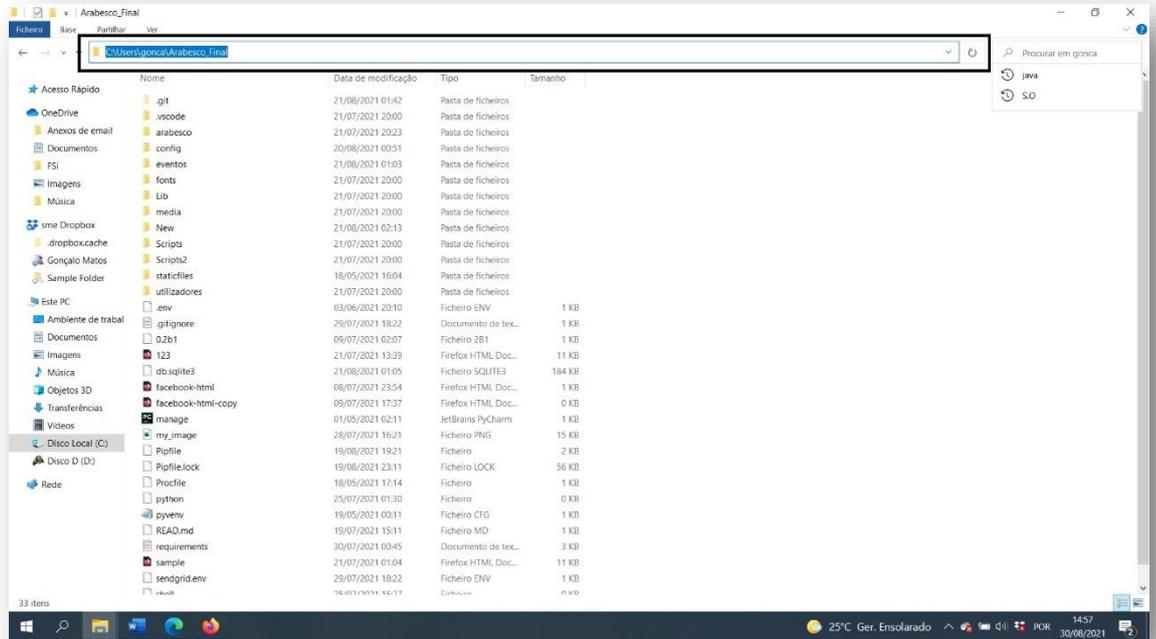


Figura 107 - Caminho para a pasta

4. Escrever “cmd” e carregar enter. Neste momento deve ter aberto este menu

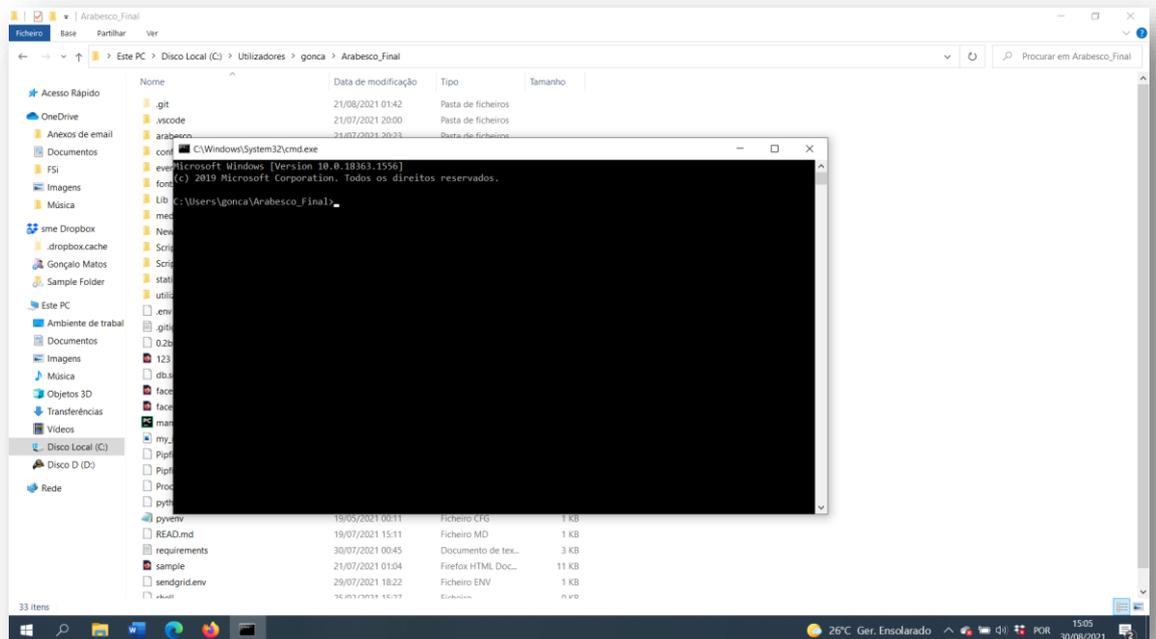
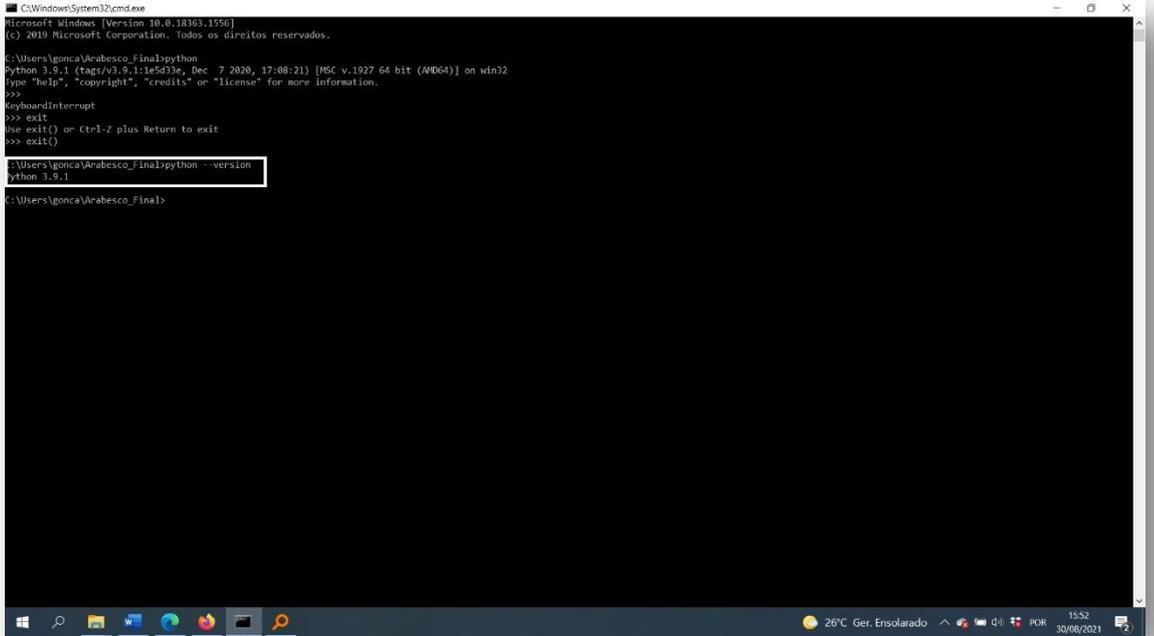


Figura 108- Menu cmd

5. Fazer download da última versão do python em <https://www.python.org/downloads/> (escolher a versão mac ou Windows consoante o caso).
6. Clicar no ficheiro executável do python e instalar.
7. Repetir os passos 3 e 4.
8. Escrever “python --version” no terminal
9. Deve aparecer a versão que instalou, neste momento tem o python instalado na sua máquina.



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.18363.1556]
(c) 2019 Microsoft Corporation. Todos os direitos reservados.

C:\Users\gonca\Arabesco_Final>python
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
KeyboardInterrupt
>>> exit()
Use exit() or Ctrl-Z plus Return to exit
>>> exit()

C:\Users\gonca\Arabesco_Final>python --version
Python 3.9.1

C:\Users\gonca\Arabesco_Final>
```

Figura 109 - Instalação de Pipenv shell

10. De seguida escrevemos no terminal “pipenv install”.
11. Depois de concluída a instalação escrevemos “pipenv shell”. Este deve ser o output esperado.

```
C:\Windows\System32\cmd.exe - pipenv shell
Microsoft Windows [Version 10.0.18363.1556]
(c) 2019 Microsoft Corporation. Todos os direitos reservados.

C:\Users\gonca\Arabesco_Final>python
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
KeyboardInterrupt
>>> exit()
Use exit() or Ctrl-Z plus Return to exit
>>> exit()

C:\Users\gonca\Arabesco_Final>python --version
python 3.9.1

C:\Users\gonca\Arabesco_Final>pipenv shell
loading env environment variables...
creating a virtualenv for this project...
Pipfile: C:\Users\gonca\Arabesco_Final\Pipfile
Using C:\Users\gonca\AppData\Local\Programs\Python\Python38\python.exe (3.8.10) to create virtualenv...
[*] Creating virtual environment...created virtual environment (Python3.9.1) in 10.4s
creator: PythonSubprocess(cwd=C:\Users\gonca\Arabesco_Final-83CFVud, clear=False, no_vcs_ignore=False, global=False)
creator: FromAppData(download=False, pip-bundle=True, setuptools-bundle=True, wheel-bundle=True, vcs-copy=True, app_data_dir=C:\Users\gonca\AppData\Local\pip\virtualenv)
added seed packages: pip==21.1.1, setuptools==57.2.0, wheel==0.36.2
activators: BashActivator, BatchActivator, FishActivator, PowerShellActivator, PythonActivator, XonshActivator

Successfully created virtual environment!
Virtualenv location: C:\Users\gonca\AppData\Local\Programs\Python\Python38\envs\Arabesco_Final-83CFVud
launching subshell in virtual environment...
Microsoft Windows [Version 10.0.18363.1556]
(c) 2019 Microsoft Corporation. Todos os direitos reservados.

Arabesco_Final-83CFVud > C:\Users\gonca\Arabesco_Final\
```

Figura 110 - Instalação do ficheiro requirements.txt

12. Escrevemos no terminal “pip install -r requirements.txt”.
13. Depois de concluídas as instalações de todas as dependências devemos escrever “python manage.py makemigrations”.
14. De seguida escrevemos “python manage.py migrate”. Neste momento temos todas as migrações feitas para a base de dados local.
15. Para correr o projeto escrevemos “python manage.py runserver”. Este deve ser o output. Devemos copiar o url assinalado e colá-lo no browser e dar *enter*

```
C:\Windows\System32\cmd.exe - pipenv shell - python manage.py runserver
Successfully installed Automat-20.2.0 CairoSVG-2.5.2 Django-3.2.3 EasyProcess-0.3 Jinja2-3.0.1 MarkupSafe-2.0.1 Pillow-8.2.0 Protego-0.1.16 PyDispatcher-2.0.5 PyPDF2-1.26.0 PyVirtualDisplay-2.2 Scrapy-2.5.0 Twilio-7.0.0 Unidecode-1.2.0 WeasyPrint-52.5 asgiref-3.4 arabic-reshaper-2.1.3 asgiref-3.4 atomicwrites-1.4.0 attrs-21.2.0 beautifulsoup4-4.9.3 bootstrap-0.11.0 bs4-0.0.1 cairocffi-1.2.0 certifi-2020.12.5 cffi-1.14.6 chardet-4.0.0 charset-normalizer-2.0.3 cloudfare-1.26.0 colorama-0.4.4 constantly-15.1.0 cryptography-3.4.7 cssselect-1.1.0 cssselect2-0.4.1 defusedxml-0.7.1 dj-database-url-0.5.0 dj-email-url-1.0.2 django-cache-url-3.2.3 django-easy-pdf-0.1.1 django-facebook-6.0.3 django-filter-2.4.0 django-heroku-0.3.1 django-installed-apps-command-2020.12.3 django-multiupload-0.5.2 django-search-view-0.3.7 django-widget-tweaks-1.4.8 django-xmltopdf-3.3.0 django-yam-0.2.2 django-yarnpkg-0.0.1 environs-0.0.0 facebook-sdk-3.1.0 fake-useragent-0.1.11 future-0.18.2 gunicorn-20.1.0 h2-3.2.0 hpack-3.0.0 html5lib-1.1 hyperlink-21.0.0 idna-2.10 inikit-1.2.2 incremental-21.3.0 iniconfig-1.1.1 iteoadapter-0.3.0 itelemadapters-1.0.4 jmspath-0.10.0 lxml-4.6.3 marshmallow-3.12.1 packaging-21.0 parse-1.19.0 parse-1.6.0 pdfkit-0.6.1 pluggy-1.0.0 priority-1.3.0 psychopg2-2.8.6 py-1.10.0 pyOpenSSL-20.0.1 pyasn1-0.4.8 pyasn1-modules-0.2.8 pycparser-2.20 pyee-8.1.0 byparsen-2.4.7 pyphen-0.11.0 pyppeteer-0.2.5 pyquery-1.4.3 pytest-6.2.4 pytest-xvfb-2.0.0 python-bidi-0.4.2 python-decouple-3.4 python-dotenv-0.17.1 python-http-client-3.3.2 pytz-2021.1 queue-lib-1.6.1 reportlab-3.5.68 requests-2.26.0 requests-html-0.10.0 schedule-1.1.0 sendgrid-6.7.1 service-identity-21.1.0 six-1.16.0 soupsieve-2.2.1 sqlalchemy-0.4.1 starkbank-ecdsa-1.1.1 tincys2-1.1.0 tomli-1.0.2 toml-1.0.10.2 tqdm-4.61.2 twisted-locsupport-1.0.1 urllib3-1.26.4 wlib-1.22.0 webencodings-0.5.1 websockets-8.1 whitenoise-5.2.0 xml2pdf-0.2.5 xvfbwrapper-0.2.9 zope.interface-5.4.0

(Arabesco_Final-03C\Fud_) C:\Users\gonca\Arabesco_Final\python manage.py runserver
Python-dotenv could not parse statement starting at line 4
Python-dotenv could not parse statement starting at line 4
Watching for file changes with StatReloader
Performing system checks...

System check identified some issues:

WARNINGS:
Arabesco.Imagens: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
HINT: Configure the DEFAULT_AUTO_FIELD setting or the AppConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.
eventos.Evento2: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
HINT: Configure the DEFAULT_AUTO_FIELD setting or the AppConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.
eventos.Eventos: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
HINT: Configure the DEFAULT_AUTO_FIELD setting or the AppConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.
eventos.Projeto: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
HINT: Configure the DEFAULT_AUTO_FIELD setting or the AppConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.
eventos.Videos: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
HINT: Configure the DEFAULT_AUTO_FIELD setting or the AppConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.
eventos.agendamento: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
HINT: Configure the DEFAULT_AUTO_FIELD setting or the AppConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.
utilizadores.Imagens: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
HINT: Configure the DEFAULT_AUTO_FIELD setting or the AppConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.

System check identified 7 issues (0 silenced).
August 30, 2021 - 16:09:42
Django version 3.2.3, using settings 'arabesco.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Figura 111 - Aplicação a correr localmente

16. Neste momento estamos na página principal do website. Agora, clique no sign up da navigation bar. Este é o output esperado.

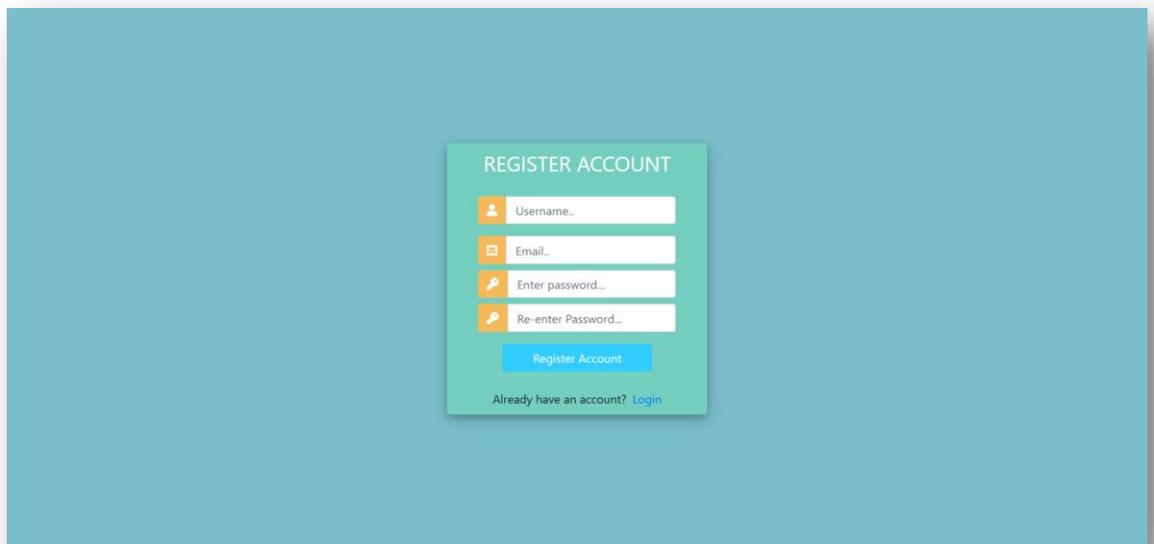


Figura 112 - Página de login

17. Crie um utilizador com username, email e password à sua escolha.
18. Vamos voltar ao menu cmd e fazer “ctrl + c” para interromper o funcionamento do servidor local.
19. De seguida verificar se o utilizador ficou guardado na base de dados, para isso, escrevemos “python manage.py createsuperuser” no menu cmd. Desta forma teremos acesso ao django admin.

```
C:\Windows\System32\cmd.exe - pipenv shell - python manage.py createsuperuser
HINT: Configure the DEFAULT_AUTO_FIELD setting or the UtilizadoresConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.

System check identified 7 issues (0 silenced).
August 30, 2021 - 16:09:42
Django version 3.2.3, using settings 'arabesco.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[30/Aug/2021 16:32:59] "GET / HTTP/1.1" 200 16144
[30/Aug/2021 16:32:59,630] - Broken pipe from ("127.0.0.1", 1486)

[30/Aug/2021 16:32:59] "GET /static/images/png-transparent-black-border-template-drawing-arabesque-art-arabesque-love-miscellaneous-text-thumbnail.png HTTP/1.1" 200 6265
[30/Aug/2021 16:32:59,748] - Broken pipe from ("127.0.0.1", 1485)

[30/Aug/2021 16:33:05] "GET /favicon HTTP/1.1" 200 0
[30/Aug/2021 16:33:05] "GET /login/ HTTP/1.1" 200 3258
[30/Aug/2021 16:33:16] "GET /registo HTTP/1.1" 200 0
[30/Aug/2021 16:33:16] "GET /registo/ HTTP/1.1" 200 4330

(Arabesco_Final-03C\Uad_) C:\Users\gonca\Arabesco_Final\python manage.py createsuperuser
Python-dotenv could not parse statement starting at line 4
System check identified some issues:

WARNINGS:
arabesco.Imagens: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
HINT: Configure the DEFAULT_AUTO_FIELD setting or the AppConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.
eventos.Evento2: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
HINT: Configure the DEFAULT_AUTO_FIELD setting or the EventosConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.
eventos.Imagens: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
HINT: Configure the DEFAULT_AUTO_FIELD setting or the EventosConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.
eventos.Projeto: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
HINT: Configure the DEFAULT_AUTO_FIELD setting or the EventosConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.
eventos.Videos: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
HINT: Configure the DEFAULT_AUTO_FIELD setting or the EventosConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.
eventos.agendamento: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
HINT: Configure the DEFAULT_AUTO_FIELD setting or the EventosConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.
utilizadores.Imagens: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
HINT: Configure the DEFAULT_AUTO_FIELD setting or the UtilizadoresConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.

Username:
```

Figura 113 - Criação de um user com premissões de admin

20. Aqui iremos registar um utilizador com username e password, este user não precisa de ser igual ao do registo anterior, da página do website, deve inclusivamente ser diferente do anterior, de forma a verificar a diferença de premissões de acesso entre os dois.
21. Depois de ter efetuado o registo vamos realizar novamente o passo número 15.
22. No endereço da barra de pesquisa devemos acrescentar a palavra admin no final do url. <http://127.0.0.1:8000/admin>
23. Aqui efetuamos o login com as credenciais que inserimos no registo do menu cmd.

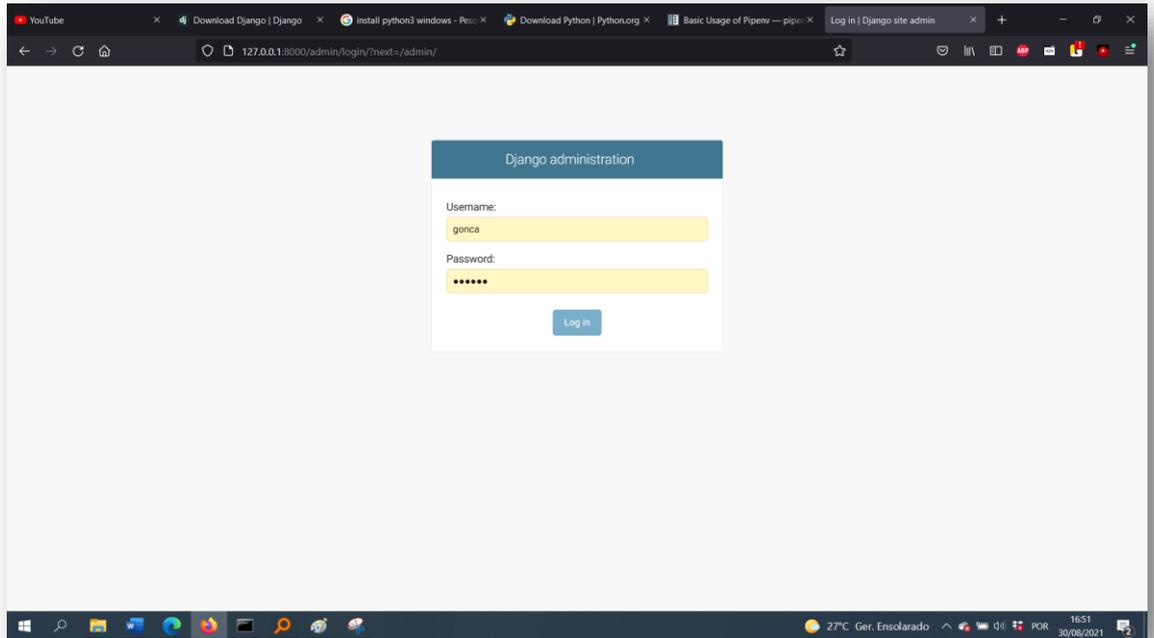


Figura 114 - Login na página de admin

24. Depois de entrar na página de admin vamos clicar em “Users”.

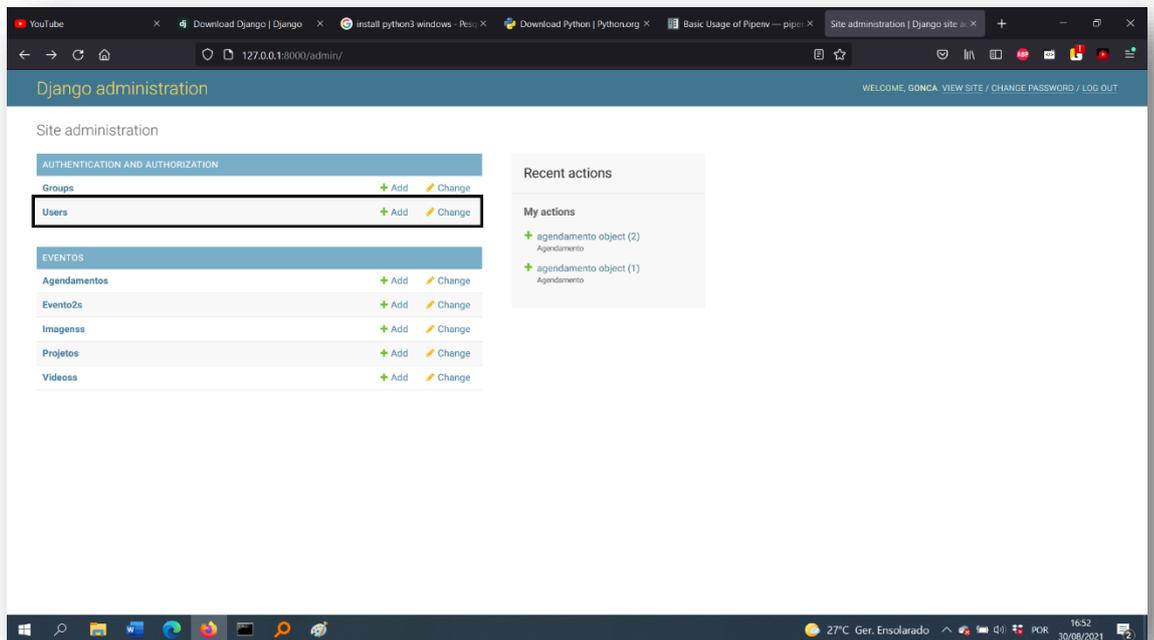


Figura 115 - Menu da página admin

25. Aqui podemos ver a lista de utilizadores registados. O user registado através do comando “python manage.py createsuperuser” deve aparecer com o status assinado com o símbolo “✓” e o user registado pelo website com o símbolo “x”, como aparece no exemplo abaixo.

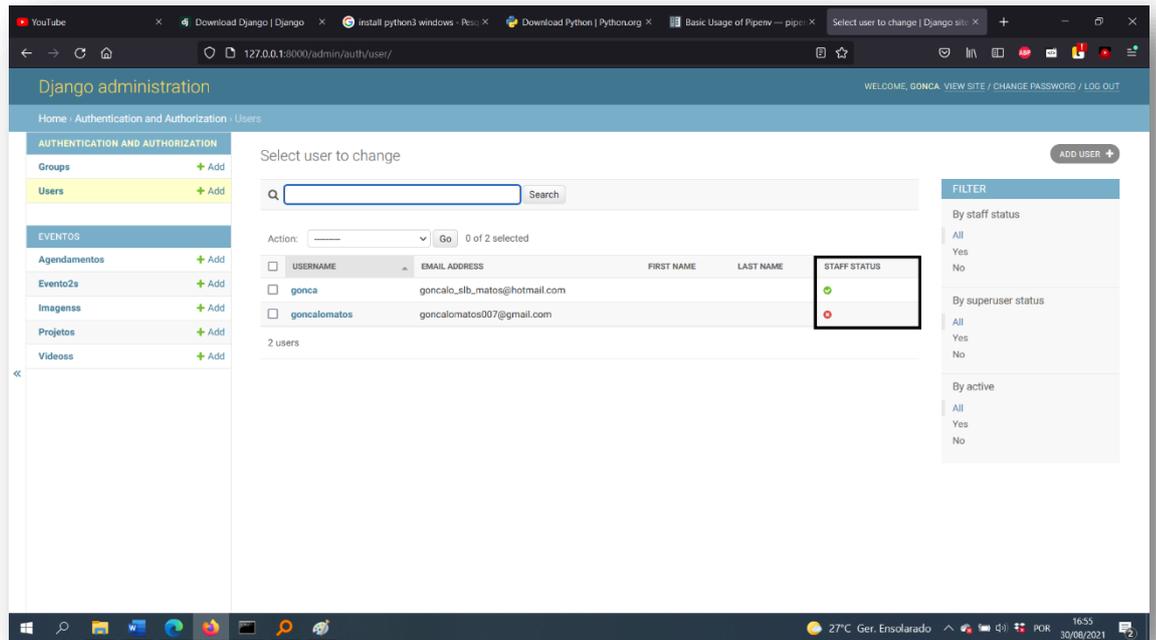


Figura 116 - Staff status

26. Agora vamos testar a criação de um novo evento. Vamos clicar onde diz "Add", na barra de eventos.
27. Agora preenchemos os campos necessários e no final fazemos *save*.
28. Agora vamos voltar ao site retirando tudo do url, deixando apenas:  
<http://127.0.0.1:8000/>
29. Vamos à parte do website onde esta presente a lista de eventos.
30. Podemos verificar que o evento que acabamos de criar já se encontra no site.

Arabesco

Home Eventos Membros História Canvas Projeto Associado Log In Sign In

## REQUIEM de MOZART

Requiem em Ré Menor KV 626, W.A. Mozart (1756-1791) versão para quarteto de cordas (1810) de P. Lichtenthal (1780-1853)

Sept. 30, 2021

Teatro Armando Cortez



**Figura 117 - Evento**

## Anexo H - Guião de Testes

### Testes de Usabilidade

#### Plataforma para gestão de agrupamento musical

O departamento de Engenharia Informática e Sistemas de Informação, no âmbito do trabalho final de curso solicita a sua participação num estudo de usabilidade à plataforma do grupo musical Quarteto Arabesco, com vista a melhorar a plataforma.

Iremos recolher informação sobre a forma como utiliza a aplicação e como esta responde às suas necessidades informativas. Após o teste vamos solicitar-lhe o preenchimento de um questionário. Utilizaremos esta informação, bem como a recolha da informação dos outros participantes, para a melhoria da aplicação. Ao assinar este acordo, o Departamento de Engenharia Informática e Sistemas de Informação garante que esta informação será utilizada apenas para efeitos de avaliação do serviço e análise dos resultados e que em caso algum o departamento cederá esta informação a terceiros.

A sua assinatura do acordo garante o seu consentimento para utilizarmos os comentários verbais ou escritos. No âmbito deste teste, vai ter de realizar um conjunto de tarefas que lhe serão apresentadas uma a uma, pelo(a) Administrador(a) de testes.

Para darmos início ao teste, informe o Administrador(a) que está preparado(a). Se tiver alguma dúvida pode colocá-la ao administrador(a) do teste, agora ou durante o mesmo.

Se está de acordo com os termos do presente acordo, por favor assine em baixo, indicando assim a sua aceitação.

DEISI

Participante 1

Participante 2

Nome:

Data:

Assinatura:

Obrigado!

Requisitos: Um computador

Tarefa 1: Configurações iniciais

Seguir todos os passos do Manual de Utilizador até ao passo número 15. Só depois de feitas estas configurações é possível realizar as restantes tarefas.

Tarefa 2: Registo sem credenciais

Fazer o registo sem credenciais. Indique o output apresentado.

R:

Tarefa 3: Registo com credenciais validas

Fazer o registo com credenciais à sua escolha.

Tarefa 4: Login sem credencias

Fazer o login sem credenciais. Indique o output apresentado.

R:

Tarefa 5: Testar a validade das credenciais

Fazer o login com outras credenciais à escolha. Descreva qual foi o output.

R:

Tarefa 6: Adicionar um novo evento sem campos preenchidos

Deve voltar a abrir o manual de utilizador e realizar os passos 18 até ao 23, inclusive. Nesta altura já entrou na página de administrador, agora, deve clicar na palavra “add” na barra de eventos. Agora deve tentar adicionar um novo evento sem preencher qualquer campo clicando diretamente em “save”. Descreva o output apresentado.

R:

Tarefa 7: Adicionar um evento preenchendo os campos. Descreva qual foi o output.

R:

Tarefa 8: Verificar se o evento foi adicionado com sucesso.

Deve entrar na zona do site que contem a lista de eventos. Entre no site através do localhost, de seguida clique em “Eventos” e verifique se o evento que criou, consta na lista.

R:

Tarefa 9: Pesquisar por data de eventos invalida.

De seguida clique em “Pesquisar” e seleccione um intervalo de datas em que não existam eventos. Indique qual foi o output.

R:

Tarefa 10: Pesquisar por data de eventos valida.

Pesquise por um intervalo de datas em que exista pelo menos um evento. Indique o output obtido.

R:

**Tarefa 11:** Pesquisar com palavra inexistente.

Pesquise por uma palavra que não conste em nenhum evento. Indique o output obtido.

R:

Tarefa 12: Pesquisar com palavra existente.

Pesquise por uma palavra que conste em algum evento. Indique o output obtido.

R:

Tarefa 13: Inserir um Canvas pelo Admin.

Entre novamente na página de Admin e insira um *Canvas* na aba correspondente.

Tarefa 14: Executar o ficheiro "post.py".

Corra o seguinte comando na linha de comandos: "python eventos/posts.py". Descreva o output obtido.

R:

Tarefa 15: Executar o ficheiro "mail-sender2.py".

Corra o seguinte comando na linha de comandos: "python eventos/mail-sender2.py". Descreva o output obtido.

R:

Tarefa 16: Utilizar o Heroku Scheduler.

Deve entrar em <https://www.heroku.com>

Depois, deve aguardar pelas credenciais dadas por um dos orientadores do teste, depois de inserir as credenciais deve entrar na app "quarteto-arabesco". Depois deve entrar em "Resources" e depois em "Heroku Scheduler", deve seleccionar "create job" e de seguida seleccionar o tempo "Every 10 minutes". Por fim deve inserir este comando na barra de comandos "python eventos/posts.py" e fazer "save job".

Deve aguardar até ao fim dos 10 minutos e verificar se o post foi efetuado.

R:



## **Glossário**

LEI	Licenciatura em Engenharia Informática
LIG	Licenciatura em Informática de Gestão
TFC	Trabalho Final de Curso
ORM	Object Relational Mapping
CMS	Custom Management System
MTV	Model Template View
MVC	Model View Control
CRUD	Create Read Update Delete
URL	Uniform Resource Locators
HTML	HyperText Markup Language
CSS	Cascading Styles Sheet

---