



UNIVERSIDADE
LUSÓFONA

Plataforma de Gestão de Eventos

TFC 04

Lucas Silva - 21703349 | Pedro Santos - 21702626

Orientador: Rodrigo Correia

Coorientador: Bruno Cipriano

TFC | Licenciatura em Engenharia Informática | 2019/2020

www.ulusofona.pt

Direitos de cópia

(Plataforma de gestão de eventos), Copyright de (Lucas Silva, Pedro Santos, Rodrigo Correia e Bruno Cipriano), 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.

Conteúdo

Índice de Tabelas	4
Índice de Figuras	4
Resumo	5
Abstract.....	5
1. Identificação do Problema	6
2. Viabilidade e Pertinência.....	7
3. Levantamento e análise dos Requisitos	9
4. Solução Proposta	17
4.1 Back-end	18
4.2 Front-end.....	19
4.3 Entidades presentes.....	20
4.4 Mapa da Aplicação	20
4.5 Implementação da Plataforma.....	21
5. Benchmarking.....	22
7.Plano de testes e validação	24
7. Calendário.....	25
Bibliografia.....	26
Anexos	27
1 – Exemplo do Doodle	27
2 – Mapa da aplicação	28
Glossário.....	29

Índice de Tabelas

Tabela 1 - Inscrições e provas ao longo dos anos.....	7
Tabela 2 – Tempo Total em Horas para a realização de inscrições	8
Tabela 3 - Comparação com a concorrência	22
Tabela 4 - Testes à inscrição e duração dos mesmos	24
Tabela 5 - Método e planeamento do projeto.	25

Índice de Figuras

Figura 1 - Dados dos Membros	6
Figura 2 - Arquitetura cliente-servidor.....	17
Figura 3 - Frameworks mais populares para desenvolvimento web em 2019.....	19
Figura 4 - Entidades e respetivos dados	20
Figura 5 - Exemplo código do backend em LISP.....	21
Figura 6 - Exemplo de um pedido POST	21
Figura 7 - Tabela User na base de dados	21
Figura 8 - Exemplo da utilização do doodle.....	27
Figura 9 - Mapa da aplicação	28

Resumo

Uma empresa tem um “clube de corrida e caminhada” o qual serve para promover a atividade física e o bem-estar dos colaboradores. O objetivo da realização deste projeto é a criação de uma plataforma que permita gerir as inscrições em eventos físicos, e facilitar o controlo destas inscrições.

A plataforma irá deixar de ser apenas para “corrida e caminhada”, e passará a ser uma plataforma de gestão de eventos, sendo possível adicionar inúmeros eventos novos, como por exemplo duatlo, triatlo, entre outros.

Irá ser possível também gerar, apenas por administradores, relatórios como por exemplo, os participantes mais ativos, orçamento gasto por membro e entre outros.

Abstract

One company has a “running and walking club” that manages registration for running or walking events. The goal is to create a platform that makes it easier to control these entries and other facilities.

The platform will no longer be just for “running and walking”, but will be for event management, and it will be possible to add numerous new events, such as duathlon, triathlon, among other.

It will also be possible only by administrators to generate reports such as the most active participants, budget spent per member and others.

1. Identificação do Problema

Nesta empresa Portuguesa que prefere manter-se anónima, os funcionários têm um certo saldo anual que pode ser gasto unicamente em eventos de atividade física (sendo estes apenas de corrida e caminhada no presente). Para isso os funcionários que queiram utilizar este saldo disponível têm que se registar no clube de corrida e caminhada. Após o seu registo (neste momento o registo é feito apenas por um administrador) os membros podem utilizar este serviço, e assim inscrever-se e participar em várias atividades de diversos eventos.

Até ao momento a gestão deste projeto é feita de maneira muito manual, pois são necessários vários ficheiros Excel para guardar e gerir as diversas informações existentes, mencionadas à frente. Posteriormente é utilizado o Doodle [1] para realizar as inscrições dos colaboradores aos eventos, ou seja, a empresa de momento está dependente de uma plataforma externa e por questões de segurança dos dados queremos manter os processos todos internos à empresa.

Apenas existem dois tipos de utilizadores, os membros e os administradores (sendo os administradores membros da empresa também).

Na figura 1 podemos verificar a informação dos membros, os dados presentes são: Sócio ID, Short Name, Nome Completo, Cartão de Cidadão, Data de Nascimento, Tamanho de T-Shirt e Escritório (localização). Neste momento estes dados são guardados numa tabela em Excel, sendo a alteração de dados muito dificultada para o membro, pois só um administrador o consegue fazer, por isso têm sempre de contactar um administrador sempre que queira alterar, por exemplo, o tamanho da sua t-shirt.

Sócio ID	Short Name	Nome Completo	Cartão de Cidadão	Data de Nascimento	Tamanho de T-shirt	Escritório
1	Zeca	José Carlos	11.111.111,00	32.874,00 €	L	Lisboa
2	Maria	Maria Madalena	22.222.222,00	25.782,00 €	XL	Porto
3	António	António Variações	33.333.333,00	27.704,00 €	M	Lisboa

Figura 1 - Dados dos Membros

A maneira de guardar dados dos eventos é semelhante à dos membros. A tabela dos eventos contém a seguinte informação: Data, Prova (nome da prova), Local, Distância Corrida, Distância Caminhada, Página Oficial, Inscrição Interna (um link para o doodle, onde é feito a inscrição dos membros nos eventos – ver anexo 1), Data Limite Inscrição, Inscrição Corrida € (custo da inscrição numa corrida), Inscrição Caminhada € (custo da inscrição numa caminhada).

Como podemos ver a gestão é um pouco limitada, pois se existir um novo tipo de atividade, por exemplo triatlo, íamos ter de inserir uma nova coluna com o preço. Isto demonstra que atualmente o sistema não tem uma boa escalabilidade.

A maneira de gestão de membro-evento também é feita numa tabela Excel, na qual cada linha representa um membro e vamos adicionando colunas com novos eventos. Esta gestão fica muito complicada para os responsáveis.

Foi-nos então proposto pelos professores a criação de um sistema que permita facilitar a gestão dos dados da plataforma, através da utilização de um website.

2. Viabilidade e Pertinência

O incentivo da prática de desporto por parte das empresas tem vindo a crescer nos últimos anos, devido a isso a empresa do professor Bruno Cipriano e do professor Rodrigo Correia juntou-se em 2017 a este incentivo.

A empresa tem dois escritórios em Portugal, um em Lisboa e outro no Porto, totalizando cento e vinte funcionários, dos quais quarenta são membros e seis administradores (três de cada escritório). Na Tabela 1 podemos verificar os dados recolhidos sobre o número de provas e de inscrições por semestres durante os 3 anos de existência do clube.

Tabela 1 - Inscrições e provas ao longo dos anos.

	2017		2018		2019	
Semestres	1º	2º	1º	2º	1º	2º
Nº de Provas	N/A	23	7	29	15	19
Nº de Inscrições	121		115		48	

Atualmente uma única inscrição demora por volta de sete minutos e meio, desta forma, é perdido bastante tempo apenas na realização de inscrições, como está demonstrado na Tabela 2. E como tempo é dinheiro, supondo que o salário médio por hora desta empresa é de 6.25€, foram gastos então 221.875€ na gestão deste projeto.

Como podemos ver mais à frente, neste momento a plataforma demora cerca de 1.20 minutos por inscrição, mantendo-se as 284 inscrições totais chegamos a 14.2 horas gastas o que nos dá apenas 88.75€ gastos, o que achamos bastante bom para a empresa pois vai ter um lucro bastante mais elevado.

Tabela 2 – Tempo Total em Horas para a realização de inscrições

	2017	2018	2019	Total
Nº de Inscrições	121	115	48	284
Nº de Horas	15.1	14.3	6	35.5

Esta redução de produtividade e de tempo vai ser um dos nossos principais objetivos para melhorar, queremos que tanto a inscrição e a criação de um evento seja o mais facilitada e rápida possível.

Vamos também com a nova plataforma tentar com que o número de membros aumente, criando uma plataforma mais agradável e fácil para todos utilizarem.

Como vamos utilizar tecnologias que são familiares às usadas na empresa, (para assim respeitar as políticas de segurança da própria empresa em que não irá ser preciso instalar nenhum software extra, sendo então preferível utilizar softwares já aprovados para uso interno) previmos que a continuação da implementação e uso do novo sistema irá ser bastante facilitado, favorecendo também, a futura manutenção do mesmo.

3. Levantamento e análise dos Requisitos

Nesta plataforma apenas irão existir dois tipos de utilizadores, os membros e os administradores. A não esquecer que os membros são obrigatoriamente funcionários da empresa e os administradores podem também ser membros. Será possível aos membros propor um evento novo, tal como procurar e inscrever-se em eventos. Mas todas estas ações do membro têm de ser aprovadas por um Admin. É importante relembrar que cada membro tem um orçamento anual que vai sendo gasto à medida que se inscreve em provas.

Requisitos Membros

Requisitos Funcionais

RFM01

Registo de Membro.

O membro tem de facultar os seguintes dados para realizar o registo.

1. Nome
2. Email
3. Nome de Utilizador
4. Palavra-passe
5. Localização Escritório
6. Tamanho T-Shirt
7. Cartão de Cidadão (CC)
8. Telefone
9. Data de Nascimento

Pré-Condição

O utilizador tem de ser colaborador da empresa.

CrITÉRIOS de aceitação

O Sistema deve permitir registar um membro.

O Administrador tem de aprovar a o registo do membro.

RFM02

Alteração de Dados do Registo.

Pré-condição

O utilizador tem de estar autenticado na sua conta.

CrITÉrios de aceitação

O Sistema deve permitir ao membro fazer alterações aos dados de registo anteriormente inseridos.

RFM03

Escolha das Atividades Preferenciais.

Pré-condição

O utilizador deverá estar autenticado na sua conta.

CrITÉrios de aceitação

O Sistema deve permitir ao membro escolher as suas atividades preferenciais.

RFM04

Inscrição num Evento.

Pré-Condição

O utilizador deverá estar autenticado na sua conta.

O utilizador deverá ter saldo suficiente para o evento em causa.

A atividade tem de existir na plataforma.

CrITÉrios de aceitação

O Sistema deve permitir ao membro fazer uma inscrição num evento.

RFM05

Proposta de Evento.

O membro tem de facultar os seguintes dados para realizar a proposta de evento.

1. Data
2. Local
3. Site URL
4. Preço
5. Atividades (corrida, triatlo, ...)

Pré-Condição

O utilizador deverá estar autenticado na sua conta.

Critérios de aceitação

O Sistema deve permitir ao membro fazer uma proposta de evento.

RFM06

Pesquisa de Evento.

O membro tem de facultar um ou mais dos seguintes tópicos de filtragem para a pesquisa de um evento aprovado.

1. Nome do Evento
2. Intervalo de Data
3. Tipo de Atividade
4. Localização do Evento

Pré-condição

O utilizador tem de estar autenticado na sua conta.

Critérios de aceitação

O Sistema deve permitir ao membro pesquisar por um evento.

Requisitos Administradores

Requisitos Funcionais

RFA01

Tornar um membro num Admin.

Pré-Condição

Tem de existir já um Admin no sistema.

CrITÉrios de aceitação

O Sistema deve permitir ao Admin promover um membro em Admin.

RFA02

Tornar um Admin num membro.

Pré-Condição

Tem de existir já um Admin no sistema.

CrITÉrios de aceitação

O Sistema deve permitir ao Admin despromover um Admin para membro.

RFA03

Aprovar um registo de um membro.

Pré-Condição

Um membro efetuou o registo (RFM01).

CrITÉrios de aceitação

O Sistema tem de permitir ao Admin aprovar o registo de um membro.

O Sistema deve atualizar o orçamento anual do membro por cada inscrição confirmada.

RFA04

Aprovar uma inscrição de um membro num evento.

Pré-Condição

Um membro efetuou uma inscrição num evento. (RFM04).

Critérios de aceitação

O Sistema deve permitir ao Admin confirmar a inscrição de um membro nos eventos.

O Sistema deve atualizar o orçamento anual do membro por cada inscrição confirmada.

RFA05

Criação de um Evento.

O Admin tem de facultar os seguintes dados para realizar a criação de um novo evento.

1. Data
2. Morada
3. Código-Postal
4. Site URL
5. Preço
6. Atividades (corrida, triatlo, ...)

Pré-Condição

Tem de existir um Admin na plataforma.

Critérios de aceitação

O Sistema deve permitir que o Admin crie um novo evento.

RFA06

Aprovação de uma proposta de evento.

Pré-Condição

Um membro efetuou uma proposta de evento (RFM05).

CrITÉrios de aceitação

O Sistema deve permitir que o Admin aprove uma proposta de evento.

RFA07

Gestão de eventos.

Pré-Condição

Tem de existir um Admin na plataforma.

Tem de existir um evento na plataforma

CrITÉrios de aceitação

O Sistema deve permitir ao Admin gerir eventos (listar, editar, procurar e remover).

RFA08

Gestão de membros.

Pré-Condição

Tem de existir um Admin na plataforma.

Tem de existir um membro na plataforma

CrITÉrios de aceitação

O Sistema deve permitir ao Admin gerir membros (adicionar, editar e remover).

RFA09

Criação de uma atividade.

O Admin tem de facultar os seguintes dados para realizar a criação de uma nova atividade.

1. Tipo de atividade (corrida, triatlo, ...)

Pré-Condição

Tem de existir um Admin na plataforma.

Critérios de aceitação

O Sistema deve permitir que o Admin crie uma atividade.

RFA10

Notificação aos membros de novo evento.

Pré-Condição

Tem de existir um Admin na plataforma.

Tem de existir um membro na plataforma.

Foi criado (RFA05) ou aceite a proposta (RFA06) de um evento.

Critérios de aceitação

O Sistema tem de permitir ao Admin enviar emails aos membros interessados quando surge um novo evento.

RFA11

Relatórios sobre estatísticas da plataforma.

Deve ser possível apresentar as seguintes métricas.

1. Orçamento usado, em média / mediana, máximo e mínimo (por ano, global).
2. Nr provas distintas (por ano, global).

3. Nr participações (por ano, global).
4. Nr provas distintas, por tipo (corrida vs caminhada) (por ano, global).
5. Nr participações, por tipo (corrida vs caminhada) (por ano, global).
6. Participantes mais activos (top 5 default, top N a pedido) (por ano, global).
7. Participantes menos activos (bottom 5, bottom N a pedido) (por ano, global).

Pré-Condição

Tem de existir um Admin na plataforma.

Critérios de aceitação

O Sistema tem de permitir ao Admin pedir as métricas acima definidas.

O Sistema tem de permitir ao Admin pedir um relatório que contenha todas as métricas acima definidas.

4. Solução Proposta

A arquitetura escolhida para esta plataforma vai ser do tipo cliente-servidor (ver figura 2). Para o front-end vamos utilizar uma tecnologia que utilize typescript e para o back-end vamos usar common lisp, vamos esclarecer mais a frente o porquê do uso de cada uma. Para comunicar entre o servidor eo cliente iremos usar pedidos JSON.

Relativamente à base de dados decidimos usar o modelo não relacional (mais sobre esta escolha no capítulo “Benchmarking”).

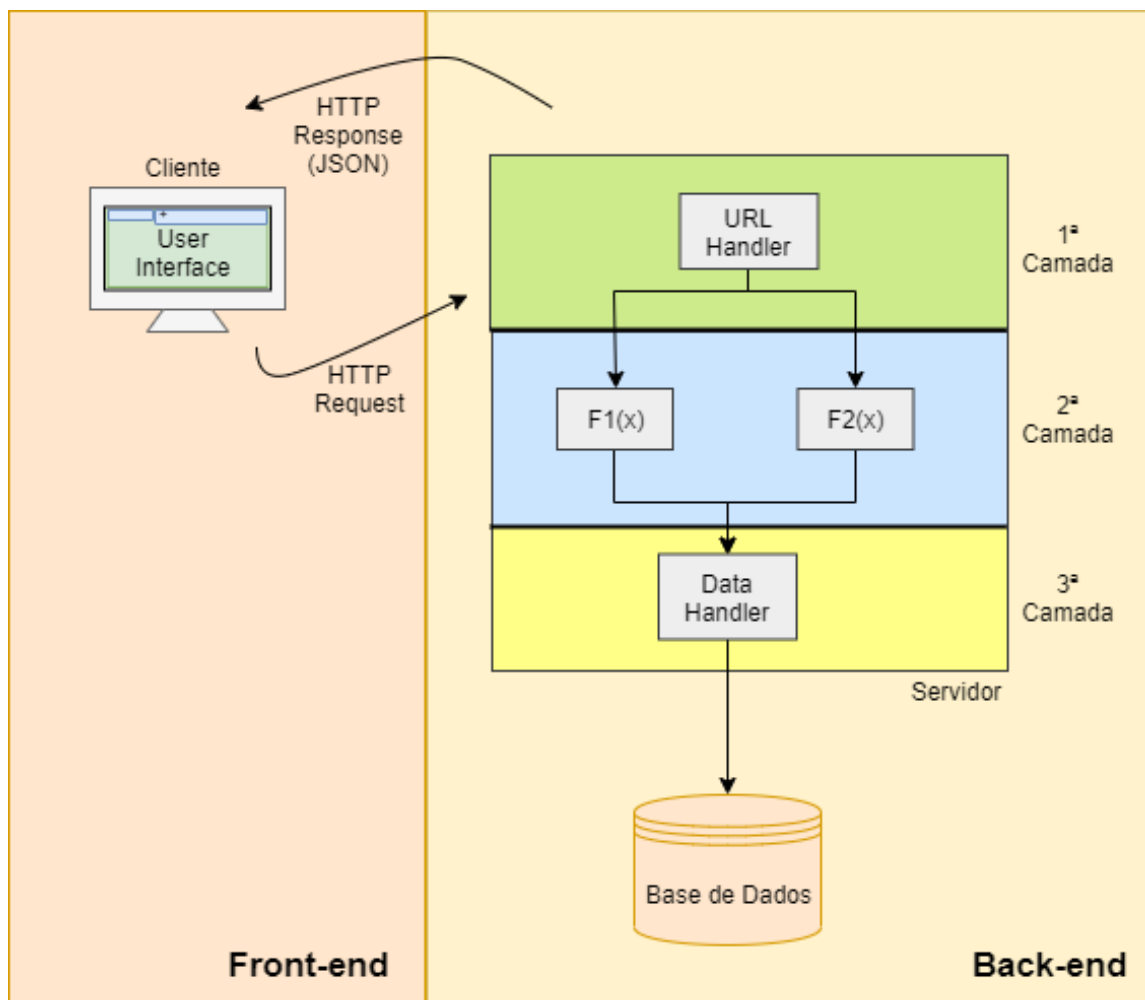


Figura 2 - Arquitetura cliente-servidor

4.1 Back-end

Para este sistema propomos usar uma arquitetura de “3 Camadas” para o lado do servidor.

A primeira camada (ver figura 2) irá ter a responsabilidade de receber os pedidos que o utilizador efetua na User Interface. Depois de ter recebido os pedidos irá direcionar para a função que trata tal pedido - esta função encontra-se na segunda camada. A terceira camada serve para fazer pedidos à base de dados. As funções da segunda camada irão pedir listas de dados à função presente nesta camada, indo esta função fazer uma primeira filtragem dos dados pedidos para poupar trabalho às camadas acima.

A escolha do uso de common lisp para linguagem na parte do servidor foi bastante simples. A empresa não gosta de produtos externos, sendo common lisp uma das mais trabalhadas pelos programadores desta empresa, a escolha foi facilitada.

4.2 Front-end

No front-end escolhemos utilizar a framework Angular, que utiliza typescript tal como nos pretendíamos. Visto que é uma framework muito utilizada mundialmente como podemos ver no gráfico da figura 3¹, e como já é utilizada internamente na empresa respeitando assim a política de segurança da empresa, foi a melhor escolha para assim também nos preparar para o mercado de trabalho, pois iremos dominar uma das mais pedidas frameworks, como já vimos anteriormente.

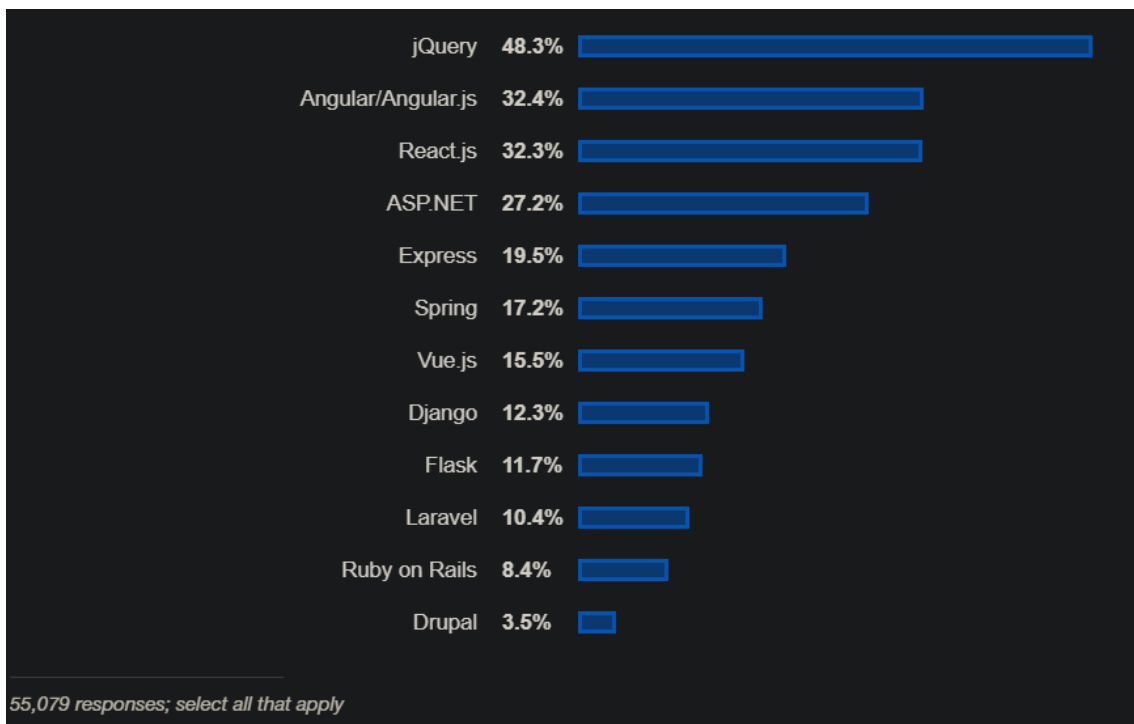


Figura 3 - Frameworks mais populares para desenvolvimento web em 2019

¹ Stack Overflow. *Stack Overflow Developer Survey 2019*. Available at: <https://insights.stackoverflow.com/survey/2019#most-popular-technologies>

4.3 Entidades presentes

Para primeiro contacto com o projeto, analisámos os dados partilhados pelos professores (ficheiros Excel que contêm os dados dos membros e eventos como vimos anteriormente), separámos em entidades e respetivos dados para facilitar a construção da base de dados. Isto facilita não só a construção da base de dados, mas também nos dá um primeiro contacto com todas as entidades presentes e que dados importantes é que cada uma irá ter (ver figura 4).

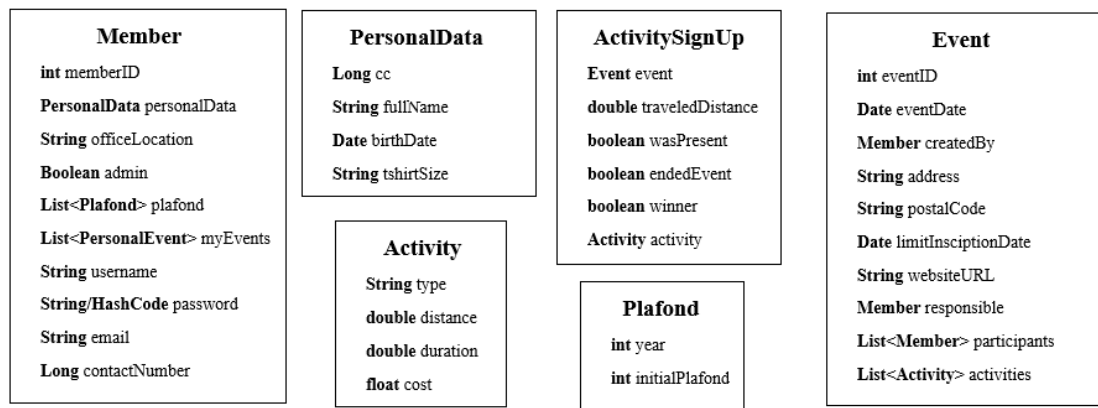


Figura 4 - Entidades e respetivos dados

4.4 Mapa da Aplicação

Elaborámos também um primeiro mapa da aplicação, que podemos observar no Anexo 2, para ser mais simples de navegar na interface do utilizador, mas também para conseguirmos detetar já alguns requisitos que talvez não são visíveis à primeira, reduzindo assim erros futuros.

4.5 Implementação da Plataforma

No seguinte link dá acesso a uma demonstração da plataforma:

- <https://youtu.be/xAZd8a54t9k>

Este capítulo irá ser dedicado a demonstração e explicação de código implementado para a construção desta plataforma.

O lado do backend tem sido o nosso principal foco e um dos maiores desafios foi a ligação à base de dados, para isso tivemos que fazer o seguinte:

```
(defun add-user (name cc birth_date t_shirt_size office plafond)
  "Add a user with the given name to the database."
  (let ((user (make-instance 'users :name name :cc cc :birth_date birth_date :t_shirt_size t_shirt_size :office office :plafond plafond)))
    (db.insert 'user-collection* (user->doc user))))
```

Figura 5 - Exemplo código do backend em LISP

Este pedaço de código é um exemplo de ligação e introdução de um membro na base de dados MongoDB, no qual é recebido todos os dados necessários para a adição na colecção “user”.

Através da introdução dos dados num formulário dado na parte da web, o backend recebe os dados através de um pedido post (como podemos ver abaixo) e introduz então na base de dados.

```
$.post('/user',{name: name,cc: cc, birth_date: birth_date, t_shirt_size: t_shirt_size, office: office, plafond: plafond},function(users){
```

Figura 7 - Exemplo de um pedido POST

Podemos verificar, através da aplicação Mongo Compass² disponibilizada pelo MongoDB, que os users criados foram introduzidos na base de dados.

tfc_Database.user

DOCUMENTS 19 TOTAL SIZE 2.0KB AVG. SIZE 109B

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER

INSERT DOCUMENT VIEW LIST TABLE

Displaying documents

	_id ObjectId	name String	cc String	birth_date String	t_shirt_size String	office String	plafond String
1	5e9afdebb34eb647365a57f9	"Jose"	"14725889"	"1999-04-15"	"S"	"Porto"	"200€"
2	5e9afe51b34eb647365a5819	"António Jesus"	"148/8652"	"1999-04-15"	"L"	"Lisboa"	"200€"

Figura 6 - Tabela User na base de dados

² Interface gráfica do utilizador do MongoDB

5. Benchmarking

A Tabela seguinte tem como objetivo representar diferentes funcionalidades que as plataformas em questão têm em comum, ou não, de forma a termos uma primeira comparação com outras plataformas já disponíveis.

Os vários exemplos usados na tabela seguinte, são também plataformas para a gestão de eventos, a maioria delas estão direcionadas para o uso de aplicações móveis e possibilitam aos utilizadores criar e/ou inscrever-se em eventos que se encontrem na plataforma.

Tabela 3 - Comparação com a concorrência

Plataformas	Private Platform	Online Registration	Website Integration	Event Reports	Custom Notifications	Types of Members	Admin Aprooval	Types of Event
Sympla [2]	✗	✓	✓	✓	✓	1	✗	✓
Arlo [3]	✗	✓	✓	✓	✓	1	✗	✓
Azavista [4]	✗	✓	✓	✓	✗	1	✗	✓
A nossa plataforma	✓	✓	✓	✓	✓	2(Member & Admin)	✓	✓

A criação de uma plataforma de gestão e a sua implementação na empresa, tem vantagens em termos de segurança e manutenção.

Ao haver restrições de instalação de software existentes na empresa, não é possível a utilização de outras plataformas que tenham de ser instaladas. Ao mesmo tempo, a empresa não fica dependente de software de outra empresa que, por exemplo, pode vir a ter algum tipo de falha na proteção de dados, isto pode implicar, termos dados de colaboradores da empresa comprometidos.

Assim para complementar o tópico anterior, apenas colaboradores da empresa podem habilitar-se a realizar o registo na plataforma. Pedido que tem de ser aprovado por um dos administradores, daí temos dois tipos diferentes de utilizadores.

No caso de manutenção e posterior desenvolvimento, é também vantajoso pois podemos ter em tecnologias usadas internamente o código usado no desenvolvimento da

plataforma. Como estamos a utilizar tecnologias que já são usadas na empresa, todo o tipo de manutenção que seja necessária é feita por colaboradores da empresa sem que seja preciso adquirir serviços de uma empresa externa para a realização do mesmo. E caso haja a intenção de desenvolver mais a plataforma, por exemplo, adicionar mais funcionalidades que possam ser consideradas benéficas para a melhoria da plataforma.

Ao usar um modelo não relacional ou NoSQL iremos permitir uma grande estabilidade para uma futura expansão da nossa plataforma. Ao usar este modelo não temos de ter já um esquema definido antes de começar a implementar, o que também vai ajudar no desenvolvimento do projeto.

A escolha da framework Angular não foi imediata, houve uma disputa entre escolher Angular ³ou NestJS⁴. A escolha foi Angular pois é utilizada já na empresa e é muito procurada por outras empresas no mercado de trabalho e existe uma enorme quantidade de documentação e bibliotecas com componentes de front-end já implementadas o que nos irá ajudar a ultrapassar diversos obstáculos, o que é ideal para o desenvolvimento da plataforma.

³ <https://angular.io/>

⁴ <https://nestjs.com/>

7. Plano de testes e validação

De momento como a plataforma não está completamente implementada, é difícil de realizar testes completos, pois vai depender também de outros vários fatores como por exemplo a dependência de a insolvência do sistema organizador da prova o que de momento não é possível.

Mas é possível simular alguns tempos de execução, que neste projeto é um dos nossos principais focos, tal como diminuição do tempo gasto e simplificação de todos os processos envolvidos.

Os seguintes testes foram feitos relativamente ao registo e depois aprovação e recorremos a algumas pessoas próximas para efetuar o registo.

Tabela 4 - Testes à inscrição e duração dos mesmos

Nº Teste	Duração
Teste 1	58 segundos
Teste 2	1 minuto e 20 segundos
Teste 3	1 minuto e 3 segundos

Como podemos verificar já se verifica uma diminuição dos tempos antes propostos, mas ainda queremos realizar mais testes que de momento alguns ainda não são possíveis de se realizar.

Do ponto de vistas de um Admin:

- Tempo total de criação de um novo evento;
- Tempo total de alerta de novo eventos aos membros interessados;
- Tempos de aprovação de membro e de evento;

Do ponto de vistas de um Admin:

- Tempo de procura de um determinado evento;
- Tempo de procura e registo num determinado evento.

7. Calendário

Em relação à organização de trabalho decidimos optar por dividir em quatro etapas, sendo elas as quatro entregas do trabalho final de curso.

A primeira etapa é uma etapa de recolha de dados e escolha de tecnologias a serem utilizadas ao longo do projeto. A segunda etapa teve como objetivo tratar dos requisitos funcionais e o começo da implementação do sistema. A penúltima etapa será a continuação e finalização da implementação de todas as funcionalidades, para na última etapa conseguirmos implementar testes e corrigir bugs detetados por estes, seguindo-se a entrega final do projeto.

Na tabela 5 podemos ver o planeamento mais detalhado.

Tabela 5 - Método e planeamento do projeto.

Período de Desenvolvimento	Objetivo
20 Outubro 2019	Primeira Reunião com os professores.
21 Outubro 2019 – 23 Novembro 2019	Estudo e escolha de tecnologias.
24 Novembro 2019	Primeira entrega do relatório
25 Novembro 2019 – 31 Dezembro 2019	Recolha e análise de Requisitos
1 Janeiro 2020 – 30 Janeiro 2020	Desenvolvimento e implementação de funcionalidades.
31 Janeiro 2020	Segunda entrega do relatório.
1 Fevereiro 2020 – 25 Abril 2020	Desenvolvimento e conclusão da implementação de funcionalidades.
26 Abril 2020	Terceira entrega do relatório.
27 Abril 2020 – 18 Junho 2020	Implementação de testes automáticos e correção de bugs.
19 Junho 2020	Última entrega do relatório.

Bibliografia

- [1] Doodle. *Doodle: easy scheduling*. [online] Available at: <https://doodle.com/> [Accessed 24 Nov. 2019].
- [2] Sympla. A Plataforma Online de Eventos - Inscrições e Venda de Ingressos", *Sympla.com.br*. [Online]. Available at: <https://www.sympla.com.br/>. [Accessed: 24- Nov- 2019].
- [3] Azavista. Event Technology", *Azavista.com*. [Online]. Available at: <https://azavista.com/>. [Accessed: 24- Nov- 2019].
- [4] Arlo. *Arlo Training & Event Management Software*. [Online]. Available at: <https://www.arlo.co/>. [Accessed: 24- Nov- 2019].

Anexos

1 – Exemplo do Doodle

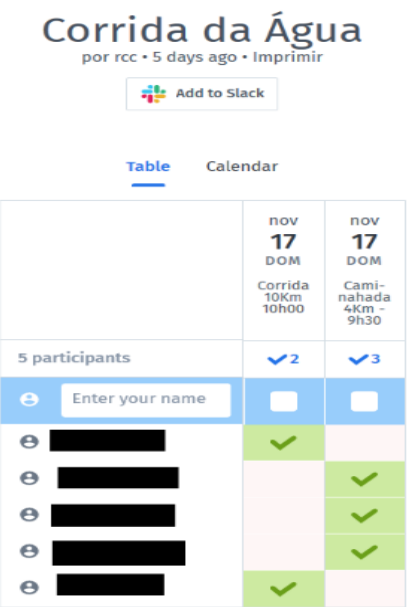


Figura 8 - Exemplo da utilização do doodle.

2 – Mapa da aplicação

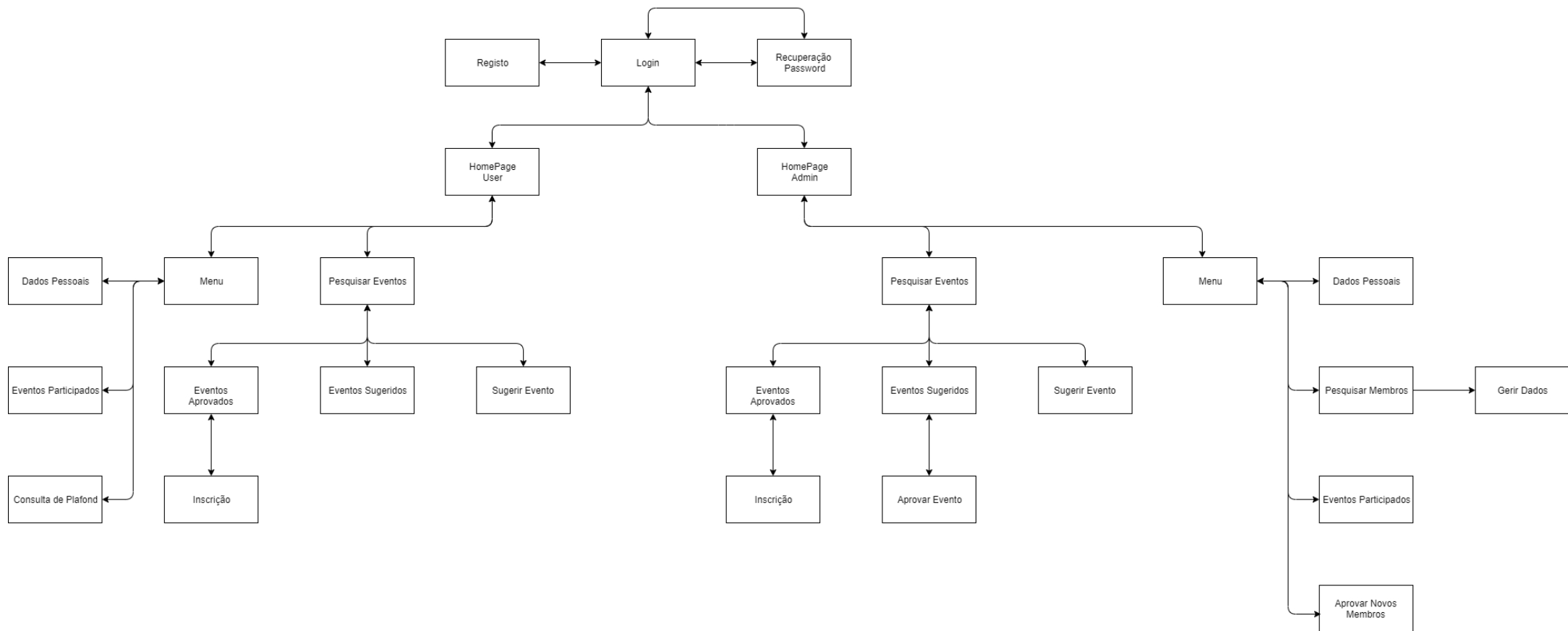


Figura 9 - Mapa da aplicação

Glossário

Angular – é uma plataforma de aplicações web de código-fonte aberto e front-end baseado em TypeScript. Desenvolvido pela Google.

Back-end – Parte de tratamento e manipulação de data de um sistema;

Common Lisp – implementação específica da linguagem de programação Lisp multi-paradigma que suporta programação funcional.

Doodle – Plataforma de criação e agendamento de eventos;

Excel – Editor de folhas de cálculo produzido pela Microsoft;

Framework - Abstração que une código de diferentes linguagens, mas comuns ao mesmo projeto e promove uma funcionalidade genérica;

Front-end – Parte de um sistema que o utilizador interage;

JSON – é um formato compacto, de troca de dados simples e rápida entre sistemas;

NestJS – Uma framework de node.js com objetivo de desenvolver aplicações eficientes e escaláveis com o uso de typescript;

NoSQL - é um termo genérico que representa o modelo de bases de dados não relacionais;

TypeScript – é um superconjunto de JavaScript desenvolvido pela Microsoft;

User Interface ou UI - Área dentro do Design que define interação entre humano e máquina;