



UNIVERSIDADE
LUSÓFONA

Plataforma de Gestão de Eventos

Lucas Silva | Pedro Santos

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	5
Resumo	6
Abstract.....	7
1. Identificação do Problema	8
2. Viabilidade e Pertinência.....	10
3. Solução Proposta	11
3.1 Back-end	12
3.2 Front-end.....	12
3.3 Entidades presentes	12
3.3 Mapa da Aplicação	13
4. Benchmarking.....	14
5. Calendário.....	16
Bibliografia.....	17
Anexos	18
Glossário	20

Índice de Tabelas

Tabela 1 - Inscrições e provas ao longo dos anos.....	10
Tabela 2 - Comparação com a concorrência	14

Índice de Figuras

Figura 1 - Dados dos Membros	8
Figura 2 - Arquitetura cliente-servidor.....	11
Figura 3 - Entidades e respectivos dados	13
Figura 4 - Método de Planeamento do projeto	16
Figura 5- Inscrição através do doodle	18
Figura 6 - Mapa da Aplicação	19

Resumo

Uma empresa tem um “clube de corrida e caminhada” o qual serve para gerir as inscrições em eventos de tipo de corrida ou de caminhada. O objetivo é criar uma plataforma que permita facilitar o controlo destas inscrições e de outras facilidades.

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

Irá ser possível também ser gerado, apenas por administradores, relatórios como por exemplo os participantes mais ativos, orçamento gasto por membro e muitos 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 diatlo, trialto, etc...

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

1. Identificação do Problema

Na empresa do professor Bruno Cipriano e do professor Rodrigo Correia, 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, participando em várias atividades de diversos eventos.

Até ao momento a gestão deste projeto é feita de maneira muito primitiva, pois são necessários vários ficheiros Excel para guardar e gerir as diversas informações existentes, que vamos mencionar à frente. Posteriormente é utilizado para realizar as inscrições dos colaboradores aos eventos através do Doodle [1], 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)

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

Na figura 1 podemos verificar a constituição dos membros, os dados presentes são: Sócio ID, Short Name, Nome Completo, Cartão de Cidadão, 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.

A maneira de guardar dados dos eventos é semelhante à dos membros. A tabela dos eventos tem então os seguintes dados: 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 anexos figura 5), 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 o projeto não tem uma boa escalabilidade.

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

Foi-nos então proposto pelos professores criarmos 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, nas quais existem no total cento e vinte funcionários, dos quais quarenta são membros e seis administradores (três de cada escritório).

Tabela 1 - Inscrições e provas ao longo dos anos.

	2017	2018	2019
Nº de Provas	10	36	25
Nº de Inscrições	121	115	48

Atualmente uma única inscrição demora por volta de sete minutos e meio, devido à procura “manual” dos dados. Este caso é um dos problemas que em conjunto com outro, como por exemplo a dificuldade de procura de eventos por parte dos membros, que em conjunto provocam a descida da aderência aos eventos, tal como podemos ver na *Tabela 1*.

Vamos então com a elaboração da 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, (devido a questões de segurança da própria empresa) previmos que a continuação da implementação e uso do novo sistema irá ser bastante facilitado, favorecendo também, a futura manutenção do mesmo.

3. Solução Proposta

A arquitetura escolhida para esta plataforma vai ser do tipo cliente-servidor (*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 e o cliente iremos usar pedidos JSON.

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

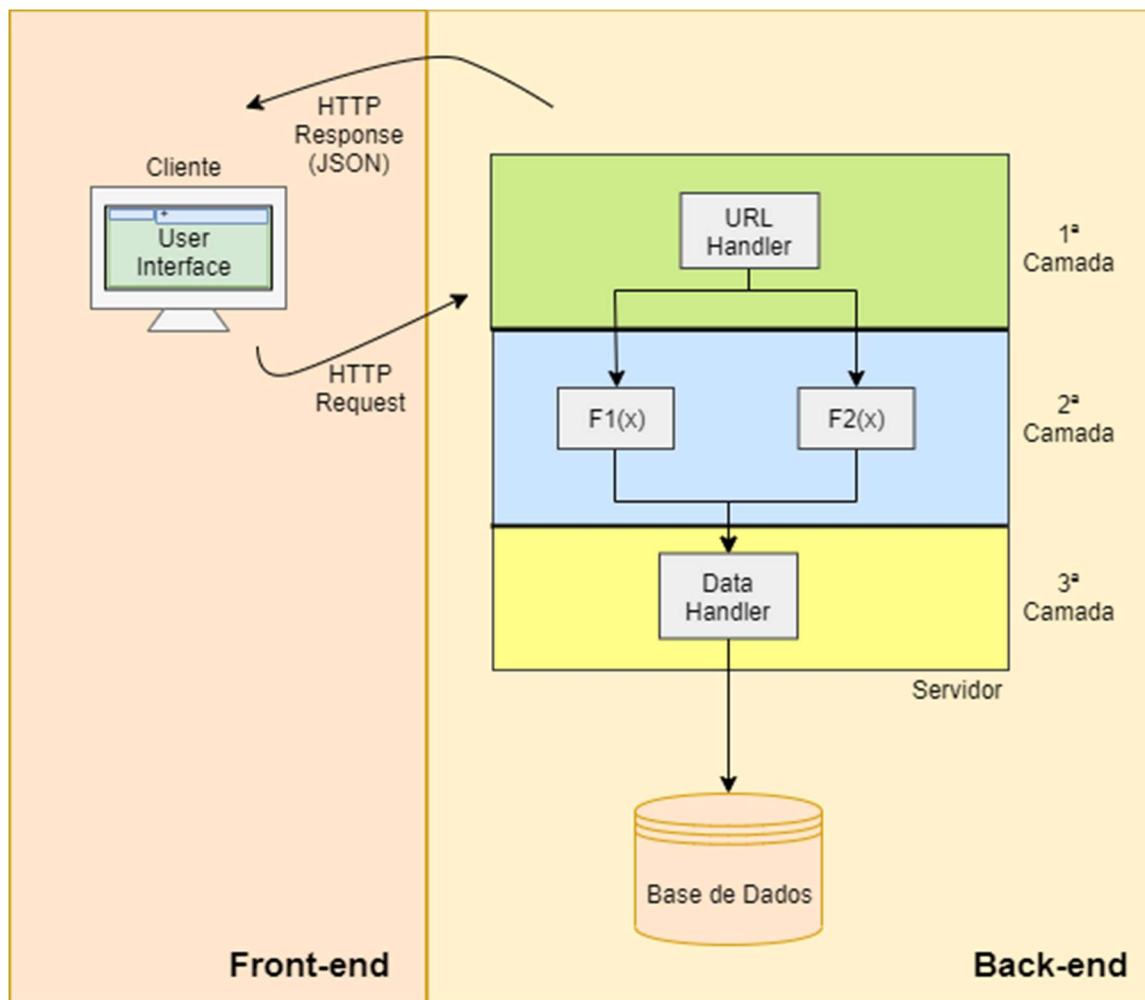


Figura 2 - Arquitetura cliente-servidor

3.1 Back-end

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

A primeira camada 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 ir buscar dados 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.

3.2 Front-end

No front-end escolhemos utilizar a framework Angular, que utiliza typescript tal como nos pretendíamos. Visto que é uma framework muito pedida no mercado de trabalho, é ideal dominar esta tecnologia, para também nos preparar.

3.3 Entidades presentes

Para primeiro contacto com o projeto, analisámos os dados partilhados pelos professores (ficheiros Excel que contém os dados dos membros e eventos como vimos anteriormente), separamos 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 3*.

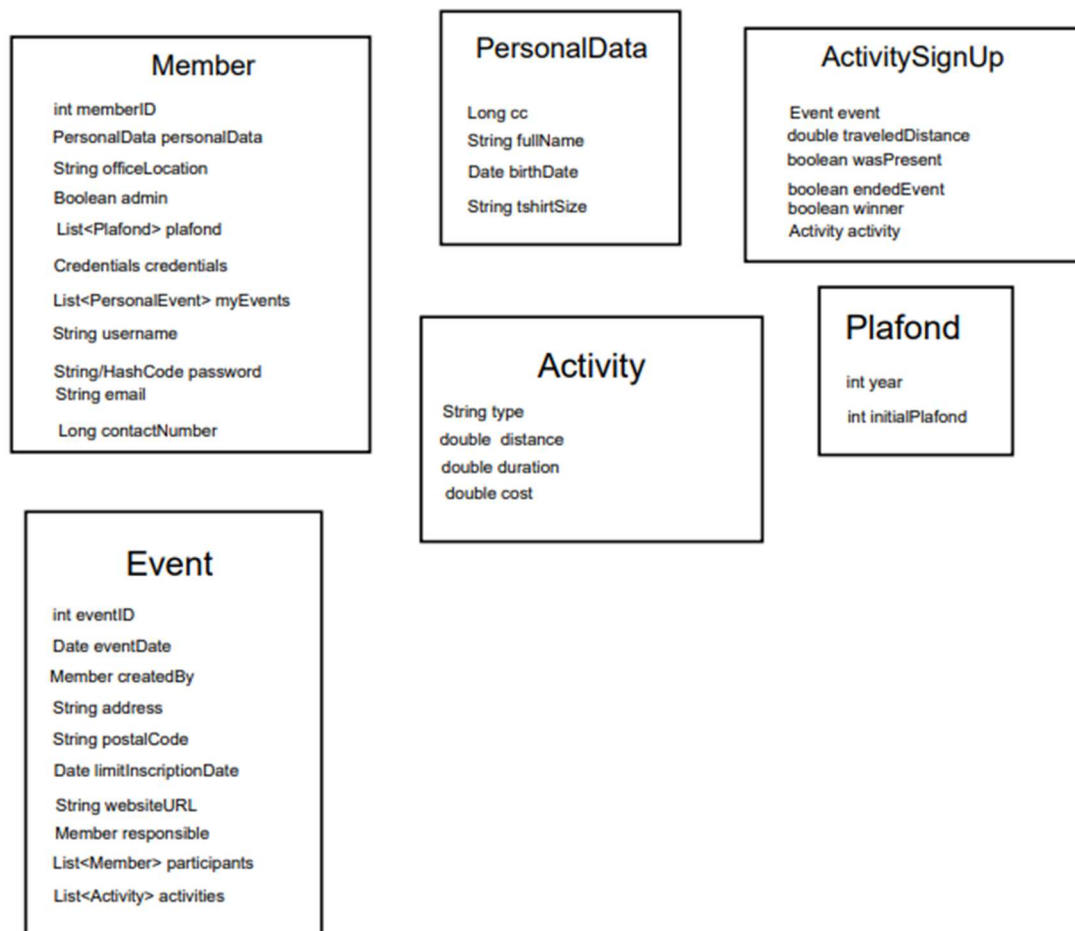


Figura 3 - Entidades e respetivos dados

3.3 Mapa da Aplicação

Elaboramos também um primeiro mapa da aplicação (ver anexos, figura 6) 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. 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 2 - Comparação com a concorrência

Plataforms	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 à plataforma. Pedido que tem de ser aprovado por um dos administradores, daí termos 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.

As vantagens de se utilizar o modelo de base de dados Relacional e não o modelo não-relacional (NoSQL), é devido a guardarmos os dados em tabelas onde os dados são organizados em colunas e linhas. Desta forma conseguimos evitar a complexidade, pois temos uma organização intuitiva para a maioria dos utilizadores.

Adicionalmente o número de utilizadores e de dados na plataforma não chega a ser justificável para utilização de um modelo não-relacional, pois o principal motivo porque se escolhe este modelo é devido à escalabilidade que se obtém.

No entanto, neste caso é preferível a confiabilidade e consistência dos dados que temos guardados, em vez da escalabilidade que poderíamos ter.

A escolha da framework Angular não foi imediata, houve uma disputa entre escolher Angular ou NestJS. A escolha foi Angular pois é muito procurada por 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.

5. 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 vai ter 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.

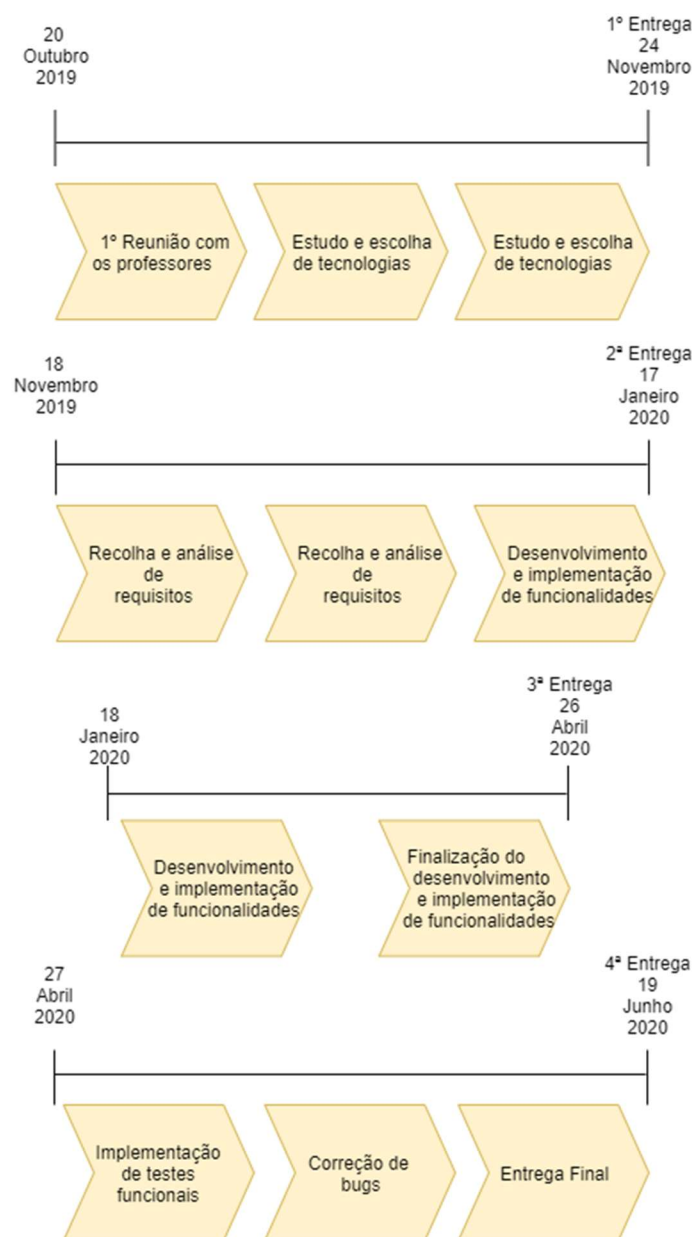


Figura 4 - Método de Planeamento do projeto

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: <https://azavista.com/>. [Accessed: 24- Nov- 2019].
- [4] Arlo", *Arlo Training & Event Management Software*. [Online]. Available: <https://www.arlo.co/>. [Accessed: 24- Nov- 2019].

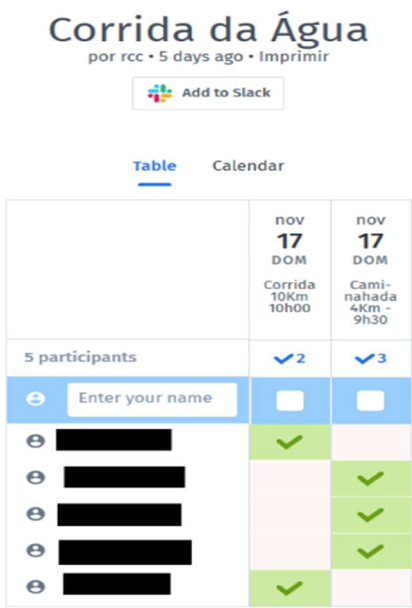


Figura 5- Inscrição através do doodle

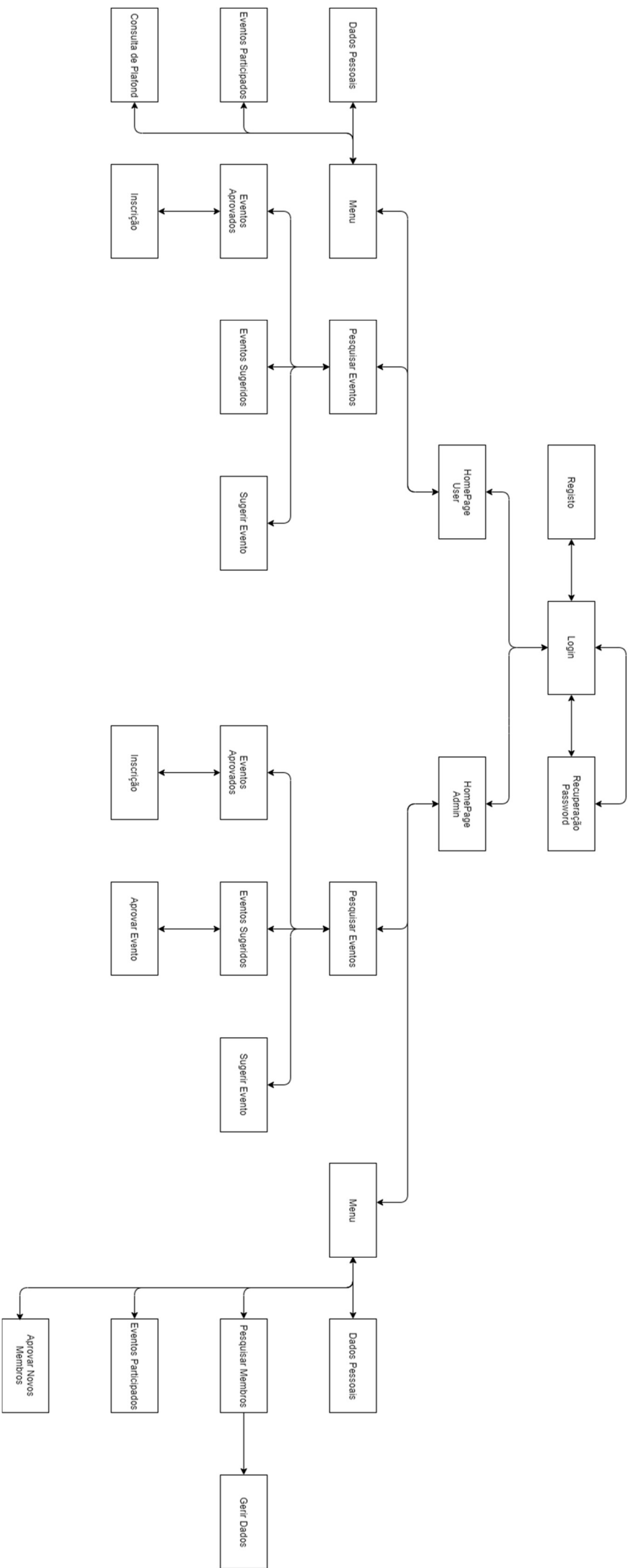


Figura 6 - Mapa da Aplicação

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 estrutura 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;