

**Universidade do Minho**

Escola de Engenharia

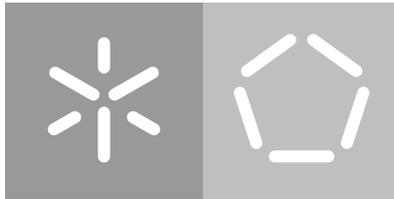
Departamento de Informática

Vasco F. S. Monteiro

**Aproximação Digital das Populações  
às suas Instituições Públicas**

**Através da Ludificação**





**Universidade do Minho**

Escola de Engenharia

Departamento de Informática

Vasco F. S. Monteiro

## **Aproximação Digital das Populações às suas Instituições Públicas**

**Através da Ludificação**

Masters dissertation

Masters Degree in Informatics Engineering

Dissertation supervised by

**Paulo Jorge Freitas de Oliveira Novais**

**Fábio André Souto da Silva**

2018



---

## AGRADECIMENTOS

---

Em primeiro lugar deixar uma grande palavra de gratidão aos meus orientadores Professor Paulo Novais e Fábio Silva pelo apoio não só ao longo do ano como nos anos anteriores. Um especial agradecimento também ao Professor Cesar Analide que, embora não sendo meu orientador, agiu como tal, prestando uma ajuda preciosa no processo de desenvolvimento e escrita desta dissertação e no caminho que levou até este ponto.

Em segundo lugar um abraço sentido aos meus amigos e colegas que, ao longo dos anos, me foram acompanhando em tristezas e alegrias no nosso percurso académico. Em especial aos grandes André Sampaio, Ricardo Ramalho, Alexey Gylytskyy e Rui Batista que, mais que amigos, são irmãos.

Um grande obrigado, também, à minha amiga Mariana Veloso, pela motivação e apoio dado em momentos cruciais não só do meu percurso académico mas também da minha vida.

Por fim, as três pessoas mais importantes do mundo: o meu pai, a minha mãe e a minha irmã, cujo amor incondicional não tem preço. A eles devo tudo e eles dedico esta dissertação como o culminar de um percurso que, por culpa própria, foi demasiado longo mas com uma reta final feliz. Tudo farei para que consiga, nesta nova etapa da minha vida, retribuir tanto ou mais o que eles fizeram por mim até agora, que foi imenso.



---

## RESUMO

---

Com o crescimento significativo de utilizadores de dispositivos móveis, nos últimos anos, mais do que a necessidade apareceu a oportunidade de criar novas plataformas e serviços digitais que não só facilitam o quotidiano das pessoas, evitando deslocamentos, filas de espera e complicações desnecessárias, como tornam a comunicação das pessoas com as instituições num processo mais rápido e cómodo. Como tal, estas alternativas estão gradualmente a complementar, em alguns casos mesmo a substituir, os métodos antigos.

Esta dissertação propõe uma solução digital, na forma de uma aplicação móvel, para aproximar as populações às suas instituições públicas recorrendo a *gamification*, isto é, transformando o processo de comunicação com uma Câmara Municipal ou Junta de Freguesia, por exemplo, recorrendo ao uso de elementos de jogos como conquistas, atribuição de recompensas, classificações, entre outras, com o objetivo de estimular a comunicação do utilizador com essas instituições permitindo que este possa ao mesmo tempo divertir-se e competir com os outros, enquanto explora e ajuda a sua cidade. Para tal, além de um serviço de participação de ocorrências na cidade, os utilizadores têm à sua disposição toda uma narrativa que os leva a completar missões, participar em eventos e conhecer gente nova bem como a conhecer verdadeiramente a sua própria cidade. Por sua vez, a instituição pública responsável pela manutenção e administração do sistema tem à sua disposição uma plataforma de administração para manter o conteúdo da aplicação atualizado para os seus utilizadores.

No decorrer desta dissertação está documentado todo o processo de desenvolvimento da aplicação móvel e do servidor *web*, que a suporta, assim como todas as decisões tomadas e as razões que as justificam. O documento contém, ainda, exemplos e explicações do funcionamento da aplicação, considerações finais sobre o projeto e ideias para trabalho futuro. Nas considerações finais é feita uma comparação dos objetivos inicialmente traçados para o projeto e o resultado alcançado, provando que os mesmos foram cumpridos com sucesso.



---

## ABSTRACT

---

The significant growth of mobile device users of the last few years, revealed the opportunity to develop new digital platforms and services which not only make people's everyday lives easier, avoiding unnecessary trips, waiting lines and complications, but also making the process behind the communication between the populations and institutions in easier and smoother. Consequently, these new alternatives are gradually complementing, in some cases even completely replacing, the old methods.

This dissertation proposes a digital solution, in the form of a mobile application, to strengthen the communication between populations and their public institutions through gamification. Turning that process into a game ensures people can have fun while they are exploring and helping their city. To this purpose, users have a report system that allows them to expose things they think are wrong with the city, such as broken structures, clogged sewers, among others. The game also has a lore that leads to extra motivation to complete missions, be part of player gatherings and events, meet new people and to better explore the cities' wonders. The entity responsible for the maintenance and administration of the system have an admin platform to better help keeping the application's content fresh and updated, allowing for a better user experience for the population.

This document addresses all the steps taken in the application's development process as well as the web server's that support said application. All the decisions have been documented and the reasons that justify them, as well as examples and *how tos* about the application, ending with some considerations about the work that has been done and some ideas for the future of the system. In the final considerations is also made a parallel between the project's goals and the final result, proving these goals were met successfully.



---

## GLOSSARY

---

**Framework** Uma framework (ou biblioteca), em desenvolvimento de software, é uma abstração que une códigos comuns entre vários projetos de software provendo uma funcionalidade genérica.

**Gamification** A aplicação de conceitos e princípios de jogos a contextos e situações onde não é usual fazê-lo. O conceito tem ganho força e notoriedade aplicado em cenários em que se pretende cativar os utilizadores, aumentar produtividade nas empresas, angariações de fundos, recrutamento e avaliação de novos trabalhadores nas empresas, entre outras situações onde se pretende motivar as pessoas a fazer algo que inicialmente não tem grandes argumentos para os convencer. Em português, Ludificação.

**Open-source** Código aberto, em português, é um modelo de desenvolvimento que promove livre licenciamento para o design, arquitectura ou funcionalidades de um produto, neste caso de software, dando a possibilidade para que qualquer pessoa possa consultar e transformar podendo ajudar a evoluir esse mesmo software.

**REST** A Representational State Transfer (REST), em português Transferência de Estado Representacional, é uma abstração da arquitectura da Web com uma sintaxe universal para identificar os recursos. Num sistema REST, cada recurso é unicamente direccionado através da sua URI (Identificador Uniforme de Recurso). É um protocolo cliente/servidor sem estado: cada mensagem HTTP contém toda a informação necessária para compreender o pedido.

**Smart Cities** Cidades inteligentes; cidades onde através do uso da tecnologia se tenta melhorar a infraestrutura urbana tornando-as mais eficientes e transformando-as em sítios melhores para viver.

**Smartphone** Palavra que não carece de grande explicação, mas representa o termo adoptado para situações em que os portugueses se referem a "telefones inteligentes" que ainda não tem termo correspondente em português.

**Tablet** Embora não precise de tradução por ser o termo utilizado para estes dispositivos em Portugal, esta palavra refere a um dispositivo pessoal em formato de prancheta que pode ser usado para acesso à Internet, organização pessoal, visualização de fotos,

vídeos, leitura de livros, jornais e revistas e para entretenimento com jogos e que apresenta uma tela sensível ao toque.

**Web/Mobile** Estes dois termos, no contexto de desenvolvimento de software, referem-se às plataformas onde o sistema/aplicação será acessível, neste caso disponível em forma de página de internet e em forma de aplicação de um dispositivo móvel como um tablet ou um smartphone, respectivamente.

---

## LISTA DE ACRÓNIMOS

---

API      Application programming interface

CSS      Cascading Style Sheets

HTML    HyperText Markup Language

HTTP    Hypertext Transfer Protocol

JPEG    Joint Photographic Experts Group

JSON    JavaScript Object Notation

PdI      Ponto de Interesse

SQL      Structured Query Language

---

## ÍNDICE

---

Agradecimentos	i
Resumo	ii
Abstract	iii
Glossário	iv
Lista de Acrónimos	vi
Índice	vi
Índice de Figuras	ix
Índice de Tabelas	xi
1 INTRODUÇÃO	1
1.1 Contextualização e Motivação	1
1.2 Objetivos	4
1.3 Estrutura do Documento	5
2 ESTADO DA ARTE	7
2.1 O que é a Ludificação/ <i>Gamification</i> ?	7
2.2 Tecnologias: Bases de Dados	10
2.2.1 Modelo Relacional	11
2.2.2 Modelo Não Relacional	12
2.3 Tecnologias: Servidor	15
2.4 Tecnologias: Aplicação Móvel	18
2.5 Trabalho Relacionado	22
3 ANÁLISE DO PROBLEMA E ESTRATÉGIA PARA LUDIFICAÇÃO DO SISTEMA	27
3.1 A Narrativa da Aplicação	28
3.2 As Tribos	29
3.3 A Loja e a Moeda do Jogo	30
3.4 Essências: os materiais do <i>gameplay</i>	31
3.5 Templos: Pontos de Interesse	32
3.5.1 Ações nos Templos	32
3.6 Funcionalidades da Aplicação	33
3.6.1 Participações	33
3.6.2 Eventos	33
3.6.3 Missões Diárias	34
3.6.4 Conquistas	34
3.6.5 Classificações	35

3.6.6	Escalões de Classificação	35
3.7	A Conta de Utilizador	37
4	IMPLEMENTAÇÃO	38
4.1	Arquitetura do Sistema	38
4.2	Base de Dados	39
4.2.1	Entidades	40
4.2.2	Preenchimento da Base de Dados: Casos de Teste	48
4.3	Servidor	52
4.3.1	API do Servidor	53
4.3.2	<i>Chat</i> e Notificações	56
4.3.3	Área de Administração	57
4.4	Aplicação Móvel	71
4.4.1	Navegação da Aplicação	72
4.4.2	Páginas da Aplicação	73
5	ANÁLISE DO TRABALHO REALIZADO E CONCLUSÕES FINAIS	87
5.1	Síntese do Trabalho	87
5.2	Ideias para Trabalho Futuro	88
6	BIBLIOGRAFIA	89

---

## ÍNDICE DE FIGURAS

---

Figura 1.1	MARKTEST: % utilizadores de smartphones em Portugal	2
Figura 1.2	PORDATA: dist. por grupos etários da população portuguesa	3
Figura 2.1	Ranking de motores de bases de dados	14
Figura 2.2	Diagrama para escolha de frameworks para <i>app development</i>	19
Figura 2.3	Exemplo: MyCity	23
Figura 2.4	Imagens das aplicações Streetspotr, FieldAgent e TaskSpotting	24
Figura 2.5	Imagens das aplicações Surveys on the go, Survey e iPoll	24
Figura 2.6	Imagens das aplicações Ingress e Pokemon GO	25
Figura 3.1	Gorintou, representação física dos cinco grandes”	29
Figura 3.2	Pirâmide de Classificação	36
Figura 3.3	Exemplo: Novo evento, parte 1	37
Figura 4.1	Arquitetura do Sistema	39
Figura 4.2	Diagrama de Relações entre Entidades da Base de Dados	40
Figura 4.3	Exemplos dos campos json, complexos, da entidade utilizador	43
Figura 4.4	Exemplo da estrutura de um ponto de interesse / templo	50
Figura 4.5	Estrutura da página de administração/painel de controlo	57
Figura 4.6	Página dos utilizadores no painel de controlo do sistema	58
Figura 4.7	Página das participações no painel de controlo do sistema	59
Figura 4.8	Estrutura da janela com detalhes da participação aberta	60
Figura 4.9	Página das parcerias no painel de controlo do sistema	62
Figura 4.10	Formulário para adicionar novo parceiro	63
Figura 4.11	Formulário para editar informações de parceira	64
Figura 4.12	Página dos locais no painel de controlo do sistema	65
Figura 4.13	Janela de edição de informações sobre local	66
Figura 4.14	Exemplo: Editar informações do local	67
Figura 4.15	Página da Loja no painel de controlo do sistema	67
Figura 4.16	Página dos Eventos no painel de controlo do sistema	69
Figura 4.17	Exemplo: Novo evento, parte 1	70
Figura 4.18	Exemplo: Novo evento, parte 2	70
Figura 4.19	Cordova: como funciona?	72
Figura 4.20	Fluxograma de navegação da aplicação	72
Figura 4.21	Aplicação: Página Inicial	73
Figura 4.22	Aplicação: Página Inicial, 2	74

Figura 4.23	Aplicação: Visitar um templo	75
Figura 4.24	Aplicação: ações nos templos	76
Figura 4.25	Aplicação: aparência do templo consoante os tributos feitos	77
Figura 4.26	Aplicação: <i>Chat</i> da aplicação	77
Figura 4.27	Aplicação: Chat da aplicação, 2	78
Figura 4.28	Aplicação: Missões Diárias	79
Figura 4.29	Aplicação: Conquistas	80
Figura 4.30	Aplicação: Classificações	80
Figura 4.31	Aplicação: Loja	81
Figura 4.32	Aplicação: Perfil do Utilizador	82
Figura 4.33	Aplicação: Perfil do Utilizador	82
Figura 4.34	Aplicação: Lista de Participações e Detalhes	84
Figura 4.35	Aplicação: Processo de nova participação	85

---

## ÍNDICE DE TABELAS

---

Tabela 3.1	As tribos e os seus ideais	30
Tabela 3.1	<i>Essências</i> , um tipo para cada elemento/tribo	31
Tabela 3.1	Escalões de classificação	36
Tabela 3.2	Pontos necessários para atingir determinado nível	37
Tabela 4.1	Atributos da entidade Tribo	41
Tabela 4.2	Atributos da entidade Freguesia	41
Tabela 4.3	Atributos da entidade Utilizador	42
Tabela 4.4	Atributos da entidade Ponto de Interesse	43
Tabela 4.5	Atributos da entidade Categoria de Ponto de Interesse	44
Tabela 4.6	Atributos da entidade Missão	44
Tabela 4.7	Atributos da entidade Evento	45
Tabela 4.8	Atributos da entidade Conquista	45
Tabela 4.9	Atributos da entidade Participação	46
Tabela 4.10	Atributos da entidade Categoria de Participação	46
Tabela 4.11	Atributos da entidade Item da Loja	47
Tabela 4.12	Atributos da entidade Parceiro	47
Tabela 4.13	Atributos da entidade Mensagem Interna	48
Tabela 4.14	Atributos da entidade Pedido de Amizade	48



---

## INTRODUÇÃO

---

O primeiro capítulo deste documento, a introdução, tem o intuito de introduzir o leitor ao contexto em que o presente trabalho se insere e a motivação para o mesmo, isto é, serve para expor as razões que levam o autor a realizar este trabalho, a que objetivos se propõe alcançar, o público alvo ao qual o produto final se destina e em que medida este produto vale a pena ser desenvolvido. Depois de explicados estes aspetos é também apresentado um resumo da estrutura do documento onde está sintetizado o conteúdo de cada um dos capítulos do mesmo.

### 1.1 CONTEXTUALIZAÇÃO E MOTIVAÇÃO

Estamos numa era em que uma grande percentagem da população, independentemente da idade, tem contacto com tecnologias das mais variadas e, em concreto, é raro encontrar aqueles que, nos dias que correm, não tenham acesso a um computador e/ou a dispositivos móveis como *tablets* ou *smartphones*. Segundo estudo realizado pelo Grupo Marktest [19] em 2015, a percentagem de portugueses que utilizam smartphones atinge os 60% e teve um crescimento de 83% em relação a estudos realizados em 2012.

Aliado a esta tendência crescente e, em particular, pela necessidade que as pessoas têm de estar em contacto constante com o resto do mundo, houve um crescimento enorme de aplicações que estão agora ao alcance de um clique em dispositivos com os quais muita gente já não consegue passar um dia sem.

Aproveitando este crescimento, surge a oportunidade de conceber novas formas de aproximar os cidadãos da cidade onde vivem às suas instituições públicas, sensibilizando e convidando todos a cuidarem da sua cidade para que esta possa ser um local cada vez mais agradável, seguro, limpo e interessante onde viver e crescer.

A par de tudo isto e com o rápido crescimento da área dos sistemas inteligentes e da inteligência artificial, o conceito, cada vez mais em voga, de Smart Cities apareceu naturalmente, muitas vezes enquadrado com os conceitos de e-Participation [7][8] e e-Government [8][9], que retratam a transição dos processos já existentes, e por vezes um pouco antiquados, para os meios digitais.

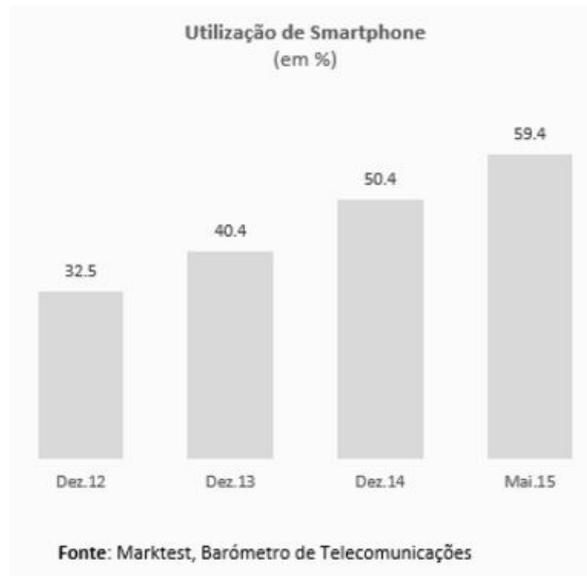


Figura 1.1: MARKTEST: % utilizadores de smartphones em Portugal

Em concreto, o conceito de e-Government ou, em português, governação eletrónica consiste na oportunidade, movida pela necessidade, de fazer chegar os produtos e serviços do Estado aos cidadãos, tirando partido da evolução da tecnologia e das próprias tecnologias de informação para encurtar a distância entre estes e o governo e tentar devolver alguma transparência a alguns destes serviços prestados pelo Estado. No fundo pode ser comparado com o comércio eletrónico na medida em que se pretende a automatização dos processos e serviços do estado, tornando-os também acessíveis em qualquer dispositivo, em qualquer lugar, vinte e quatro horas por dia, todos os dias. Este conceito tem como principal objetivo tornar o trabalho das instituições públicas significativamente mais célere e eficaz, aumentando, como supracitado, a transparência mas também a capacidade de resposta e de gestão de recursos de cada uma destas instituições. No entanto, o crescimento desta tendência está bastante condicionado pela decisão dos responsáveis pelas instituições públicas que muitas vezes tomam essas decisões com base em interesses políticos [20]; por outro lado temos a aceitação dos próprios clientes, os cidadãos. Não é novidade que o ser humano é avesso à mudança, Rosabeth Moss Kanter escreve no seu artigo [21] dez razões pelas quais as pessoas resistem a mudanças, neste caso mudanças de liderança, mas as ideias por trás mantêm-se, isto é, uma mudança é bastante significativa na medida em que:

- As pessoas estão habituadas ao atendimento em pessoa e valorizam, portanto, o poderem olhar nos olhos das pessoas que lhes estão a tratar de assuntos, por vezes extremamente importantes. Com isto, ao serem atendidos por um mecanismo automático e impessoal consideram estarem a ser menosprezadas ou que os seus problemas poderão estar a ser categorizados como menos importantes;

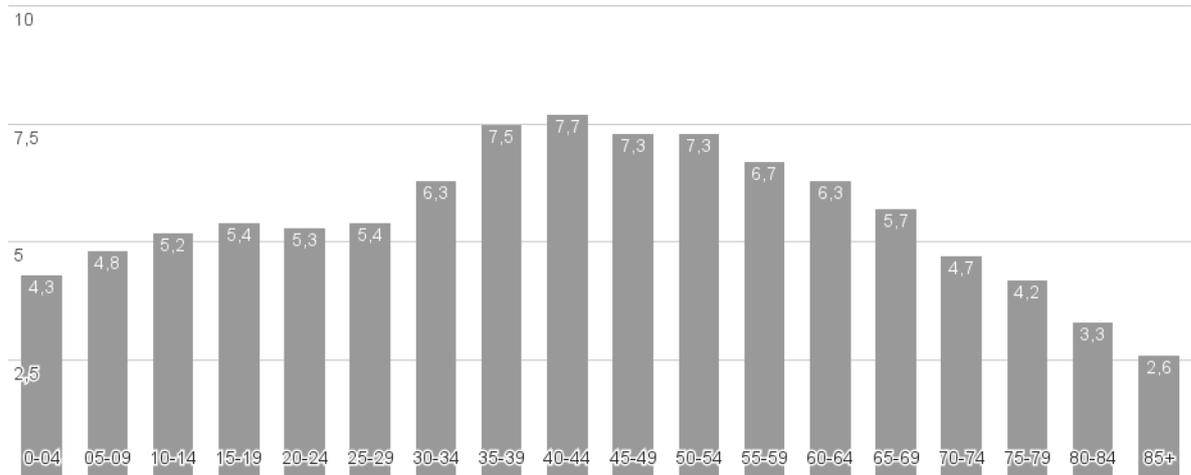


Figura 1.2: PORDATA: dist. por grupos etários da população portuguesa

- A resistência à evolução tecnológica é, também, um grande impasse para a evolução destes sistemas e processos. Tendo em conta que Portugal é dos países com a média de idades mais elevada na União Europeia e que 40.8 por cento (ver figura 1.2) da população portuguesa tem mais de 50 anos de idade, isto faz com que Portugal seja um país com uma população pouco instruída, arriscando a generalização, no uso e adaptação às tecnologias mais recentes;

Esta aversão à mudança, por vezes confundida com nostalgia, contribui juntamente com os habituais entraves políticos, como a excessiva burocracia que acompanha este e qualquer tipo de mudanças no funcionamento dos órgãos públicos portugueses, para o atraso na evolução e aceitação de propostas desta natureza. Apesar do mencionado anteriormente e no contexto apresentado, é proposto o desenvolvimento de uma plataforma que permita às instituições públicas das cidades criar eventos ou desafios para os seus munícipes participarem e se divertirem enquanto cuidam da sua cidade. Pretende-se que uma Câmara Municipal, por exemplo, consiga aproximar-se de todos os seus munícipes, incluindo os que moram na periferia da cidade e que, no caso dos utilizadores, estes se sintam importantes, ouvidos e envolvidos no que diz respeito ao funcionamento da sua cidade. Com isto, estas instituições ganham a possibilidade de estudar o apoio que os seus munícipes poderão dar a certas iniciativas como, por exemplo, construção de algo novo ou alteração de algo existente, podendo perceber que caminhos se poderão percorrer em direção ao futuro como uma espécie de estudo de mercado diário e constante, aliado a uma ferramenta de promoção não só da própria cidade e dos seus valores como também dos seus espaços comerciais, culturais, artísticos e de lazer. Para tal pretende-se desenvolver um sistema não intrusivo [69] que, mesmo sendo orientado e pensado para benefício do utilizador, possa oferecer estas informações às entidades administradoras do sistema.

## 1.2 OBJETIVOS

Este projeto tem como principal objetivo desenvolver e apresentar uma proposta viável para uma aplicação móvel, com suporte *web*, que permita a instituições públicas ou privadas como, por exemplo, câmaras municipais, aproximar-se das pessoas e estudar as suas tendências através do uso métodos de ludificação, isto é, usando este processo componentes naturalmente associadas aos jogos, onde os utilizadores podem ser recompensados por executar tarefas. Pretende-se que esta plataforma esteja acessível ao maior número de pessoas, seja o mais intuitiva possível, com um design simples e apelativo para que a interação com a mesma não seja uma tarefa fastidiosa, complicada e excessivamente complexa. O utilizador terá de cumprir missões que serão constituídas por um conjunto de tarefas a cumprir, com recompensas finais que serão proporcionais à missão cumprida. A plataforma funcionará com um sistema de pontos e conquistas aliado a uma narrativa envolvente, cujo objetivo é tornar as atividades apelativas e instigar um espírito competitivo saudável, para manter o interesse dos utilizadores na aplicação.

Delineado o objetivo principal, o conjunto de objetivos complementares que o compõem são:

1. O sistema deve compreender uma base de dados de pontos de interesse ou estratégicos que servirão de ponto de partida para as missões e eventos;
2. Os utilizadores da plataforma devem ser capazes de juntar pontos a uma velocidade, suficientemente, grande para que os possam trocar por recompensas na loja *in-app*, e ao mesmo tempo poderem divertir-se no processo;
3. O sistema deve dar à entidade responsável pela manutenção do mesmo, por exemplo uma câmara municipal, a possibilidade de gerir os eventos, parceiros, produtos da loja, missões;
4. O sistema deverá permitir aos utilizador participarem ocorrências de algo que achem estar mal, errado ou estragado na cidade, bem como de permitir à entidade administradora fazer a gestão dessas participações;
5. A plataforma deve estar construída de forma à gestão da mesma estar construída de forma hierárquica, permitindo à entidade principal criar sub-entidades que possam ficar responsáveis pela atualização mais regular ou introdução de conteúdo mais específico, evitando assim que o conteúdo da aplicação fique estagnado.
6. A interface deverá ser simples e intuitiva;
7. A entidade responsável pelo sistema deverá ser capaz de conseguir coletar informações sobre os seus utilizadores, como tendências ou opiniões, tendo sempre

em consideração o enquadramento legal deste tipo de ambientes, assegurando a privacidade dos utilizadores e a proteção dos dados adquiridos sobre os mesmos, na linha do apresentado em [68].

É com este conjunto de objetivos em mente que irá ser criada uma narrativa que motive o envolvimento dos utilizadores com as atividades da cidade e, conseqüentemente, a plataforma que, através dessa narrativa, tenha meios para cativar o interesse desses utilizadores e, por fim, permitir à entidade administradora extrapolar sobre as tendências da população com a utilização prolongada do sistema. Através desta observação dos seus utilizadores, a entidade administradora do sistema tem, também, a possibilidade de estar continuamente a melhorar o sistema para que este possa oferecer cada vez mais e melhores serviços a esses utilizadores [71, 72].

### 1.3 ESTRUTURA DO DOCUMENTO

A estrutura do presente documento está dividida em cinco capítulos principais. Começa com uma introdução do assunto e com um estudo sobre o estado da arte. De seguida é feita uma análise do problema e delineada uma estratégia para a ludificação do sistema e a sua implementação. Por fim é realizada uma análise do trabalho realizado e tecidas as conclusões finais sobre o mesmo.

No primeiro capítulo é feita a introdução ao tema do trabalho, bem como o contexto onde se enquadra e a motivação por trás do mesmo. É o capítulo onde também são definidos as metas, principais e secundárias, que este trabalho pretende alcançar e uma breve e superficial descrição da estrutura do presente documento.

No capítulo do estado da arte, pretende-se explicar cuidadosamente os conceitos teóricos que sustentam a proposta apresentada no presente documento, bem como exemplos práticos, e já existentes, que explicam de melhor forma alguns dos componentes que se pretende implementar na plataforma a desenvolver; exemplos que ilustram como algumas das técnicas a utilizar são aplicadas em situações do dia-a-dia, com pessoas reais.

De seguida é proposta uma solução para o problema, levantado na introdução, que envolve a ludificação do sistema e é exposta uma narrativa que pretende envolver o utilizador num mundo virtual com o objetivo de o cativar a realizar tarefas em troca de recompensas e de ajudar a sua cidade. Depois de explicada a solução encontrada segue-se o capítulo da implementação, onde são apresentadas e justificadas as escolhas relativas à tecnologia utilizada, bem como diagramas detalhados da arquitetura geral do sistema e das entidades que o tornam possível. É, também, apresentada a API do servidor que suporta a aplicação móvel e são, ainda, apresentados exemplos tanto da área de administração do sistema como do funcionamento da aplicação móvel.

O capítulo final apresenta uma avaliação ao trabalho realizado, mediante os objetivos

propostos no capítulo introdutório e são, ainda, apresentadas algumas conjeturas sobre possível trabalho futuro.



---

## ESTADO DA ARTE

---

Neste capítulo é elaborada uma análise sobre os conceitos fundamentais das tecnologias necessárias à realização do projeto, bem como aos conceitos fundamentais que servirão para encontrar uma solução para os problemas expostos, nomeadamente a ludificação. Inicialmente é brevemente analisado o conceito de *gamification*, ou ludificação, uma vez que o conceito está na base de todo o projeto. De seguida é feita uma síntese das tecnologias utilizadas no mercado de trabalho para cada um dos componentes previstos na arquitetura do sistema: base de dados, servidor e aplicação móvel. Por último, são avaliadas aplicações ou conceitos do mercado que poderão servir de inspiração para funcionalidades da aplicação.

### 2.1 O QUE É A LUDIFICAÇÃO/*gamification*?

*Gamification* ou, em português, Ludificação é um termo relativamente informal para descrever o uso de elementos de jogos, como recompensas, classificações e missões, em sistemas ou ambientes sem qualquer relação com jogos, com o objetivo de melhorar a experiência do utilizador e, também, o seu grau de envolvimento e motivação [28]. Apesar de implementações da ludificação em novas áreas e com novas ideias aparecerem todos os dias, o conceito em si tem mais de cem anos.

Entre as muitas empresas que usam esta técnica, houve um conjunto de empresas que criaram plataformas de ludificação como *Bunchball*[66] que, em 2007, foi a primeira que ofereceu mecanismos ou componentes de jogos como serviços, com clientes como a *Playboy* e a *USA Network*. Outro exemplo é a empresa *Badgeville* que começou em 2010 e apenas num ano juntou milhões de dólares em capital de risco e que também oferece serviços de ludificação. Grandes nomes do mercado a Microsoft, IBM e a Deloitte, entre outras, começaram a aplicar a ludificação a diversas aplicações e processos. O MIT (Massachusetts Institute of Technology), uma das mais conceituadas universidades não só nos Estados Unidos da América mas também a nível mundial, tem um programa chamado *Education Arcade* cujo objetivo é explorar o interesse natural das pessoas nos jogos e transformar o processo de educação de maneira a motivar os alunos. "As pessoas que jogam exibem regularmente persistência, atenção a pormenores/detalhes e capacidade de tomar inicia-

tiva, arriscar e resolver problemas, tudo comportamentos que, idealmente, deveriam ser demonstrados na escola.”[58]. Todos os anos milhares de alunos por todo o mundo exibem dificuldades em passar de ano ou de acabar cadeiras ou disciplinas devido ao ambiente que geralmente se vive nas escolas que leva a desencorajamento, tentativas de contornar as regras e que, muitas vezes, levam à desistência de cursos ou planos escolares.

Estudos realizados, a nível mundial, revelam que mais de 1200 milhões de pessoas jogam algo, dos quais perto de setecentos milhões jogam *online*[59]. Só nos EUA, para cima de cinco milhões de pessoas despendem mais de quarenta e cinco horas semanais a jogar e estima-se que a população mundial, no seu todo, passe mais de três mil milhões de horas a jogar em consolas, computadores, telemóveis ou noutros dispositivos. Há inúmeros artigos e discussões sobre os benefícios que a prática dos videojogos, em geral, pode providenciar às pessoas. A repetição de ações, prática muito comum nos videojogos, ajuda a desenvolver e treinar a memória e há ainda casos onde pode ajudar na recuperação das vítimas de acidentes vasculares cerebrais. Jogos que requeiram reações rápidas ajudam a treinar os reflexos e a também a capacidade de um indivíduo de tomar decisões rápidas e em ambientes de stress. Jogos de lógica ajudam a ativar o córtex pré-frontal, área do cérebro humano responsável pela tomada de decisões. Entre estes benefícios podem também encontrar-se a possibilidade de retardar o envelhecimento, ajudam na coordenação óculo-manual, a possibilidade de potencialização do interesse por história e cultura (através das narrativas dos jogos), a possibilidade de tratamento de pacientes com dislexia, treino de matemática e línguas (no caso de jogos online) bem como a melhoria das suas habilidades sociais e mesmo a criação de novos relacionamentos de amizade, tipicamente nos jogos que envolvem trabalho de equipa ou trocas de bens. Os videojogos podem, também, treinar a capacidade de atenção e são já usados em certas clínicas e hospitais para ajudar no processo de fisioterapia ou em casos de lesões, por ajudarem os pacientes a abstraírem-se da dor e do desconforto [61, 62, 63].

A acrescentar a todos estes benefícios, completar tarefas, subir de nível, ganhar uma recompensa, ou jogo ou um torneio de videojogos faz aumentar o nível de dopamina, um neurotransmissor que tem influência nos níveis de atenção, capacidade de aprendizagem e na memória de um ser humano [64]. Este químico regula, como já foi referido, o prazer no cérebro obtido pela prática de uma atividade ou ocupação e que motiva o ser humano a procurar repetir a experiência que lhe proporcionou esse prazer. Como tal, o uso de elementos dos videojogos em cenários onde não é habitual o seu uso (*gamification*), como nas escolas ou nos locais de trabalho, tem como objetivo oferecer esse mesmo tipo de experiência para motivar as pessoas na realização de tarefas tipicamente fastidiosas.

Posto isto, que tipo de elementos de videojogos podem ser aproveitados para serem aplicados a áreas onde não é normal fazê-lo?

1. **Progressão:** Provas visuais ou palpáveis da evolução e do sucesso

- **Níveis:** Evoluir e desbloquear conteúdo;
  - **Pontos:** Servem de contador visual ou prova da evolução.
2. **Investimento:** sentir orgulho no trabalho realizado e, principalmente, sentir que o tempo foi bem despendido
- **Conquistas:** reconhecimento público pelo trabalho realizado;
  - **Colaboração:** trabalhar em equipa para alcançar objetivos;
  - **Objetivos significativos:** sentir que o trabalho levou à realização de algo incrível ou transcendente;
  - **Viralidade:** sentir vontade de envolver outros, sejam amigos, familiares ou conhecidos.
3. **Cascading Information Theory:** a teoria de que a informação deve ser divulgada em quantidades pequenas para que seja entendida devidamente, a cada instante, durante a narrativa de um videojogo [65]
- **Bónus:** receber recompensas das quais não se está à espera;
  - **Exploração:** investigar o desconhecido com o objetivo de aumentar a cultura e o conhecimento;
  - **Contagens decrescentes / Eventos Temporizados:** resolver problemas com tempo contado;
  - **Aversão à perda:** participar para evitar perder o que foi conquistado até ao momento;
  - **Continuidade:** jogar/participar até alcançar um ou mais objetivos para poder abraçar novos;
  - **Multidisciplinaridade:** trabalhar para objetivos que requeiram conhecimentos em várias áreas para serem alcançados.

Certas tarefas podem ser dadas a um indivíduo sem ele reparar que as está a realizar quando são contextualizadas em ambientes de jogo como, por exemplo, motivar uma pessoa a criar algo seja escrito ou visual como criar cenários no jogo *Minecraft*; permitir que um aluno aprenda alguma da história do império Otomano ao incorporar o personagem principal do jogo *Assassin's Creed Revelations* enquanto vagueia pelas ruas da antiga Constantinopla; por um indivíduo a treinar e a pensar em probabilidades em jogos como *Dungeons & Dragons*; ao jogar xadrez os jogadores estão constantemente a analisar as jogadas do seu adversário tentando, não só, pensar como eles mas também perceber os padrões das suas ações e decisões para tomarem as suas próprias; a ver o mundo com outros olhos incorporando diferentes papéis: na perspetiva de um presidente da câmara em *SimCity*,

como um treinador de futebol em *Football Manager* ou a pensar como um detetive no jogo de tabuleiro *Cluedo*. Estes são apenas alguns exemplos de como se pode envolver os jogos, ou elementos destes, nas tarefas do quotidiano, onde as pessoas podem ganhar ou desenvolver competências enquanto estão abstraídos na narrativa do jogo e se divertem.

Como já foi referido, a gamification pode ser aplicado em diversas áreas como no trabalho, na educação, na política, no desenvolvimento de tecnologia, na saúde e *fitness*, no marketing e até no processo criativo ou na discussão de novas ideias. Cada vez mais são realizados estudos e experiências com o objetivo de descobrir novas aplicações para a ludificação, de preferência em áreas onde ainda não foi aplicada.

Em particular, o caso de estudo desta dissertação passa por tentar aplicar o conceito de gamification à interação e à participação dos cidadãos na governação de uma cidade, também conhecido como e-participation ou e-government [7, 8, 27], investigando como é que elementos de videojogos, como premiar o utilizador pelas suas conquistas promovendo sistemas de recompensa e oportunidades para interações sociais, podem ser utilizados para estimular um diálogo contínuo entre os cidadãos e a sua cidade [29].

Atualmente existem municípios que já reconheceram a ludificação da participação dos cidadãos como uma boa forma de envolverem os mesmos, de forma ativa na vida política (e não só) da cidade e, como resultado disso, começaram a desenvolver aplicações que utilizam essas metodologias, como é o caso da câmara municipal de Guimarães.

## 2.2 TECNOLOGIAS: BASES DE DADOS

Desde os primórdios da informática que um dos aspetos mais importantes e mais focados pelos investigadores é a memória e a capacidade de armazenagem de informação. Embora diferentes, por vezes, em tecnicidades e métodos de implementação a maior parte dos computadores vem equipada com *hardware* que permite o processamento de informação e guardar essa mesma informação para futuros e alternativos usos. Com o passar dos anos foram desenvolvidos muitos métodos para armazenar informação e as primeiras bases de dados computadorizadas começaram a aparecer nos anos sessenta. Nos anos setenta apareceram as primeiras propostas para as bem conhecidas bases de dados relacionais, até que nos anos oitenta foi instituído o SQL como a linguagem padrão para inquirir as bases de dados. As bases de dados relacionais rapidamente se tornaram um sucesso mundial e no início dos anos noventa apareceram diversas ferramentas para uso pessoal como ODBC e Excel/Acess. O aparecimento da Internet levou a um crescimento exponencial da indústria das bases de dados o que conduziu a um crescimento do investimento em negócios *online* ou mesmo na construção dos primeiros web sites como cara das empresas na Internet.

No mundo atual tornou-se quase impossível pensar numa aplicação que não utilize pelo menos uma base de dados, desde ferramentas ligadas à indústria até a um simples jogo

milhares de aplicações tratam, transformam, guardam e reutilizam dados múltiplas vezes. Cada sistema de base de dados implementa um diferente modelo que determina a estrutura lógica da forma como os dados vão ser tratados. A determinação destes modelos é, geralmente, o primeiro passo no processo de desenvolvimento de qualquer aplicação e, como tal, o fator mais determinante na forma como a aplicação, com base de dados, vai trabalhar e manusear a informação com a qual lida.

Apesar das bases de dados relacionais oferecerem muita flexibilidade e serem extremamente poderosas contêm certos problemas para os quais ainda não há solução, dentro deste tipo de modelo de base de dados; mais recentemente, as bases de dados não relacionais, **NoSQL**, começaram a ganhar popularidade, com a promessa de resolverem esses problemas, oferecendo um conjunto muito interessante de funcionalidades extra, eliminando por completo o estilo estritamente estruturado do modelo relacional. Estes sistemas de bases de dados oferecem uma maneira, livre de forma e estrutura, de trabalhar com a informação e isso traz um nível de flexibilidade significativamente maior que o modelo relacional.

### 2.2.1 Modelo Relacional

Primeiro, e como já foi referido, uma das grandes diferenças entre os dois sistemas, relacional e não relacional, é a forma como está estruturada a informação. Uma base de dados relacional, ou SQL, é de uma natureza mais rigorosa e estrita na maneira como armazena a informação. Estas bases de dados consistem num conjunto de tabelas, com linhas e colunas, onde cada linha representa uma entrada ou instância da entidade ao passo que cada coluna representa um atributo dessa entidade. A relação entre as tabelas e os tipos dos campos (colunas) chama-se *schema*, e estas esquema tem de ser claramente definido antes de qualquer tipo de informação possa ser armazenada.

Para que uma base de dados relacional seja eficiente, a informação nela armazenada tem de estar estruturada de um modo extremamente organizado. Um esquema bem planeado tem como objetivo minimizar a redundância da informação (informação repetida) e previne que as tabelas não estejam sincronizadas. Um esquema mal planeado poderá exigir mudanças ao longo de tempo que podem mexer com o funcionamento de toda a aplicação.

A linguagem de programação utilizada pelas pessoas que trabalham com bases de dados chama-se **Structured Query Language** ou SQL. Numa base de dados SQL como MySQL, Oracle, PostgreSQL ou a IBM DM2, essa linguagem permite executar pedidos, conhecidos como *queries*, através dos quais se pode fazer pedidos de informação, atualizações de dados e criar ou apagar entradas nas tabelas. Hoje em dia continuam a ser extremamente usadas devido ao imenso suporte que ainda têm, através da enorme comunidade de programadores que as usam e as continuam a desenvolver e a ajudar na sua evolução, o que é uma vantagem enorme para qualquer programador, novato ou experiente, ao deparar-se

com problemas.

A seguinte lista apresenta algumas das mais populares alternativas de bases de dados SQL e sistemas de administração de bases de dados relacionais:

- **MySQL**[39] - A base de dados, *open-source*, mais popular do mercado, excelente para projetos pequenos e médios como web sites e blogues, por exemplo;
- **PostgreSQL**[42] - Sistema de administração de bases de dados, orientado a objetos, que usa linguagens como Perl e Python além de código SQL;
- **Oracle**[40] - Um sistema de administração de bases de dados, orientado a objetos, escrito em C++. É uma solução extremamente completa que, para quem tiver orçamento, oferece um ótimo serviço de apoio ao cliente e confiabilidade;
- **IMB DB2**[41] - Uma família de produtos da IBM que está concebida para o manuseamento e análise avançada de enormes quantidades de dados;
- **MS SQL Server**[43] - Ferramenta de administração de bases de dados, de nível empresarial, desenvolvida pela MicroSoft.

Quando se trabalha com dados consistentes e quando não se espera um crescimento muito acentuado o uso de bases de dados relacionais é aconselhado, mais uma vez, pela sua vasta documentação e suporte.

### 2.2.2 Modelo Não Relacional

A abordagem do modelo não relacional consiste em abandonar as restrições descritas no modelo relacional, aumento imenso o grau de flexibilidade na maneira como é armazenada, requisitada e usada a informação. Quando a natureza dos dados não é totalmente conhecida, ao planear o projeto, ou quando se lida com quantidades astronómicas de dados não estruturados torna-se impossível desenvolver um esquema relacional que defina esses dados, e é esta a motivação para as bases de dados não relacionais. Pense-se nas bases de dados não relacionais como pastas onde se armazena informação relacionada de todos os tipos.

Informação não estruturada pode incluir dados de sensores, fotografias, informação baseada em localização, atividade *online*, métricas de utilização, entre outras. A necessidade de armazenar este tipo de informação levou á criação das bases de dados não relacionais, conhecidas como **NoSQL** (Not Only SQL).

Estas bases de dados, em vez de tabelas, usam *documentos*; desta forma dados não estruturados como artigos, fotografias, informação de redes sociais, vídeos e o conteúdo de uma

publicação num blogue, por exemplo, podem ser armazenados num único documento e podem ser facilmente consultados mas, no entanto, não estão categorizados em campos como numa base de dados relacional. Contudo, este tipo de armazenagem requer processamento posterior extra além de ocupar mais espaço do que se estivesse organizado em SQL.

Outra grande vantagem das bases de dados NoSQL, particularmente para os programadores, é a facilidade de acesso. As bases de dados SQL estão normalmente associadas a aplicações que têm de ser escritas em linguagens de programação orientadas a objetos como Java, PHP ou Python, por exemplo, ao passo que as bases de dados NoSQL ultrapassam esta limitação através do uso de APIs o que permite aos programadores executarem pedidos, à base de dados, sem terem de aprender SQL ou perceber a arquitetura do seu sistema de base de dados. Atualmente existem quatro grandes tipos de bases de dados não relacionais que são:

1. **Modelo Chave-Valor:** A opção NoSQL menos complexa que armazena a informação indexando chaves e valores respetivos. Alguns exemplos deste tipo de bases de dados são **Cassandra, Azure e LevelDB**;
2. **Armazenamento em Colunas:** Armazena em tabelas, como o relacional, mas em colunas em vez de linhas, no entanto é mais que uma tabela invertida uma vez que o seccionamento das tabelas possibilita grande escalabilidade e performance. Exemplos: **HBase, BigTable, HyperTable**;
3. **Orientada a Documentos:** é uma versão mais complexa do modelo Chave-Valor onde cada documento tem a sua própria informação e a sua própria chave, usada para o reconhecer e o requisitar. Exemplos: **MongoDB, CouchDB**;
4. **Orientada a Grafos:** para a informação que está inter-relacionada e que é melhor representada como um grafo, permite um grau de complexidade enorme. Exemplos: **Polyglot, Neo4J**.

Categorizados os tipos de bases de dados relacionais pode-se falar sobre as soluções mais utilizadas no mercado com uma breve explicação do motivo dessa utilização:

- **MongoDB**[44] - o sistema NoSQL mais utilizado, especialmente entre *startups*. Como já foi referido, é uma base de dados orientada ao documento com documentos dinâmicos ao estilo dos documentos JSON. É utilizado em sites como Ebay e Foursquare. É gratuito e tem muito suporte e uma comunidade cada vez maior;
- **Oracle NoSQL**[47] - A abordagem NoSQL da Oracle para acompanhar as tendências do mercado, não é gratuito;

- **Apache Cassandra**[48] - base de dados distribuída com a capacidade de tratar enormes quantidades de dados, altamente escalável, ótima solução para quando se antecipa um grande crescimento numa aplicação. É usado por empresas como Instagram, Comcast, Apple e Spotify;
- **Apache CouchDB**[45] - uma base de dados ótima para a web, armazena os dados em documentos formatados em JSON, usa JavaScript para indexar, combinar e transformar os documentos, e HTTP para a sua API;
- **Apache Hbase**[46] - uma base de dados não relacional de armazenamento em colunas escrita em Java, apropriada para trabalhar com quantidades enormes de dados.

Quando se está a desenvolver uma aplicação do lado de servidor, concebida e esquematizada para ser rápida e eficaz, uma base de dados não relacional evita que a informação atrase essa rapidez. A grande motivação deste tipo de bases de dados é, no entanto, o *Big data*, isto é, a quantidade enorme de dados que está disponível *online* para tratar e extrapolar que tem de ser armazenada algures. Portanto quando se quer armazenar quantidades muito grandes de informação, com pouca ou nenhuma estrutura, quando se quer tirar o maior proveito do armazenamento e computação na *cloud* e, principalmente, quando se pretende um arranque rápido e um desenvolvimento rápido num projecto, as bases de dados não relacionais são a escolha ideal.

315 systems in ranking, January 2017

Rank			DBMS	Database Model	Score		
Jan 2017	Dec 2016	Jan 2016			Jan 2017	Dec 2016	Jan 2016
1.	1.	1.	Oracle +	Relational DBMS	1416.72	+12.32	-79.36
2.	2.	2.	MySQL +	Relational DBMS	1366.29	-8.12	+67.03
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1220.95	-5.70	+76.89
4.	↑5.	4.	MongoDB +	Document store	331.90	+3.22	+25.88
5.	↓4.	5.	PostgreSQL	Relational DBMS	330.37	+0.35	+47.97
6.	6.	6.	DB2	Relational DBMS	182.49	-1.85	-13.88
7.	7.	↑8.	Cassandra +	Wide column store	136.44	+2.16	+5.49
8.	8.	↓7.	Microsoft Access	Relational DBMS	127.45	+2.75	-6.59
9.	9.	↑10.	Redis +	Key-value store	118.70	-1.20	+17.54
10.	10.	↓9.	SQLite	Relational DBMS	112.38	+1.54	+8.64

Figura 2.1: Ranking de motores de bases de dados

Como se pode verificar na figura 2.1, retirada do *web site* <http://db-engines.com/en/ranking>, o *top* dez das soluções de bases de dados mais utilizadas é maioritariamente constituído por soluções baseadas no modelo relacional, em grande parte porque as soluções NoSQL ainda são bastante recentes. Na mesma figura pode verificar-se que as três alternativas NoSQL presentes, MongoDB, Redis e Cassandra, estão em ascensão e a tendência do

mercado é que esta ascensão seja cada vez mais evidente devido ao volume de dados, cada vez maior, que há necessidade de armazenar e conseqüentemente de tratar e utilizar.

### 2.3 TECNOLOGIAS: SERVIDOR

As *frameworks web* para o lado do servidor disponibilizam funcionalidades que tornam facilitada a tarefa de manter e escalar aplicações web. Elas oferecem ferramentas e bibliotecas que simplificam certas tarefas no processo de desenvolvimento de uma aplicação web como o tratamento das rotas (URLs), interação com as bases de dados, autenticação do utilizador através de sessões e a formatação dos resultados em JSON, HTML ou XML, por exemplo, e ajudam ainda a melhorar a segurança contra ataques web. Não é obrigatório utilizar uma *framework* do lado do servidor, no entanto o uso de uma garantirá que o processo de desenvolvimento seja mais célere e descomplicado.

Os servidores web e os *browsers* comunicam através do protocolo HTTP, os servidores esperam por pedidos e retornam informação através de respostas HTTP. Estas *frameworks* permitem, através de uma sintaxe pouco complexa, gerar código para trabalhar estes pedidos e respostas. Maior parte dos sites e aplicações web terão disponíveis vários recursos, acessíveis através de endereços diferentes e tratar todos estes caminhos numa só função pode ser complicado gerir. Novamente, estas *frameworks* incluem mecanismos para facilitar a gestão destas rotas ou caminhos.

Cada pedido pode estar codificado de várias maneiras: um *GET*, para obter ficheiros ou informação do servidor, pode conter codificada a informação necessária para realizar a operação em parâmetros ou na estrutura do endereço; um *POST*, para atualizar conteúdo no servidor por exemplo, poderá conter a informação necessária a essa atualização codificado no corpo do pedido. Cada pedido HTTP pode também conter informação sobre uma sessão ou sobre um utilizador em forma de *cookie*, extremamente útil para cenários onde é necessário distinguir que utilizador está a realizar uma dada acção.

Estas *frameworks*, além de possuírem mecanismos para aceder facilmente à informação contida nestes pedidos, providenciam também uma camada de abstração da base de dados que facilita, geralmente, as operações de leitura, escrita e consulta da base de dados; esta camada é geralmente conhecida como Object-Relation Mapper (ORM) ou, em português, um tradutor das relações entre os objetos ou entidades da base de dados. O uso de uma ORM traz duas grandes vantagens: pode ser alterada a tecnologia da base de dados por baixo, por vezes sem ser necessário alterar uma única linha de código e permite implementar uma validação básica da informação, facilitando e tornando mais segura a verificação da informação que vai ser armazenada, isto é, se tem a informação tem o tipo e formato corretos, por exemplo.

Existem algumas destas *frameworks* em quase todas as linguagens de programação e,

com tanta escolha, pode ser uma tarefa complicada selecionar a mais apropriada para determinado projeto. Como tal, a decisão deve ter em conta fatores como:

- **Dificuldade de aprendizagem:** a dificuldade de aprendizagem de uma *framework web* geralmente depende no quão familiar o programador está com a linguagem na qual ela está desenvolvida, na consistência da sua API, na qualidade da sua documentação e no tamanho e atividade da sua comunidade;
- **Produtividade:** a produtividade é medida na rapidez com que é possível criar novas funcionalidades, uma vez que o programador está familiarizado com a *framework*, e tem em conta a quantidade de código que é necessário escrever e também a sua manutenção. Neste ponto cada *framework* depende muito do propósito para o qual foi criada, por exemplo, Django foi criada para suportar o desenvolvimento de sites que envolvam publicações em contraste com o Flask que é muito mais leve e ótimo para o desenvolvimento de aplicações web para correr em dispositivos embebidos. Há também *frameworks* que incluem ferramentas e bibliotecas que cobrem todas as necessidades básicas dos programadores ao passo que noutras, mais leves, o programador tem de escolher bibliotecas externas para resolver certos problemas.
- **Performance da *framework* e da própria linguagem de programação:** este é um ponto importante mas na generalidade dos casos talvez o menos importante uma vez que para a maior parte das aplicações de dimensões médias tempos de execução lentos como os do Python, por exemplo, são mais do que suficientes para cumprir os requisitos. Neste ponto tem de ser analisado o quando se ganha em velocidade em relação ao que se perde em facilidade de aprendizagem e manutenção do código;
- **Suporte a *Caching*:** à medida que um *web site* vai crescendo em utilizadores e visitas, pode haver situações em que comece a não conseguir atender os clientes da mesma maneira. Nesta altura o suporte a *caching* permite armazenar os conteúdos de um pedido que é efetuado muitas vezes para não ter de ser recalculado cada vez que é requisitado em pedidos consecutivos;
- **Escalabilidade:** novamente com o crescimento da população dos utilizadores de um *web site* ou aplicação web mudar o *hardware* onde está a correr o servidor ou adicionar o suporte a *caching* pode não ser suficiente para atender todas as solicitações. Neste passo poderá ser necessário distribuir a carga por um número mais vasto de servidores e bases de dados ou escalar o servidor por zonas geográficas. A escolha da *framework web* pode influenciar significativamente a facilidade com que se pode fazer crescer o sistema;

- **Segurança Web:** algumas frameworks oferecem melhor suporte para lidar com os comuns ataques à segurança dos *web sites*, ao validar toda a informação inserida pelo utilizador, limpando informação que possa correr código JavaScript inserido por esse utilizador que poderá desestabilizar o sistema ou comprometer a sua base de dados.

Outros fatores que são importantes referir prendem-se com a licença para uso da framework, isto é, se é gratuita ou não e também com a sua aceitação pela comunidade e se tem uma equipa ativa e dedicada ao seu desenvolvimento. A lista que se segue contém algumas das frameworks para servidores web mais utilizadas e uma breve descrição sobre as mesmas:

- **Django**[49] e **Flask**[38]

As duas alternativas mais populares, em Python. A primeira é uma framework de alto nível ao passo que a segunda é uma micro-framework extremamente leve que, embora minimalista, contém muitas funcionalidades extra através de bibliotecas complementares. Django contém quase tudo o que os programadores precisam e mais algumas coisas e tem uma documentação rica e em constante atualização, é rápido, seguro e bastante escalável. O Flask tem boa documentação, uma comunidade muito ativa e ficou extremamente popular particularmente entre os programadores que precisavam de serviços web em dispositivos pequenos ou de recursos limitados, como Raspberry Pi ou controladores para *drones*, por exemplo. Sendo desenvolvidas em Python, ambas as frameworks têm código de leitura e manutenção fácil.

- **Laravel**[52] e **Slim**[35]

Estas duas frameworks são um paralelo às anteriores mas em PHP. Laravel ocupa um lugar de topo no ranking das *framework web* mais utilizadas, em grande parte pelo facto de muitos provedores de alojamentos web tradicionais só disponibilizarem PHP. Laravel é o equivalente ao Django, na medida em que é uma framework super completa ao passo que o Slim é, tal como o Flask, uma micro-framework mas apropriada para projetos de pequena ou média dimensão.

- **Ruby on Rails**[50]

Framework escrita na linguagem de programação Ruby que segue uma filosofia muito semelhante ao Django e à Laravel, encorajando também metodologias DRY ("don't repeat yourself") que visam a escrita do código apenas uma vez sempre que possível.

- **ExpressJS**[36]

Express é uma micro-framework rápida, minimalista e flexível para Node.js (ambiente que permite executar JavaScript sem recurso a um browser). Inclui um conjunto robusto de funcionalidades para o desenvolvimento de sites e aplicações móveis e

oferece métodos de utilidade para tratar os pedidos HTTP. Esta framework é extremamente popular devido à facilidade de migração dos programadores JavaScript da vertente de cliente para a vertente servidor e também porque é bastante eficiente, uma vez que o ambiente node segue uma filosofia multi tarefa através do uso de *threads* em vez de executar um processo para cada pedido que chega. Como é uma micro-framework, tal como o Flask e a Slim tem de recorrer a bibliotecas externas para suportar certas funcionalidades como acesso à bases de dados e o suporte para sessões de utilizadores, por exemplo.

- **ASP.NET**[51]

A alternativa *open source* criada pela Microsoft para o desenvolvimento de aplicações e serviços web modernos. Esta framework possibilita a criação rápida de *web sites* baseados em HTML, CSS e JavaScript, escaláveis para o uso de milhões de utilizadores e contém funcionalidades complexas. O que diferencia esta framework das outras é que permite aos programadores desenvolver código em qualquer linguagem que suporte .NET como C#, Visual Basic, entre outras. Como os restantes produtos da microsoft tem excelentes ferramentas à sua disposição (nem sempre gratuitas), uma comunidade ativa e uma documentação rica e detalhada.

- **Spring**[53] e **Spark**[37]

Estas duas frameworks partilham correspondem aos pares Django/Flask e Laravel/Slim mas em Java. A Spark é uma micro-framework que tal como o Flask, a Slim e a ExpressJS necessita de bibliotecas extra para implementar certas funcionalidades, ao passo que a Spring é uma framework completa e preparada para o maior número de cenários que um projeto de desenvolvimento normalmente enfrenta.

A oferta é grande neste campo, há muitas frameworks que não são mencionadas mas que serviriam para realizar as mesmas tarefas tão bem como as que estão listadas anteriormente. É uma questão de escolher a melhor para o trabalho que se pretende realizar tendo em conta os fatores como a documentação disponível, se a framework tem uma comunidade ativa e disposta a ajudar, se o próprio desenvolvimento da framework não está estagnado, a quantidade de bibliotecas auxiliares disponíveis para cobrir o maior número possível de funcionalidades e até um pequeno toque de preferência pessoal, seja pela facilidade de desenvolvimento e de aprendizagem da linguagem em que está desenvolvida a framework ou pela complexidade da sua API.

## 2.4 TECNOLOGIAS: APLICAÇÃO MÓVEL

As aplicações móveis têm crescido drasticamente não só em quantidade como em qualidade, nos últimos anos. Como tal muitos tipos diferentes de aplicações têm surgido no mercado

mas, geralmente, podem categorizar-se entre dois paradigmas de desenvolvimento mobile: nativo e híbrido. O desenvolvimento nativo é baseado na plataforma, e a forma original de desenvolver aplicações móveis, ao passo que o paradigma híbrido é uma abordagem, mais recente, que tem como objetivo permitir o desenvolvimento de aplicações compatíveis com mais que uma plataforma (geralmente focadas em Android e iOS) usando tecnologias web como HTML, CSS e JavaScript. As diferenças entre estas duas principais alternativas prendem-se na escolha entre ter maior suporte, velocidade e maior acesso ao dispositivo ou desenvolver mais fácil e rapidamente com vista a colocar o produto no mercado antes de um concorrente, por exemplo.

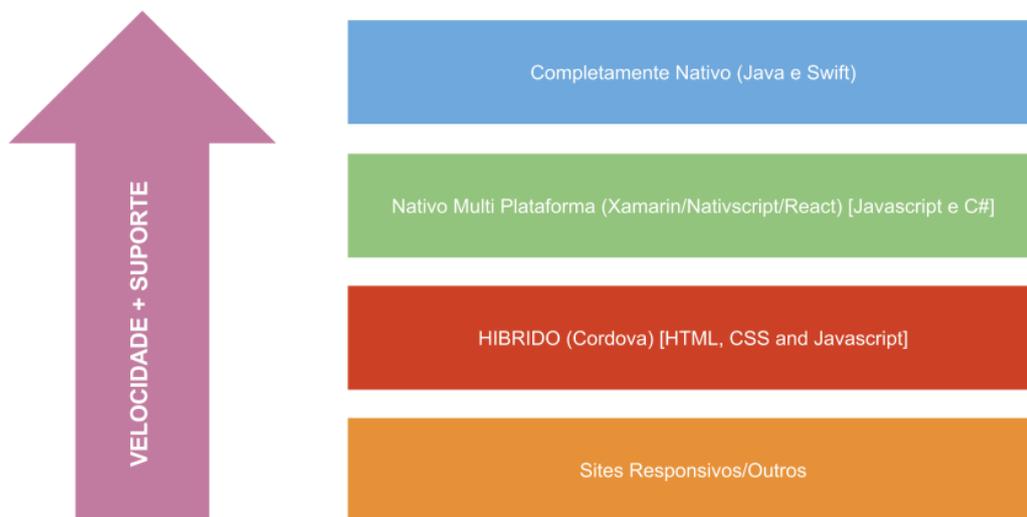


Figura 2.2: Diagrama para escolha de frameworks para *app development*

Na figura 2.2 está ilustrada a grande diferença, referida anteriormente, entre as categorias de frameworks existentes para desenvolvimento de aplicações móveis. Ao afastar-se do nativo a framework está a perder em velocidade e em suporte, isto é, em relação ao acesso que tem às funcionalidades do dispositivo e do sistema operativo do mesmo. Como tal, ao escolher uma framework há que ter em conta fatores como: Velocidade e suporte mas também o custo de desenvolver numa dada framework ou o risco de aprender uma nova quando o mercado está constantemente a receber novas alternativas e a que estamos a aprender pode deixar de ser utilizada.

Dada a natureza do projeto desta dissertação, faz sentido investigar melhor o paradigma de desenvolvimento híbrido uma vez que o acesso às funcionalidades do dispositivo que é necessário é mínimo. Atualmente, existe uma grande oferta de bibliotecas de desenvolvi-

mento híbrido e a seguinte listagem contém alguns dos nomes, lembrando algumas das suas vantagens e desvantagens:

- **Apache Cordova**[30]

Cordova é uma framework em JavaScript, open-source, que permite ao programador desenvolver aplicações móveis multi-plataforma tirando partido de tecnologias web existentes. Envolve a aplicação num recipiente de aplicações nativo que permite ao JavaScript aceder às bibliotecas do dispositivo, da mesma maneira que uma aplicação nativa permite. Com Cordova é possível criar aplicações com tecnologias web muito usadas e bem conhecidas (HTML5, CSS3 e JavaScript) e o mesmo código funciona em diferentes plataformas como iOS, Android, Windows Phone, BlackBerry, Firefox OS, entre outros. No entanto é muitas vezes usado apenas como base para outras frameworks como Ionic ou PhoneGap, ou usado em conjunto com frameworks como Framework 7 e VueJS.

- **Appcelerator Titanium**[31]

Ao contrário de soluções como o PhoneGap, Ionic Framework ou Meteor.js, a Appcelerator não envolve as aplicações em web containers. É uma framework, em JavaScript, que oferece ferramentas para um desenvolvimento rápido e tem bastantes características e pré-configurações com algumas das mais conhecidas e usadas ferramentas de bases de dados. Todavia está conhecida como tendo alguns problemas de latência e de instabilidade, embora tenham sido resolvidos nas versões mais recentes, e é conhecida por ter pouco suporte da sua equipa de desenvolvimento.

- **Xamarin**[32]

Framework em C#, poderosa e com IDE próprio. Através das suas aplicações exemplo e tutoriais é possível aprender e arrancar com um projeto rápido. Oferece testes de funcionalidade e de qualidade para inúmeros dispositivos que permite a integração sem problemas. Porém, muitas bibliotecas open-source para iOS e Android não podem ser usadas devido a problemas de compatibilidade com o Xamarin e como não é uma framework livre, a sua versão gratuita é extremamente limitada para projetos mais complexos.

- **Monocross**[33]

Framework open-source para desenvolvimento móvel multi-plataforma que possibilita a criação de aplicações para iOS, Android, Windows Phone e dispositivos Webkit. Monocross usa C#, Microsoft .NET e a framework Mono para suportar o desenvolvimento destas aplicações e viabiliza o acesso total às APIs nativas dos dispositivos. Contudo a sua documentação, recursos disponíveis e suporte na Internet são extremamente limitados, tornando a aprendizagem mais complicada.

- **Framework 7**[34]

Uma das melhores características desta framework é não ter quaisquer dependências externas como o React ou Angular conseguindo, no entanto, que as aplicações pareçam nativas com as suas animações apelativas e componentes devidamente desenhados. Qualquer programador com conhecimento de CSS, HTML e JavaScript pode criar uma aplicação sem ter que escrever uma grande quantidade de código, no entanto para interagir com o dispositivo tem de ser combinada com PhoneGap ou Cordova.

- **Ionic**[54]

Uma framework popular e a primeira escolha de muitos programadores, que permite criar aplicações com design nativo com bastante facilidade. Tem uma comunidade bastante grande e ativa e contém uma linha de comandos que traz consigo uma quantidade de funcionalidades extra, úteis para o desenvolvimento. No entanto precisa de conhecimento em Angular para desenvolver funcionalidade mais complexas.

- **React Native**[55]

Como o nome indica, esta framework serve o propósito de desenvolver aplicações nativas em vez de desenvolver seguindo o paradigma híbrido. No entanto, maior parte do desenvolvimento é feito em JavaScript e React. Tem uma enorme comunidade assegurando que a falta de apoio e documentação não sejam um problema e é apoiado pelo Facebook, que de si já é uma garantia da sua qualidade. No entanto, a framework não está orientada para os principiantes do desenvolvimento web e tem uma curva de aprendizagem algo acentuada.

- **Kendo UI**[56]

Principalmente focada no mercado empresarial, esta framework permite o desenvolvimento de aplicações web ou móveis híbridas usando JavaScript, HTML e CSS. jQuery constitui uma grande componente da Kendo e inclui mais de setenta *widgets*. Vem com um grande conjunto de temas e contém um editor de temas para uma fácil customização para cada aplicação. A Kendo está inserida numa comunidade grande de desenvolvimento open source e tem um conjunto de opções de utilização e licença. Empresas como a Sony, NASA, Microsoft, Volvo e Toshiba, entre outras, usam a Kendo.

- **Onsen UI**[57]

Uma alternativa recente mas que está a competir diretamente com a Ionic. É open-source, suportada pela Apache e é uma framework que não está orientada para os componentes da interface do utilizador, facilitando a escolha e a mudança entre Angular, Angular 2, React, Vue.js ou até JavaScript puro para desenhar as aplicações. Permite a construção de aplicações usando HTML, CSS e JavaScript através do uso

de Cordova ou PhoneGap para compilar a aplicação para Android e/ou iOS. É uma framework de licença gratuita, fácil aprendizagem e utilização.

Estas são algumas das melhores alternativas, para desenvolvimento híbrido de aplicações móveis, disponíveis no mercado e a escolha deve recair sobre aquela que facilitar o desenvolvimento e que permita o acesso a algumas funcionalidades básicas e sensores do dispositivo, como o GPS e a bússola, sem comprometer a *performance* e a experiência final do utilizador.

## 2.5 TRABALHO RELACIONADO

Atualmente existem milhares de aplicações para quase todos os tipos de utilidades, desde jogos de todos os géneros a ferramentas úteis para o dia a dia. Estar informado sobre o que existe numa determinada área pode ajudar não só a perceber o que faz falta mas também como fonte de inspiração. Algumas componentes interessantes de certas aplicações podem ser aproveitados, depois de repensados e adaptados, assim como os erros ou más decisões encontradas numa ou outra aplicação poderão de servir de exemplo ao planear uma nova. A lista que segue contém uma breve síntese de aplicações, existentes no mercado, que têm componentes interessantes que poderão ser reutilizadas no processo de ludificação da interação das pessoas com a cidade e as suas instituições:

- **MyCity**[5]

Criada pelo Departamento de Sistemas de Informação da sua Câmara Municipal de Guimarães, MyCity é uma aplicação que oferece aos seus utilizadores a possibilidade de reportar ocorrências, com categorias devidamente adaptadas à cidade. Isto faz com que a população tenha uma forma pró-ativa de ajudar a cidade e a câmara uma forma reativa de movimentar as entidades responsáveis por tratar cada um destes tipos de ocorrência de uma maneira célere e eficaz. É importante referir que este conceito vai servir de base de partida para a plataforma que se pretende desenvolver. Todas as funcionalidades estão disponíveis tanto para a versão móvel e para browser, e o aspeto geral da aplicação é simples e de utilização intuitiva, como se pode verificar na figura 2.3.

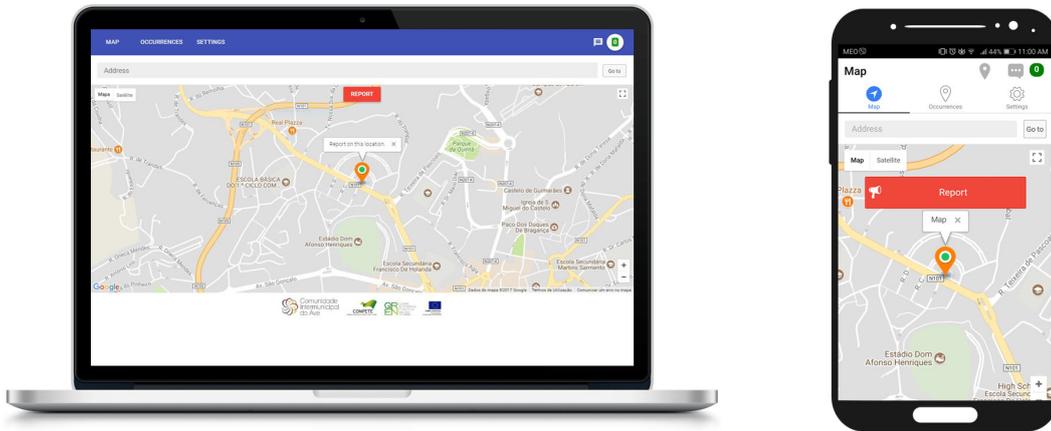


Figura 2.3: Exemplo: MyCity

- **Streetspotr[13], FieldAgent[13], TaskSpotting[13]**

Estas aplicações partilham bastantes características e operam de maneira semelhante. É frequente serem usadas por empresas em operações de marketing, onde estas utilizam esta tecnologia para fazer auditorias de mercado, para tentarem perceber a receção do público de determinadas zonas a um novo produto, mudança de visual de um já existente ou até mesmo da abertura e localização de uma nova loja. Os utilizadores têm de se deslocar até aos locais assinalados no mapa e executar tarefas como responder a questionários, tirar fotografias e partilhá-las nas redes sociais, entre outras, tudo isto usando apenas um dispositivo móvel como o smartphone ou um tablet.

A ideia geral destas aplicações é que o utilizador cumpra missões (tarefas), como se incorporasse o papel de um agente secreto, em troca de recompensas, geralmente, monetárias.

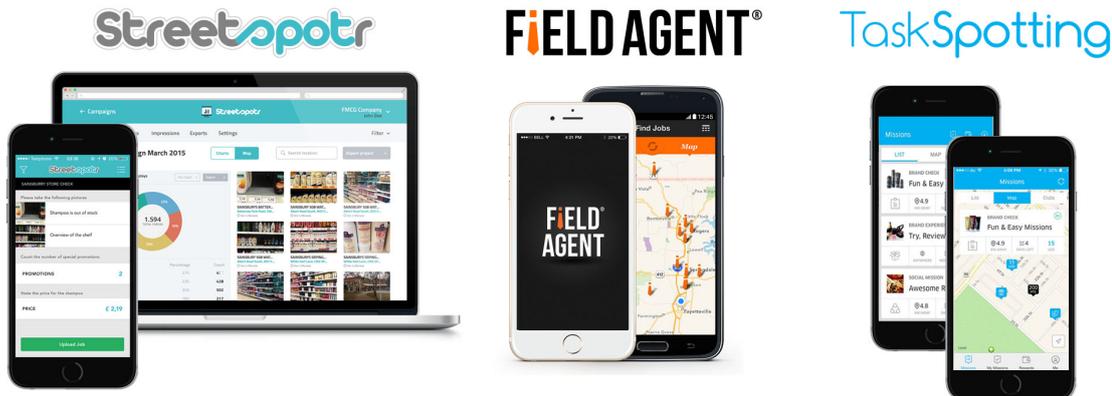


Figura 2.4: Imagens das aplicações Streetspotr, FieldAgent e TaskSpotting

- **Surveys on the go[14], Survey[15], iPoll[18]**

A aplicações Surveys on the go, Survey e iPoll têm uma lógica bem mais simples, quando comparadas com as faladas anteriormente, na medida em que não exigem ao utilizador que se desloque e as missões pedem apenas que respondam a questionários ou votações em troca de recompensas, monetárias ou não.

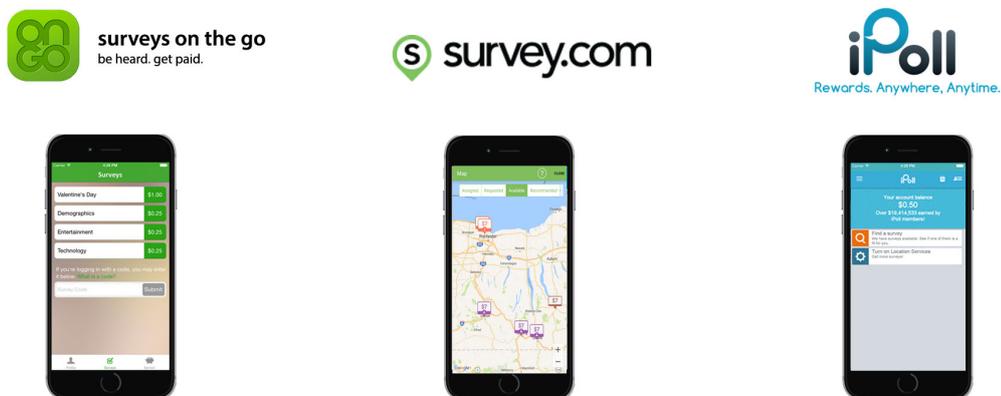


Figura 2.5: Imagens das aplicações Surveys on the go, Survey e iPoll

- **Ingress[22]**

Um dos mais complexos e completos jogos de georreferenciação e realidade alternativa do mercado que consolidou a estrutura para o muito conhecido *Pokemon GO*, da mesma empresa. O Ingress[22] é mundo virtual paralelo, escondido no mundo real, onde cada utilizador é um agente secreto que luta pela resistência ou pelo regime e

tem como sua principal função conquistar torres (pontos de interesse) para dominar o mundo para a sua facção.



Figura 2.6: Imagens das aplicações Ingress e Pokemon GO

Alguns aspetos destas aplicações, como foi mencionado anteriormente, vão servir de inspiração para o projeto proposto por esta dissertação. Os conceitos de **evento** e **missão** da aplicação StreetSpotr[13], quando modificados e enquadrados no contexto do projeto, podem ser muito interessantes para providenciar conteúdo dinâmico à aplicação. No presente caso de estudo não seriam, pelo menos inicialmente, os parceiros comerciais a entidade responsável pela criação de eventos, mas sim a instituição pública responsável pela administração do sistema; os parceiros poderiam, no entanto e através da sua parceria com a instituição, propor a criação de um novo evento que lhe conviesse, fosse pelo contexto do evento ou pela localização do mesmo. Adicionalmente, a moeda da aplicação seria fictícia e os utilizadores poderiam receber quantidades variáveis dessa moeda, através dos vários eventos, missões disponíveis e/ou outras atividades dentro da aplicação, para trocar por recompensas reais que poderiam ou não estar associadas aos parceiros.

Na íntegra, o conceito da aplicação desenvolvida para câmara municipal de Guimarães (MyCity[5]) será, certamente, adotado. Esta aplicação, como já foi referido, serve para os cidadãos fazerem participações de ocorrências, estragos, danos na via pública, entre outras coisas. O conteúdo da mesma será muito útil para a aplicação que está a ser desenvolvida, uma vez que o caso prático adotado é mesmo a cidade de Guimarães; desse conteúdo serão aproveitadas, principalmente, as categorias e sub-categorias das participações que são resultado de um estudo desenvolvido para responder às necessidades, em particular, da cidade de Guimarães.

Por último, a aplicação Ingress[22] tem vários aspetos interessantes que poderão acrescentar bastante interesse à aplicação deste projeto, principalmente o conceito do utilizador pertencer a uma equipa e competir com os outros utilizadores e as suas respetivas equipas,

colaborando com os seus amigos e colegas enquanto realizam micro ações nos vários pontos de interesse espalhados pela cidade. Contudo, tem outros componentes que tornam o jogo demasiado complexo, como os vários itens que o utilizador tem de aprender a utilizar e que variam de potência e utilidade com os níveis e à medida que o utilizador vai evoluindo, o que pode fazer com que o utilizador se assuste com a quantidade de informação que tem de assimilar em pouco tempo. A acrescentar a este fator, fica também o facto de que não há um sistema de recompensas no jogo, fazendo ao mesmo tempo com que quem joga não tenha grande sentido de continuidade e de aproveitamento do tempo despendido no mesmo.



---

## ANÁLISE DO PROBLEMA E ESTRATÉGIA PARA LUDIFICAÇÃO DO SISTEMA

---

A motivação principal para esta dissertação, como já foi referido anteriormente, é aproximar as pessoas da sua cidade, umas das outras, e dos órgãos públicos da mesma, como as juntas de freguesia e as câmaras municipais, num esforço conjunto para o bem do presente e futuro da cidade. Esta afirmação é válida não só para os turistas mas, e principalmente, para os locais e os que, mesmo não sendo da cidade, passam maior parte do seu dia nela. O intuito deste projeto é, então, apresentar um caso de estudo e uma prova de conceito de uma aplicação móvel que consiga oferecer uma oportunidade realista e prática para essa aproximação.

Como tal, o principal problema está na aplicação em si e em como a desenvolver e que conteúdo terá disponível para os utilizadores, não esquecendo, também, que tem de cumprir certos requisitos como:

- Ser intuitiva e fácil de utilizar;
- Ter um design funcional mas apelativo;
- Estar disponível para o maior número de pessoas e plataformas móveis;
- Ter conteúdo atualizado e em constante atualização;
- Facilitar a comunicação e certos processos entre os cidadãos e as instituições públicas da cidade.

No mercado atual, com os milhares de aplicações existentes, é um desafio cada vez maior aparecer algo que despolete a atenção das pessoas por ser útil ou interessante e que, por outro lado, lhes ofereça uma experiência nova. Este é o ponto mais complicado deste projeto e, nesse sentido e como já foi referido anteriormente, foi aplicada a técnica de *gamification* ao conceito inicial, isto é, a decisão de transformar o processo de comunicação das pessoas com as instituições da cidade num jogo onde as pessoas também tenham o objetivo de a explorar e conhecer melhor, bem como terem objetivos e competirem para ganhar prémios. Com este objetivo em mente a aplicação vai girar em torno de dois conceitos: **competição**

e **recompensa**. A ideia final de como vai funcionar o jogo foi influenciada por algumas aplicações já existentes, como foi referido no capítulo anterior do estado da arte. Como tal, resolveu-se desenvolver um jogo de georreferenciação e realidade aumentada onde as pessoas possam conhecer gente nova ou partilhar a experiência com conhecidos, familiares ou amigos, enquanto se divertem a explorar e a ajudar a sua cidade, com a possibilidade de interagir pró-ativamente no estado limpeza e normalidade da cidade através de um sistema de participações.

### 3.1 A NARRATIVA DA APLICAÇÃO

Foi concebido um mundo virtual onde a sociedade é gerida por um conselho de tribos que, entre elas, ditam democraticamente o rumo da cidade. Este conselho reúne-se mensalmente e o nível de influência que cada tribo tem nas decisões tomadas vai variar com o apoio que tiver da população em cada mês. Cada uma destas tribos representa uma filosofia ou vertente diferente da cidade. Embora as tribos não sejam necessariamente rivais, vão competir pela atenção, respeito e dedicação dos cidadãos.

O que isto significa:

- *No jogo* - isto traduz-se em criar um *ranking* de tribos/equipas, que vão ser classificadas pelo apoio (número de pontos) conseguido num dado mês, que servirá para a atribuição de prémios mensais;
- *No seu dia a dia* - A ideia por trás do conceito é dar à instituição responsável pela manutenção e administração da plataforma a perceção, ainda que ligeira, das tendências e/ou opiniões dos utilizadores, uma vez que cada um terá de escolher uma tribo para representar.

Em primeiro lugar, a escolha da tribo representa a escolha de uma filosofia ou ideia para o futuro da cidade com que o utilizador mais se identifica e, em segundo lugar, permite identificar estas tendências por regiões da cidade. Isto significa que o evoluir do jogo, vai permitir, no caso prático deste projeto, à Câmara Municipal de Guimarães perceber que a população de uma freguesia A tem certas preferências em relação ao futuro da cidade enquanto a população de uma freguesia B tem uma visão completamente diferente.

Pretende-se, portanto, que esta componente possa ajudar a instituição a tomar decisões em relação a determinada freguesia ou mesmo a perceber o que está mal e precisa de ser mudado ou atualizado e, ainda que isto não seja totalmente direto e perceptível, é mais uma forma de obter a opinião dos cidadãos de uma forma não invasiva como seria um questionário ou uma sondagem, por exemplo.

## 3.2 AS TRIBOS

Muitas vezes as pessoas afirmam conhecerem extremamente bem a sua cidade só pelo simples facto de que nasceram, vivem ou passam uma quantidade significativa de tempo das suas vidas nela. Por incrível que pareça, isso não é verdade na maior parte dos casos.

Foi adotado o termo **tribo** seguindo a ideia de fazer com que as pessoas sejam nómadas na sua própria cidade, que abandonem as suas vidas, muitas vezes, sedentárias de cidade e passem a explorar e conhecer melhor não só a zona que habitam mas também o resto da cidade de onde são naturais, residentes ou visitantes. Neste sentido, as tribos têm o função adicional de formar grupos ou equipas com objetivos e filosofias comuns, para que as pessoas tenham ou possam ter com quem sair à rua e jogar, seja com amigos, conhecidos ou conhecendo gente nova.

Como tal, foram criadas cinco tribos baseadas na filosofia budista dos cinco elementos, representados na figura 3.1. Cada elemento diz-se representar uma tendência no mundo, seja na física, na espiritualidade ou mesmo dentro de cada um de nós. Juntos, estes elementos explicam a natureza das coisas, por trás de cada ação, sociedade ou pessoa.

No contexto deste projeto cada tribo vai representar não só um aspeto da cidade mas também uma ideia ou visão para o futuro da mesma, como se pode verificar na tabela 3.1.

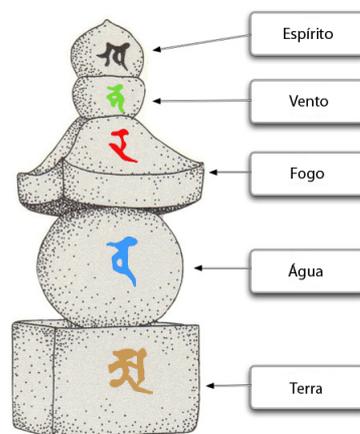


Figura 3.1: Gorintou, representação física dos cinco grandes''

Juntas, e apesar das diferenças nas suas filosofias, as tribos representam o equilíbrio da cidade como um todo, uma vez que elas precisam da cidade assim como a cidade precisa de cada um dos aspetos que elas representam.

**Terra**

A tribo da terra acredita que o povo e a cidade devem muito à história da mesma e à dureza e força das pessoas que ajudaram a construí-la. Vê o abraçar e aceitar a herança cultural e histórica da cidade, aprendendo com os erros e sucessos dos seus antecessores como o caminho certo para um futuro melhor da cidade.

**Água**

Representado pelos rios, lagos e oceanos, este elemento simboliza a capacidade de mudança e adaptação. A tribo da água promove a ciência e o avanço tecnológico como os aspetos mais importantes da sociedade.

**Fogo**

Simboliza a energia e paixão do povo da cidade. A tribo do fogo aposta no turismo e no comércio local e, principalmente, em mostrar o valor da cidade aos seus e aos visitantes.

**Vento**

A tribo do vento pede cidadãos que mantenham uma mente aberta e que tenham a sabedoria de perceber que a saúde e a educação devem ser as principais preocupações da cidade. Criar futuras gerações mais sábias, atentas e com melhores condições de vida.

**Espírito**

Promovendo a criatividade, espontaneidade e a inventividade, a tribo do espírito tem as artes e o desporto como o estandarte do brilho e alma da cidade.

Tabela 3.1: As tribos e os seus ideais

### 3.3 A LOJA E A MOEDA DO JOGO

Para o jogo funcionar e os utilizadores se sentirem recompensados foi desenvolvida uma loja virtual onde os utilizadores poderão trocar os pontos acumulados por produtos reais adicionados à loja pela administração do sistema, fruto das parcerias que esta for formando. Estes produtos terão, logicamente, preços e níveis de interesse diferentes e poderão ter

também limite no número de vezes que podem ser comprados num espaço de tempo. A título de exemplo, imagine-se que a administração do sistema estabelece uma parceria com um clube de futebol da cidade, isto poderia dar a origem a inúmeros itens para colocar na loja como: bilhetes duplos para assistir a um jogo da equipa, equipamentos, bolas autografadas, entre outros. Outro exemplo seria estabelecer parcerias com companhias de teatro, cinema ou espetáculos ao vivo, dando acesso aos utilizadores a bilhetes através da aplicação para *shows* nos estabelecimentos desses parceiros.

A moeda do jogo vão ser os pontos que os utilizadores vão recebendo, chamados *nomads*. Eles vão servir, como foi dito anteriormente, para trocar por itens na loja e quase todas as ações realizadas pelos utilizadores serão recompensadas com estes pontos, sempre proporcionalmente ao trabalho realizado numa missão, evento ou ao desbloquear uma conquista. A quantia de cada recompensa terá de fazer sentido e terá de estar de acordo com o valor real e preço, na aplicação, dos itens da loja.

Como tal, a loja terá de ter itens apelativos para que angariar *nomads* seja a principal motivação dos utilizadores, a juntar ao facto de que estão a ajudar e a contribuir para a sua cidade.

### 3.4 ESSÊNCIAS: OS MATERIAIS DO *gameplay*

Para introduzir um bocado de dinâmica no jogo que não seja só angariar pontos, foram criados cinco materiais, representando cada um uma tribo, denominados *essências*:



Tabela 3.1: *Essências*, um tipo para cada elemento/tribo

Ao criar a conta, o utilizador terá já alguns destes materiais que irá necessitar para realizar certas ações nos templos (pontos de interesse da cidade) e é também nos templos onde poderão coletar mais. A quantia, de cada material, que os utilizadores irão conseguir retirar de cada vez que executarem essa ação nos templos dependerá da tribo que representam e de um pouco de sorte.

Como funciona?

Um nómada (utilizador) de fogo irá sempre retirar, e em maior quantidade, *essências* de fogo e uma quantia aleatória das restantes, quantia essa que será sempre menor, podendo ir de 0 até metade das recebidas do elemento principal.

Estas *essências* poderão, também, ser convertidas em pontos do jogo (*nomads*), num rácio que seguirá a seguinte fórmula:

$$X \text{ essências de elemento escolhido} + 2 \cdot X \text{ essências de cada um dos restantes elementos} = X \text{ pontos}$$

Se o utilizador for da tribo da água, por exemplo:

$$500 \text{ essências de água} + 1000 \text{ de fogo} + 1000 \text{ de vento} + 1000 \text{ de espírito} + 1000 \text{ de terra} = 500 \text{ pontos}$$

### 3.5 TEMPLOS: PONTOS DE INTERESSE

Cada ponto de interesse da cidade será tratado como um *templo*. Estes templos não têm qualquer tipo de relação com alguma religião, que não existe neste mundo virtual, e como tal, eles representam locais onde os utilizadores prestarão homenagem à cidade e farão contributos em nome da sua tribo.

#### 3.5.1 Ações nos Templos

Uma vez no raio de alcance de um templo o utilizador tem várias ações à sua disposição:

- **Ver informações:** as informações disponíveis sobre o ponto de interesse (templo) como as imagens, descrição, categorias do ponto de interesse e estatísticas como o número de visitas por tribo, nesse templo, e últimos visitantes, por exemplo;
- **Explorar:** quando o utilizador está a visitar pela primeira vez um dado tempo, a única opção disponível é **explorar**. Isto é uma ação simbólica que tem como único objetivo ser contabilizado para conquistas;
- **Conversão:** converte *essências* em pontos, como foi descrito anteriormente. O utilizador tem a oportunidade de escolher a quantia de *essências* que quer converter em pontos;

- **Recolha:** o utilizador colhe o fruto do seu trabalho na cidade, em nome da sua tribo, na forma de essências que servirão para poder realizar outras ações;
- **Tributo:** o utilizador oferece *essências* como tributo ao templo da cidade, em nome da sua tribo, isto traduz-se em contribuir para a contagem de tributos globais (em todos os templos da cidade), afetando o *ranking* mensal das tribos.  
Esta ação vai consumir *essências* de todos os elementos mas em maior quantidade do elemento da tribo que o utilizador representa.  
O templo brilhará com a luz do elemento da tribo que tenha a maior contagem no momento, representando a força dessa tribo na zona onde o templo está situado.

### 3.6 FUNCIONALIDADES DA APLICAÇÃO

Descrito o cenário da cidade virtual que dá contexto à aplicação, passa-se então a definir as funcionalidades e curiosidades da mesma.

#### 3.6.1 Participações

Os utilizadores poderão em qualquer sítio, dentro dos limites territoriais da cidade, comunicar algo à instituição pública que administra a aplicação. Cada participação será analisada sendo, primeiro, feita uma pré filtragem baseada na sua legitimidade e facilidade de resolução, sendo posteriormente orientada para o respetivo organismo ou departamento responsável pela resolução do problema em questão.

Devido ao caso prático deste projeto, as categorias usadas para catalogar as participações são, exatamente, as mesmas encontradas na aplicação *MyCity*[5], uma vez que elas foram especificamente pensadas para a cidade de Guimarães fazia sentido aproveitar esse trabalho previamente realizado pela Câmara Municipal.

Este sistema serve não só para ajudar a manter a cidade como o utilizador a quer, limpa e tratada, mas também vai servir para o premiar quando o faz. Para evitar que sejam feitas participações falsas, exageradas ou em excesso, o utilizador só recebe pontos quando a participação começa a ser analisada e, posteriormente, quando é resolvida. No fim do processo o utilizador é notificado de que a situação está resolvida.

#### 3.6.2 Eventos

A instituição pública encarregue pela aplicação terá à sua disposição todas as ferramentas para controlar a evolução da mesma. Isso inclui uma ferramenta para criar e gerir eventos relacionados com a aplicação. Estes eventos podem ser mais ou menos complexos, depen-

dendo da imaginação, apoios e parcerias e/ou objetivos da instituição. Podem também ser eventos orientados à tribo (para os utilizadores de uma dada tribo se conhecerem), puramente de lazer ou então eventos de promoção de algo como, por exemplo, um produto de um parceiro ou uma ideia para a cidade. Com isto a instituição tem mais uma oportunidade de reunir alguns cidadãos e tentar perceber a opinião deles sobre algo, através de pequenas missões que levam ao cumprimento dos objetivos do evento. Como tal, cada evento consistirá num conjunto de pequenas missões ou tarefas que os utilizadores terão de realizar e validar com o administrador do sistema, no local. O utilizador arrecada pontos e ou prémios por participar nestes eventos, assim como teve a oportunidade de estabelecer contacto com gente nova e, quem sabe, conhecer aspetos que até então lhe eram desconhecidos da sua cidade.

### 3.6.3 Missões Diárias

Todos os dias, os utilizadores terão à sua disponibilidade um conjunto de missões diárias. Idealmente, estas missões serão simples e rápidas de fazer que, ao contrário das missões dos eventos, não necessitarão de validação por parte de um administrador.

O objetivo destas missões é fornecer conteúdo diário para que o utilizador considere vantajoso abrir a aplicação pelo menos para fazer estas missões. O utilizador ganhará pontos e outras recompensas que serão sempre proporcionais ao esforço e tempo despendidos em cada missão.

### 3.6.4 Conquistas

O sistema compreenderá também um sistema de conquistas (*achievements*) que é um aspeto que, quase sem exceção, todos os jogadores gostam de ter num jogo, uma vez que lhes dá um sentido de continuidade e recompensa pelo esforço e tempo despendido, assim como uma noção de evolução do seu perfil como jogador dentro do próprio jogo. Como exemplos de conquistas podem listar-se: número de templos descobertos e visitados, quantidade de vezes que cumpriu todas as missões diárias, entre outros.

Desbloqueando uma conquistas, o utilizar será recompensado com *nomads* (pontos) e não poderá ganhar essa recompensa novamente. As recompensas por desbloquear conquistas serão ligeiramente mais significativas que as que o utilizador receberá por completar missões, por exemplo.

### 3.6.5 Classificações

Os utilizadores serão classificados com o intuito de poderem comparar o seu progresso com os demais. Eles serão classificados segundo alguns critérios como pontos acumulados, número de templos visitados, quantidade de participações submetidas e resolvidas, entre outros.

Além destes critérios, haverá as seguintes categorias:

- **Individuais: Totais**

As classificações *totais* serão baseadas nos dados dos utilizadores, reunidos desde o primeiro dia em que for lançada a aplicação. Estas não servem para atribuir prémios aos utilizadores, servem apenas para manter um *ranking* ao estilo *hall of fame*, para criar um espírito competitivo no jogo. Não seria justo atribuir prémios por esta classificação uma vez que nem todos os utilizadores vão começar a jogar ao mesmo tempo.

- **Individuais: Semanais**

Ao contrário dos *rankings* totais, os semanais servirão para atribuir prémios e/ou pontos aos utilizadores. No fim de cada semana, mediante a sua classificação, o utilizador receberá mais ou menos pontos e as classificações serão reiniciadas.

- **Tribos: Mensais**

Para simbolizar a reunião mensal das tribos, todos os meses é feita uma atribuição de prémios aos utilizadores da tribo com o maior número de tributos realizados durante um dado mês. Depois de feita a atribuição dos prémios esta classificação é reiniciada.

### 3.6.6 Escalões de Classificação

Os utilizadores serão distribuídos, pelo menos nesta primeira versão do jogo, por oito escalões diferentes que são os seguintes:

Como se pode ver na tabela 3.1, os escalões representam papéis a desempenhar numa tribo, mas esses papéis são puramente simbólicos e são somente uma maneira de distinguir os escalões com terminologia associada a tribos.

Quando um utilizador começar a jogar, ele vai estar automaticamente no escalão *Bardo* até fazer a sua primeira ação no jogo, momento em que será devidamente classificado entre os demais. Os primeiros quatro escalões compreendem sessenta e cinco por cento da população da aplicação. Os restantes trinta e cinco por cento ocupam os escalões de Caçador até Chefe (como se pode ver na figura 3.2). Cada escalão terá acesso a melhores

			
Bardo	Coletor	Batedor	Guerreiro
			
Caçador	Curandeiro	Xamã	Chefe

Tabela 3.1: Escalões de classificação

recompensas que o anterior, para que a motivação de subir na classificação seja mais elevada, que é o objetivo da mesma.

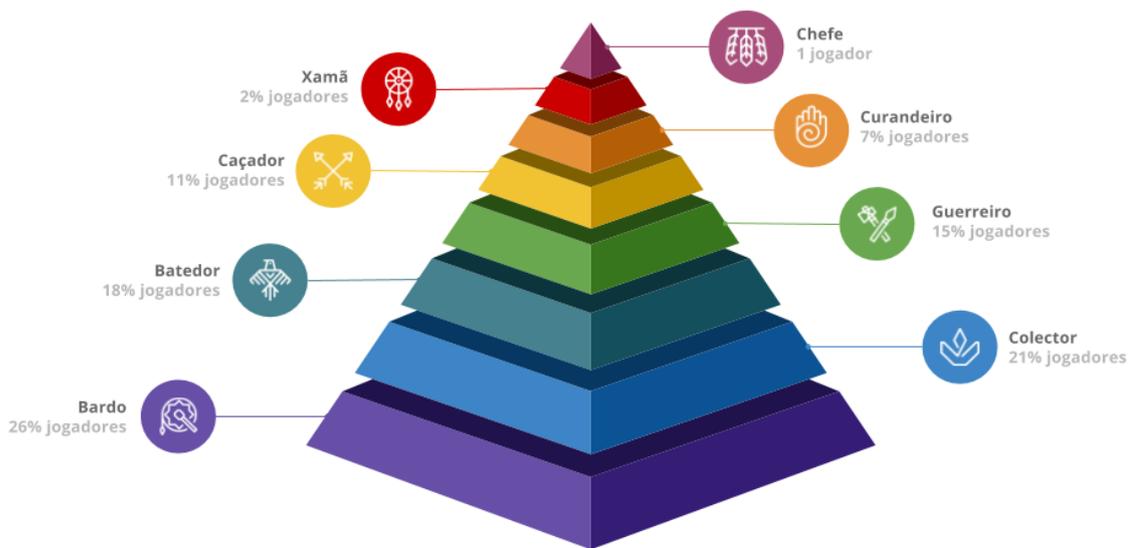


Figura 3.2: Pirâmide de Classificação

Para clarificar, o utilizador colocado no escalão *Chefe* é selecionado escolhendo, entre os utilizadores que estão no escalão *Xamã*, o utilizador com maior classificação.

### 3.7 A CONTA DE UTILIZADOR

Como em qualquer jogo, os utilizadores gostam de ver e sentir a sua progressão. Para tal foi implementado outro uso para os pontos que o utilizador vai recebendo, a progressão de nível, isto é, ao começar cada conta vai encontrar-se no nível um (1) e ao arrecadar pontos o utilizador vai subindo o nível da sua conta.

Os pontos necessários para atingir cada nível seguem a seguinte fórmula matemática:

$$nível = \begin{cases} 150x^2 + 1050x & \text{if } 1 \leq x \leq 7 \\ 200x^2 + 1050 + -2450 & \text{if } 8 \leq x \leq 11 \\ 50x^2 + 1750x + 9800 & \text{if } 12 \leq x \leq 22 \\ 250x^2 - 1500x - 22750 & \text{if } 23 \leq x \leq 30 \end{cases}$$

Figura 3.3: Exemplo: Novo evento, parte 1

Nesta versão inicial da aplicação, o nível máximo será trinta (30) que, mesmo assim, não será rápido de atingir. A equação na figura 3.3 mostra uma contagem de níveis a contar do 0. Então, seguindo a fórmula tem-se a lista completa, representada na tabela 3.2:

Nível	Pontos	Nível	Pontos	Nível	Pontos
1	0	11	28,050	21	64,800
2	1,200	12	33,300	22	68,600
3	2,700	13	38,000	23	72,500
4	4,500	14	41,000	24	75,000
5	6,600	15	44,100	25	85,250
6	9,000	16	47,300	26	96,000
7	11,700	17	50,600	27	119,000
8	14,700	18	54,000	28	131,250
9	18,750	19	57,500	29	131,250
10	23,200	20	61,100	30	144,000

Tabela 3.2: Pontos necessários para atingir determinado nível

Para melhor interpretar esta lista, ela funciona da seguinte maneira: para chegar a nível dois (2) o utilizador terá de angariar mil e duzentos (1200) pontos, para chegar a nível três (3) terá de angariar outros dois mil e setecentos (2700) e a lógica mantém-se para os restantes níveis. Como tal, feitas as contas o utilizador terá de juntar um total de um milhão quinhentos e dez mil novecentos e cinquenta (1510950) pontos para atingir o nível máximo.



---

## IMPLEMENTAÇÃO

---

O presente capítulo apresenta o planeamento do projeto que envolve a aplicação móvel e o servidor *web*. O propósito das secções que se seguem é mostrar o processo usado para desenvolver estes dois componentes do sistema de maneira a que sejam cumpridos os objetivos e requisitos estabelecidos nas secções anteriores e, também, tendo em consideração que o projeto se insere no âmbito das *smart applications* [70].

Começa-se por descrever a arquitetura do sistema seguida pela modelação da base de dados e das especificações do servidor *web* e da aplicação móvel.

### 4.1 ARQUITETURA DO SISTEMA

Este projeto requer o estabelecimento de comunicações entre a aplicação móvel e um servidor *web* central que é o provedor de toda a informação necessária para o correto funcionamento da aplicação, comunicação essa que é realizada, na totalidade. Esta comunicação cliente-servidor é realizada através de pedidos, HTTP, feitos ao servidor, por parte da aplicação, que despoletam ações que, por sua vez, devolvem uma resposta.

Não existe, de facto, uma conexão estabelecida entre cliente e servidor, na realidade a identificação da origem de um pedido, isto é, a determinação de que utilizador está a fazer um determinado pedido, é feita reconhecendo um *token* de autenticação, enviado no cabeçalho de todos os pedidos. Este *token* é atribuído a um utilizador quando este inicia sessão com as suas credenciais, nome de utilizador ou *email* e *password*.

O cliente tem de saber comunicar com o servidor, através do seu endereço e porta, ao passo que o servidor não necessita de informação nenhuma sobre como comunicar com o utilizador, tem apenas de responder ao pedido efetuado.

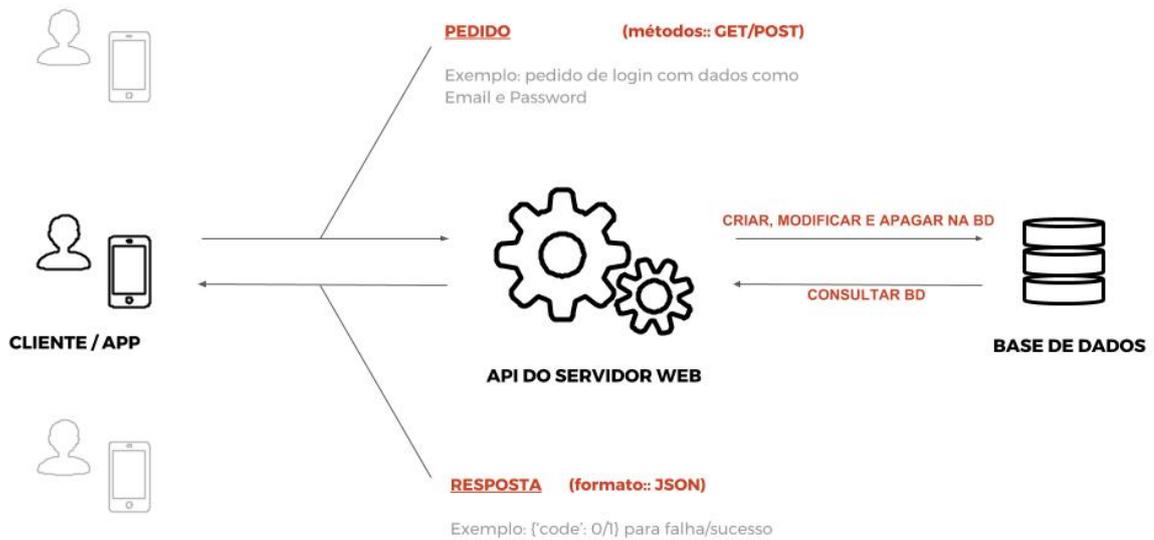


Figura 4.1: Arquitetura do Sistema

## 4.2 BASE DE DADOS

Esta secção fala sobre a escolha da tecnologia para a base de dados da aplicação e sobre as entidades que foram necessárias criar. Foi escolhida a tecnologia PostgreSQL, por ser uma ferramenta que reúne todas as condições necessárias à realização do projeto e também por motivos de familiaridade do autor com a mesma. Em particular, foi escolhida a versão 9.5 que permite ter campos do tipo JSON que possibilitam a armazenagem de estruturas dinâmicas, à semelhança das bases de dados não relacionais, fazendo com que esta base de dados use uma metodologia híbrida mas principalmente evitando a criação de tantas tabelas de relação secundárias e, por consequência maior facilidade de extração e armazenamento da informação em certos casos.

Começou-se, então, por definir as entidades necessárias à construção da API do servidor *web* e ao funcionamento da aplicação móvel.

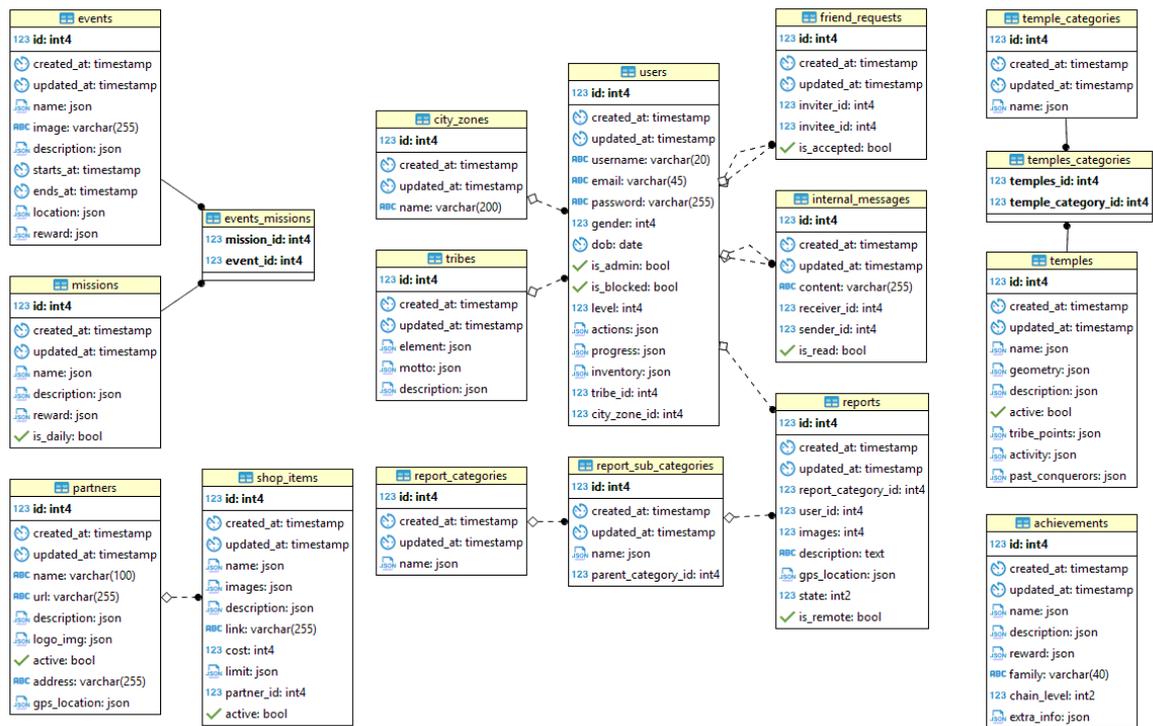


Figura 4.2: Diagrama de Relações entre Entidades da Base de Dados

É fácil identificar na figura 4.2 as tabelas que representam as entidades principais e as tabelas que representam as relações N para N dessas mesmas entidades. As subsecções que se seguem vão detalhar e explicar os campos que constituem essas entidades principais. Os campos **id**, **created\_at** e **updated\_at** são comuns a todas essas entidades e representam, respetivamente, o identificador único da entidade e a data e hora em que a entidade foi criada e atualizada e, como tal, não aparecem nas tabelas que se seguem.

#### 4.2.1 Entidades

Nesta subsecção estão detalhados os atributos de cada entidade do sistema, para melhor entender como irá funcionar o mesmo e para explicar o porquê do uso do tipo JSON para alguns dos atributos das entidades.

##### Tribo e Freguesia

As primeiras entidades a ser definidas, **tribo** e **freguesia**, são a base para o registo do utilizador e, em particular, a tribo é importante para toda a narrativa da aplicação e, como tal, foi a primeira a ser definida com os seguintes atributos:

Atributo	Tipo	Descrição
element	json	Nome do elemento da tribo
motto	json	Frase, dizer ou <i>slogan</i> da tribo
description	json	Pequeno texto sobre a da tribo

Tabela 4.1: Atributos da entidade Tribo

Na tabela 4.1 pode constatar-se que os campos **element**, **motto** e **description** são do tipo JSON, a razão para assim ser é análoga para todos os atributos das tabelas de entidades presentes nesta sub-secção que representem nomes ou descrições e que, por isso, o expectável é que fossem atributos do tipo *varchar* ou *text*, os tipos geralmente usados para representar texto. Estes campos estão representados neste tipo para que os seus valores tomem o formato de uma estrutura em JSON (pares chave-valor) que possibilitam guardar a informação em mais que uma língua (português, inglês, espanhol, entre outras) sem a necessidade de criar novos campos e, conseqüentemente, ter que alterar código para corresponder à nova estrutura da tabela atualizada. A solução adotada permite associar aos comuns códigos de dois dígitos dos países o seu valor correspondente (Exemplo: "pt": "descrição em português de Portugal", "en": "descrição em inglês").

De seguida foi definida a entidade **freguesia** que é de maior importância para os administradores do sistema uma vez que permitir-lhes-á analisar os dados de utilização da aplicação com a informação da freguesia de cada utilizador e, dessa forma, associar tendências e escolhas a cada área da cidade.

Atributo	Tipo	Descrição
name	varchar	Nome da freguesia

Tabela 4.2: Atributos da entidade Freguesia

A tabela 4.2 representa os atributos da entidade "freguesia" que é, sem dúvida, a entidade mais simples da bases de dados deste projeto, contendo apenas o identificador e o nome.

### Utilizador

Ao contrário da freguesia, a entidade **utilizador** é a mais complexa da aplicação, uma vez que está ligado a quase todas as outras, a sua estrutura está representada na seguinte lista de atributos:

Atributo	Tipo	Descrição
username	varchar	Alcunha do utilizador
email	varchar	Email do utilizador
password	varchar	Password cifrada do utilizador
gender	smallint	Género do utilizador, se tiver valor "-1" significa "não determinado"
dob	date	Data de nascimento do utilizador
is_admin	boolean	Atribui estatuto de administrador ao utilizador
is_blocked	boolean	Determina se utilizador pode ou não fazer <i>login</i>
actions	json	Estrutura que guarda informação sobre as ações <i>visita</i> , <i>recolha</i> , <i>tributo</i> e <i>troca</i> , em que templo foi executada e a data e hora da ação
progress	json	Estrutura que guarda informação sobre o progresso do utilizador em eventos, missões diárias e conquistas
inventory	json	Inventário do utilizador
tribe_id	int	Chave estrangeira da tribo a que utilizador pertence
city_zone_id	int	Chave estrangeira da freguesia a que utilizador diz pertencer no momento do registo

Tabela 4.3: Atributos da entidade Utilizador

Na tabela 4.3 podem verificar-se vários campos do tipo JSON, utilizados para manter a informação necessária para mostrar ao utilizador na aplicação e também para gerir o progresso do mesmo no jogo. O campo "progress" contém uma estrutura capaz de manter o progresso de cada missão diária, evento e conquista, por exemplo no caso de uma conquista informa se utilizador a completou e, caso tenha, se já coletou o seu prémio. O campo "actions" contém uma estrutura JSON que tem como chaves os tipos de ação que o utilizador pode realizar: "visit", "collect", "tribute" e "exchange"; a cada uma destas chaves corresponde um outro objecto JSON cujas chaves são agora os identificadores dos templos às quais correspondem *arrays* com datas e horas dessas ações. O campo "city\_zone\_id" que representa a freguesia escolhida pelo utilizador pode estar com valor "nulo", isto significa que o utilizador afirma ser turista ou não morar na cidade. Por fim o campo "inventory" contém a contagem de pontos totais e atuais do utilizador, bem como a contagem de essências de cada tipo que ele possui no momento e é atualizado sempre que o utilizador realizar alguma operação na loja ou num templo. Por último o campo "inventory" representa, como o nome indica, o inventário (ou carteira) do utilizador, constituído por uma estrutura do tipo json que contém informação sobre os valores totais e atuais dos pontos e materiais do jogo que o utilizador vai juntando.

```

actions = {
  'collect': {
    'count': 3,
    '1': ["2017-12-23 15:41:16.916884+00:00"],
    ...
    '151': ["2017-12-23 15:45:16.916884+00:00", "2017-12-23 12:15:16.916884+00:00"],
    ...
  },
  'exchange': { ... },
  'tribute': { ... },
  'visit': { ... }
}

progress = {
  'achievements': {
    // família de achievements
    'collect': {
      // nível atual de progressão dentro da família
      'current_chain_level': 0,

      // identificador de um achievement
      '1': {
        'completed': true,
        'collected': false
      },
      '5': { ... },
      ...
    },
    'visit': { ... },
    'tribute': { ... },
    'level': { ... }
  },
  'dailies': { ... }, // progressão nas missões diárias
  'events': { ... } // progressão nos eventos
}

inventory = {
  'points': {'total': 0, 'current': 0},
  'essences': {
    'fire': {'total': 0, 'current': 0},
    'water': {'total': 0, 'current': 0},
    'earth': {'total': 0, 'current': 0},
    'spirit': {'total': 0, 'current': 0},
    'wind': {'total': 0, 'current': 0}
  }
}

```

Figura 4.3: Exemplos dos campos json, complexos, da entidade utilizador

Na figura 4.3 estão representados os campos json mais complexos da entidade utilizador "actions", "progress" e "inventory" que, como foi dito anteriormente, permitem armazenar a informação necessária de uma forma dinâmica e de fácil acesso e manipulação.

### Pontos de Interesse e Categorias

Para possibilitar toda jogabilidade foi necessário estabelecer a entidade *Ponto de Interesse* onde a maior parte da ação da aplicação vai decorrer.

Atributo	Tipo	Descrição
name	json	Nome do ponto de interesse
images	json	Lista dos nomes dos ficheiros correspondentes às imagens do ponto de interesse
geometry	json	Coordenadas GPS e, caso haja informação, do noroeste e sudeste da área delimitadora do ponto de interesse
description	json	Texto sobre o ponto de interesse
active	boolean	Usar ou não o ponto de interesse na aplicação
tribe_points	json	Contador de tributos que cada tribo tem no ponto de interesse
past_conquerors	json	Array com o histórico das tribos vencedoras em cada mês

Tabela 4.4: Atributos da entidade Ponto de Interesse

Na tabela 4.4, referente à entidade que representa os pontos de interesse, pode ver-se os campos "name" e "description" com o tipo JSON pelas razões explicadas no início desta secção, permitir ter estas informações em várias línguas. O campo "images" contém uma estrutura em forma de *array* com o nome e formato das imagens desse ponto de interesse, o campo "geometry" uma estrutura em forma de objeto que contém os valores para as coordenadas GPS do ponto de interesse. Por fim o campo "tribe\_points" contém um objeto com a contagem dos tributos realizados em nome de cada tribo nesse ponto de interesse, contagem essa que é reinicializada no fim de cada mês e dá origem à atribuição de recompensas referida no capítulo anterior.

Foi também definida uma entidade **Categoria de Ponto de Interesse** para adicionar informação para o utilizador que tem interesse em conhecer realmente o sítio que está a visitar. É uma entidade puramente informativa e complementar às imagens e descrições dos pontos de interesse e que, no estado atual da aplicação, não tem qualquer uso adicional.

Atributo	Tipo	Descrição
name	json	Nome da categoria

Tabela 4.5: Atributos da entidade Categoria de Ponto de Interesse

A tabela 4.5 demonstra o quão simples é esta entidade cujo campo "name" segue a mesma lógica referida anteriormente para permitir ter o nome da categoria em mais que uma língua.

### Missão, Evento e Conquista

Uma vez que os eventos são constituídos por missões e que as conquistas são consequência de completar missões e eventos, maioritariamente, faz sentido começar por definir os atributos da entidade **Missão**, listados na tabela 4.6.

Atributo	Tipo	Descrição
name	json	Nome da missão, permite várias línguas
description	json	Descrição da missão, permite várias línguas
reward	json	Recompensa da missão
is_daily	boolean	Distinção entre as missões diárias e as demais

Tabela 4.6: Atributos da entidade Missão

Os campos "reward" das entidades **Missão** e **Evento** estão representados numa estrutura JSON que permite determinar recompensas que podem diferir em tipo e quantidade.

Atributo	Tipo	Descrição
name	json	Nome do evento
description	json	Detalhes do evento
reward	json	Recompensas do evento
location	json	Estrutura com as coordenadas GPS do centro da área do evento
image	varchar	Nome do ficheiro da imagem do evento
starts_at	datetime	Data e hora do início do evento
ends_at	datetime	Data e hora em que o evento termina

Tabela 4.7: Atributos da entidade Evento

Pode ver-se na tabela 4.7 os atributos da entidade **Evento**. As datas de início e fim servem para controlar se um evento já terminou, está a decorrer ou ainda não começou. Os restantes campos com o tipo JSON seguem o mesmo raciocínio explicado anteriormente para campos semelhantes de outras tabelas. Como consequência de participar em eventos e completar missões, o utilizador vai desbloquear conquistas que atribuem recompensas ao utilizador e servem como marcos significantes alcançados pelo mesmo. A entidade **Conquista** tem os seguintes atributos:

Atributo	Tipo	Descrição
name	json	Nome da conquista
description	json	Pequeno texto sobre a da tribo
reward	json	Informação sobre as recompensas desta conquista
family	varchar	Família da conquista (exemplo: "collect", "visit", "level", ...)
chain_level	int	"Andar" dentro da família
extra_info	json	Informação extra

Tabela 4.8: Atributos da entidade Conquista

A tabela 4.8 demonstra os campos da entidade **Conquista**, que está estruturada de uma forma bastante simples: nome e descrição (em json para permitir tê-los em várias línguas), família a que pertence e nível/andar dentro dessa família e, por fim, dois campos mais complexos que permitem representar as recompensas por completar essa conquista e informação extra para auxiliar no processo de validação de desbloqueio dessa conquista pelos utilizadores.

### Participação e Categoria de Participação

A participação pró-ativa de ocorrências na cidade, referidas na secção anterior, é uma das funcionalidades mais importantes da aplicação, uma vez que permite ao utilizador comu-

nicar com a câmara municipal tendo um papel ativo no estado de limpeza da sua cidade, por exemplo. Os atributos que definem cada participação estão representados na tabela 4.9.

Atributo	Tipo	Descrição
report.type_id	int	Chave estrangeira da categoria de participação
images	int	Contagem de imagens da participação
state	smallint	Estado da participação. 0 à espera, 1 em análise, 2 resolvido, 3 caso especial (contactar entidade)
description	text	Descrição submetida pelo utilizador
gps.location	json	Coordenadas GPS do local da participação

Tabela 4.9: Atributos da entidade Participação

O campo "images", como é referido na tabela 4.9, é apenas uma contagem das imagens submetidas na participação; as imagens em si estão guardadas numa pasta com o id da participação e numeradas a partir do 0, depois de serem submetidas a um processo de conversão para JPEG e de serem redimensionadas para não ocuparem tanto espaço no servidor. Como foi referido anteriormente, a câmara municipal realizou um estudo sobre quais as categorias fazem sentido disponibilizar, com o objetivo de conduzir o processo de cada participação para a entidade mais indicada para tratar da mesma. Esse trabalho foi aproveitado e está representado pela entidade **Categoria de Participação** com os seguintes atributos:

Atributo	Tipo	Descrição
parent_id	int	Identificador da categoria mãe
name	JSON	Nome da categoria, permite várias linguas

Tabela 4.10: Atributos da entidade Categoria de Participação

O atributo "parent\_id", listado na tabela 4.10, serve para hierarquizar as categorias. Atualmente existem apenas dois níveis na hierarquia: as categorias principais e as secundárias. Este campo pode tomar o valor "nulo" significando que trata de uma categoria principal; caso contrário é significa que é uma sub-categoria e portanto o valor representa o identificador da sua categoria mãe.

### Item da Loja e Parceiro

Especificadas as entidades que tornam possível ao utilizador ganhar pontos passa-se à especificação daquilo que os utilizadores podem fazer com esses mesmos pontos, isto é, trocar por itens na loja da aplicação. Cada item presente na loja é representado pela entidade **Item da Loja** e pode, ou não, estar associado a um **Parceiro**.

Atributo	Tipo	Descrição
name	json	Nome do item
images	json	Imagens do item
description	json	Texto sobre item
cost	int	Custo do item
limit	json	Limite de disponibilidade do item
partner_id	int	Chave estrangeira para relacionar com tabela dos parceiros
active	boolean	Permite mostrar ou esconder item na loja

Tabela 4.11: Atributos da entidade Item da Loja

Na tabela 4.11 pode observar-se a lista de atributos da entidade **Item da Loja** e entre eles é importante mencionar que o custo está expresso em *nomads* (pontos da aplicação) e que o campo "limit" permite estabelecer um limite para a compra do item, pelo utilizador, em termos temporais e em quantidade (Exemplo: utilizador só poder comprar um por mês). Em relação ao campo "partner\_id", este pode ter o valor "nulo" o que significa que o item não estará associado a nenhum parceiro.

Já a entidade **Parceiro** é constituída pelos seguintes campos:

Atributo	Tipo	Descrição
name	varchar	Nome do parceiro
url	varchar	Link para site do parceiro
description	json	Texto sobre parceiro
logo_img	json	Imagem ou logótipo do parceiro
active	boolean	Permite ativar ou desativar uma parceria
address	varchar	Morada do parceiro
gps_location	json	Coordenadas GPS do parceiro

Tabela 4.12: Atributos da entidade Parceiro

O campo "logo\_img", listado na tabela 4.12, está no formato JSON para permitir representar a imagem através de um link ou diretamente do servidor do sistema, exemplo {"type" "url/static", "src" "origem da imagem"}. Já o campo "gps\_location" terá uma representação semelhante aos restantes campos do género presentes noutras tabelas (Exemplo {"lat" 41.445943, "lng" -8.300960}).

### Mensagem Interna e Pedido de Amizade

Estas duas entidades, juntamente com o *chat*, representam o lado social da aplicação, ou seja, uma forma de ligar os utilizadores além das tribos. Os atributos que constituem a uma

**Mensagem Interna** e um **Pedido de Amizade** estão representados nas tabelas 4.13 e 4.14, respetivamente.

Atributo	Tipo	Descrição
content	varchar	Conteúdo da mensagem
receiver_id	int	Id do utilizador que recebeu a mensagem
sender_id	int	Id do utilizador que enviou a mensagem, se for nulo significa que é mensagem enviada pelo sistema
is_read	boolean	Indica se mensagem já foi lida ou não

Tabela 4.13: Atributos da entidade Mensagem Interna

O serviço de mensagens internas funcionará de forma semelhante a um email e os utilizadores terão acesso a uma caixa de entrada e de saída, com históricos das mensagens recebidas e enviadas, respectivamente. Este serviço funcionará como complemento ao *chat* da aplicação e não como principal meio de contacto entre os utilizadores. Servirá também para que o sistema consiga contactar o utilizador de uma forma não-intrusiva, isto é, sem notificações externas no dispositivo.

Atributo	Tipo	Descrição
inviter_id	int	Id do utilizador que enviou o pedido
invitee_id	int	Id do utilizador que recebeu o pedido
is_accepted	boolean	Indica se foi aceite ou não

Tabela 4.14: Atributos da entidade Pedido de Amizade

Os pedidos de amizade requerem autorização de ambas as partes, sendo necessário que ao pedido de amizade de um utilizador haja aceitação do utilizador para o qual o convite foi enviado, só então serão considerados *amigos* no contexto da aplicação.

Há ainda quatro tabelas que representam as relações N para N entre as entidades, como se pode ver na imagem 4.2, e estas representam: a lista das categorias de todos os pontos de interesse (**spots\_spot\_types**), a lista dos pontos de interesse visitados pela primeira vez pelos utilizadores (**user\_discovered\_spots**), a lista das conquistas desbloqueadas pelos utilizadores (**unlocked\_achievements**) e a lista de todas as missões de todos os eventos (**events\_missions**).

#### 4.2.2 Preenchimento da Base de Dados: Casos de Teste

Esta subsecção contém exemplos dos casos de teste utilizados. Uma vez que a cidade de Guimarães foi escolhida como exemplo prático para a aplicação que foi desenvolvida, fez

sentido preencher a base de dados com alguma informação real, como é o caso dos pontos de interesse, das categorias dos pontos de interesse, das categorias de participações e das freguesias.

### **Pontos de Interesse e Categorias**

Para obter o maior número possível de pontos de interesse, foram usados dois scripts para obter os mesmos: um através da API do *Google Places* e outro através da API do *OpenStreetMaps*. Depois de reunidos os dados em bruto, no formato JSON, foi utilizado um algoritmo de comparação das duas listas de informação com o objetivo de fundir toda a informação numa só listagem. Este algoritmo comparou a proximidade das coordenadas de GPS dos objetos (pontos de interesse), de uma lista e de outra, e efetuou um teste para encontrar uma correspondência nos nomes dos locais. Quando todo este processo dava origem a uma correspondência positiva, era criado um objeto novo que continha a maior quantidade possível de informação reunida dos dois objetos correspondidos.

Após todo este processo foram filtrados os locais que representavam cafés, discotecas, bares, restaurantes, hotéis e semelhantes, deixando apenas locais com significado histórico, igrejas, museus, monumentos, escolas, estações de comboio, entre outras. Informações como contactos, comentários e classificações (no caso do *Google Places*) e horários de funcionamento (quando aplicável) foram excluídas por não constituírem interesse para a aplicação.

O resultado final deixou um registo de cento e sessenta e cinco (165) pontos de interesse espalhados pelo centro e arredores da cidade de Guimarães e vilas vizinhas. Cada um dos pontos de interesse finais foram sujeitos a uma verificação manual das suas coordenadas GPS bem como da atribuição, também manual, de pelo menos uma fotografia.

Como exemplo de um ponto de interesse tem-se o Castelo de Guimarães, cuja informação está representada na figura 4.4.

```

castelo_de_guimaraes = {
  'name': {'pt': 'Castelo de Guimarães'},
  'categories': ["tourism", "historical"],
  'images': ["0.jpg", "1.jpg", "2.JPG", "3.jpg", "4.jpg", "5.jpg"],
  'geometry': {
    // coordenadas do centro
    "location": {
      "lat": 41.4478935,
      "lng": -8.2903191
    },
    // área definida pelos extremos a sudoeste e a nordeste
    "viewport": {
      "northeast": {
        "lat": 41.4493750802915,
        "lng": -8.288587369708498
      },
      "southwest": {
        "lat": 41.4466771197085,
        "lng": -8.291285330291503
      }
    }
  },
  'description': {'pt': 'u'No século X a Condessa Mumadona Dias, após ter ficado viúva ...', 'en': ... }
}

```

Figura 4.4: Exemplo da estrutura de um ponto de interesse / templo

Deste processo resultou, também, a lista de tipos que são usados para categorizar os pontos de interesse. Esta lista contém os seguintes tipos: "biblioteca", "bombeiros", "cemitério", "desportos", "educação", "entretenimento", "escola", "estação de comboio", "estátua", "eventos", "governo local", "histórico", "lazer", "museu", "parque", "religião", "tribunal", "turismo" e "universidade".

### Freguesias

Os dados utilizados para preencher a tabela de freguesias, da base de dados, foram retirados do site oficial da câmara municipal de Guimarães [26]. Neste site estão listadas todas as freguesias, no seu estado atual e após uniões, do município de Guimarães. Mais uma vez, este dado sobre o utilizador vai permitir à câmara municipal perceber como se distribuem os utilizadores do sistema, isto é, vai poder extrapolar a percentagem de utilizadores que são de fora ou efetivamente da cidade, bem como entre os que são, saber a que freguesia pertencem, assumindo que os utilizadores indicam essa informação de uma forma honesta.

### Categorias de Participação

Como foi referido no capítulo da análise do problema, as categorias de participação foram literalmente adaptadas da aplicação existente MyCity [5]. Estas categorias são fruto da análise realizada pela câmara municipal e estão portanto talhadas para a cidade de Guimarães:

- Animais
  - Vespas Velutinas/Ninhos
  - Cadáveres

- Outros
- **Sinalização de Trânsito**
  - Desaparecimento de Sinal
  - Sinal Derrubado
  - Sinal Vandalizado
- **Semáforos**
  - Desligados
  - Mau Funcionamento
  - Óticas Inoperacionais
  - Derrube
- **Placa de Rua**
  - Vandalizada
  - Derrubada
  - Inexistente
- **Segurança Rodoviária**
  - Rails Danificados
  - Outros
- **Equipamentos Urbanos Danificados**
  - Abrigo de Passageiros
  - Banco de Jardim
  - Vedação de Madeira
  - Papeleiras
  - Parque Infantil
  - Ecoponto, Contentor ou Molok
- **Limpeza de Via Pública**
  - Lixo à Volta dos Contentores/Ecopontos
  - Deposição de Lixo Fora de Horas
  - Contentores Cheios
  - Varredura de Via Pública

- Lavagem de Passeios
- Recolha de Monstros na Via Pública

- **Danos no Espaço Público**

- Talude ou Muro em Risco
- Estrada ou Passeio Danificado
- Rotura Conduta
- Tampas de Saneamento
- Tampas Águas Pluviais
- Candeeiros Danificados
- Outros

- **Limpeza Linhas de Água**

- Lixo nas Margens Ribeirinhas
- Foco de Poluição
- Descarga

Cada participação poderá ter apenas uma categoria e uma sub-categoria, para que possa ser encaminhada e, posteriormente, resolvida pela entidade pública adequada à sua resolução.

### 4.3 SERVIDOR

Nesta secção estão explicadas as escolhas para a tecnologia do servidor e uma amostra da API implementada para suportar a aplicação móvel.

Com base no estudo apresentado no capítulo do estado da arte sobre as tecnologias para o servidor *web*, por terem uma curva de aprendizagem pouco acentuada permitindo um desenvolvimento mais rápido do protótipo final e por preferência pessoal do autor, a decisão final de qual tecnologia utilizar foi feita entre as micro-frameworks Flask[38] e Express.js[36]. Acabou por ser usada a Flask, por ser em Python, linguagem que permite desenvolver mais com menos código (o que ajuda a acelerar bastante o processo de desenvolvimento), por necessitar de menos bibliotecas externas que a Express.js para realizar as mesmas operações e por ter um *setup* mais simples e direto. Os módulos externos utilizados foram: *flask-login* para a gestão das sessões de utilizador, *flask-socketio* para os web sockets que permitirão o funcionamento do chat e das notificações, *arrow* para a gestão das datas e horas, *flask-sqlalchemy* como biblioteca de ORM (mapeamento da base de dados em entidades), *Pillow* para manipulação de imagens, *passlib* para encriptação das passwords

dos utilizadores, *geopy* para o cálculo de distâncias entre pares de coordenadas GPS e *httpagentparser* para a determinação do origem de cada pedido (Android, iOS, browser ou outro).

#### 4.3.1 API do Servidor

Como foi dito anteriormente, a aplicação vai comunicar com o servidor através de pedidos HTTP: GET ou POST. A API disponibiliza as seguintes ações:

- **Verificação do estado de login de um utilizador/dispositivo**

Quando a aplicação é ligada, este é o primeiro pedido efetuado ao servidor. O servidor, por sua vez, responde com um objeto JSON que efetivamente responde sim ou não e se responder sim envia, também, informações necessárias para ajudar a aplicação a renderizar a sua página inicial;

- **Registar**

O servidor recebe pedido para registar com *username*, email e outras informações incluídas, e faz a verificação da existência de utilizadores registados ou com esse *username* ou esse email. Caso o teste seja negativo, isto é, caso ambas as credenciais estejam desocupadas o registo prossegue. É criada um diretório para o utilizador dentro do diretório das imagens de utilizadores e, caso o utilizador não tenha submetido nenhuma fotografia para usar como *avatar*, é escolhido o *avatar* generalista para homem ou mulher consoante o género escolhido pelo utilizador. Mais uma vez, se o utilizador não tiver escolhido nenhum género é-lhe atribuído o *avatar* de homem, por definição, pelo simples facto de que os estudos observados [23, 24, 25] indicam que a maioria dos jogadores em plataformas móveis são do sexo masculino, embora a diferença não seja tão grande quanto se possa pensar;

- **Login**

Pedido normal de *login* com credencial, nome de utilizador ou email, e palavra chave. Se for encontrado um utilizadores com essas credenciais é feita a verificação e dada uma resposta que corresponda com a verificação;

- **Logout**

Pedido normal de terminar a sessão, servidor termina a sessão do utilizador no dispositivo em causa através do seu token de sessão;

- **Listagem de Participações**

Ao receber este pedido o servidor responde com a lista de participações submetidas

pelo utilizador que a pediu, que vai servir para desenhar na aplicação essa mesma lista;

- **Submissão de uma Participação**

Utilizador envia pedido de submissão de nova participação. Esse pedido traz informações como coordenadas de GPS, descrição da cena, categoria de participação e pode ou não conter imagens. Se tiver imagens é criada uma pasta para esta nova participação onde serão depositadas, depois de otimizadas e redimensionadas para ocuparem o menor espaço possível no servidor;

- **Ações da Página de Administração sobre:**

- Utilizadores:

- \* Apagar: Apaga utilizador(es) selecionados da base de dados;
- \* Bloquear: Coloca estado *bloqueado* nos utilizadores selecionados;
- \* Desbloquear: Coloca estado *ativo* nos utilizadores selecionados.

- Participações:

- \* Apagar: Apaga as participações selecionadas
- \* Analisar: Coloca estado *em análise* nas participações selecionadas, atribui uma quantia de pontos ao utilizador e notifica-o quando entrar novamente na aplicação através de uma mensagem interna enviada pela sistema;
- \* Concluído: Coloca estado *concluída* em cada participação selecionada, atribui uma quantia de pontos ao utilizador e notifica-o quando entrar novamente na aplicação através de uma mensagem interna enviada pela sistema;
- \* Especial: Coloca estado *especial* nas participações selecionadas.

- Pontos de Interesse / Templos:

- \* Apagar: Apaga os pontos de interesse selecionados
- \* Esconder/Mostrar: Altera o estado para *inativo* ou *ativo* dos pontos de interesse selecionados, respetivamente;
- \* Guardar: Guarda novas informações sobre o ponto de interesse selecionado com a informação enviada no pedido com o formulário preenchido pelo administrador;
- \* Criar: Guarda novo ponto de interesse, na base de dados, com a informação enviada no pedido com o formulário preenchido pelo administrador;
- \* Listagem das categorias dos pontos de interesse.

- Imagens dos pontos de interesse:

- \* Apagar: Apaga as imagens selecionadas do ponto de interesse em questão, eliminando os ficheiros e o registo das mesmas na base de dados;
- \* Adicionar: Adiciona as novas imagens submetidas, copiando os ficheiros enviados para a pasta que corresponde a esse ponto de interesse. Cada imagem é submetida a um processo de verificação de tamanho (em bytes) e dimensões, sendo redimensionada se necessário e passando por um processo de otimização para ocupar menos espaço. No fim deste processo é também adicionado o nome desse ficheiro, também ele alterado, à lista de imagens do ponto de interesse na base de dados.

– Parceiros:

- \* Apagar: Apagar parceiros selecionados, bem como os itens da loja que dizem respeito a esse parceiro;
- \* Esconder/Mostrar: Altera o estado para *inativo* ou *ativo* dos parceiros selecionados, respetivamente;
- \* Detalhes: Retorna detalhes do parceiro, para possível edição;
- \* Editar: Guarda novas informações submetidas pelo administrador sobre determinado parceiro;
- \* Adicionar: Guarda novo parceiro na base de dados com as informações submetidas no pedido.

– Eventos:

- \* Apagar: Apaga eventos selecionados, atualiza progresso do evento de todos os utilizadores inscritos, que estavam a participar ou que tinham participado no evento. Se evento for apagado antes de terminar, os utilizadores participantes recebem as recompensas das missões que tinham concluído e uma compensação pelas que não conseguiram ou tiveram tempo de concluir;
- \* Editar: Atualiza as informações do evento com os novos dados submetidos pelo administrador;
- \* Criar: Guarda novo parceiro na base de dados com as informações submetidas no pedido;
- \* Criar Missão: Cria e adiciona missão a evento, com todos os detalhes enviados no pedido (tarefa e recompensa), e adiciona missão ao evento.

– Loja:

- \* Apagar: Coloca os itens selecionados com o estado *inativo* e, como tal, invisíveis para os utilizadores;

- \* Editar: Atualiza informações sobre um item como nome, imagem do item, preço, se está ou não em promoção, entre outras coisas;
- \* Adicionar: Adiciona novo item à loja com informações passadas no pedido.

- **Pedidos Relacionados com Pontos de Interesse:**

- **Listagem dos Templos/Pontos de Interesse**

Geralmente pedido uma única vez quando o utilizador vê pela primeira vez a página inicial da aplicação, depois de efetuar *login*. Devolve a lista de pontos de interesse pouco detalhada, apenas com o nome e as coordenadas GPS de cada um;

- **Ações do Utilizador no Templo/Ponto de Interesse:**

- \* Explorar: Adiciona o templo à lista de templos descobertos pelo utilizador e devolve resposta com sucesso/insucesso;
- \* Conversão: Transforma a quantidade pedida de *essências* do utilizador na quantidade correspondente de pontos e atualiza perfil do utilizador, devolve resposta com sucesso/insucesso;
- \* Recolha: Calcula quantidades aleatórias de cada *essência* e atribui-as ao utilizador, atualiza o perfil do mesmo e devolve resposta com sucesso/insucesso e novas quantidades resultantes no inventário do utilizador;
- \* Tributo: Desconta quantidade de *essências* necessárias para tributo e atualiza perfil de utilizador, atualiza contagem de tributos no templo para a tribo do utilizador e devolve resposta com sucesso/insucesso, novas contagens de tributos do tempo e inventário do utilizador atualizado após a operação.

#### 4.3.2 Chat e Notificações

Como qualquer aplicação deste género, não pode faltar uma maneira de notificar em tempo real o utilizador quando recebe uma mensagem ou quando algo aconteceu na aplicação que lhe possa suscitar interesse. Isto é importante por dois motivos:

- **Se o utilizador for bastante ativo:** Este tipo de notificações ajudam a manter o interesse a avisá-lo que algo está disponível, como um novo evento, ou o já poder fazer tributo num templo por ter terminado o tempo limite;
- **Se o utilizador ainda estiver a ganhar o interesse pela aplicação:** Sem grande intrusão ou insistência, pode ser uma boa forma de lhe lembrar de abrir a aplicação novamente.

Isto acrescenta uma nova dinâmica que com a estrutura inicial do servidor não era possível, uma vez que este só era capaz de enviar informação para o cliente mediante um pedido do mesmo.

Para resolver este problema foi usada uma extensão do *Flask* chamada *flask-socketio*. O que esta extensão faz é criar um *websocket*, onde de facto há conexão do cliente ao servidor ao contrário do que acontecia até agora, e que permite através de uma API muito simples criar canais de conversação, como um *chat*, que permitem enviar notificações para a aplicação, caso esta esteja ligada.

Esta extensão é, por isso, perfeita para fazer canais de *chat* e vai ser utilizada para oferecer ao utilizador algumas soluções para envio instantâneo de mensagens dos tipos "um para um" ou "um para muitos", através do método *emit* com ou sem a opção *broadcast* ativa. Como tal, o utilizador terá à possibilidade de falar com os outros utilizadores individualmente ou de se dirigir aos utilizadores da sua tribo ou da aplicação em geral, que estejam *online*, através dos canais respetivos.

#### 4.3.3 Área de Administração

Esta subsecção contém imagens que representam o estado atual do painel de controlo que os administradores do sistema têm à sua disposição para controlar o conteúdo da aplicação. Nele eles podem controlar os utilizadores, as participações, os eventos, os parceiros, os produtos da loja, os pontos de interesse (templos), categorias de pontos de interesse e toda a informação relativa a estas entidades.

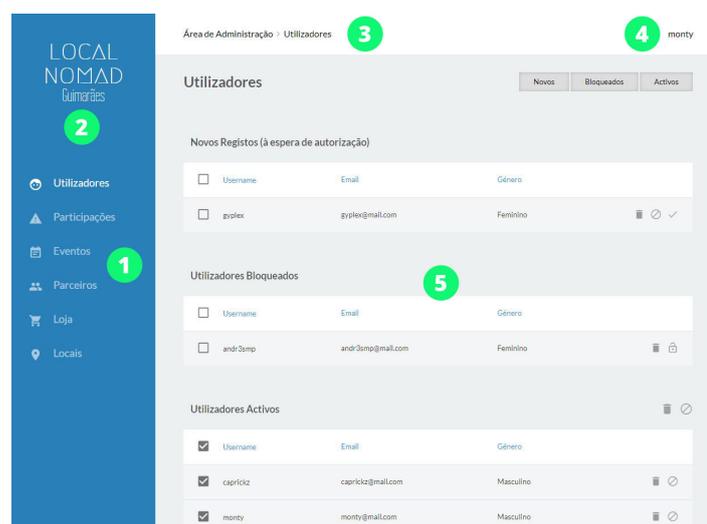


Figura 4.5: Estrutura da página de administração/painel de controlo

Na figura 4.5 pode ver-se uma versão básica da área de administração onde se podem ver círculos verdes numerados que identificam os diferentes componentes da página:

1. Navegação entre *views*;
2. Logótipo com *link* para página inicial de um futuro site que poderá ser desenvolvido (não está nos planos para o decorrer do projeto, estará anotado como trabalho futuro);
3. *Breadcrumbs*, apenas uma nota de que *view* se está a ver, neste caso puramente estético;
4. Navegação de utilizador administrador, com link para o seu perfil (não vai ser implementado) e opção para *logout*;
5. Corpo da página onde estão as *views*, este conteúdo é dinâmico mas todo renderizado no carregar da página.

### Utilizadores

Como foi detalhada na subsecção da API do servidor, os administradores do sistema têm à sua disposição a capacidade de controlar que utilizadores podem ou não utilizar o sistema, não podem editar os dados desses utilizadores mas podem apagá-los ou bloqueá-los caso estes violem alguma regra do sistema ou tenham um nome ou email ofensivos.

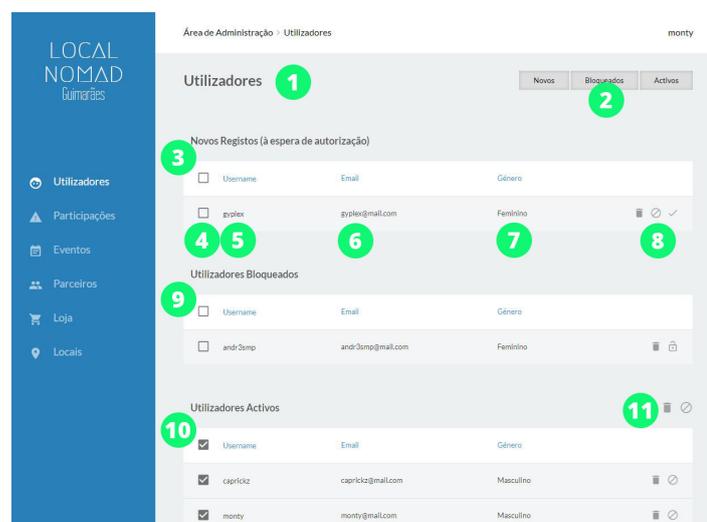


Figura 4.6: Página dos utilizadores no painel de controlo do sistema

Os componentes importantes/principais da *view* dos utilizadores estão evidenciados na figura 4.6 e têm os seguintes significados:

1. Título da *view*;
2. Botões para esconder/mostrar tabelas de utilizadores;
3. Tabela de utilizadores registados que ainda não foram aceites no sistema;

4. *Checkbox* para seleccionar linha da tabela que serve para as opções da tabela (ponto 11);
5. Alcunha do utilizador;
6. Email do utilizador;
7. Género do utilizador;
8. Opções sobre conta do utilizador
  - Opções da tabela **Novos Registos**: Apagar, Bloquear e Aceitar;
  - Opções da tabela **Utilizadores Bloqueados**: Apagar e Desbloquear;
  - Opções da tabela **Utilizadores Ativos**: Apagar e Bloquear.
9. Tabela dos utilizadores que foram bloqueados;
10. Tabela dos utilizadores ativos, isto é, que estão aceites pelo sistema, a jogar e, portanto, não estão bloqueados;
11. Opções de tabela, iguais às opções sobre conta do utilizador explicadas no ponto (8).

### Participações

A secção das participações mantém o mesmo esquema da dos utilizadores, desta vez com quatro tabelas diferentes que representam os quatro estados em que uma participação pode passar: *Por aceitar*, *Em análise*, *Resolvida* e *Caso especial*.

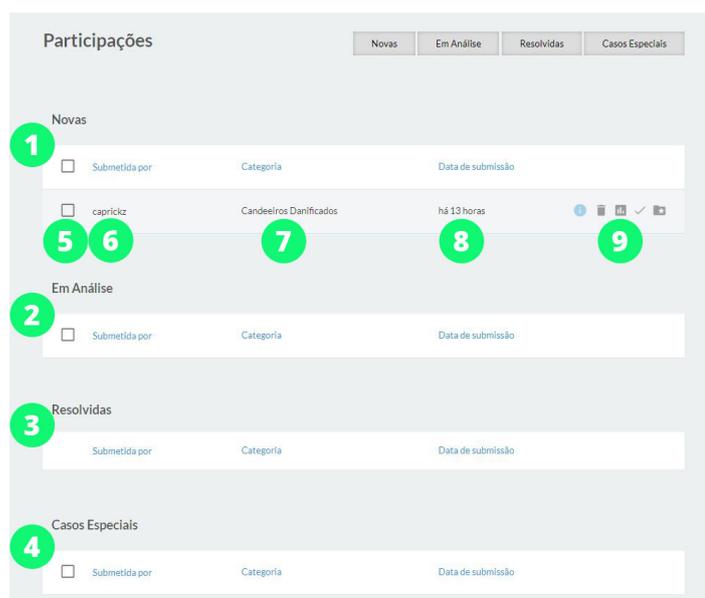


Figura 4.7: Página das participações no painel de controlo do sistema

Os componentes principais da *view* das participações estão evidenciados na figura 4.7 e os seus significados são:

1. Tabela das participações submetidas que ainda não foram aceites;
2. Tabela das participações em análise;
3. Tabela das participações que já foram resolvidas/tratadas;
4. Tabela das participações que serão tratadas como casos especiais pela entidade administradora do sistema;
5. *Checkbox* para seleccionar linha da tabela;
6. Nome do utilizador que submeteu a participação;
7. Sub-categoria da participação;
8. Quando foi submetida a participação;
9. Opções da participação (ver imagem 4.8).

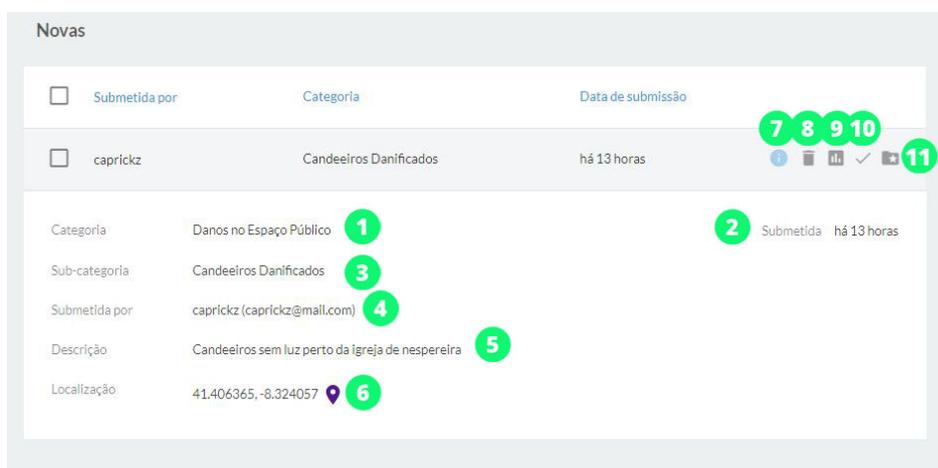


Figura 4.8: Estrutura da janela com detalhes da participação aberta

Ao carregar no botão de informações no campo 9 da figura 4.7 faz com que abra ou feche a janela que podemos ver na figura 4.8, cuja estrutura é a seguinte:

1. Categoria mãe da participação;
2. Quando foi submetida a participação;
3. Sub-categoria da participação;

4. Autor da participação (nome de utilizador e email caso haja necessidade de o contactar);
5. Descrição que o autor providenciou na participação;
6. Local da participação (com link para google maps);
7. Abrir/Fechar janela de detalhes (abrir uma fecha outra que estivesse aberta);
8. Apagar participação;
9. Mudar estado da participação para *em análise*;
10. Mudar estado da participação para *resolvida*;
11. Mudar estado da participação para *caso especial*.

Como é de esperar, as opções de cada participação não são idênticas para todas as tabelas e, como tal, as opções para cada tabela, excluindo o botão de informações e o de apagar que são comuns a todas, são:

- **Tabela novas participações:**
  - Passar para *em análise*;
  - Passar para *resolvidas*;
  - Passar para *casos especiais*.
- **Tabela de participações em análise:**
  - Passar para *resolvidas*;
  - Passar para *casos especiais*.
- **Tabela de participações resolvidas:**
  - Sem outras opções.
- **Tabela de participações tratadas como casos especiais:**
  - Passar para *resolvidas*.

### Parceiros

A secção das parceiros segue a mesma arquitetura das anteriores. Desta feita, é representada por duas tabelas que fazem a separação, evidente, entre parceiros *ativos* e *inativos*.

Na figura 4.9 podem ver-se enumerados os seguintes componentes:

1. Botão para aparecer formulário para adicionar novo parceiro (ver figura 4.10);

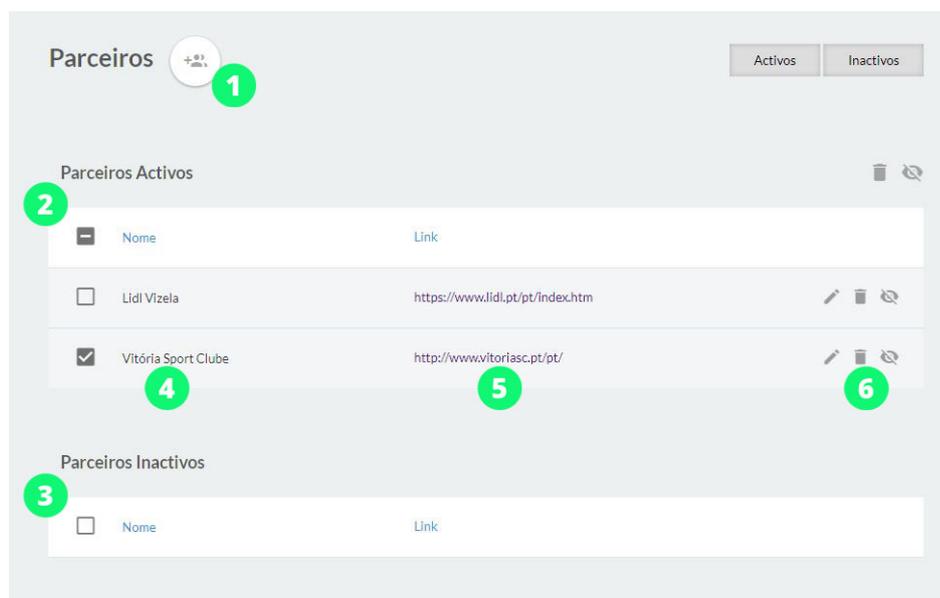


Figura 4.9: Página das parcerias no painel de controlo do sistema

2. Tabela das parcerias ativas;
3. Tabela das parcerias inativas;
4. Nome do parceiro;
5. Link para site do parceiro;
6. Opções sobre parceria (Editar, Apagar e Ativar/Desativar).

Esta secção tem duas funcionalidades novas (em relação às anteriores), a de adicionar novo parceira e a de editar informações de parceiro existente, como se pode ver nas figuras 4.10 e 4.11.

The image shows a web application interface for 'LOCAL NOMAD Guimarães'. On the left is a dark blue sidebar with navigation options: Utilizadores, Participações, Eventos, Parceiros, Loja, and Locais. The main content area is a modal form for adding a new partner. The form has the following fields and features:

- 1**: 'Nome (obrigatório)' - A text input field for the partner's name.
- 2**: 'Link do site' - A text input field for the partner's website link.
- 3**: 'Descrição' - A large text area for the partner's description.
- 4**: 'Imagem' - A section with radio buttons for 'URL' and 'Ficheiro Local' to select the image source.
- 5**: 'URL da imagem' - A text input field for the image URL.
- 6**: 'Localização' - A section with 'Latitude' and 'Longitude' labels and input fields. Example values are 'Exemplo: 41.447877' and 'Exemplo: -8.290308'.
- 7**: 'Morada' - A text input field for the partner's address.

At the bottom of the form are two buttons: 'CANCELAR' (grey) and 'CRIAR' (blue). The background shows a list of existing partners with columns for 'Nome', 'Link do site', 'Descrição', 'Imagem', and 'Localização'.

Figura 4.10: Formulário para adicionar novo parceiro

O formulário representado na figura 4.10 que aparece quando se carrega no botão(1) da figura 4.9 tem sete campos: **Nome (1)**, **Link para site (2)**, **Descrição (3)**, **Tipo e origem (url ou local) da imagem (4)**, **Pré-visualização da imagem escolhida no campo 4 (5)**, **Latitude e Longitude da sede/loja/armazém do parceiro (6)** e **Morada (7)**.

Uma vez que o campo *descrição* do parceiro na base de dados é do tipo JSON, o servidor trata de maneira especial o pedido de edição ou de criação de nova parceria. É feita a avaliação do conteúdo da descrição, determinando a língua que está a ser utilizada. É, também, verificada da origem do IP local se for necessário ter um fator adicional para ajudar a decidir, caso o primeiro método não tenha funcionado.

Determinada a origem, o campo *descrição* do parceiro é atualizado adicionando a chave da língua (por exemplo: "pt", "de" ou "en") ao objeto JSON, caso este ainda não exista, com o respetivo valor da descrição submetida no pedido. Se já existir, no caso de se estar a editar uma parceria, o valor, correspondente à chave, é simplesmente atualizado.

Figura 4.11: Formulário para editar informações de parceira

## Locais

Esta *view* segue uma lógica semelhante mas com ligeiras diferenças. Aqui há apenas uma tabela onde o estado de *ativo* ou *inativo* de um Local/ponto de interesse se distingue pela cor, vermelho se o local estiver com estado inativo e verde se estiver com o estado ativo, como se pode ver na figura 4.12.

A legenda da figura 4.12, onde se pode distinguir 10 componentes importantes, é a seguinte:

1. Botão para janela de adicionar novo ponto de interesse;
2. Botão para janela de gerir categorias de pontos de interesse;
3. Esconder/Mostrar pontos de interesse ativos na tabela;
4. Esconder/Mostrar pontos de interesse inativos na tabela;
5. Nome do ponto de interesse;
6. Link para google maps (coordenadas do local);
7. Categorias do local;

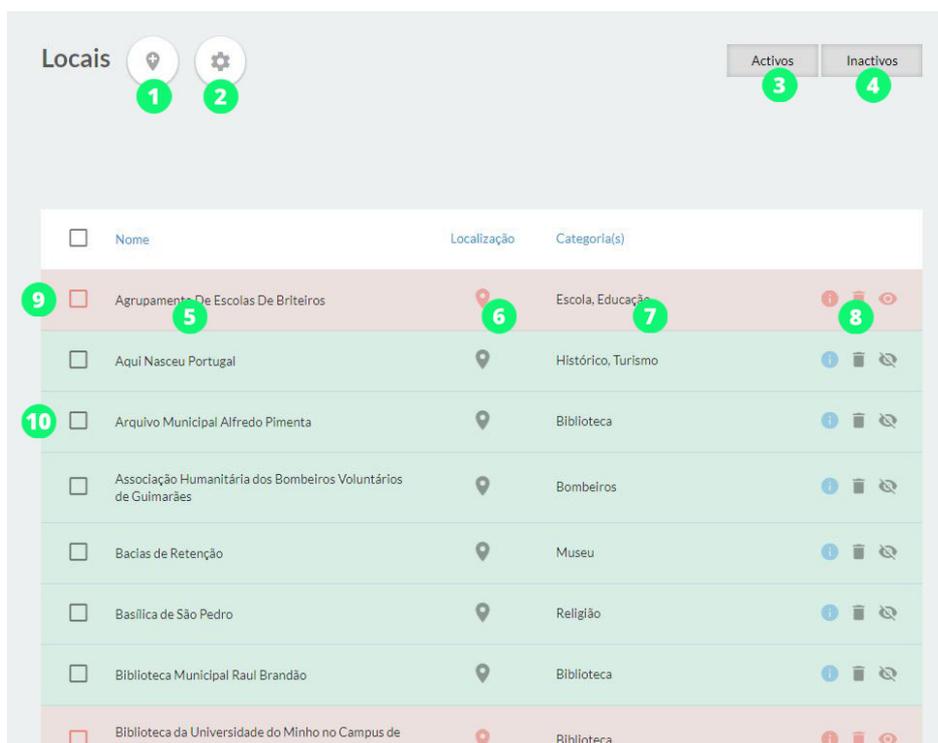


Figura 4.12: Página dos locais no painel de controlo do sistema

#### 8. Opções do local

- Botão para abrir/fechar janela de detalhes do local (ver figura 4.13);
- Apagar local;
- Tornar local ativo/inativo.

9. Exemplo de um local inativo (linha avermelhada);

10. Exemplo de um local ativo (linha esverdeada).

Nas opções de cada local pode ver-se um botão semelhante ao encontrado na secção das participações, que é o botão das informações ou detalhes. Este botão, também à semelhança do que acontece nas participações, abre uma janela de detalhes, neste caso, do local. Neste janela há duas subsecções, relacionadas com o local: a primeira serve para editar os dados do local e a segunda para editar as imagens do local, como se pode constatar na figura 4.13.

Ao abrir a janela de edição de um local tem-se os seguintes componentes:

1. Campo do nome;
2. Botão para editar informações textuais e categorias de local (ver figura 4.14);

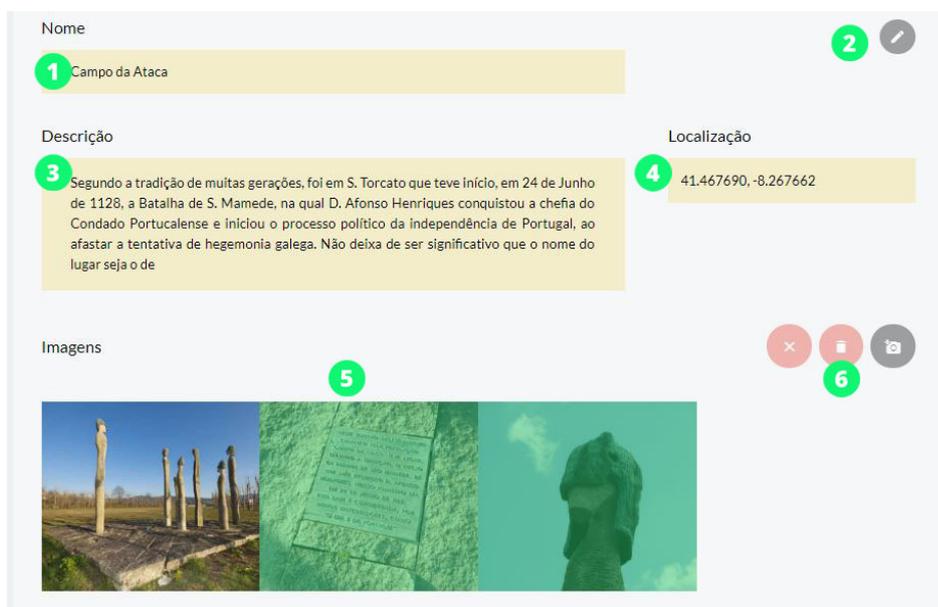


Figura 4.13: Janela de edição de informações sobre local

3. Campo da descrição;
4. Campo das coordenadas GPS do local;
5. Lista de imagens. Clicar numa imagem seleciona-a ou desseleciona-a e quando uma ou mais imagens estão selecionadas o botões vermelho do ponto 6 aparecem;
6. Opções da lista de imagens
  - Limpar seleção: todas as imagens deixam de estar selecionadas e opções a vermelho desaparecem;
  - Apagar: todas as imagens selecionadas são apagadas, isto significa que os ficheiros das imagens são apagados do diretório respetivo assim como o campo *imagens* do local na base de dados é atualizado para não as conter;
  - Carregar (opção a cinzento): serve para adicionar imagens ao local, esta opção está sempre visível.

Como foi referido no ponto 2 da explicação anterior, carregar no botão de editar informações do local permite editar os campos que estão bloqueados. Estes campos são então desbloqueados e é revelada também a lista completa dos tipos ou categorias de local para que o utilizador administrador possa escolher os que mais se adequam ao local, como se pode ver na figura 4.14.

### Loja

A secção que permite a gestão da loja, isto é, aos itens que vão estar disponíveis para

Nome

Campo da Ataca

Descrição

Junho de 1128, a Batalha de S. Mamede, na qual D. Afonso Henriques conquistou a chefia do Condado Portucalense e iniciou o processo político da independência de Portugal, ao afastar a tentativa de hegemonia galega. Não deixa de ser significativo que o nome do lugar seja o de "Campo da Ataca" - ou do ataque - designação guerreira bem sugestiva. Em 1996 foi inaugurado o actual arranjo artístico-monumental, que celebra este acontecimento.

Localização

41.467690, -8.267662

Tipos

biblioteca  bombeiros  cemitério  desportos

educação  entretenimento  escola  estação de comboio

estátua  eventos  governo local  histórico

lazer  museu  parque  religião

tribunal  turismo  universidade

Figura 4.14: Exemplo: Editar informações do local

os utilizadores trocaram por pontos, segue um *design* semelhante às demais. Esta secção contempla duas tabelas que separam os itens ou produtos ativos dos inativos, neste caso isto significa aqueles produtos que o utilizador vai ver na loja e os que não vão aparecer.

Loja  1

Activos Inactivos

Produtos activos

<input type="checkbox"/>	Nome	Parceiro	Custo (em pontos)	Limite	
<input type="checkbox"/>	Equipamento principal VSC	Vitória Sport Clube	2250	Nenhum	  
<input type="checkbox"/>	Bilhete Duplo VSC	Vitória Sport Clube	5500	1 por mês	  

Figura 4.15: Página da Loja no painel de controlo do sistema

Na figura 4.15 pode-se distinguir seis componentes:

1. Botão para adicionar novo item à loja;
2. Nome do item;
3. Parceiro ao qual o item está associado;
4. O valor, em pontos, que o item vai custar na loja *in-app*;

5. O limite para compra do item. Se tiver limite este está separado em dois campos **quantidade** e **prazo/duração**. No caso da figura 4.15 pode ver-se um item com limite de *1 por mês*, como a frase o indica, quer dizer que **num mês** cada utilizador só poderá comprar este item **uma vez**;
6. Opções do item
  - Ver informações: esta opção funciona de maneira análoga à dos locais, por exemplo. É aberta uma janela auxiliar onde estão dispostas as informações do item, bem como as suas imagens e, tal como no caso dos locais, opções que permitem ao utilizador editar essas informações e imagens;
  - Apagar: Apaga o item da loja, utilizadores que já o tenham comprado poderão na mesma levantar o item uma vez que um comprovativo de compra é enviado para o email do utilizador com a data e hora de quando foi comprado, responsabilizando a entidade administradora do sistema pelo cumprimento da transação realizada;
  - Mostrar/Esconder: Como nas secções demonstradas anteriormente, esta opção serve para cativar ou desativar um item, colocando-o disponível na loja para todos os utilizador ou escondendo-o dos mesmos, respetivamente.

### Eventos

Esta é a secção mais complexa da página de administração. É onde se criam e geram eventos e missões para os mesmos. Ora se uma missão tem apenas três campos (nome, descrição e recompensas), como foi mencionado anteriormente, o evento é uma entidade mais complexa que pressupõe, também, datas de início e de fim, uma imagem e um par de coordenadas, latitude e longitude, que representam o ponto central da área do evento.

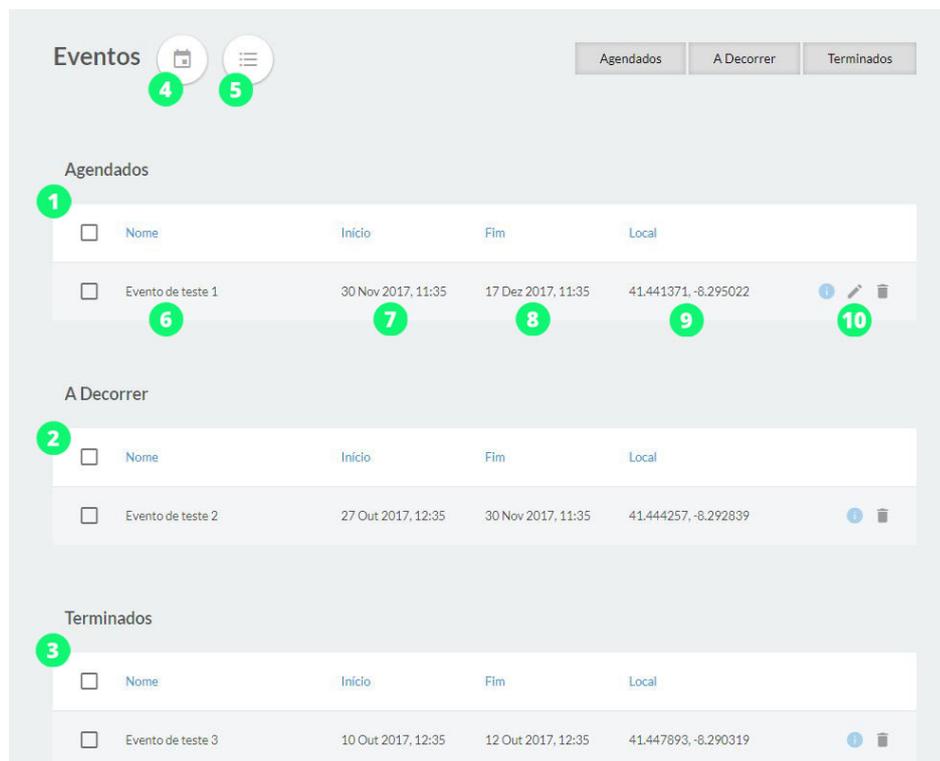


Figura 4.16: Página dos Eventos no painel de controlo do sistema

Como se pode ver na figura 4.16, a página segue um *design* semelhante às outras com os seguintes componentes:

1. Tabela dos eventos agendados;
2. Tabela dos eventos que estão a decorrer;
3. Tabela dos eventos que já terminaram;
4. Botão para agendar novo evento;
5. Botão para gerir missões de eventos;
6. Nome do evento;
7. Data e hora do início do evento;
8. Data e hora do fim do evento;
9. Coordenadas do local do evento;
10. Opções do evento:
  - Ver informações;

- Editar evento;
- Apagar evento.

Carregando no botão de agendar novo evento surge um formulário para o administrador personalizar o nome, a descrição, a localização, as recompensas, as datas e as missões do evento.

Agendar novo evento

Nome

1 Conhecer o centro

Descrição

2 Evento para todos os jogadores e de todas as idades, o objectivo é fazer missões enquanto conhecem melhor o centro da cidade e os seus mistérios

Período

Início 2017/12/06 17:03

Fim 2017/12/21 17:03

4

Localização

5

Mapa Satélite

Dezembro - 2017 -

Dom	Seg	Ter	Qua	Qui	Sex	Sab
						17.03
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

Figura 4.17: Exemplo: Novo evento, parte 1

Período

Início 2017/12/06 17:03

Fim 2017/12/21 17:03

Localização

6

Mapa Satélite

7 41.44391540391101, -8.293722867965698

Recompensas

Essências de Fogo	Essências de Água	Essências de Vento	Essências de Espírito	Essências de Terra	Pontos
8 25	25	25	25	25	2000

Escolher missões

9

Tributos à cidade	Homenagens, muitas homenagens
Fazer 10 tributos em templos da cidade <input type="checkbox"/>	Prestar 40 homenagens em templos da cidade <input checked="" type="checkbox"/>

CANCELAR AGENDAR

Figura 4.18: Exemplo: Novo evento, parte 2

Nas figuras 4.17 e 4.18 podem-se ver os seguintes componentes, que permitem essa personalização:

1. Nome do novo evento;
2. Descrição do novo evento;
3. Data do início do novo evento;
4. Data do fim do novo evento;
5. Mapa para seleção do centro da área do novo evento;
6. Exemplo de um local escolhido para o centro da área do evento;
7. Coordenadas GPS resultantes da escolha do local exemplo;
8. Seleção das recompensas para o novo evento;
9. Seleção das missões a ser incluídas no novo evento.

Este é o conteúdo da página de administração, com estas funcionalidades um administrador poderá controlar facilmente o conteúdo que está disponível na aplicação para os utilizadores verem e usufruírem.

#### 4.4 APLICAÇÃO MÓVEL

Esta secção contém informação sobre as tecnologias utilizadas para desenvolver a aplicação móvel. Contém ainda um fluxograma que explica a navegação na aplicação bem como exemplos legendados e detalhados das páginas da aplicação que dão a ideia da experiência que o utilizador terá ao utilizar a mesma.

Depois de analisadas várias alternativas e tendo em conta a preferência do autor por tecnologias *web*, foi escolhido o Cordova[30] como *framework* de desenvolvimento para a aplicação móvel. Córdoba não só permite implementar toda a aplicação usando tecnologias *web* (HTML, CSS e JavaScript), como tem ainda um suporte mínimo para os sensores do dispositivo, neste caso o GPS e a bússola que são os únicos que interessam para o desenvolvimento desta aplicação. Permite ainda compilar o mesmo código para Android e iOS, evitando a obrigação de ter que desenvolver aplicações separadas caso se tivesse optado por desenvolver a aplicação nativamente.

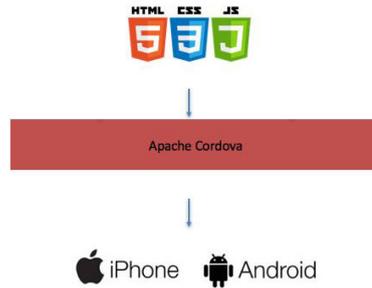


Figura 4.19: Cordova: como funciona?

Como tal, além de reunir todas as condições necessárias para o desenvolvimento da aplicação móvel deste projeto, o facto de permitir desenvolvê-la com recurso a tecnologias *web* vai acelerar substancialmente o processo de desenvolvimento, o que é vital num projeto desta natureza.

#### 4.4.1 Navegação da Aplicação

A página principal será o nó central ou a referência na ligação a qualquer outra página da aplicação. O fluxograma da figura 4.20 representa como funciona a navegação dentro da aplicação.

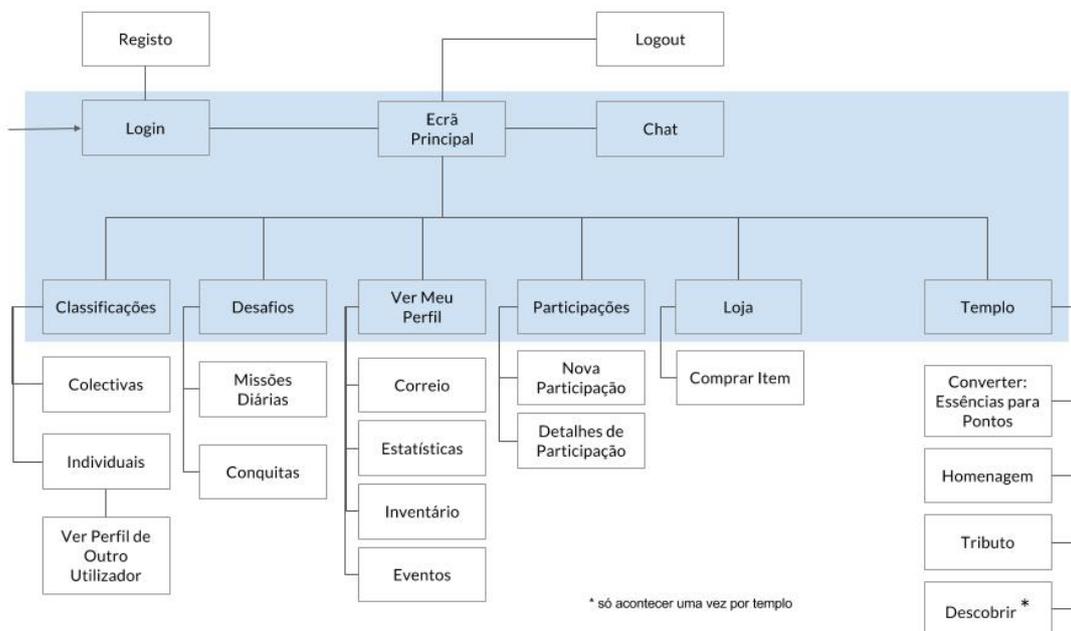


Figura 4.20: Fluxograma de navegação da aplicação

#### 4.4.2 Páginas da Aplicação

Depois de esquematizada a navegação da aplicação, falta falar sobre o *design* da aplicação e os componentes de cada página ou *view*.

Os *ícones* da aplicação foram, cuidadosamente, escolhidos entre os universalmente reconhecidos para a função que vão desempenhar e as cores suficientemente contrastantes com o fundo escuro do mapa, com o intuito de realçar os botões e componentes importantes de cada página.

Em relação ao tipo de letra, este é a mesma escolhida para a parte de administração do sistema, Lato das *Google Fonts*.

##### Página Inicial

Começando pela página inicial, após efetuar registo e *login*, o utilizador depara-se com um ecrã relativamente limpo e minimalista que muda de esquema de cores consoante a tribo que escolheu, ao efetuar o primeiro *login* na aplicação.

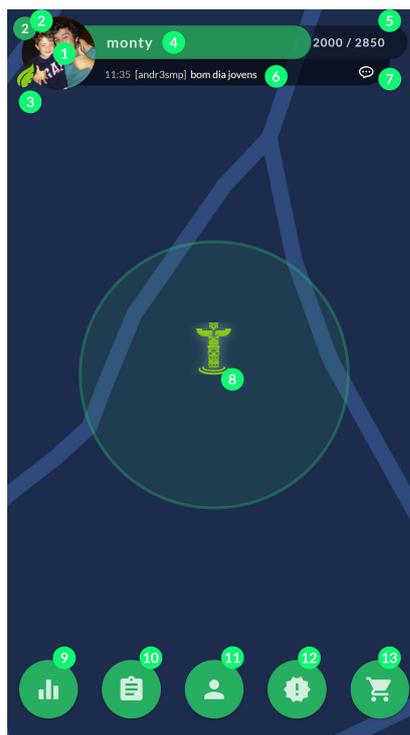


Figura 4.21: Aplicação: Página Inicial

Na figura 4.21 destacam-se os treze (13) componentes da página inicial que são:

1. Zona do *avatar* /foto de perfil;

2. Nível atual do utilizador;
3. Ícone da tribo do utilizador;
4. Nome do utilizador diretamente por cima da barra de progresso para o próximo nível;
5. Progresso para próximo nível (pontos atuais / pontos totais para próximo nível);
6. Última mensagem do *chat*;
7. Botão para abrir página do *chat*;
8. Exemplo de um templo a aparecer no mapa (ícone do templo da imagem está a verde porque, no momento em que a imagem foi criada, o templo tinha mais tributos da tribo da terra);
9. Botão que abre página das classificações;
10. Botão que abre página dos desafios;
11. Botão que abre página de perfil pessoal do utilizador;
12. Botão que abre página das participações;
13. Botão que abre página da loja.



Figura 4.22: Aplicação: Página Inicial, 2

Na figura 4.22 pode ver-se o UI da página inicial mudar dependendo da tribo que o utilizador escolher.

### Visitar um Templo

Ao carregar num templo (ponto 8 na figura 4.21) surge a página do mesmo. Nela podem ver-se as informações sobre o templo como o nome, imagens e a contagem de tributos feitos pelas diversas tribos no mesmo.

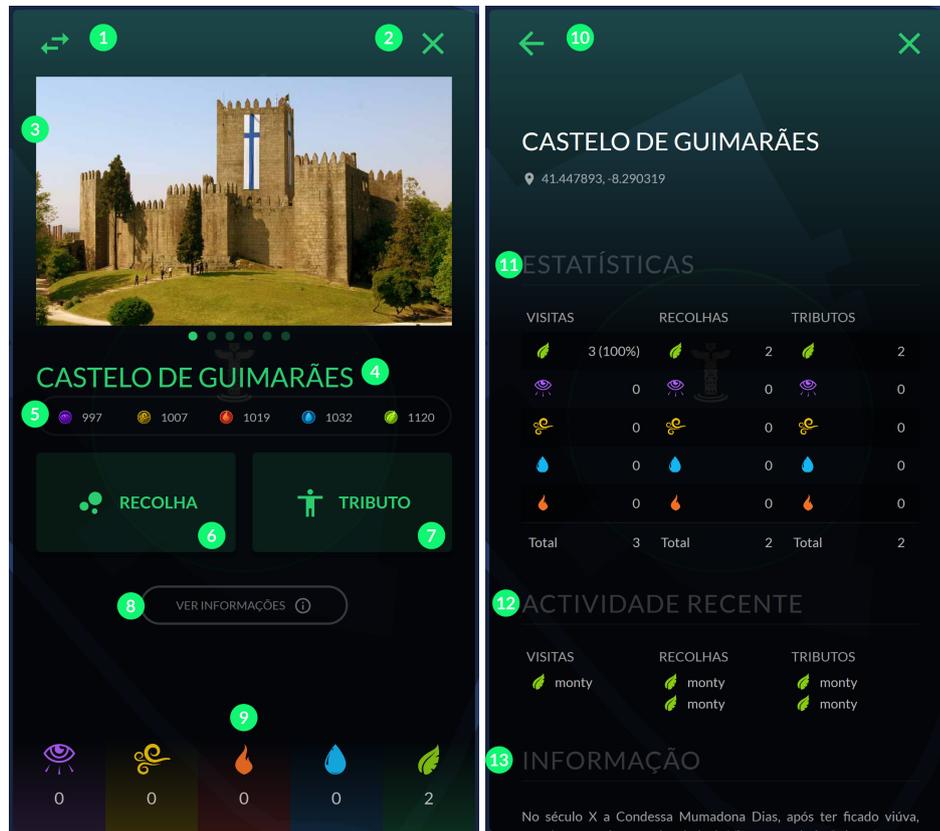


Figura 4.23: Aplicação: Visitar um templo

Na figura 4.23 destacam-se dez (13) componentes importantes da página do templo:

1. Botão para fazer conversão, abre pequena janela onde utilizador pode escolher a quantidade de essência que quer trocar por pontos;
2. Botão para fechar janela do tempo e regressar à página inicial;
3. Imagem ou imagens do templo;
4. Nome do templo;
5. Inventário de essências atual do utilizador;
6. Botão para fazer uma recolha;
7. Botão para fazer um tributo;

8. Botão para ver página de informações do templo (imagem da direita na figura 4.23);
9. Contagem dos tributos feitos por cada tribo neste templo;
10. Botão para regressar á página principal do templo;
11. Informações estatísticas de utilização do templo: visitas, recolhas e tributos;
12. Registo de atividade recente: utilizadores que visitaram, fizeram recolhas ou tributos;
13. Descrição do templo.

Ao carregar no botão de fazer uma recolha é apresentada uma pequena janela para informar o utilizador sobre quantas essências resultaram da ação, como se pode ver na imagem da esquerda da figura 4.24, e despoleta uma contagem de três minutos ao fim dos quais o utilizador poderá fazer uma nova recolha, como se pode ver na imagem do meio da figura 4.24. De carregar no botão de tributo resultam vários resultados visuais. O primeiro está representado na imagem da direita da figura 4.24, onde é apresentado o símbolo da tribo do utilizador em grande no ecrã. O segundo depende da contagem de tributos para cada tribo nesse templo, caso após a ação a tribo do utilizador fique com mais tributos que todas as outras a aparência geral da janela do templo é alterada para condizer com as cores da tribo. Por último, a contagem dos tributos realizados no templo é atualizada (em baixo).



Figura 4.24: Aplicação: ações nos templos

A aparência geral da página do templo muda consoante o número de tributos das tribos nesse templo, tal como é demonstrado na imagem 4.25.



Figura 4.25: Aplicação: aparência do templo consoante os tributos feitos

### Chat

Como foi referido na secção do servidor, o utilizador tem à sua disposição um *chat* para falar com os outros utilizadores. Este *chat* tem dois canais distintos: um onde fala para toda a população do jogo e outro onde fala apenas para os elementos da sua tribo.

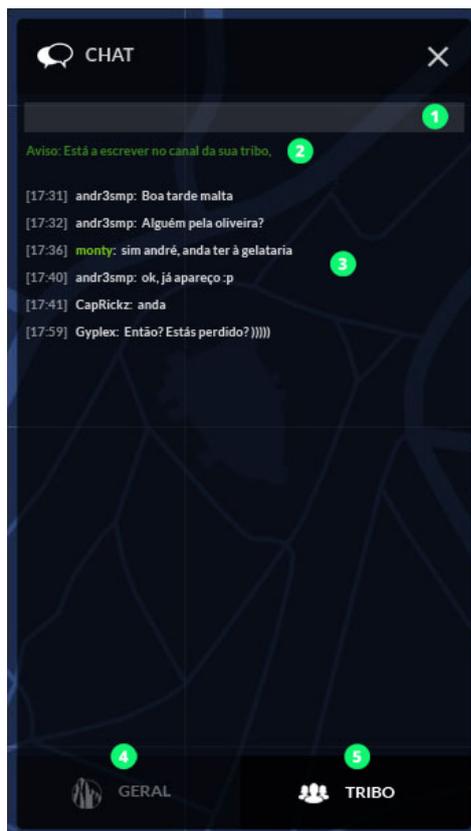


Figura 4.26: Aplicação: Chat da aplicação

O *chat* tem, também, um *design* simples e minimalista, para a melhor leitura possível das mensagens, onde se destacam apenas cinco (5) componentes:

1. Caixa de texto onde utilizador escreve as suas mensagens, botão de enviar aparece quando selecionado (ver figura 4.27);
2. Aviso para que utilizador saiba em que canal está a falar;
3. Corpo do *chat*;
4. Botão para conversar no canal global/geral;
5. Botão para conversar no canal da sua tribo.

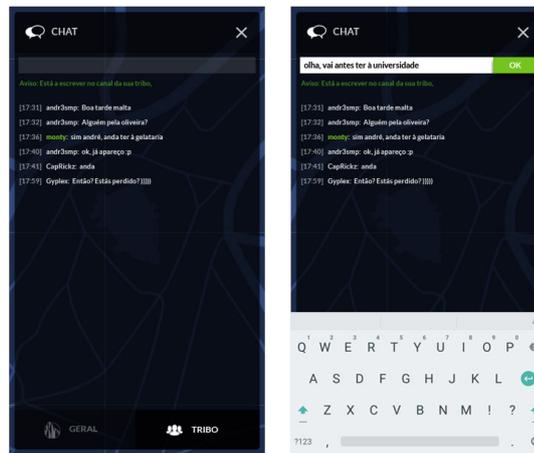


Figura 4.27: Aplicação: Chat da aplicação, 2

Na figura 4.27 pode ver-se um exemplo da aparência do *chat* com o teclado ativo/aberto num dispositivo Android.

### Desafios

A página dos desafios contém duas componentes da aplicação: as missões diárias e as conquistas. É nesta página que o utilizador vai poder consultar as tarefas que tem que desempenhar para ganhar as recompensas respetivas e também acompanhar o seu progresso nas mesmas. Em termos de *design*, esta página contém dois *tabs* onde o utilizador alterna entre a visualização da lista de missões diárias ou da lista de conquistas.



Figura 4.28: Aplicação: Missões Diárias

Como se pode ver na figura 4.28, a lista tem, também ela, um *design* minimalista, contendo apenas a informação necessária para que o utilizador saiba o que tem de fazer, o seu progresso em cada missão e as recompensas que o esperam. Os componentes da *tab* das missões diárias são:

1. Título da *tab*;
2. Nome da missão;
3. Recompensa da missão;
4. Progresso na missão;
5. Botão para ir para *tab* das missões diárias;
6. Botão para ir para *tab* das conquistas.

Dentro da mesma página, a *tab* das conquistas segue uma arquitetura semelhante, como se pode ver na figura 4.29.



Figura 4.29: Aplicação: Conquistas

## Classificações

Como foi falado no capítulo de apresentação da solução encontrada, o utilizador vai ser classificado, com os demais, pelos pontos e ações realizadas no jogo. Como também já foi referido, há três classificações distintas: duas individuais (Geral e Semanal) e uma coletiva (Tribos).



Figura 4.30: Aplicação: Classificações

Na figura 4.30 pode ver-se que a página segue a mesma filosofia das restantes, com três (3) *tabs* distintas para separar as três tabelas de classificação. As duas tabelas de classificação individuais têm *designs* idênticos onde mostram algumas informações sobre cada classificado como: A fotografia, a tribo a que pertencem, nome de utilizador, pontos atuais e o escalão que ocupam na classificação. O ranking coletivo e mensal, das tribos, tem um design semelhante mas mostra apenas o *ícone* da tribo, o seu nome e o total de pontos reunidos pelos seus utilizador.

### Loja

A loja apresenta, também ela, um *design* minimalista seguindo a filosofia do resto da aplicação. É uma página de dois passos, isto é, há uma primeira página que lista os itens da loja com *links* que levam a uma segunda página com as informações, em detalhe, do produto selecionado.

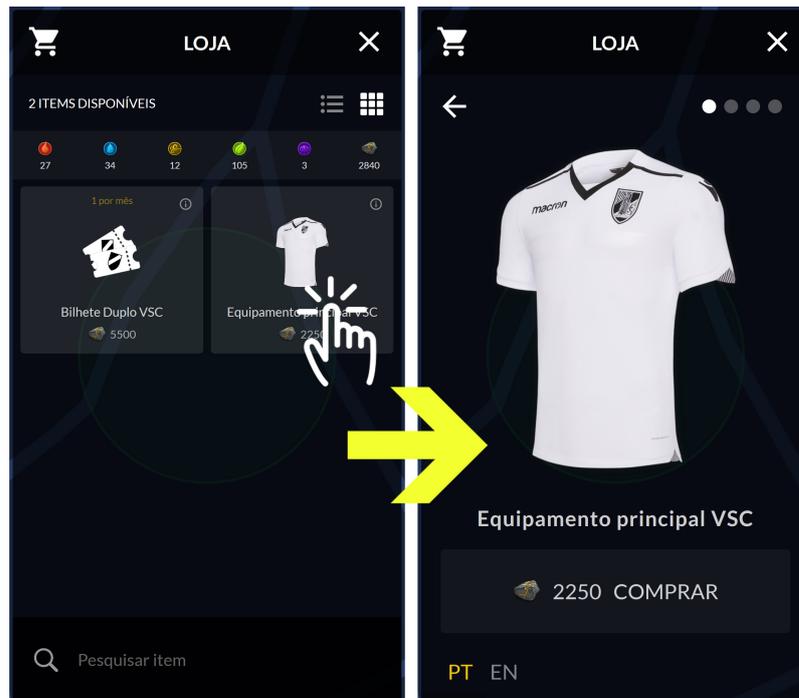


Figura 4.31: Aplicação: Loja

Como se pode verificar na figura 4.31, a página de listagem dos produtos permite ter uma breve introdução a cada produto como o seu nome, uma imagem, o seu custo e limite por utilizador.

Em termos de funcionalidades é possível ver os produtos em lista ou em grelha e pesquisar um produto pelo seu nome.

Carregando num dos produtos o utilizador é, então, redirecionado para a página desse

produto onde poderá comprá-lo, caso tenha a quantidade necessária de pontos para o fazer.

### Perfil do Utilizador

A página de perfil do utilizador servirá para este poder verificar o seu inventário, estatísticas de utilização, registo de atividade e lista de amigos. Na figura 4.32 podem ver-se as quatro *tabs* que o utilizador vê quando escolhe visitar o seu próprio perfil; a *tab* do inventário não é visível caso esteja a visitar o perfil de outro utilizador, assim como as opções disponíveis no cabeçalho de navegação são diferentes.

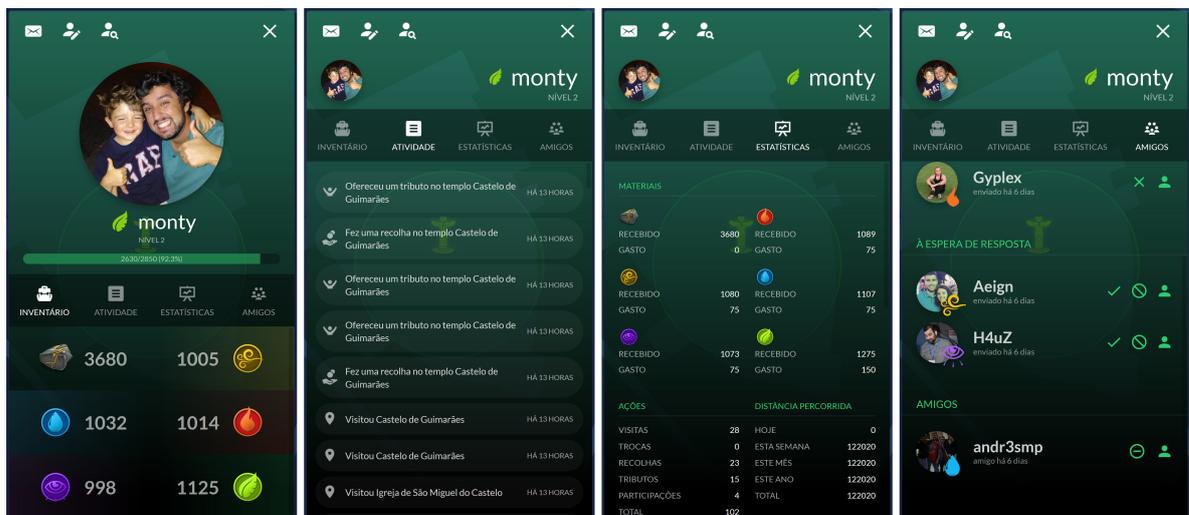


Figura 4.32: Aplicação: Perfil do Utilizador

Na figura 4.32, da esquerda para a direita, estão as representadas as *tabs*: inventário (*tab* principal), registo de atividade, estatísticas de utilização e gestão de pedidos de amizade. Como foi referido anteriormente, ao visitar o perfil de outro utilizador o conteúdo disponível é diferente. A grande diferença está nas opções do cabeçalho de navegação, sendo que as únicas que se mantêm são a de fechar a janela e a de procurar utilizador.

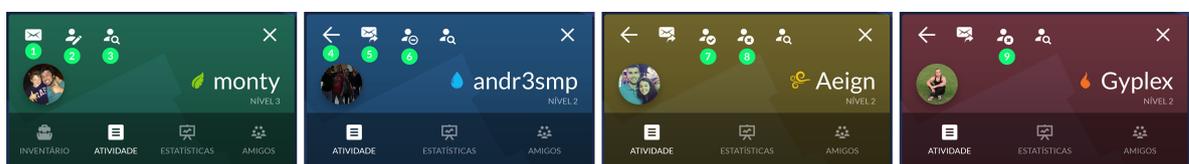


Figura 4.33: Aplicação: Perfil do Utilizador

Na figura 4.33 estão, da esquerda para a direita, representados os cabeçalhos de navegação de quatro perfis distintos: do próprio utilizador, de um utilizador de quem é amigo, de um utilizador que lhe enviou um pedido de amizade e, por último, de um uti-

lizador a quem enviou um pedido de amizade. Nesta mesma figura estão enumeradas as 9 opções diferentes que se identificam nos quatro perfis nela representada:

1. Abrir caixa de mensagens interna (só disponível no próprio perfil);
2. Editar dados pessoais (só disponível no próprio perfil);
3. Procurar utilizador;
4. Retroceder (só disponível ao navegar para outro perfil);
5. Enviar mensagem interna (só disponível ao visitar perfis de outros utilizadores);
6. Remover amigo (quando se visita um perfil de um amigo);
7. Aceitar pedido de amizade (quando se visita o perfil de um utilizador que nos enviou um pedido de amizade e este está pendente);
8. Rejeitar pedido de amizade (análogo ao anterior);
9. Cancelar pedido de amizade (quando se visita o perfil de um utilizador ao qual se enviou um pedido de amizade que ainda está pendente).

O perfil e o *chat* da aplicação são os dois elementos da aplicação que ajudam a promover a componente social da mesma, permitindo a criação de uma rede de amigos com quem se pode comunicar e combinar estratégias para o jogo em equipa e permitindo, também, estabelecer novos contactos.

### **Participações**

Esta página teve um tratamento diferente das outras e, como tal, é falada em último lugar porque tem um carácter mais sério. A página das participações é onde utilizador começa por ver a lista e o estado de participações realizadas por ele e pode optar por ver os detalhes de uma delas ou submeter uma nova.

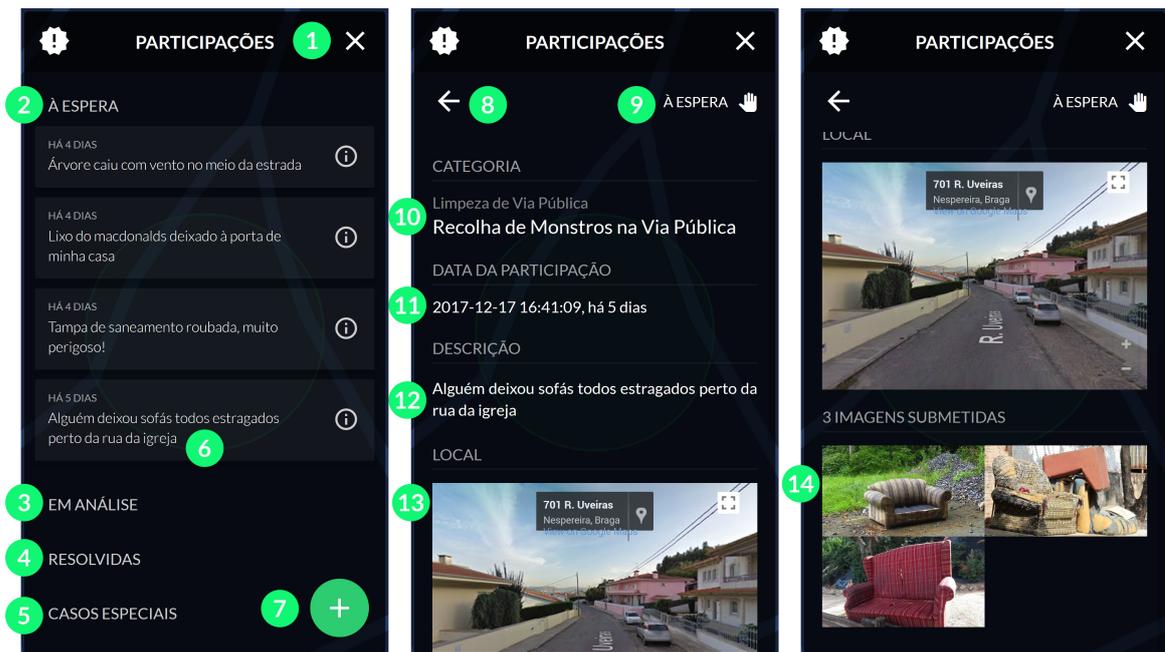


Figura 4.34: Aplicação: Lista de Participações e Detalhes

Na figura 4.34 pode ver-se a listagem de participações submetidas pelo utilizador, separadas em tabelas que traduzem o estado das participações que nelas estão listadas. Pode também ver-se, na mesma figura, um exemplo da página de informação da participação que o utilizador poderá ver ao carregar num dos itens da lista. Nesta figura (4.34), estão evidenciados catorze componentes enumerados, cujas funções são:

1. Botão para regressar à página inicial;
2. Lista das participações à espera de serem revistas;
3. Lista das participações que estão em análise;
4. Lista das participações que estão resolvidas;
5. Lista das participações que foram catalogadas como casos especiais;
6. Breve informação sobre participação e ao mesmo tempo um botão que direciona o utilizador para a página de detalhes da mesma;
7. Botão para abrir formulário de preenchimento de nova participação;
8. Botão para retroceder para lista de participações;
9. Estado da participação;
10. Categoria e sub-categoria da participação;

11. Data da participação;
12. Descrição da participação;
13. *Google Street View* do local da participação (interativo);
14. Fotografias submetidas na participação, se não tiver fotografias não aparece a secção.

Se o utilizador carregar no botão para submeter uma nova participação, representado pelo número 7 na figura 4.34, é-lhe apresentado um formulário que terá de preencher com um comentário, localização (que pode ser selecionada manualmente ou automaticamente através da localização GPS do dispositivo que o utilizador estiver a usar para submeter a participação), escolher uma categoria onde a participação se insira e poderá, caso considere oportuno, submeter imagens para oferecer credibilidade à sua participação.

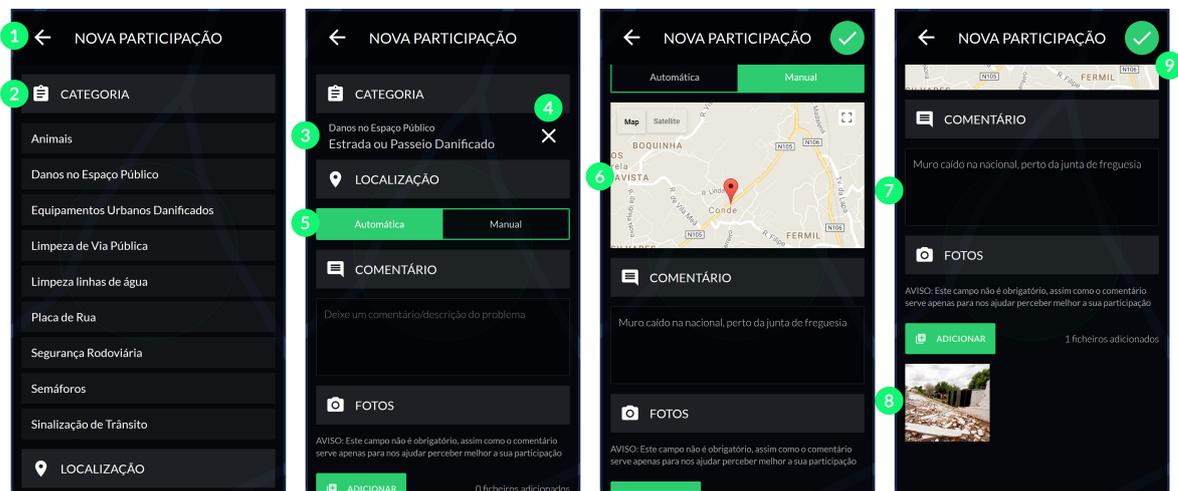


Figura 4.35: Aplicação: Processo de nova participação

Este processo está ilustrado na figura 4.35, onde estão representados os 9 componentes que fazem parte do formulário que deve ser preenchido caso se queira submeter uma nova participação, estes componentes são:

1. Regressar à lista de participações;
2. Escolha da categoria;
3. Informação depois de escolhida a categoria;
4. Cancelar escolha atual de categoria, volta a apresentar como anteriormente;
5. Escolha da localização (automática ou manual);
6. Google maps para escolha manual do local da participação;

7. Comentário/descrição da participação;
8. Fotos adicionadas;
9. Submeter participação, este botão só aparece quando a categoria, a localização e o comentário estão devidamente preenchidos.

Ao submeter a participação, o utilizador recebe uma mensagem que lhe indica se esta foi ou não submetida com sucesso. Caso tudo tenha corrido bem ele é redirecionado para a sua listagem, agora atualizada, de participações, caso tenha ocorrido um erro no servidor ao receber a participação (por alguma razão), o utilizador é informado de que houve um problema no processo e é aconselhado a submeter a participação mais tarde.



---

## ANÁLISE DO TRABALHO REALIZADO E CONCLUSÕES FINAIS

---

Neste último capítulo da dissertação é feita uma avaliação ao trabalho realizado ao longo deste ano, atendendo aos objetivos propostos no capítulo introdutório. Esta avaliação leva em consideração o produto final obtido, o seu design e complexidade de utilização, funcionalidades extra, interesse e utilidade.

É ainda realizado um planeamento futuro tendo em consideração as áreas para onde o sistema se poderá entender e o quanto poderá ser melhorado.

### 5.1 SÍNTESE DO TRABALHO

Terminado o projeto, o produto final tem todas as suas vertentes a funcionar, desde os eventos e missões até à narrativa estruturada que pretende estimular os utilizadores a usar a aplicação e a perceber que o contributo para o coletivo compreende mais vantagens e, como tal, vai compensar muito mais quando comparado com o trabalho que é desenvolvido apenas em prol do indivíduo.

Os componentes aplicação, área de administração e servidor *web* que suporta e alimenta ambas, estão implementados de forma suficientemente modular para permitir aplicar o sistema a qualquer cidade do mundo, carecendo sempre do preenchimento de informação inicial sobre pontos de interesse e freguesias ou outro tipo de categorização que se queira utilizar para distinguir as zonas da cidade.

Relativamente ao *design*, a aplicação segue uma filosofia minimalista que traz vantagens em termos de leitura e redução na complexidade de utilização. As diversas páginas seguem o mesmo *layout* com um aspeto generalista quanto à colocação dos componentes como botões, imagens, símbolos e ligações. Na generalidade, a aplicação segue um tema mais escuro que ajuda a realçar os componentes mais importantes de cada página bem como ajuda, mesmo que minimamente, a resguardar a visão dos utilizadores.

A escolha da freguesia e da tribo, seguindo uma filosofia, aliada à funcionalidade das participações dos utilizadores permitiu cumprir todos os objetivos que foram propostos na introdução do projeto, na medida em que possibilitam à entidade responsável pelo sistema retirar informações sobre as tendências da população por freguesia.

Falando em interesse e utilidade, pensa-se ter alcançado uma solução que tem como objetivo incentivar os cidadãos a sair de suas casas e a conhecer melhor a sua cidade, ao mesmo tempo construindo uma ponte entre o processo fastidioso de participar ocorrências e a diversão com um objetivo em mente: as recompensas.

## 5.2 IDEIAS PARA TRABALHO FUTURO

Tecidas as considerações finais sobre o projeto resta evidenciar ideias para o possível trabalho futuro. Embora se tenha alcançado uma solução completamente funcional e, na opinião do autor, interessante, há muito que pode ser melhorado não só no ponto de vista de *design* mas também em termos funcionais, do que está implementado e do que ainda pode ser implementado.

Com uma revisão da narrativa e da arte da aplicação por parte de uma equipa especializada, com tempo, ponderação e sempre com o melhor interesse do utilizador e da cidade em mente, a aplicação poderia ficar ainda mais apelativa aos olhos dos utilizadores e, também, da própria entidade administradora. Em termos funcionais seria muito interessante, com um uso mais avançado do sistema, implementar um sistema de sugestão de visita de pontos de interesse baseado na proximidade e tipos de sítios visitados por cada utilizador. Já na área de administração os alicerces estão construídos para um sistema capaz de gerar eventos autonomamente, baseado nos *heatmaps* de utilização das pessoas. A título de exemplo, a criação de eventos com recompensas mais apelativas em zonas de densidade populacional baixa (na aplicação), com vista a promover a homogeneidade na distribuição dos jogadores pelas freguesias da cidade em dias de eventos. Seria também interessante a construção de um algoritmo de sugestão para valores de itens da loja, isto é, quando a entidade administradora estivesse a acrescentar itens à loja ou a editar informações de um já existente, sabendo o valor real do item (em dinheiro) o sistema ser capaz de sugerir um preço (em pontos da aplicação) para esse item tendo em conta os valores que os utilizadores conseguem juntar, em média, durante uma semana ou um mês de utilização regular da aplicação.

---

**BIBLIOGRAFIA**

---

- [1] Fábio Silva, Cesar Analide, *"Gamification and the Improvement of Urban Sustainability"*, Ambient Intelligence and Smart Environments, Volume 21: Intelligent Environments, 2016
- [2] Cedrid Pimenta, Fábio Silva, Cesar Analide, *"Gamification in social environments to promote behaviours regarding energetic sustainability"*, 2013
- [3] Zachary Fitz-Wlater, Peta Wyeth, Dian Tjondronegoro, Bridie Scott-Parker, *"Driven to drive: Designing gamification for a learner logbook smartphone application"*, 2013
- [4] Heong Weng Mak, *"5 Gamified Environmental Apps for Sustainable Living"*, 2015
- [5] MyCity Guimarães, [http://www.cm-guimaraes.pt/frontoffice/pages/991?news\\_id=2534](http://www.cm-guimaraes.pt/frontoffice/pages/991?news_id=2534)
- [6] Orçamento Participativo, Câmara de Guimarães, <http://op.cm-guimaraes.pt/pagegen.aspx>
- [7] Gajendra Sharma, Purusottam Kharel, *"E-Participation Concept and Web 2.0 in E-government"*, General Scientific Researches, Vol(3), No (1), 2015
- [8] Wikipedia, *"E-Participation"*, <https://en.wikipedia.org/wiki/E-participation>, 2016
- [9] Portal do cidadão, *"E-Government - Portugal novamente em destaque na prestação de serviços públicos online"*, <https://www.portaldocidadao.pt/web/entidade-gestora/portugal-novamente-em-destaque-na-prestacao-de-servicos-publicos-online>, 2015
- [10] Whitehouse.gov, *"Digital Government: Building a 21st century platform to better serve the american people"*, <https://www.whitehouse.gov/sites/default/files/omb/egov/digital-government/digital-government.html>

- [11] RTP Notícias: *"Lisboa quer criar projecto digital para aproximar cidadãos"*, [http://www.rtp.pt/noticias/pais/lisboa-quer-criar-projecto-digital-para-aproximar-cidadaos\\_n4159](http://www.rtp.pt/noticias/pais/lisboa-quer-criar-projecto-digital-para-aproximar-cidadaos_n4159).
- [12] Geocaching, <http://www.geocaching.pt/>.
- [13] Streetspotr, <https://streetspotr.com/>
- [14] Surveys on the go, <http://www.surveyonthego.com/>
- [15] Survey, *"Retail intelligence & merchandising services"*, <https://survey.com/>
- [16] FieldAgent, *"Mobile audits & research"*, <http://www.fieldagent.net/>
- [17] TaskSpotting, *"Create, manage & track brand advocacy program"*, <https://taskspotting.com/business/>
- [18] iPoll, *"Rewards. Anywhere, Anytime."*, <https://www.ipoll.com/en>
- [19] Grupo Marktest, *"Estudo Barómetro de Telecomunicações da Marktest"*, <http://www.marktest.com/wap/a/n/id~1f2a.aspx>, 2015
- [20] José Pinheiro Neves, Joel Felizes, *"E-participation in Portuguese Local Governments: a Sociological Approach"*, 7<sup>a</sup>CAPSI – 7<sup>a</sup> Conferência da Associação Portuguesa de Sistemas de Informação, 2007
- [21] Rosabeth Moss Kanter, *"Ten Reasons People Resist Change"*, Harvard Business Review, <https://hbr.org/2012/09/ten-reasons-people-resist-chang>, 2012
- [22] Ingress, *"The world around you is not what it seems."*, <https://www.ingress.com/>
- [23] Statista.com, *"Distribution of mobile gamers worldwide as of 2nd quarter 2014, by gender"*, <https://www.statista.com/statistics/272327/mobile-social-and-traditional-gaming-users-by-gender/>
- [24] BigFishGames, *"2017 Video Game Trends and Statistics – Who's Playing What and Why?"*, <https://www.bigfishgames.com/blog/2017-video-game-trends-and-statistics-whos-playing-what-and-why/>
- [25] NewZoo, *"Male and Female Gamers: How Their Similarities and Differences Shape the Games Market"*, <https://newzoo.com/insights/articles/male-and-female-gamers-how-their-similarities-and-differences-shape-the-games-market/>
- [26] Município de Guimarães, *"Freguesias de Guimarães"*, <http://www.cm-guimaraes.pt/pages/844>

- [27] Jesús Eduardo Coronado Escobar, Angela Rocio Vasquez Urriago, "Gamification: an effective mechanism to promote civic engagement and generate trust?", ICEGOV '14 Proceedings of the 8th International Conference on Theory and Practice of Electronic Governance, Pages 514-515, 2014
- [28] Sebastian Deterding, Miguel Sicart, Lennart Nacke, Kenton O'Hara, Dan Dixon "Gamification. using game-design elements in non-gaming contexts", CHI EA '11 CHI '11 Extended Abstracts on Human Factors in Computing Systems, Pages 2425-2428, 2011
- [29] Sarah-Kristin Thiel "Investigating the Influence of Game Elements in Civic Engagement", CHI PLAY '15 Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play, Pages 415-418, 2015
- [30] Apache Cordova, "Mobile apps with HTML, CSS & JS. Target multiple platforms with one code base. Free and open source.", <https://cordova.apache.org/>
- [31] Appcelerator, "Build great mobile experiences faster. Native apps. Mobile APIs. Real-time analytics. One Platform.", <http://www.appcelerator.com/>
- [32] Xamarin, "Everything you need to deliver great mobile apps.", <https://www.xamarin.com/>
- [33] Monocross, "An Open Source Cross-Platform Mobile Framework.", <http://monocross.net/>
- [34] Framework 7, "Full Featured HTML Framework for Building iOS & Android Apps.", <https://framework7.io/>
- [35] Slim Framework, "Slim is a PHP micro framework that helps you quickly write simple yet powerful web applications and APIs.", <https://www.slimframework.com/>
- [36] Express.js, "Fast, unopinionated, minimalist web framework for Node.js", <https://expressjs.com/>
- [37] Spark, "A micro framework for creating web applications in Kotlin and Java 8 with minimal effort", <http://sparkjava.com/>
- [38] Flask, "Web development, one drop at a time", <http://flask.pocoo.org/>
- [39] MySQL, <https://www.mysql.com/>
- [40] Oracle, "The World's #1 Database Is Now the World's First Self-Driving Database", <https://www.oracle.com/database/index.html>
- [41] IBM DB2, "Use your data on your own terms.", <https://www.ibm.com/analytics/us/en/db2/>

- [42] PostgreSQL, *"The world's most advanced open source database"*, <https://www.postgresql.org/>
- [43] Microsoft SQL Server, *"Made to help you make more."*, <https://www.microsoft.com/pt-pt/sql-server/sql-server-2016>
- [44] MongoDB, *"Move at the speed of your data."*, <https://www.mongodb.com/>
- [45] Apache CouchDB, *"Seamless multi-master sync, that scales from Big Data to Mobile, with an Intuitive HTTP/JSON API and designed for Reliability."*, <http://couchdb.apache.org/>
- [46] Apache HBase, *"Distributed, scalable, big data store."*, <https://hbase.apache.org/>
- [47] Oracle NoSQL, *"Horizontally Scaled, Key-Value Database for Web Services and Cloud"*, <https://www.oracle.com/database/nosql/index.html>
- [48] Apache Cassandra, *"Manage massive amounts of data, fast, without losing sleep."*, <http://cassandra.apache.org/>
- [49] Django, *"The web framework for perfectionists with deadlines."*, <https://www.djangoproject.com/>
- [50] Ruby on Rails, *"Imagine what you could build if you learned Ruby on Rails..."*, <http://rubyonrails.org/>
- [51] ASP.NET, *"Open source web framework for building modern web apps and services."*, <https://www.asp.net/>
- [52] Laravel, *"The PHP Framework For Web Artisans."*, <https://laravel.com/>
- [53] Spring, *"The right stack for the right job."*, <https://spring.io/>
- [54] Ionic, *"The top open source framework for building amazing mobile apps."*, <https://ionicframework.com/>
- [55] React Native, *"Learn once, write anywhere: Build mobile apps with React"*, <https://facebook.github.io/react-native/>
- [56] Kendo UI, *"Build Better Web Apps Faster."*, <https://www.telerik.com/kendo-ui>
- [57] Onsen UI, *"The most beautiful and efficient way to develop HTML5 hybrid and mobile web apps."*, <https://onsen.io/>
- [58] Sara de Freitas, Paul Maharg and Contributors, *"Digital Games and Learning"* page 152-171, 2011

- [59] Tailor Soper, GeekWire.com, "Study 1.2 billion people are playing games worldwide 700M of them are online", <https://www.geekwire.com/2013/gaming-report-12-billion-people-playing-games-worldwide/>
- [60] Javier Torrente, Pablo Moreno-Ger, Iván Martínez-Ortiz & Baltasar Fernandez-Manjon, "Integration and Deployment of Educational Games in e-Learning Environments The Learning Object Model Meets Educational Gaming", Journal of Educational Technology & Society Vol. 12, No. 4, New Directions in Advanced Learning Technologies, pages 5-13, 2009
- [61] Positive Health Wellness, "12 Surprising Health Benefits of Playing Video Games", <https://www.positivehealthwellness.com/fitness/12-surprising-health-benefits-playing-video-games/>
- [62] Acer for education, "The benefits of Gaming in Education: the Build A World case", [http://eu-acerforeducation.acer.com/education-trends/the-benefits-of-gaming-in-education-the-build-a-world-case/?gclid=EAIaIQobChMIiKbQk43B1wIVYhbTCh3P4A2REAAAYASAAEgKHQvD\\_BwE](http://eu-acerforeducation.acer.com/education-trends/the-benefits-of-gaming-in-education-the-build-a-world-case/?gclid=EAIaIQobChMIiKbQk43B1wIVYhbTCh3P4A2REAAAYASAAEgKHQvD_BwE)
- [63] Dr. Mark Griffiths, theconversation.com, "Playing video games is good for your braind - heres how", <http://theconversation.com/playing-video-games-is-good-for-your-brain-heres-how-34034>
- [64] Dr. Ananya Mandal, MD, "Dopamine Functions", <https://www.news-medical.net/health/Dopamine-Functions.aspx>
- [65] Information Resources Management Association, "Gamification: Concepts, Methodologies, Tools, and Applications", 4 Volumes, 2211 Pages, 2015
- [66] Bunchball, "The engagement and performance solution, powered by gamification", <http://www.bunchball.com/>
- [67] Badgeville, "The leader in enterprise gamification", <https://badgeville.com/>
- [68] Costa A., Yelshyna A., C. Moreira T., Andrade F., Julián V., Novais P., "A Legal Framework for Elderly Healthcare Platform: Privacy and Data Protection Overview, The Computer Law & Security Review", Elsevier, ISSN: 0267-3649 Vol. 33, Issue 5, pp 647-658, 2017
- [69] Novais P., Carneiro D., "The Role of Non-Intrusive Approaches in the Development of People-Aware Systems, PAI - Progress in Artificial Intelligence", Springer, ISSN 2192-6352, vol. 5, issue 3, pp 215-220, 2016
- [70] Preuveneers D., Novais P., "A survey of software engineering best practices for the development of smart applications in Ambient Intelligence", Journal of Ambient Intelligence and Smart Environments, IOS Press, ISSN: 1876-1364, Volume 4, Number 3 , pp 149-162, 2012

- [71] Carneiro D., Pimenta A., Gonçalves S., Neves J., Novais P., *"Monitoring and improving performance in human–computer interaction"*, *Concurrency and Computation: Practice and Experience*, ISSN: 1532-0634, Vol. 28, n<sup>o</sup>4, pp 1291-1309, 2016
- [72] Carneiro D., Novais P., *"Quantifying the effects of external factors on individual performance, Future Generation Computer Systems"*, *The International Journal of eScience*, Elsevier Science BV, ISSN: 0167-739X, Vol. 66, pp 171–186, 2017

