



Mário Pinto Amorim

Licenciado em Ciências da Engenharia Eletrotécnica e de Computadores

Aplicação Móvel para Circuitos Turísticos Baseada em Algoritmos de Otimização de Percursos

Dissertação para obtenção do Grau de Mestre em
Engenharia Eletrotécnica e de Computadores

Orientador: Pedro Miguel Ribeiro Pereira, Prof. Doutor, FCT-UNL

Coorientador: João Francisco Alves Martins, Prof. Doutor, FCT-UNL

Júri:

Presidente: Doutor Luís Augusto Bica Gomes de Oliveira

Vogais: Doutor Rui Manuel Leitão Santos Tavares
Doutor Pedro Miguel Ribeiro Pereira



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Janeiro de 2019

Aplicação Móvel para Circuitos Turísticos Baseada em Algoritmos de Otimização de Percursos

Copyright © Mário Pinto Amorim, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Aos meus Pais, Tios e
Avós
À Mariana

Agradecimentos

Quero agradecer ao meu orientador Prof. Doutor Pedro Miguel Ribeiro Pereira, pela disponibilidade, conhecimentos e ajuda dada ao longo da realização desta dissertação. Ao meu coorientador Prof. Doutor João Francisco Alves Martins pela simpatia, esclarecimento de dúvidas e disponibilidade.

Ao Departamento de Engenharia Eletrotécnica da Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, o meu sincero agradecimento aos docentes pela disponibilidade e conhecimentos partilhados ao longo de todo o curso.

A todos os parceiros que fizeram parte do projeto *Smart Heritage City*, à Fundação Santa Maria Real do Património Histórico, à AIDIMME, à Tecnalia, à Câmara Municipal de Ávila, à Nobatek e a CARTIF.

Aos meus colegas e amigos de laboratório pela simpatia, boa disposição e ajuda em diversas tarefas, em particular à Adriana Mar e ao Fábio Januário pela amizade, pela ajuda, partilha e discussão de ideias neste último ano. Aos meus colegas e amigos de curso o meu grande obrigado pela amizade, pelo trabalho, pelos ensinamentos, pela confiança e por todos os momentos proporcionados em noites de estudo, dias de trabalho ou momentos de diversão ao longo destes anos.

À família BEST Almada, em particular à Carolina Amorim, à Maria Inês Pires, a Inês Cruz e à Catarina Marques, por tudo o que me ensinaram e partilharam ao longo de 2 anos. Obrigada por toda a partilha, por cada gargalhada ao vosso lado. Foi sem dúvida muito bom conhecer-vos e crescer com vocês.

Aos amigos em particular ao Pedro Rocha, à Carolina Amorim, ao Tiago Ramalho, ao João Rodrigo, ao Artur Saramago e ao Ricardo Fernandes por continuarem sempre ao meu lado e pelo apoio constante nesta importante etapa.

A toda a família Nascimento, por todo o apoio que me deram nesta importante etapa da minha vida, obrigado por tudo.

Aos meus avós por todo o carinho e apoio dado e por todos os ensinamentos que me transitaram ao longo de toda a minha vida.

Aos meus tios por todo o apoio e confiança dada, por estarem sempre presentes nas etapas importantes da minha vida.

Aos meus pais um agradecimento muito especial. Pelos valores que sempre me transmitiram, pelo sentido de responsabilidade, pela liberdade que sempre me deram e por toda a confiança que sempre depositaram em mim para tomar as minhas decisões. Obrigada por todo o amor. Sem o vosso esforço apoio e carinho não estaria onde estou neste momento. Sem vocês nada disto teria sido possível.

À Mariana, aquela que atura todas as minhas preocupações, todos os maus feitios e que continua sempre ao meu lado. Por me motivar todos os dias a ser melhor e nunca me deixar desistir. Sem todo o teu apoio, carinho e amor nada disto tinha sido possível. Obrigado por tudo.

Resumo

O crescimento da atividade turística criou a necessidade de um maior investimento por parte das cidades na restauração de monumentos e na criação e melhoramento de infraestruturas de apoio ao turismo. Face a este panorama pretende-se com este trabalho desenvolver um algoritmo de otimização de percursos que permita otimizar o tempo de visita do utilizador tendo em conta as suas necessidades e informações em tempo real disponíveis acerca dos locais a visitar. Como caso de estudo deste algoritmo pretende-se desenvolver uma aplicação móvel para turistas que terá como zona piloto a cidade histórica de Ávila, em Espanha, e como principal objetivo melhorar a experiência do turista durante a visita à cidade de Ávila.

Neste contexto, foi desenvolvida, na plataforma *Outsystems* uma aplicação móvel SHCity-Turista, a qual apresenta 5 funções principais: uma lista com todos os monumentos/lugares disponíveis para visita com informação estática e em tempo real acerca dos mesmos; um conjunto de circuitos pré-definidos; a criação de circuitos personalizados ou otimizados; a localização de outros serviços como farmácias, hospitais e pontos de descanso e ainda vídeos acerca da história da cidade. Face às funcionalidades da plataforma proposta foi desenvolvido um algoritmo de otimização de percursos, o qual foi implementado na funcionalidade de criação de circuitos em que, o utilizador apenas escolhe quais os monumentos/lugares que pretende visitar e, através deste algoritmo de otimização é apresentado o circuito com a ordem dos monumentos/lugares seleccionados otimizada.

Dos testes realizados resultou a perceção de que a aplicação desenvolvida é intuitiva, completa em termos de funcionalidades, e útil no auxílio ao turista durante a visita à cidade de Ávila. O algoritmo desenvolvido apresentou em termos de resultados uma otimização na duração total do circuito para todos os testes realizados.

Palavras-Chave: Turismo, Aplicação móvel, *Outsystems*, Algoritmo de otimização, Informação em tempo real.

Abstract

The growth of tourist activity has created the need for a bigger investment by the cities in the restoration of monuments and in the creation and improvement of tourism support infrastructures. In view of this situation, this work intends to develop an optimization algorithm that optimizes the user's visiting time taking into account their needs and real-time information about the places available to visit. As a case study of this algorithm we intend to develop a mobile application for tourists that will have as a pilot city, the historical city of Avila in Spain and its main objective is to improve the tourist experience during the visit to Avila.

In this context, SHCity-Tourist mobile application was developed using the Outsystems platform, it has 5 main functionalities, a list with all the monuments/places available to visit with static and real time information about them, a set of predefined circuits, the possibility to create personalized or optimized circuits, the location of other services such as pharmacies, hospitals and resting places and videos about the city and its history. Given the features of the proposed platform, a route optimization algorithm was developed, which was implemented in the create circuit functionality in which the user only chooses which monuments/places he wants to visit and through this optimization algorithm the circuit with the optimized order of the selected monuments/places is presented.

From the tests carried out, it was concluded that the application developed is intuitive, complete in terms of functionalities, and useful in assisting the tourist during the visit to the city of Ávila. The developed algorithm presented in terms of results an optimization in the total duration of the circuit for all the performed tests.

Key-Words: Tourism, Mobile application, Outsystems, Optimization algorithm, Real time information.

Índice de Matérias

1. Introdução	1
1.1 Motivação	1
1.2 Objetivos	2
1.3 Contribuição Original.....	3
2. Revisão Bibliográfica.....	5
2.1 Enquadramento	5
2.2 FindNatal	7
2.3 SMARTAPPCITY	8
2.4 Places and Trails	9
2.5 Edinburgh - World Heritage City.....	10
2.6 Norwich: Heritage City.....	11
2.7 Ahmedabad World Heritage City Guide	12
2.8 Comparação	13
2.9 Algoritmos de Otimização	14
3. Arquitetura e Implementação.....	19
3.1 Arquitetura da Aplicação	19
3.1.1 Arquitetura do Algoritmo de Otimização.....	21
3.2 Implementação dos Menus da Aplicação Móvel.....	22
3.3 Implementação do Algoritmo de Otimização	34
3.4 Síntese	39
4. Validação e Testes da Aplicação	41

4.1	Simulações em Ambiente de Simulação.....	41
4.2	Testes da Aplicação	46
4.3	Validação por Parte dos Utilizadores	50
4.4	Síntese	52
5.	Conclusões e Trabalho Futuro	53
5.1	Conclusões	53
5.2	Trabalho futuro	54
	Bibliografia	55
	Anexos	57

Índice de Figuras

Figura 1.1 - Evolução percentual das chegadas internacionais no Mundo, Europa e Espanha, entre (2006-2016). [2]	2
Figura 2.1- Categorias mais populares na <i>Apple App Store</i> em 2017. Adaptado de [5].	6
Figura 2.2 - As cinco etapas vividas pelo turista durante uma viagem. Adaptado de [6].	6
Figura 2.3- Aplicação <i>FindNatal</i> . [9]	7
Figura 2.4- Aplicação <i>Places and Trails</i> . [11]	9
Figura 2.5- Mapas Ilustrativos utilizados pela aplicação <i>Places and Trails</i> . [11]	9
Figura 2.6- Aplicação <i>Edinburgh - World Heritage City</i> . [13]	10
Figura 2.7- Aplicação <i>Norwich: Heritage City</i> . [14]	11
Figura 2.8- Aplicação <i>Ahmedabad Heritage City Guide</i> . [9]	12
Figura 2.9 - Categorias de Computação Evolucionária. Adaptado de [15].	14
Figura 2.10 - Exemplo de Codificação Binária. Adaptado de [21].	15
Figura 2.11 - Recombinação a partir de codificação binária. Adaptado de [23].	16
Figura 2.12 - Recombinação utilizando permutações.	16
Figura 2.13 – Exemplo de mutação em codificação binária. Adaptado de [23].	17
Figura 2.14 - Mutação por troca.	17
Figura 2.15 - Mutação por inserção.	17
Figura 2.16 - Mutação por mistura.	17
Figura 2.17 - Mutação por inversão.	18
Figura 3.1 - Fluxograma da Aplicação Móvel SHCity-Turista.	21
Figura 3.2 - Fluxograma do Algoritmo de Otimização.	22
Figura 3.3 - Ecrã inicial da aplicação móvel SHCity-Turista.	23
Figura 3.4 - Ecrã lista de monumentos/lugares.	24
Figura 3.5 - Ecrã mapa de monumentos/lugares.	25
Figura 3.6 - Ecrã monumento <i>Catedral Y Plaza De La Catedral</i>	26
Figura 3.7 - Ecrã localização local a) e direções até ao local b).	27
Figura 3.8 - Ecrã lista de circuitos a) e circuito A b).	28
Figura 3.9 - Ecrã mapa do circuito a) e trajeto para cada monumento/lugar b).	29

Figura 3.10- Ecrã criar circuito.	29
Figura 3.11- Ecrã criar circuito otimizado.	30
Figura 3.12 - Ecrã criar circuito otimizado com monumentos/lugares selecionados.	31
Figura 3.13 - Ecrã circuito otimizado obtido.	32
Figura 3.14 - Ecrã escolha monumento/lugar a incluir no circuito personalizado.	32
Figura 3.15 - Ecrã escolha do segundo monumento/lugar a incluir no circuito personalizado.	33
Figura 3.16 - Ecrã multimédia.	34
Figura 3.17 - Simulação da seleção dos monumentos e tempo de visita disponível.	35
Figura 3.18 - Simulação no caso de se exceder o tempo de visita disponível.	37
Figura 3.19 - Simulação da finalização da otimização.	38
Figura 3.20 - Implementação do algoritmo de otimização em <i>Outsystems</i>	39
Figura 4.1 - Tempos médios de espera para a <i>Catedral y Plaza de la Catedral</i>	41
Figura 4.2- Circuito não otimizado obtido.	43
Figura 4.3- Circuito otimizado com início às 10 horas obtido.	44
Figura 4.4- Circuito otimizado com início às 13 horas obtido.	44
Figura 4.5 - Comparação entre a duração dos circuitos não otimizados e otimizados.	45
Figura 4.6 - Circuito personalizado obtido na aplicação móvel.	47
Figura 4.7 - Seleção dos monumentos a incluir no circuito otimizado.	48
Figura 4.8 - Circuito otimizado obtido na aplicação móvel.	49
Figura 4.9 - Comparação entre a duração dos circuitos personalizados e otimizados obtidos na aplicação móvel.	50
Figura 4.10 - Formulário de validação por parte dos utilizadores.	50
Figura 4.11 - Resultados das questões do formulário.	51
Figura 4.12 - Resultado da questão relativa à usabilidade.	51

Índice de Tabelas

Tabela 2.1 - Comparação entre as diferentes aplicações móveis.	13
Tabela 2.2 - Propriedades dos principais algoritmos evolucionários. Adaptado de [20].	15
Tabela 4.1 - Tempos médios de espera dos monumentos 2, 8, 9 e 11 utilizados na simulação.	42
Tabela 4.2 - Tempos médios de visita dos monumentos 2, 8, 11 e 9.	42
Tabela 4.3 - Duração total obtida para o circuito não otimizado.	43
Tabela 4.4 - Duração total obtida para os circuitos otimizados.	45
Tabela 4.5 - Tempos de espera dos monumentos utilizados no algoritmo de otimização da aplicação móvel.	46
Tabela 4.6 - Durações obtidas do circuito personalizado da aplicação móvel.	47
Tabela 4.7 - Durações obtidas do circuito otimizado da aplicação móvel.	49

Introdução

Neste capítulo é apresentada a motivação, na qual será feito um enquadramento atual acerca da situação turística e a sua evolução. Serão, igualmente, referidos os principais objetivos desta dissertação bem como o contributo original, ou seja, o que este trabalho terá de original e diferente em relação a outros projetos já realizados e quais as publicações resultantes no âmbito desta dissertação.

1.1 Motivação

A atividade turística é um dos ramos que tem vindo a crescer mais rapidamente tanto a nível do número de turistas como a nível de infraestruturas turística. Este crescimento criou a necessidade de um maior investimento por parte das cidades na restauração de monumentos e na criação e melhoramento de infraestruturas de apoio ao turismo.

No caso específico da europa, a atividade turística é uma fonte de rendimento económico considerável que tem vindo a aumentar nos últimos anos em alguns países europeus. No caso de Portugal, registou-se em 2016 um aumento de 1,4% no PIB (Produto Interno Bruto) [1], representando assim uma das maiores fontes de emprego nas principais cidades europeias. Em Espanha entre 2014 e 2015 verificou se um aumento de cerca de 13% no número de empregos ligados ao turismo [2].

Pode-se verificar pela Figura 1.1 que a atividade turística a nível europeu e mundial mantem a partir de 2011 um crescimento constante, variando entre os 4% e os 5% anuais. Mais especificamente em Espanha este crescimento tem sido notável verificando-se um aumento considerável na percentagem de crescimento turístico ano após ano tendo este atingido o máximo de 11% no ano de 2016.

Com este aumento do número de turistas e infraestruturas surge a dificuldade por parte dos turistas em decidir quais os pontos turísticos a visitar e na recolha de informação relativamente a esses mesmos pontos devido à vasta variedade de fontes existentes. Por outro lado, a informação disponível acerca dos pontos turísticos não é uma informação em tempo real encontrando-se muitas vezes já desatualizada.

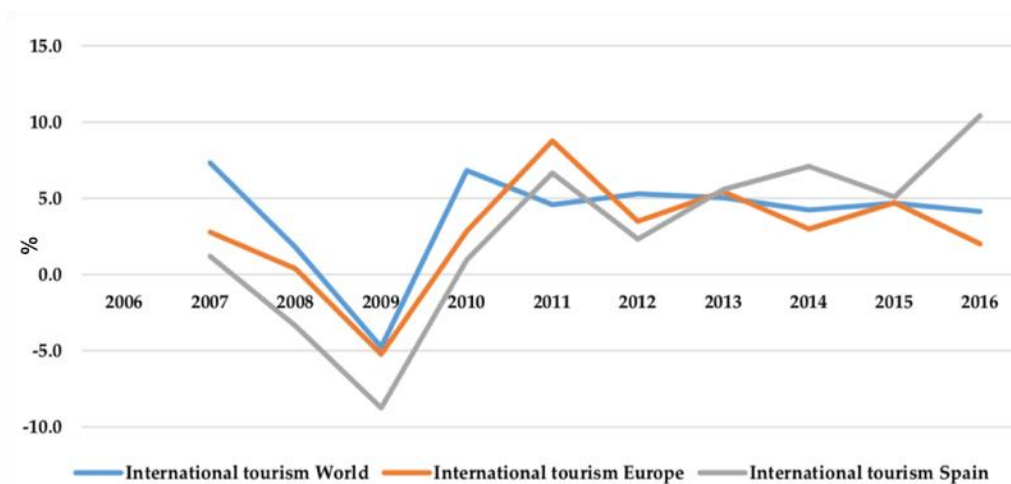


Figura 1.1 - Evolução percentual das chegadas internacionais no Mundo, Europa e Espanha, entre (2006-2016). [2]

Não só os turistas, mas também as entidades gestoras (ex. Municípios, entidades regionais) têm problemas a nível logístico no controlo das infraestruturas, número de visitantes diários, taxa de ocupação, tempos de espera, entre outros fatores, onde a informação em tempo real é atualmente inexistente. Outro dos problemas encontrados por parte dos Municípios é a partilha de informação com os turistas, como o elevado número de plataformas possíveis para partilhar informação sobre as atrações turísticas tornando-se cada vez mais difícil concentrar a informação num único local acessível a todos os turistas [3].

Face ao panorama descrito anteriormente pretende-se desenvolver um algoritmo de otimização de percursos que permita otimizar o tempo de visita do utilizador tendo em conta as suas necessidades e informações em tempo real acerca dos locais a visitar como tempo de espera, tempo de visita e a percentagem de ocupação dos mesmos.

Pretende-se assim como caso de estudo deste algoritmo desenvolver uma aplicação móvel para turistas por forma a melhorar a sua experiência. A aplicação a desenvolver terá como zona piloto a cidade histórica de Ávila, em Espanha, devido à sua geometria em muralha bem como a distribuição dos principais pontos históricos que favorece e facilita o desenvolvimento e teste desta aplicação.

1.2 Objetivos

Esta dissertação tem como objetivo principal o desenvolvimento de um algoritmo de otimização de percursos. O algoritmo que se pretende desenvolver no âmbito desta dissertação tem como principais características:

- Auxiliar e melhorar a experiência do turista;
- Otimizar o tempo de visita do turista;
- Utilizar informação em tempo real acerca da ocupação dos diferentes pontos turísticos;
- Ter em conta o tempo disponível por parte do turista para a visita.

Como caso de estudo será desenvolvida uma aplicação móvel inteligente inserida no projeto *Smart Heritage City* (SHCity) de apoio ao turista na cidade de Ávila, em Espanha.

Esta aplicação tem como principais objetivos:

- Criar uma lista com os monumentos disponíveis para visita;
- Partilhar informação em tempo real com os turistas acerca das condições do monumento tais como: ocupação, tempo de espera, tempo de visita, temperatura interior, entre outros;
- Disponibilizar direções desde a localização atual do utilizador até ao monumento que pretende visitar;
- Possibilitar a criação de circuitos pré-definidos com diferentes durações, distâncias e graus de dificuldade:
 - Criar circuitos com ordem e monumentos selecionados pelo utilizador;
 - Criar circuitos otimizados com passagem por monumentos selecionados pelo utilizador.

Entende-se que cumpridos os objetivos acima indicados, se obtém uma aplicação móvel exclusiva que vai proporcionar uma melhor experiência ao turista.

1.3 Contribuição Original

De acordo com o exposto nos subcapítulos anteriores, vê-se como contributo original desta dissertação para a temática estudada, a implementação de otimização de percursos, navegação em tempo real incorporada na aplicação bem como a comunicação com uma base de dados onde estará guardada toda a informação relativa aos pontos turísticos.

Para a otimização dos percursos foi desenvolvido um algoritmo evolutivo que permite a otimização do tempo total de visita do utilizador com base em informação em tempo real nomeadamente a percentagem de ocupação e o tempo de espera de cada monumento/lugar.

No âmbito da presente dissertação resultaram as seguintes publicações:

- Amorim M., Mar A., Monteiro F., Sylaiou S., Pereira P., Martins J. (2018) Smart Tourism Routes Based on Real Time Data and Evolutionary Algorithms. In: Ioannides M. et al. (eds) Digital Heritage. Progress in Cultural Heritage: Documentation, Preservation, and Protection. EuroMed 2018. Lecture Notes in Computer Science, vol 11196. Springer, Cham.
https://doi.org/10.1007/978-3-030-01762-0_36
- Monteiro F, Mar A, Amorim M, et al (2018) Smart Heritage Management - The SHCity Project Approach. In: Proc. of International Conference on Intelligent Systems (IS), pp. 677-684.
<https://doi.org/10.1109/IS.2018.8710497>

Revisão Bibliográfica

Neste capítulo é inicialmente feito um enquadramento geral da utilização de aplicações móveis na atividade turística. É também apresentado um levantamento das aplicações móveis existentes e as suas respetivas funcionalidades no melhoramento da experiência do turista, bem como uma comparação entre as mesmas e a aplicação a ser desenvolvida. Por fim serão descritos alguns métodos de algoritmos evolucionários e a escolha daquele que melhor se encaixa nas necessidades da aplicação móvel SHCity-Turista.

2.1 Enquadramento

O desenvolvimento e a utilização de aplicações móveis têm vindo a crescer ao longo dos últimos anos. Desde o aparecimento da primeira *App Store*, em julho de 2008, que as aplicações móveis são cada vez mais utilizadas nas diversas indústrias e atividades [4].

Com o crescimento da atividade turística aliada à evolução de aplicações móveis surgiu a necessidade de criar novas ferramentas tecnológicas de apoio ao turista. Estas aplicações estão atualmente no topo de aplicações mais populares na *App Store*, como representado na Figura 2.1.

De acordo com a Figura 2.1, verifica-se que cerca de 3.9% do total de aplicações ativas na *Apple App Store* são aplicações relacionadas com turismo, o que deixa o turismo como a sétima categoria mais popular.

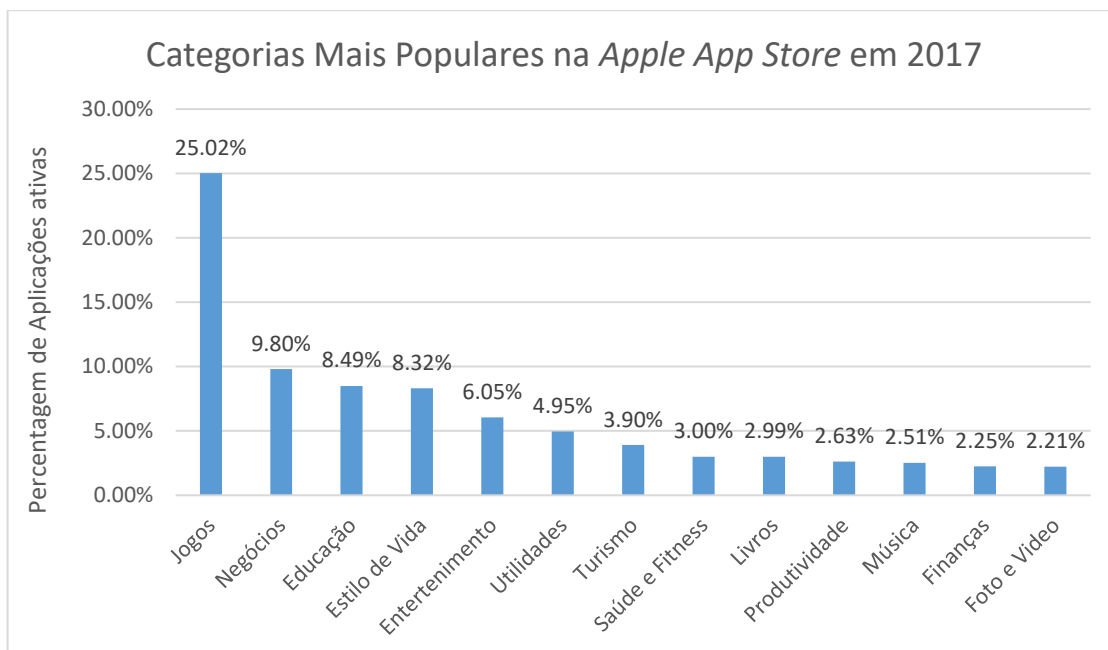


Figura 2.1- Categorias mais populares na *Apple App Store* em 2017. Adaptado de [5].

Estas aplicações podem ser divididas pelas cinco fases de viagem experienciadas pelo turista, representadas na Figura 2.2, que vão desde a escolha do destino até a partilha da experiência.

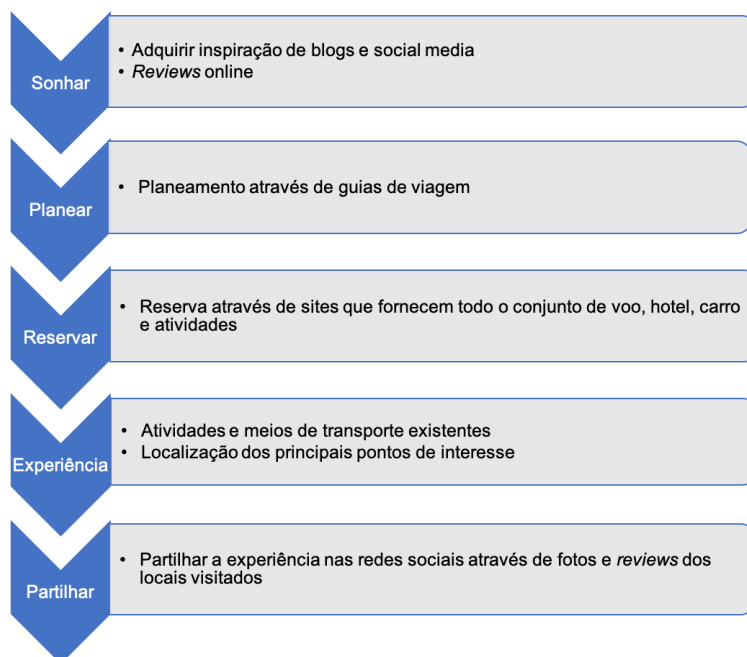


Figura 2.2 - As cinco etapas vividas pelo turista durante uma viagem. Adaptado de [6].

A fase de experiência, fundamental para o turista, visa fornecer através de aplicações móveis informação acerca da rede de transportes públicos, quais os principais pontos de interesse e informação estática acerca desses locais. Algumas destas aplicações possuem também direções GPS para orientar o turista na sua visita e utilizam a aplicação como plataforma de interação entre diferentes utilizadores.

Estas aplicações encontram-se implementadas em algumas cidades a nível mundial, cidades estas consideradas *Smart Cities*.

Smart City é um termo que tem vindo a expandir-se a nível mundial desde 1990 [7], e a grande questão é o que torna uma cidade inteligente. Mobilidade, ambiente, governo, economia, população e estilo de vida inteligente, são as 6 dimensões que, de acordo com [7], tornam uma cidade inteligente. A implementação deste tipo de aplicações móveis de auxílio à experiência do turista torna estas cidades mais inteligentes a nível de mobilidade, população e estilo de vida.

Nos seguintes subcapítulos serão analisadas algumas aplicações móveis implementadas em *Smart Cities*.

2.2 *FindNatal*

FindNatal é uma aplicação móvel disponível para IOS e Android criada em 2014 que visa enriquecer a experiência do turista ao visitar a cidade de Natal, Brasil. Foi criada para trabalhar de forma independente e offline apresentando, na sua versão atual as seguintes funcionalidades [8]:

- Inglês e Português como idiomas disponíveis;
- Exibe para cada atração fotos, descrição. Localização, email, página web e horários de abertura e fecho;
- Mostra a partir da localização atual do utilizador, qual o trajeto que deve ser seguido para chegar ao destino selecionado;
- Envia um alerta ao utilizador quando este estiver a 500m de um ponto turístico;
- Permite que o utilizador efetue uma avaliação tanto dos pontos de interesse visitados como da cidade. Esta avaliação pode ser feita a nível de limpeza pública, segurança, acessibilidade e serviços turísticos.

Esta aplicação é utilizada não só para auxiliar o turista na localização dos principais pontos turísticos, mas também mostra os principais bares, restaurantes e hotéis da cidade, como representado na Figura 2.3.

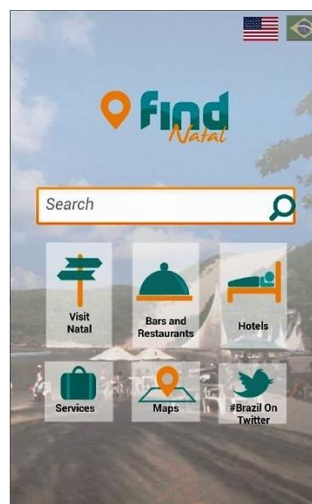


Figura 2.3- Aplicação *FindNatal*. [9]

Pontos positivos da aplicação:

- Alerta para quando o utilizador estiver a 500m de um ponto turístico;
- Direções GPS para os pontos turísticos;
- Avaliação dos serviços existentes na cidade;
- Idiomas Português e Inglês disponíveis.

Pontos negativos da aplicação:

- Aplicação estática, não recebe informação em tempo real;
- Número elevado de funcionalidades;
- Não possui navegação em tempo real;
- *Design* pouco atrativo;
- Inexistência de circuitos pré-definidos.

2.3 SMARTAPPCITY

Primeira aplicação móvel disponível para IOS e Android em Espanha que junta todos os serviços da cidade numa única plataforma. Através de uma parceria com os municípios ou entidades privadas, os dados são fornecidos pelas entidades e inseridos numa base de dados ficando posteriormente disponíveis para os utilizadores da aplicação. Possui uma grande quantidade de serviços como turismo, informações de trânsito, transportes públicos entre outros, [10].

O serviço dedicado ao turismo permite ao utilizador visualizar numa lista ou num mapa quais são as principais atrações turísticas. Todos essas atrações possuem informação como a descrição, fotos e direções.

Esta aplicação é adaptável a qualquer cidade e pode ser adquirida com uma versão mais reduzida que possui apenas alguns dos serviços disponíveis. Possui também informação dinâmica como meteorologia, trânsito e notícias [10].

Pontos positivos da aplicação:

- Facilmente adaptável a qualquer cidade;
- Recebe informação em tempo real;
- Descrição dos principais pontos turísticos.

Pontos negativos da aplicação:

- Número elevado de funcionalidades;
- Não possui navegação em tempo real;
- Inexistência de circuitos.

2.4 Places and Trails

Aplicação móvel para IOS e Android, desenvolvida para motivar e aumentar a exploração turística por todo o Reino Unido. Apresenta duas vertentes, uma designada de *Places*, onde os principais pontos históricos são apresentados em forma de lista, como representado na Figura 2.4, ou num mapa. Estes locais podem ser selecionados manualmente a partir da lista ou pode ser enviada uma notificação quando o utilizador está próximo de um destes pontos [11]. Cada ponto possui informação como imagens, descrição e ficheiros áudio e vídeo. Outra vertente disponível é a de *Trails*, onde, em forma de lista, são mostrados os trilhos disponíveis e os utilizadores podem adicionar os seus próprios trilhos, ficando estes disponíveis para outros utilizadores usufruírem.

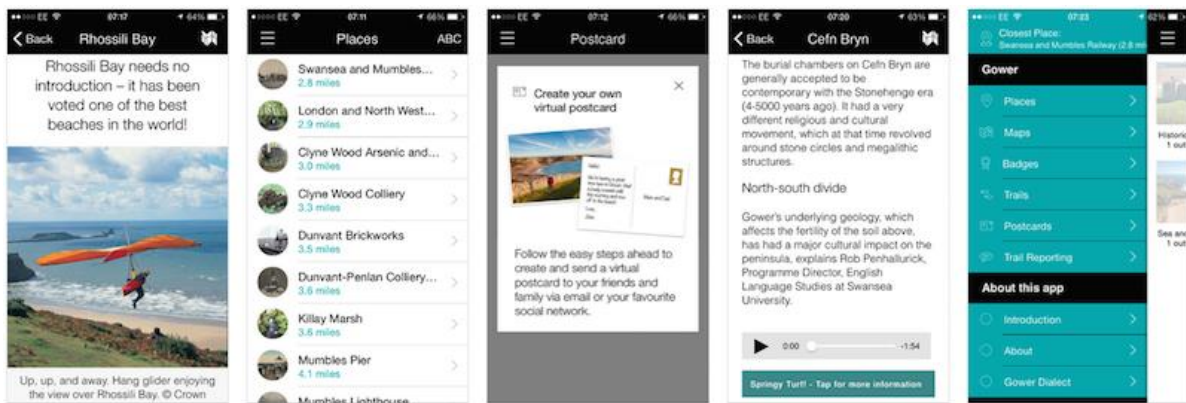


Figura 2.4- Aplicação *Places and Trails*. [11]

Os mapas utilizados quer na plataforma IOS quer na plataforma Android são da *Google Maps*. Quando offline a aplicação possui mapas ilustrativos como os da Figura 2.5 para guiar o turista pelos trilhos.

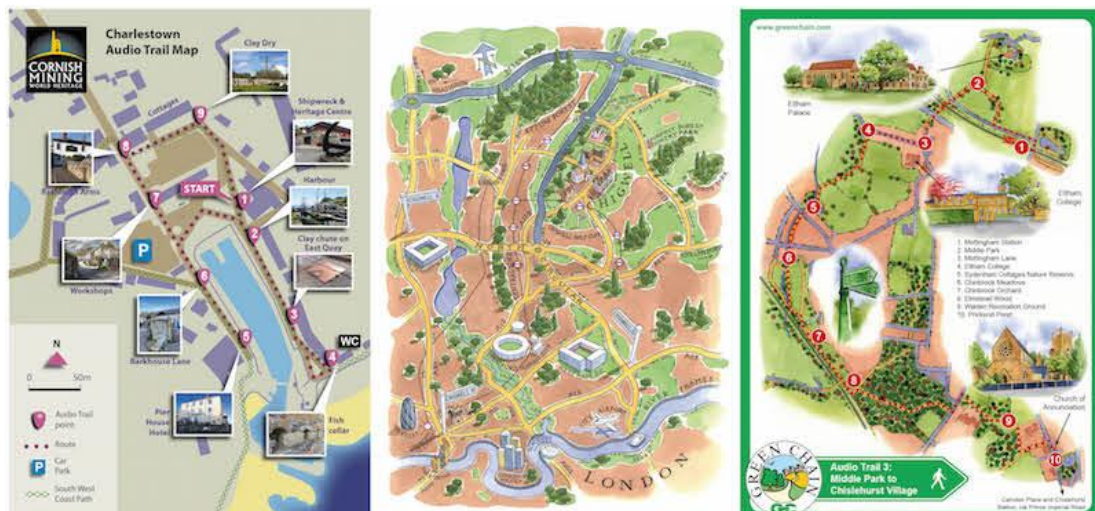


Figura 2.5- Mapas Ilustrativos utilizados pela aplicação *Places and Trails*. [11]

Pontos positivos da aplicação:

- Utilização de guias áudio;
- Descrição dos principais pontos turísticos;
- Utilização offline;
- Possibilidade de o utilizador acrescentar pontos de interesse.

Pontos negativos da aplicação:

- Não possui navegação em tempo real;
- Não recebe informação em tempo real;
- Inexistência de circuitos.

2.5 *Edinburgh - World Heritage City*

Aplicação móvel para IOS, possui quatro circuitos pré-definidos com guia áudio, onde o utilizador pode escolher o circuito a partir de uma lista que contem os quatro circuitos pré-definidos como representado na Figura 2.6. Cada um dos circuitos possui uma breve descrição e algumas fotos dos locais que o utilizador terá hipótese de visitar [12].

O utilizador pode ainda adicionar comentários e fotos dos locais visitados, tendo a possibilidade de enviar um postal do local que está a visitar para outros utilizadores, promovendo assim uma maior interação entre os utilizadores da aplicação.

Possui ainda um jogo interativo com um conjunto de objetivos, onde o utilizador que conseguir alcançar mais objetivos e visitar mais locais tem a maior pontuação, a qual fica guardada numa tabela juntamente com as pontuações dos restantes participantes [13].



Figura 2.6- Aplicação *Edinburgh - World Heritage City*. [13]

Pontos positivos da aplicação:

- Atribuição de postais dos locais visitados;
- Guias áudio;
- Jogo interativo com outros utilizadores.

Pontos negativos da aplicação:

- Possui apenas quatro circuitos pré-definidos;
- Não existe a possibilidade de apenas visitar um dos pontos turísticos, o utilizador é sempre obrigado a escolher um dos quatro circuitos;
- Não possui navegação em tempo real;
- Não possui informação em tempo real.

2.6 Norwich: *Heritage City*

Aplicação móvel para IOS e Android, desenvolvida para dar a conhecer a história dos edifícios da cidade de Norwich, Inglaterra. Pode-se observar pela análise da Figura 2.7 que possui uma lista e um mapa com os monumentos históricos disponíveis para visita, cada um com breve descrição e fotos, enviando uma notificação quando o utilizador se encontra próximo de um destes monumentos

Contem ainda uma descrição da cidade de Norwich com alguns factos históricos à cerca da cidade e uma caderneta onde o utilizador coloca os *badges* que vai ganhando à medida que visita os locais, motivando o utilizador a explorar e visitar a cidade [14].

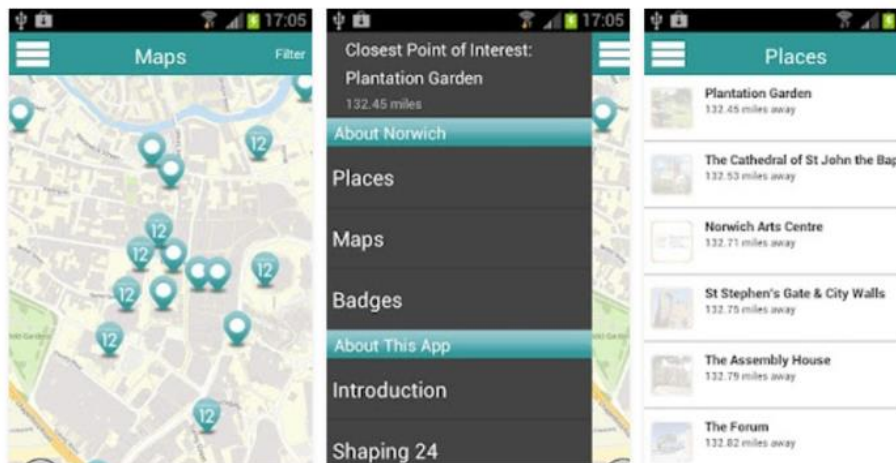


Figura 2.7- Aplicação Norwich: *Heritage City*. [14]

Pontos positivos da aplicação:

- Atribuição de *badges* por cada objetivo alcançado;
- Notificação quando o utilizador se encontra próximo de um monumento;
- Descrição da cidade de Norwich;
- Informação explicativa dos monumentos presentes na aplicação.

Pontos negativos da aplicação:

- Não possui navegação em tempo real;
- Não possui informação em tempo real;
- Impossibilidade de fazer um circuito pelos diversos monumentos.

2.7 Ahmedabad World Heritage City Guide

Aplicação para Android e IOS desenvolvida pela Câmara Municipal de Ahmedabad, Índia, criada com o objetivo de impulsionar os turistas e a população local a preservar os locais mais históricos e antigos da cidade [14].

Possui como principais funcionalidades:

- Divisão dos locais por categorias religiosa, institucional e política, alimentação e mercados como representado na Figura 2.8.
- Fotos e direções desde a localização do utilizador até ao local selecionado.
- Circuitos pré-definidos que incluem passagem pelos locais mais históricos da cidade.
- Lista com informação acerca das principais festividades que decorrem na cidade.



Figura 2.8- Aplicação Ahmedabad *Heritage City Guide*. [9]

Pontos positivos da aplicação:

- Divisão dos locais por categorias;
- Guias explicativos dos locais visitados;
- Inclusão da componente cultural como as principais feiras e festivais da cidade.

Pontos negativos da aplicação:

- Não possui navegação em tempo real;
- Não possui informação em tempo real;
- Possui apenas circuitos pré-definidos, não possibilitando ao utilizador a capacidade de fazer o seu próprio circuito.

2.8 Comparação

Neste capítulo procedeu-se ao levantamento bibliográfica do crescimento da utilização de aplicações móveis, mais especificamente na área de turismo, e da divisão dos diferentes tipos de aplicações turísticas existentes. Apresentou-se um enquadramento na utilização destas aplicações, ou seja, em que tipo de cidades ou locais é possível e faz sentido implementar este tipo de tecnologias. Fez-se um levantamento das principais aplicações móveis, para turistas que têm como objetivo auxiliar e fornecer uma melhor experiência durante a visita a um determinado local, bem como vantagens e desvantagens de cada uma destas aplicações. Por fim realizou-se uma comparação entre as aplicações móveis analisadas e a aplicação a desenvolver, SHCity-Turista, relativamente às funcionalidades de cada uma. Esta comparação é apresentada na Tabela 2.1.

Tabela 2.1 - Comparação entre as diferentes aplicações móveis.

	<i>FindNatal</i>	<i>SMARTAP PCITY</i>	<i>Places and Trails</i>	<i>Edinburgh -World Heritage City App</i>	<i>Norwich: Heritage City</i>	<i>Ahmedabad Heritage City Guide</i>	<i>SHCity-Turista</i>
Informação em tempo real	✗	✓	✗	✗	✗	✗	✓
Listagem dos pontos turísticos	✓	✓	✓	✗	✓	✓	✓
Circuitos personalizados	✗	✗	✗	✗	✗	✗	✓
Offline	✓	✗	✓	✗	✓	✓	✓
Avaliação dos monumentos	✓	✗	✗	✗	✗	✗	✗
Guia áudio	✗	✗	✓	✓	✗	✓	✗
Localização de Farmácias e Hospitais	✗	✓	✗	✗	✗	✗	✓
Circuitos pré-definidos	✗	✗	✓	✓	✗	✓	✓
Direções para os pontos turísticos	✓	✓	✗	✗	✓	✓	✓
Navegação em tempo real	✗	✗	✗	✗	✗	✗	✓
Informação à cerca de cada ponto turístico	✓	✓	✓	✗	✓	✓	✓
Notificação de proximidade de ponto turístico	✓	✗	✗	✗	✓	✗	✗

Ao analisar a Tabela 2.1 verifica-se que já existem algumas aplicações de apoio à experiência do turista, algumas com mais funcionalidades que outras. À exceção da *Edinburgh-World Heritage City App* todas as aplicações apresentam uma lista dos principais pontos turísticos, disponibilizando ao utilizador informação à cerca desses mesmos pontos. Destas aplicações referidas anteriormente apenas uma não disponibiliza direções desde a posição atual do utilizador até ao ponto selecionado.

De todas as aplicações analisadas apenas a SHCity-Turista, aplicação a ser desenvolvida, possui circuitos personalizados, ou seja, nenhuma das outras aplicações possibilita que o utilizador selecione monumentos para incluir num circuito. Apenas quatro possuem circuitos pré-definidos para o utilizador realizar. Quanto à informação em tempo real, apenas uma possui essa funcionalidade, disponibilizando para os seus utilizadores informação meteorológica e de trânsito em tempo real.

Quanto a aplicação SHCity-Turista, conclui-se que em comparação com as restantes é a aplicação mais completa em termos de funcionalidades. Destaca-se em relação às restantes aplicações pela utilização de informação em tempo real como ocupação, tempo de espera, temperatura e tempo de visita, informação que será fornecida à aplicação móvel por um conjunto de sensores existentes em todos os pontos turísticos da cidade. Outro aspeto único nesta aplicação será a existência de navegação em tempo real, ou seja, à medida que o utilizador se desloca a localização atual será atualizada, possibilitando assim que o utilizador siga a rota até ao monumento selecionado

2.9 Algoritmos de Otimização

Alguns algoritmos de otimização baseiam-se em técnicas de computação evolucionária. As principais técnicas de computação evolucionária encontram-se representadas na Figura 2.9.

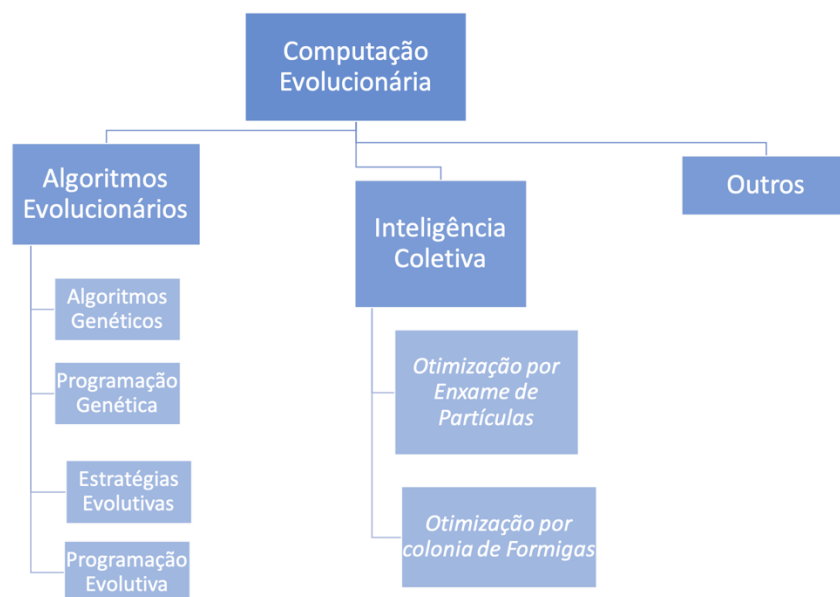


Figura 2.9 - Categorias de Computação Evolucionária. Adaptado de [15].

Pela análise da Figura 2.9 pode-se verificar que alguns autores dividem as principais técnicas de computação evolucionária em algoritmos evolucionários e inteligência coletiva [15] [16].

A área de inteligência coletiva simula o comportamento coletivo dos indivíduos de uma população e a forma de interagirem com o meio ambiente e entre si. Este subgrupo de computação evolucionária pode ser dividido em dois métodos, a otimização por enxame de partículas e a otimização por colónia de formiga. Estes métodos são de fácil implementação e têm a grande vantagem de permitir

o fácil ajuste de alguns parâmetros apesar de não possuírem operadores de recombinação e mutação [17].

Os algoritmos evolucionários são desenvolvidos com o objetivo de atingir soluções adequadas para problemas complexos [18]. Estes algoritmos têm uma grande vantagem em relação a outros métodos de resolução de problemas complexos que é o facto de necessitarem de muito pouca informação acerca do problema e de serem flexíveis o suficiente para serem aplicados numa vasta gama de problemas devido aos operadores evolutivos que apresenta tais como a recombinação e a mutação [19]. Podem ser divididos em algoritmos genéticos (AG), programação genética (PG), estratégias evolutivas (EE) e em programação evolutiva (PE).

As propriedades destes quatro algoritmos evolucionários encontram se numeradas na Tabela 2.2

Tabela 2.2 - Propriedades dos principais algoritmos evolucionários. Adaptado de [20].

	AG	EE	PE	PG
Representação	Cadeias binárias/Permutações	Vetores reais	Vetores reais	Árvores
Fitness	Valor proporcional à função de custo	Valor da função de custo	Valor proporcional à função de custo	Valor proporcional à função de custo
Recombinação	Operador principal	Diferentes variações	Nenhuma	Um dos operadores
Mutação	Operador secundário	Operador principal	Único operador	Um dos operadores
Seleção	Probabilística	Determinística	Probabilística	Probabilística

Pela análise da Tabela 2.2 verifica se que a representação, ou seja, a ponte entre o contexto original do problema e o espaço de soluções onde o algoritmo evolutivo trabalha é uma das propriedades que apresenta maior variação entre os diferentes algoritmos, sendo que nos AG a codificação das variáveis de decisão é feita através de uma cadeia binária como o exemplo apresentado na Figura 2.10 resultando numa maior precisão numérica [21], podendo codificar variáveis inteiras ou reais, ou através de permutações onde dados n objetos distintos, existem n! permutações destes objetos.

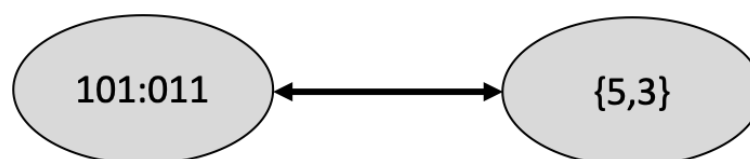


Figura 2.10 - Exemplo de Codificação Binária. Adaptado de [21].

Quanto às EE e à PE a codificação é feita através de um vetor de reais, onde cada elemento do vetor corresponde a uma variável de decisão do problema de otimização, sendo esta representação mais rápida de manipular [20]. Por fim, na PG a representação faz-se a partir de uma árvore, este método é normalmente utilizado na evolução de programas computacionais, podendo variar o seu tamanho ao longo do processo evolutivo.

Analisando agora a propriedade de *fitness*, a qual atribui uma medida de qualidade ao genótipo pode ser um valor proporcional à função de custo, ou diretamente o valor da função de custo, este último apenas acontece nas EE.

A propriedade de recombinação pela qual é criado um novo descendente, é considerado por alguns um dos operadores mais importantes visto ser aquele que confere diversidade às soluções [22]. No caso dos AG este é o operador principal, ou seja, aquele que tem mais probabilidade de acontecer.

Esta recombinação pode ser feita através de codificação binária como representado na Figura 2.11, em que são criados dois filhos partindo de duas soluções pais.



Figura 2.11 - Recombinação a partir de codificação binária. Adaptado de [23].

Neste caso da Figura 2.11 escolheu-se de forma aleatória um número no intervalo, dividiram-se os cromossomas nesse ponto e criaram-se os respectivos filhos trocando as caudas de cada um deles.

Pode ainda na recombinação ser utilizada a codificação por permutações como representado na Figura 2.12.



Figura 2.12 - Recombinação utilizando permutações.

Neste caso começou-se por escolher aleatoriamente dois pontos de recombinação e copiar o segmento do primeiro pai para a mesma posição do primeiro filho. De seguida partiu-se do segundo ponto de recombinação do segundo pai e copiou-se os restantes valores utilizados não para o primeiro filho pela ordem que aparecem no segundo pai.

Contrariamente à recombinação, a mutação permite a geração de um ou mais descendentes partindo de um único progenitor, nas EE existe maior probabilidade de acontecer relativamente à recombinação.

Também na mutação tal como na recombinação podemos ter uma mutação por codificação binária em que neste caso como representado na Figura 2.13 partindo de um único progenitor obtemos um descendente.



Figura 2.13 – Exemplo de mutação em codificação binária. Adaptado de [23].

No caso da Figura 2.13 verifica-se que o terceiro, quarto e oitavo bits sofreram mutação.

Podemos ainda ter mutações aplicadas a uma permutação em que uma permutação tem que gerar obrigatoriamente uma estrutura de dados que seja também uma permutação. Neste caso deixa de ser possível considerar cada gene individualmente, sendo assim necessário ter em conta todo o cromossoma. Existem 4 tipos diferentes de mutações aplicadas a permutações, sendo elas:

Mutação por troca- Como representado na Figura 2.14 escolhem-se aleatoriamente duas posições e trocam-se essas posições.



Figura 2.14 - Mutação por troca.

Mutação por inserção- Neste caso são escolhidos dois genes aleatoriamente e move um para a posição imediatamente a seguir ao outro como representado na Figura 2.15.



Figura 2.15 - Mutação por inserção.

Mutação por mistura- São escolhidos um subconjunto de valores e os mesmos são reordenados de forma aleatória como apresentado no exemplo da Figura 2.16.



Figura 2.16 - Mutação por mistura.

Mutação por inversão- Na mutação por inversão são selecionados aleatoriamente dois pontos e reverte-se a ordem entre esses dois pontos como apresentado no exemplo da Figura 2.17



Figura 2.17 - Mutaç o por invers o.

Quanto a seleç o, a mais usada   a seleç o a partir de probabilidades, o que privilegia os indiv duos mais aptos, que possuem um maior fitness, existindo sempre uma pequena probabilidade de seleç o dos indiv duos menos aptos.

De acordo com as caracter sticas anteriormente apresentadas, o algoritmo evolucion rio selecionado para a otimizaç o dos percursos na aplicaç o SHCity-Turista   o algoritmo gen tico. Este algoritmo possibilita a codificaç o de problemas em forma de permutaç es bem como dos respetivos mecanismos de recombinaç o e mutaç o. A grande vantagem da utilizaç o de permutaç es   facilitar a codificaç o problema sob estudo, uma vez que se pretende obter uma sequ ncia otimizada de monumentos sem que nenhum se repita dentro da mesma visita. A utilizaç o deste algoritmo baseia-se na utilizaç o dos indiv duos com maior fitness o que permite obter no final uma soluç o satisfat ria num intervalo de tempo reduzido, quando comparado com outros algoritmos evolucion rios [22]

Arquitetura e Implementação

Neste capítulo procede-se à descrição das diferentes arquiteturas utilizadas, ao nível da aplicação móvel serão descritas as 5 principais funcionalidades da aplicação móvel e a ligação entre os seus diferentes menus. É apresentada uma descrição detalhada da arquitetura do algoritmo utilizado e o seu funcionamento.

Procede-se também à descrição da implementação dos menus da aplicação móvel e dos percursos de otimização. Proceceu-se ao levantamento de qual a ferramenta a utilizar para o desenvolvimento da aplicação móvel SHCity-Turista. Para a escolha desta ferramenta teve-se em conta as funcionalidades necessárias e a otimização necessária para o bom funcionamento da aplicação. Outsystems foi a plataforma utilizada devido à sua simplicidade de utilização, às funcionalidades que possui e ao suporte disponível aos utilizadores da mesma.

3.1 Arquitetura da Aplicação

Com a ambição de melhorar a experiência do turista ao visitar a cidade de Ávila, Espanha, foi desenvolvida a aplicação móvel SHCity-Turista. Esta aplicação pretende disponibilizar ao utilizador de uma forma simples e intuitiva diferentes funcionalidades, as quais se encontram representadas na **Error! Reference source not found.**, funcionalidades estas que irão melhorar a visita do utilizador à cidade. Possui também quatro idiomas disponíveis, sendo eles o espanhol, português, francês e inglês.

A arquitetura da aplicação móvel é dividida em cinco principais funcionalidades, sendo elas:

- **Lista de Monumentos/Lugares** - Uma lista de todos os monumentos/lugares disponíveis para visita esta acessível ao utilizador. Estes monumentos podem também estar representados num mapa onde a localização de cada um dos monumentos é apresentada. Em ambos os casos o utilizador poderá escolher um dos monumentos/lugares disponíveis para visita, sobre o qual obterá fotos, uma breve descrição histórica e informações em tempo real são mostradas. O utilizador terá então a opção de obter direções desde a sua localização atual para o monumento escolhido ou pode simplesmente voltar atrás e escolher outro monumento/lugar.

- **Lista de Circuitos-** Uma lista de circuitos pré-definidos com diferentes durações e graus de dificuldade está disponível. Quando um desses circuitos é escolhido, fotos e informações sobre o circuito são mostradas, o utilizador pode então ver o circuito num mapa e iniciar a visita, onde as direções para cada monumento são dadas.
- **Criar Circuito-** Nesta opção o utilizador tem a possibilidade de criar o seu próprio circuito. O utilizador pode escolher entre criar um circuito otimizado ou personalizado. No primeiro caso, depois de escolher os monumentos/lugares que deseja visitar, um circuito otimizado é criado, tendo em consideração fatores como o tempo de visita e de espera num determinado monumento, permitindo assim ao utilizador minimizar o tempo de visita, não porque deixa de visitar algum local, mas porquê otimiza o percurso e o tempo de espera. Na segunda opção de criação de um circuito personalizado, o utilizador seleciona os monumentos/lugares por ordem de visita, obtendo no final um circuito com a ordem escolhida. Em ambos os casos, a duração total e a distância são mostradas e quando o circuito é iniciado a aplicação móvel mostra as direções para cada monumento/lugar.
- **Outros Serviços-** Pretende se ainda apresentar num mapa a localização das principais farmácias, hospitais e áreas verdes existentes na cidade de Ávila. O utilizador pode também obter direções para cada um destes locais.
- **Multimédia-** Por fim, será disponibilizada uma lista de vídeos sobre a cidade de Ávila e os seus principais pontos turísticos.

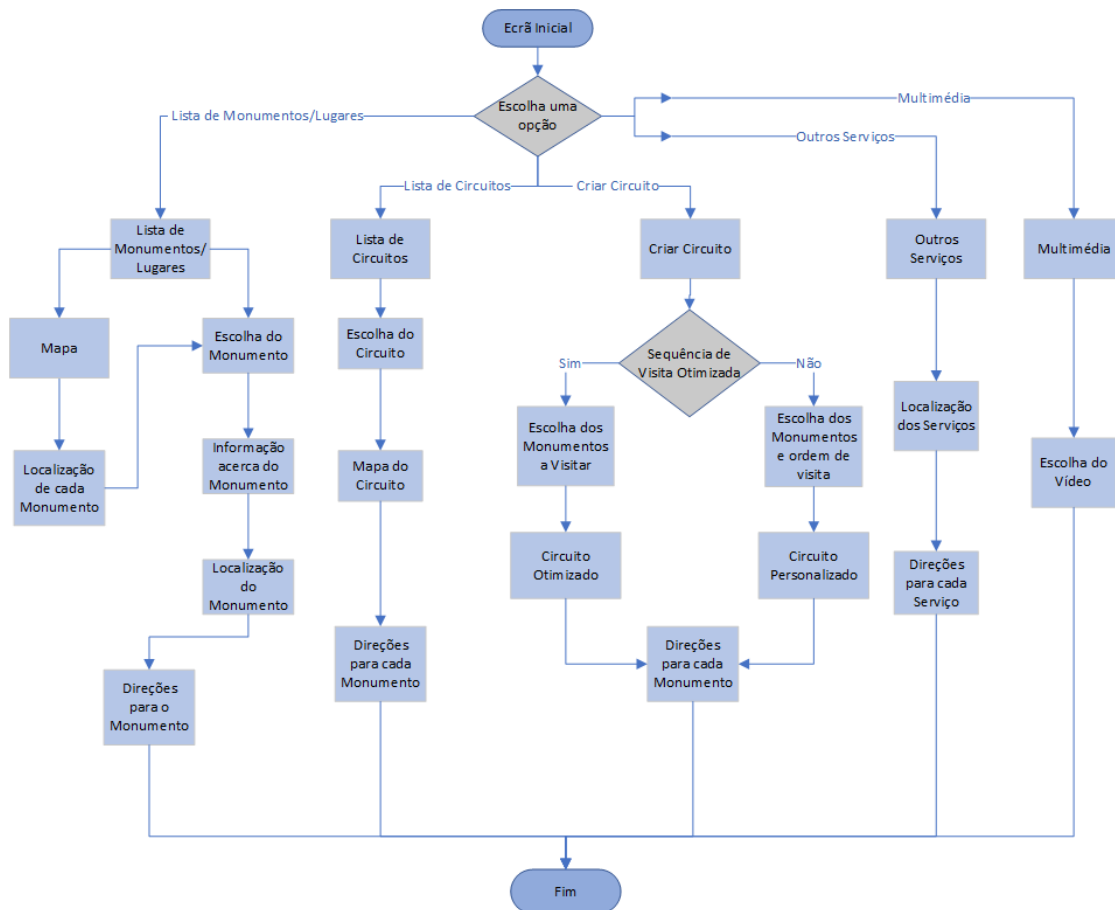


Figura 3.1 - Fluxograma da Aplicação Móvel SHCity-Turista.

3.1.1 Arquitetura do Algoritmo de Otimização

Na Figura 3.2 encontra-se ilustrado o fluxograma do algoritmo de otimização que será implementado, tendo por objetivo minimizar o tempo de visita do turista, considerando informação em tempo real acerca dos monumentos/lugares. Convém realçar que a otimização da visita não tem por base ver muitos locais em pouco tempo, mas sim, assenta no pilar de minimizar o percurso entre monumentos/lugares e os tempos de espera para entrar nesses mesmos espaços.

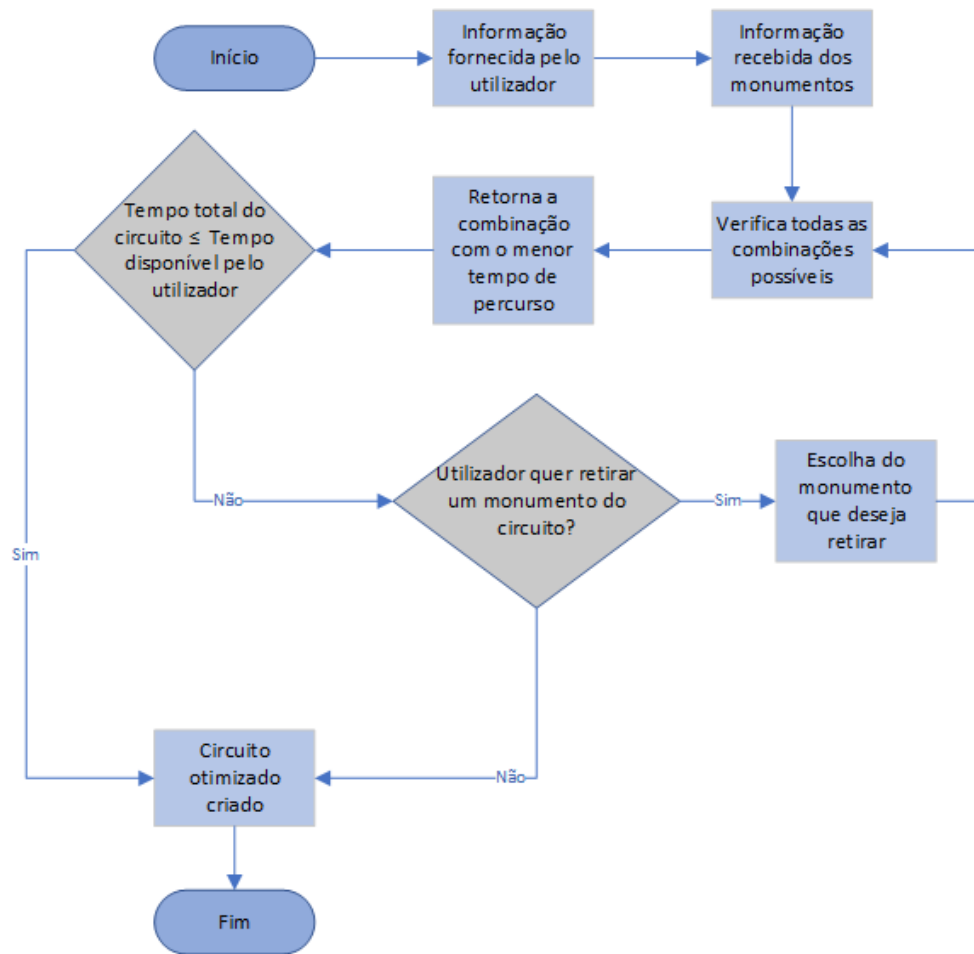


Figura 3.2 - Fluxograma do Algoritmo de Otimização.

Este algoritmo começa por receber como parâmetros de entrada quais os monumentos/lugares que o utilizador deseja visitar, os tempos de visita e de espera em cada um destes monumentos e o tempo total de visita disponível. Após ter recebido esta informação o algoritmo verifica todas as combinações de percurso possíveis e retorna aquela que verificar o menor tempo de visita tendo em conta a informação em tempo real dos monumentos. O tempo total de circuito retornado é então comparado com o tempo total de visita disponível. Se for menor é devolvido o circuito otimizado criado, caso seja maior o utilizador tem a opção de remover um monumento/lugar da sua visita de modo a baixar o tempo total do circuito ou a continuar com o tempo total obtido.

3.2 Implementação dos Menus da Aplicação Móvel

A plataforma *Outsystems* disponibiliza alguns *templates* de ecrãs que podem ser usados de modo a facilitar a construção e implementação da aplicação móvel. No caso da aplicação a desenvolver nesta Dissertação, no âmbito do projeto SHCity, existem diferentes ecrãs com estilos e organização de conteúdos diferentes, o que fez com que fossem desenvolvidos estilos personalizados de modo a implementarmos os ecrãs pretendidos.

Na Figura 3.3 encontra-se representado o ecrã principal da aplicação móvel SHCity-Turista, o qual apresenta no título o nome da aplicação, no cabeçalho as 4 opções de língua disponíveis, sendo elas o espanhol, o português, o francês e o inglês e como conteúdo as opções disponíveis que permite ao turista selecionar a opção de acordo com os seus interesses.

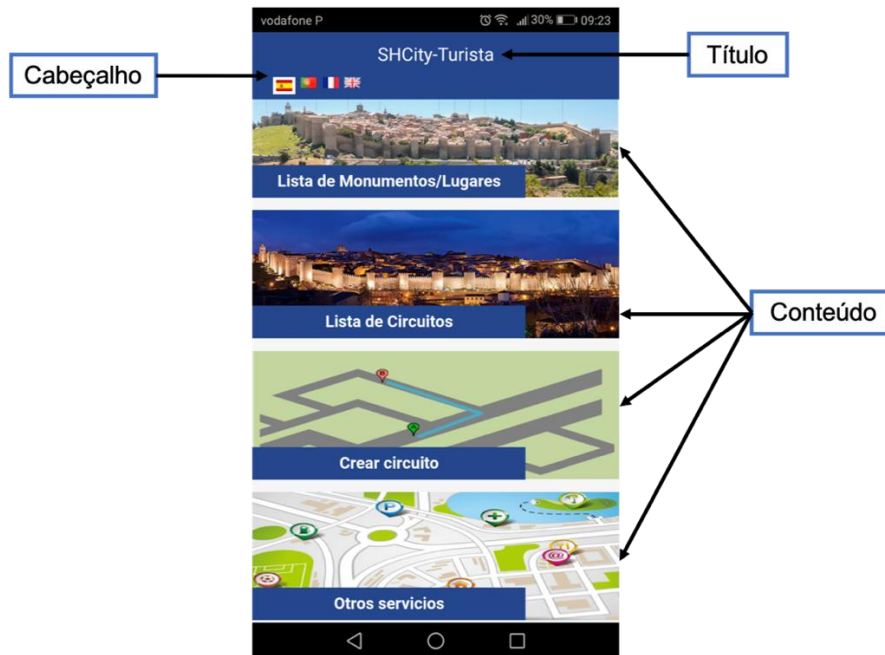


Figura 3.3 - Ecrã inicial da aplicação móvel SHCity-Turista.

Para a implementação do ecrã ilustrado na Figura 3.3 a plataforma *Outsystems* possui um editor CSS (*Cascading Style Sheets*) que permite personalizar todos os aspetos do ecrã. Quanto ao título utilizou-se como fundo uma cor de código #24468D, letra do tipo Arial e tamanho 18 pixéis. Relativamente ao cabeçalho, utilizaram-se 4 *links* com imagens das bandeiras das 4 línguas disponíveis, cada bandeira possui uma margem à esquerda de 20 pixéis e quando uma das línguas é selecionada aparece em torno da bandeira uma fronteira em branco com dimensão de 3 pixéis. O conteúdo é composto por 5 *links* em forma de caixa que quando selecionados transitam para os respetivos ecrãs. Cada um destes links possui uma imagem e por cima dessa imagem uma caixa em que a sua formatação em CSS se encontra explicada pelo pseudo-código do Algoritmo 3.1

Algoritmo 3.1 - Pseudo-código conteúdo ecrã inicial.

```

Fazer
Para cada caixa
  Definir cor de fundo #24468D
  Definir cor de letra branca
  Alinhar texto ao centro
  Definir posição relativa
  Padding top e bottom de 5 pixeis
  Definir posição relativa ao topo de 35 pixeis
Fim
  
```

Escolhendo a primeira opção é apresentado ao utilizador uma lista com os monumentos/lugares disponíveis para visita como se encontra representado na Figura 3.4.



Figura 3.4 - Ecrã lista de monumentos/lugares.

Quando à lista de monumentos/lugares em termos de implementação, este ecrã possui duas diferenças relativamente ao ecrã inicial apresentado anteriormente, na Figura 3.3. Foi criado do lado esquerdo do cabeçalho um botão que permite ao utilizador voltar para a página anterior, este botão consiste simplesmente num símbolo, já existente na plataforma *Outsystems*, com um *link* para a página anterior. Em termos de conteúdo, a lista foi implementada da mesma forma que no ecrã inicial, sendo que neste caso a seleção de um monumento abre um novo ecrã com informação acerca do mesmo. Foi também criado na parte inferior do ecrã uma barra que neste caso serve de botão para um mapa, a formatação deste botão encontra-se explicada no Algoritmo 3.2.

Algoritmo 3.2 - Pseudo-código barra inferior lista de monumentos/lugares.

```

Para Cada barra inferior
  Definir cor de fundo #FFA500
  Altura de 50 pixeis
  Alinhar texto ao centro
  Definir cor de letra branca
Fim

```

Carregando no botão mapa um novo ecrã é apresentado com a localização de cada monumento/lugar como exibido na Figura 3.5.

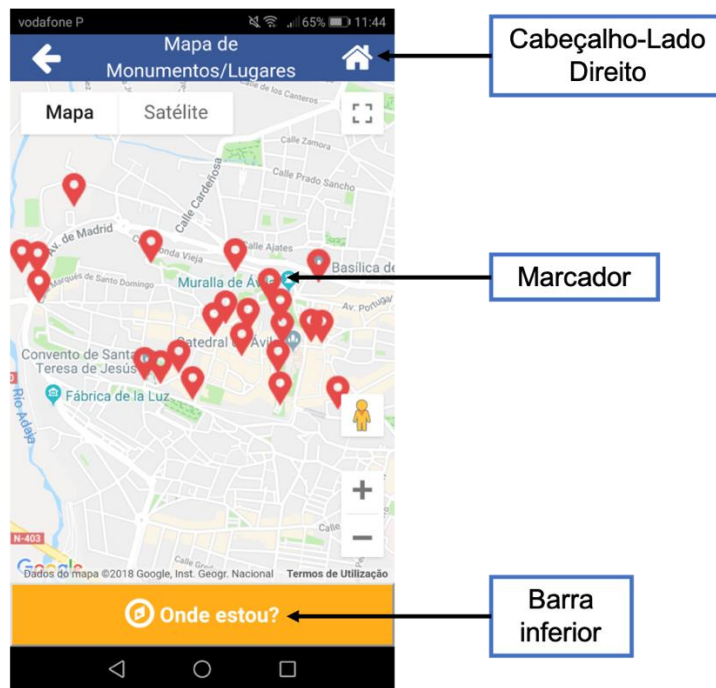


Figura 3.5 - Ecrã mapa de monumentos/lugares.

Neste ecrã existe no cabeçalho do lado direito a opção de voltar á página inicial, consiste tal como no botão de voltar para trás de um símbolo com um *link* para o ecrã inicial da aplicação. Para o conteúdo deste ecrã utilizou-se a API da *Google Maps* para mostrar o mapa, definindo-se uma localização onde o mapa fica centrado. Para os marcadores foi necessário fazer um levantamento da localização de cada monumento/lugar, tendo sido posteriormente criada uma lista de marcadores os quais se encontram representados no mapa. Estes marcadores ao serem seleccionados abrem um novo ecrã com a informação de cada um dos monumentos/lugares. O botão situado na barra inferior permite ao utilizador saber a sua localização relativamente aos marcadores, para isso utilizou se uma ação desenvolvida pela *Outsystems* designada de *GetLocation* a qual retorna à localização atual do dispositivo, esta informação é depois atribuída a um marcador que fica visível no mapa. Como referido anteriormente cada monumento/lugar contém informação acerca do mesmo, a qual é apresentada num ecrã como o representado na Figura 3.6.



Figura 3.6 - Ecrã monumento *Catedral Y Plaza De La Catedral*.

Um carrossel de fotos mostra algumas imagens do monumento/lugar, sendo que imediatamente abaixo é apresentada informação acerca do mesmo, esta está disposta num contentor com cor de fundo de código #DCDCDC e com dimensão variável dependendo da informação contida nele. Alguma desta informação é estática como uma breve descrição dos monumentos e o tempo médio de visita. O utilizador dispõe ainda de informação em tempo real como ocupação, temperatura, humidade, tempo de espera e horário de abertura e fecho do local, estes dados são medidos através de sensores instalados nos monumentos/lugares, são depois guardados numa base de dados, a qual comunica com a aplicação através de uma *Rest API*.

Ao carregar no botão mapa é mostrado num mapa a localização do monumento/lugar relativamente aos restantes monumentos, como ilustrado na Figura 3.7 a), este marcador encontra-se com uma cor diferente dos restantes com o objetivo de o destacar. O utilizador tem também a opção de obter direções desde a sua localização até ao local escolhido como ilustrado na Figura 3.7 b).

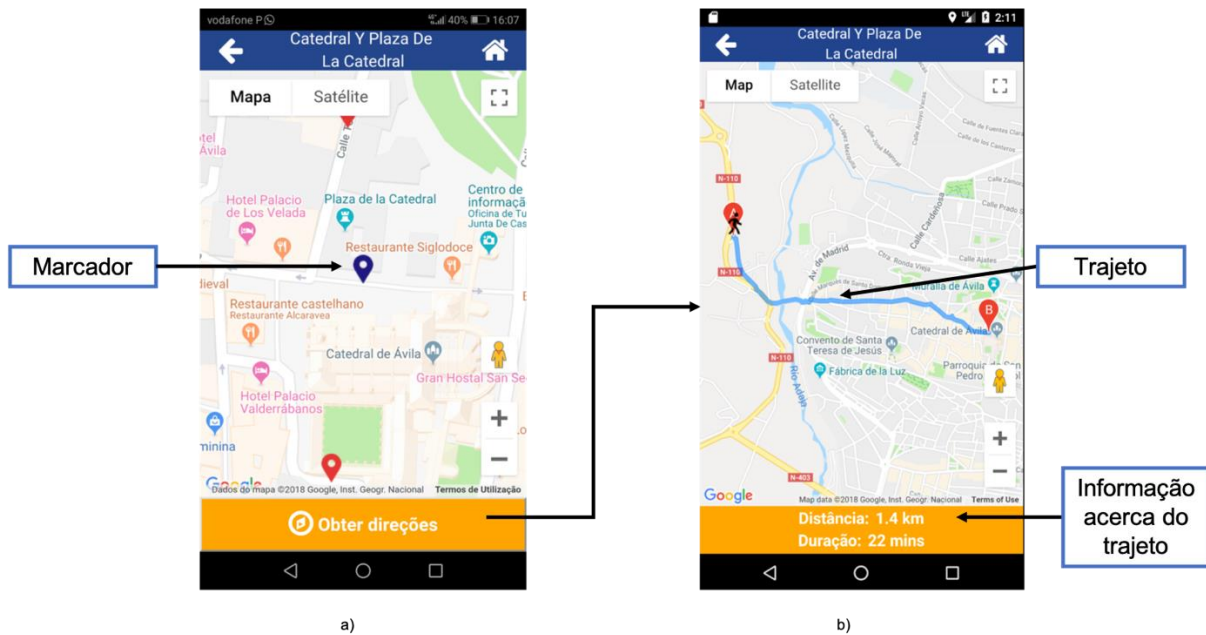


Figura 3.7 - Ecrã localização local a) e direções até ao local b).

Para obter o trajeto representado na Figura 3.7 b) desde a localização atual do utilizador até ao local selecionado utilizou-se a API de direções da Google. Como esta API não permite navegação em tempo real, para contornar esse problema criou-se um marcador que a cada segundo recebe a localização atual do dispositivo, permitindo assim ao utilizador percorrer o trajeto em tempo real. A aplicação móvel fornece ainda ao utilizador informação acerca do trajeto como a distância e duração total, esta informação é fornecida através da API de distância da Google.

Quanto aos circuitos pré-definidos, o utilizador pode escolher de uma lista de 5 circuitos qual aquele que pretende fazer. Como representado na Figura 3.8 a) cada um destes circuitos possui informação à cerca da duração e número de monumentos/lugares que o constituem. Ao selecionar um dos circuitos, o utilizador encontra um carrossel de fotos de alguns dos monumentos que constituem o circuito, a duração total e quais os monumentos/lugares pelos quais o circuito passa como ilustrado na Figura 3.8 b)

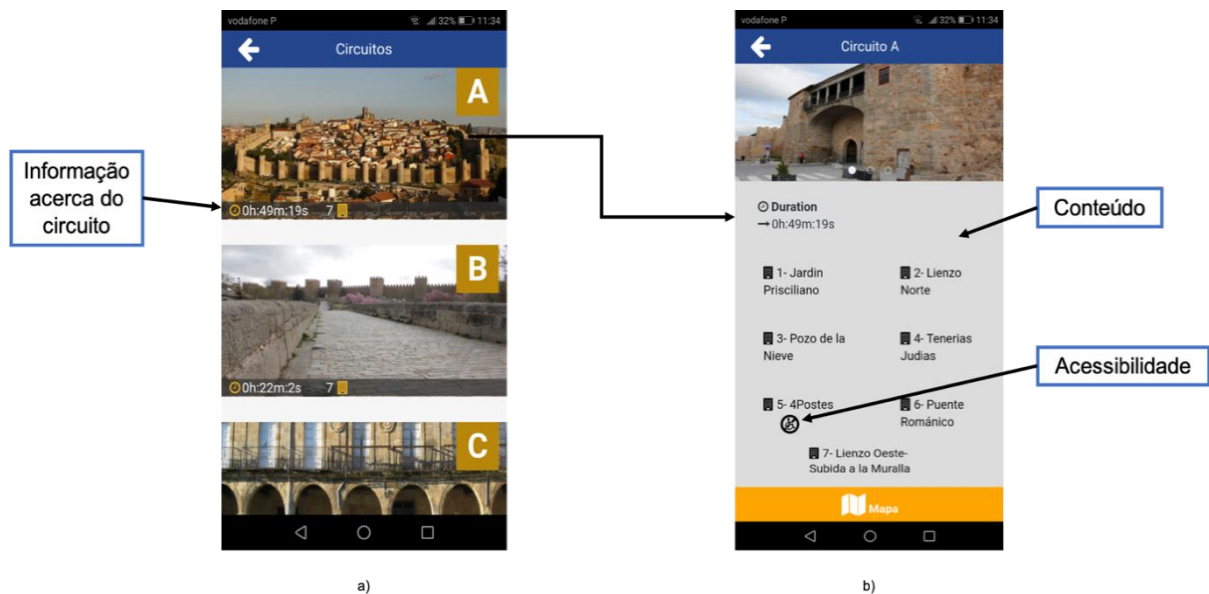


Figura 3.8 - Ecrã lista de circuitos a) e circuito A b).

Para obtenção do tempo total do circuito utilizamos mais uma vez a API de distância da Google, Equação 3.1 sendo que neste caso utilizamos a funcionalidade de acrescentar pontos intermédios ao trajeto. A API devolve então o tempo em segundos entre cada um dos trajetos, sendo depois necessário somar todos os tempos e converter em hora pela Equação 3.1, minutos pela Equação 3.2 e segundos pela Equação 3.3.

$$\text{Horas} = \frac{\text{Tempo total}}{3600} \quad \text{Equação 3.1}$$

$$\text{Minutos} = \frac{\text{Resto}(\text{Tempo total}/3600)}{60} \quad \text{Equação 3.2}$$

$$\text{Segundos} = \text{Resto}(\text{Tempo total}/60) \quad \text{Equação 3.3}$$

O conteúdo como exemplificado na Figura 3.8 b) para o circuito A possui para além da duração total, todos os monumentos/lugares incluídos no circuito, estes monumentos/lugares ao serem selecionados abrem a página com a informação à cerca do mesmo, para além disso existe informação acerca da acessibilidade, como por exemplo neste caso o monumentos *4Postes* não possui acessibilidade.

Ao carregar em mapa, é apresentado ao utilizador um novo ecrã com o mapa do circuito, a duração e distância total do circuito como representado na Figura 3.9 a). Ao carregar em iniciar, é apresentado ao utilizador o ecrã da Figura 3.9 b) que contém o trajeto com informação acerca da distância e duração desde a localização do dispositivo até cada um dos monumentos/lugares do circuito, para navegar entre os seguintes monumentos/lugares basta carregar no botão próximo monumento.

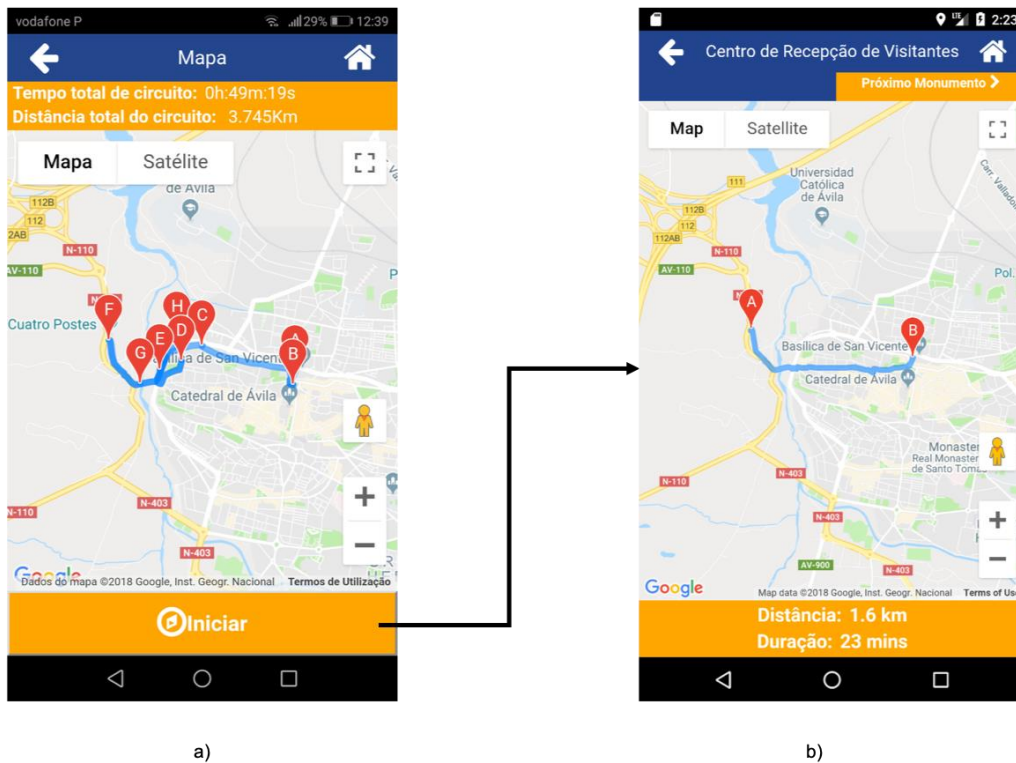


Figura 3.9 - Ecrã mapa do circuito a) e trajeto para cada monumento/lugar b).

Quando o utilizador pretende criar o seu próprio circuito tem duas opções disponíveis, como ilustradas na Figura 3.10, de criar um circuito otimizado em que a ordem dos monumentos/lugares é otimizada de modo a rentabilizar ao máximo o tempo de visita tendo em conta dados em tempo real, como a ocupação e tempo de espera de cada monumento/lugar ou um circuito personalizado em que o utilizador escolhe a ordem pela qual quer visitar os monumentos/lugares.

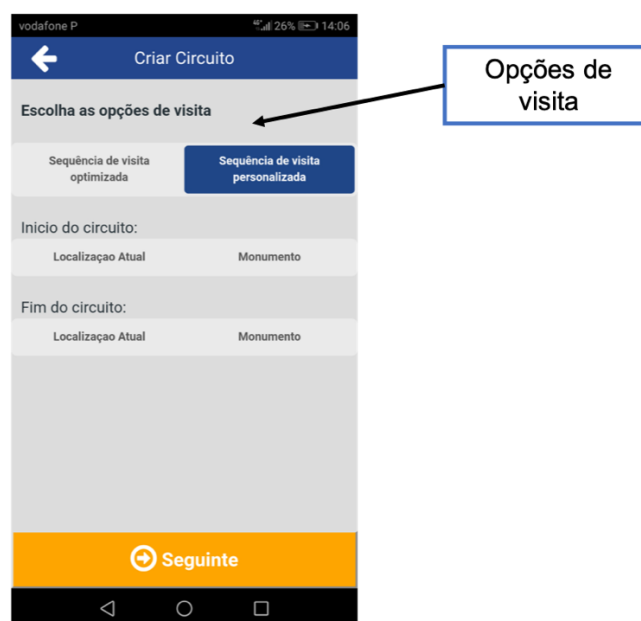


Figura 3.10- Ecrã criar circuito.

A aplicação permite ainda a opção de escolher se pretende iniciar ou finalizar o circuito na sua localização atual ou num determinado monumento/lugar.

A implementação destas opções foi feita através de botões de grupo em que a sua lógica de funcionamento se encontra explicada no Algoritmo 3.3.

Algoritmo 3.3- Pseudo-código da lógica dos botões de grupo.

```
Para botão grupo1
  Se opção de Sequência de visita otimizada
    Variável opção ordem=1
  Se não
    Variável opção ordem=2
Fim

Para botão grupo2
  Se opção de Inicio do circuito igual a Localização Atual
    Variável opção origem=1
  Se não
    Variável opção origem=2
Fim

Para botão grupo3
  Se opção de Fim do circuito igual a Localização Atual
    Variável opção fim =1
  Se não
    Variável opção fim=2
Fim
```

Ao escolher a opção de visita otimizada é feita a transição para o ecrã apresentado na Figura 3.11, os monumentos/lugares disponíveis são apresentados através de uma lista, implementada utilizando um carrossel de objetos, e num mapa com a localização de cada um dos monumentos/lugares.



Figura 3.11- Ecrã criar circuito otimizado.

O utilizador pode adicionar um máximo de oito monumentos/lugares ao seu circuito, esta limitação deve se à utilização da versão gratuita da API da Google, que limita o número de pontos

intermédios. Para facilitar a escolha de quais os monumentos/lugares a acrescentar ao circuito o utilizador dispõe do tempo de visita médio de cada monumento e do tempo total estimado do circuito que vai variando de acordo com os monumentos/lugares selecionados.

Ao selecionar quais os monumentos/lugares a adicionar ao circuito, os marcadores dos mesmos no mapa mudam de cor como ilustrado na Figura 3.12, auxiliando assim o utilizador. Para a implementação deste sistema foram utilizadas variáveis em cada uma das caixas de seleção que ao serem selecionadas atribuem o valor de verdadeiro às mesmas e alteram o marcador correspondente para azul. Verifica-se também que o tempo total do circuito é o resultado da soma dos tempos médios de visita, não tendo em conta os tempos de espera nem a duração dos percursos, como tal, este tempo é apenas indicativo.

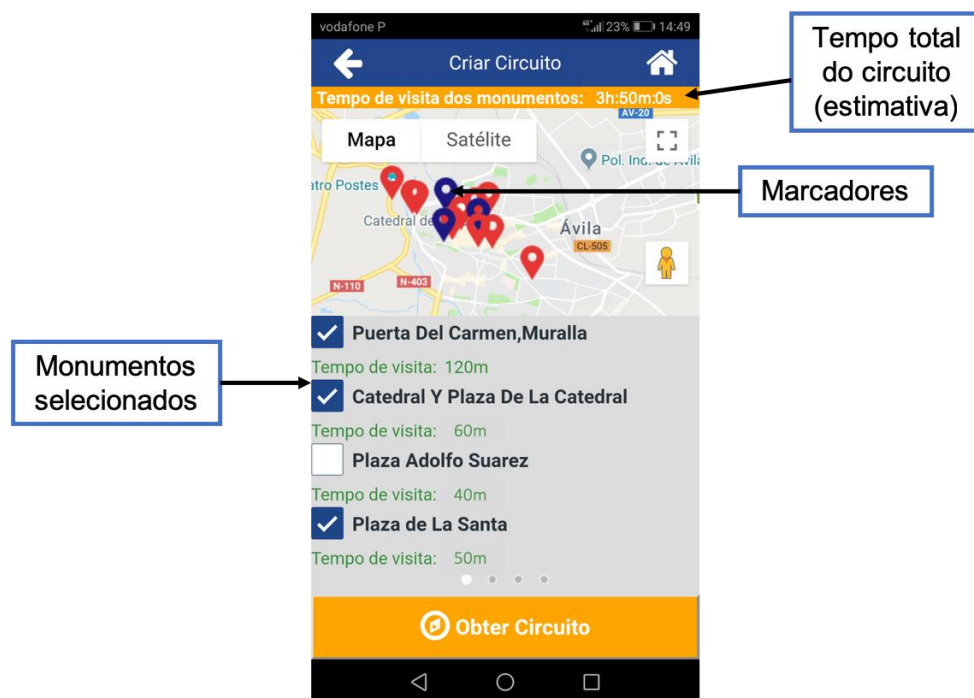


Figura 3.12 - Ecrã criar circuito otimizado com monumentos/lugares selecionados.

Ao carregar em obter circuito é mostrado num mapa como representado na Figura 3.13, o circuito otimizado. No topo do ecrã o tempo total otimizado, incluindo já o tempo de percurso e o tempo de espera em cada um dos monumentos/lugares e a distância total do circuito estão disponíveis. Ao iniciar o circuito é mostrado o trajeto para cada um dos monumentos/lugares tal como acontece para os circuitos pré-definidos, explicado anteriormente.



Figura 3.13 - Ecrã circuito otimizado obtido.

Quanto ao circuito personalizado o utilizador escolhe os monumentos/lugares pela ordem de visita pretendida, começa por escolher o primeiro como representado na Figura 3.14 e assim sucessivamente. Tal como acontece no caso do circuito otimizado a API da Google permite apenas a implementação de 8 pontos intermédios.



Figura 3.14 - Ecrã escolha monumento/lugar a incluir no circuito personalizado.

Os monumentos/lugares disponíveis encontram-se dispostos num carrossel de objetos onde está também disponível o tempo médio de visita de cada monumento/lugar. O utilizador apenas pode passar à escolha do próximo monumento/lugar quando já selecionou um monumento, esta verificação é feita através das variáveis globais presentes nas caixas de seleção em que ao serem selecionadas atribuem o valor de verdadeiro a essa variável, sendo que cada caixa possui uma variável diferente.

Quando o utilizador passa para a escolha do próximo monumento, como ilustrado na Figura 3.15, o marcador do monumento/lugar selecionado anteriormente fica a verde e na lista fica bloqueado, impossibilitando que o utilizador o volte a escolher, para isto utilizam-se as mesmas variáveis globais utilizadas anteriormente. A aplicação apresenta sempre no topo do ecrã a informação referente ao circuito, nomeadamente a duração e a distância total do circuito. Quando o utilizador finaliza a sua escolha basta carregar no botão de obter circuito e será apresentado o circuito com a ordem dos monumentos/lugares escolhida.



Figura 3.15 - Ecrã escolha do segundo monumento/lugar a incluir no circuito personalizado.

A funcionalidade multimédia apresentada pelo ecrã da Figura 3.16 permite ao utilizador visualizar vídeos da cidade de Ávila. O conteúdo deste ecrã consiste na implementação de caixas com imagens retiradas dos vídeos e o título do vídeo em baixo, em que cada caixa corresponde a um *link* que ao ser selecionado redireciona para uma página na internet.



Figura 3.16 - Ecrã multimédia.

3.3 Implementação do Algoritmo de Otimização

O algoritmo genético de otimização de percursos começou por ser desenvolvido em ambiente de simulação, onde diferentes simulações foram feitas. Para a implementação deste algoritmo utilizou-se a ferramenta *toolbox GA (Genetic Algorithm)*, a qual encontra o valor mínimo de uma função utilizando algoritmos genéticos, esta ferramenta já inclui as funções de criação, mutação e recombinação de permutações.

Para a utilização da *toolbox GA*, é necessário desenvolver uma função através da qual a *toolbox GA* encontra um valor mínimo, a qual se designa por função de custo. Esta função tem como parâmetro de entrada um conjunto de vetores com as soluções possíveis para o problema, onde a dimensão deste conjunto é definida durante a definição do problema e representa o tamanho total da população. Foi estabelecido que se o número de monumentos/lugares incluídos no circuito fosse inferior a 6 o tamanho da população inicial é de 40 soluções, caso fosse superior o tamanho é de 60 soluções, esta diferença deve-se à precisão do algoritmo genético, em que com um maior número de monumentos/lugares de modo a obter uma solução satisfatória é necessária uma população inicial superior. Esta função tem como variáveis globais, a duração em minutos entre cada monumento escolhido e o tempo de espera e visita de cada um dos monumentos/lugares. O tempo de espera de cada um dos monumentos/lugares encontra-se dividido em 3 horários, sendo o primeiro das nove horas ao meio dia, o segundo do meio dia às dezasseis horas e o último das dezasseis às vinte horas, em que em cada um desses intervalos encontra-se a média dos tempos de espera a cada hora até à data de criação do circuito otimizado. Esta divisão dos tempos de espera permite prever qual vai ser o tempo de espera de um determinado monumento/lugar à hora que o utilizador passa no mesmo. O pseudo-código da função de custo encontra-se representado no Algoritmo 3.4.

Algoritmo 3.4 - Pseudo-código da função de custo.

```
Para cada solução
  Ver o número e ordem dos monumentos/lugares

  Para cada monumento/lugar

    Verificar a hora atual
    Se Hora atual entre as 9h e as 12h
      Buscar à tabela o tempo de espera das 9h às 12h
    Se não Hora atual entre as 12h e as 16h
      Buscar à tabela o tempo de espera das 12h às 16h
    Se não Hora atual superior às 16h e as 20h
      Buscar à tabela o tempo de espera das 16h às 20h

    Tempo total=tempo total + tempo de visita + tempo de espera + duração
    percurso até ao próximo monumento/lugar

  Fim

  Guardar o tempo total para todas as soluções
Fim
```

Concluído o desenvolvimento da função de custo, passou-se ao desenvolvimento da função principal, através da qual é feita a interação com o utilizador, o cálculo de todas as distâncias e duração entre monumentos, a utilização da ferramenta GA e a formatação de todos os gráficos utilizados.

Como representado pela simulação da Figura 3.17 começa-se por perguntar ao utilizador quais os monumentos que pretende visitar de 1 a 13, guardando-se num vetor as escolhas, de seguida pergunta-se o tempo disponível para visita, onde as horas são convertidas em minutos e o tempo total é guardado numa variável local.

```
Que monumentos pretende visitar?
4 7 9 10
Tempo total de visita disponivel
Horas:
4
Minutos:
0
```

Figura 3.17 - Simulação da seleção dos monumentos e tempo de visita disponível.

Com os monumentos a incluir no circuito guardados num vetor, e as coordenadas dos mesmos adquiridas a partir de um ficheiro Excel, pode-se através do pseudo-código representado no Algoritmo 3.5 calcular a distância em quilómetros entre cada um dos monumentos.

Algoritmo 3.5 - Pseudo-código do cálculo das distâncias.

```
Para cada monumento
    Para cada monumento seguinte
        Guardar as coordenadas dos monumentos em variáveis locais
        Calcular distância entre os dois monumentos
        Guardar valor em vetor
    Até chegar ao último monumento
Fim

Até chegar ao último monumento
Fim
```

Para o cálculo da duração dos percursos assumiu-se que em média as pessoas andam a uma velocidade de 4 quilómetros por hora. Sendo assim através da Equação 3.4 conseguiu-se calcular a duração em minutos de cada um dos troços.

$$Duração(\text{minutos}) = \frac{Distância(\text{km})}{4\left(\frac{\text{km}}{\text{hora}}\right)} \times 60 \quad \text{Equação 3.4}$$

Tendo a sequência de monumentos escolhida, o tempo disponível para visita e a duração entre cada troço calculada, pode-se agora utilizar a *toolbox* GA. No caso do problema abordado, esta função vai ter como parâmetros de entrada a função de custo explicada anteriormente, o número de monumentos e as opções que pretendemos usar no algoritmo genético. Quanto a estas opções utilizamos um máximo de 500 gerações, uma população variável conforme o número de monumentos escolhidos em que se for inferior a 6 temos uma população de 40 soluções iniciais, caso seja superior a 6 de 60 soluções, de modo a obtermos uma solução mais satisfatória para um maior número de monumentos. Quanto à tolerância da função usou-se o valor definido de 1e-6 e um limite no número de gerações de 60, ou seja se ao fim de 60 gerações não existir uma variação entre cada geração superior a 1e-6, o algoritmo para e devolve a solução mais satisfatória que encontrou, caso não se verifique esta condição serão geradas as 500 gerações máximas definidas anteriormente.

Com os parâmetros anteriormente definidos a função GA irá então retornar o vetor otimizado e o tempo total em minutos desse vetor, sendo que é necessário converter esse tempo total em horas e minutos e atribuir o vetor otimizado aos respetivos monumentos/lugares. É depois feita uma verificação através de um conjunto de condições representadas no Algoritmo 3.6 entre a duração total do circuito otimizado e o tempo disponível pelo utilizador.

Algoritmo 3.6 - Pseudo-código da verificação entre duração e tempo disponível.

```
Se Duração total do circuito >= Tempo total disponível

  Enquanto (Duração total do circuito >= Tempo total disponível)
    Utilizador tem a opção de remover um monumento/lugar ou continuar
    com a sequência obtida

    Se utilizador pretender remover um monumento/lugar
      Escolha de qual o monumento/lugar a remover
      Algoritmo devolve nova duração otimizada sem o monumento/lugar selecionado
    Se não
      Sai do ciclo

  Fim
  Algoritmo finalizado

Se não
  Algoritmo finalizado
```

Como representado no Algoritmo 3.6 caso o tempo disponível pelo utilizador seja inferior à duração total do circuito otimizado é dada a opção de remover um monumento/lugar ou de continuar com a ordem e duração obtidas como apresentado na Figura 3.18 em que o utilizador apenas dispõe de 3 horas para visitar os monumentos/lugares 4, 9, 7 e 10 e a duração do circuito otimizado é de 3 horas e 45 minutos.

```
ordem_optimizada =
  4   9   7   10
|
duracao_optimizada =
03:45:00

duration_opt =
  225

Excedeu o seu tempo de visita disponível. Deseja continuar com este tempo[Y] ou retirar um monumento[R]?
```

Figura 3.18 - Simulação no caso de se exceder o tempo de visita disponível.

Ao ser finalizada a otimização do percurso é apresentado ao utilizador a ordem final dos monumentos, o tempo total de visita e um mapa do circuito otimizado como apresentado pela simulação da Figura 3.19. Neste mapa estão distribuídos os monumentos/lugares disponíveis para visita, em que esta distribuição é correspondente à real distribuição dos monumentos em Ávila, sendo que a escala utilizada de latitude e longitude foi adaptada correspondendo o (0,0) no mapa da Figura 3.19 a (40.65,-4.72) em coordenadas reais de Ávila.

```

ordem_optimizada =
    4    9    7    10

duracao_optimizada =
03:45:00

duration_opt =
    225

```

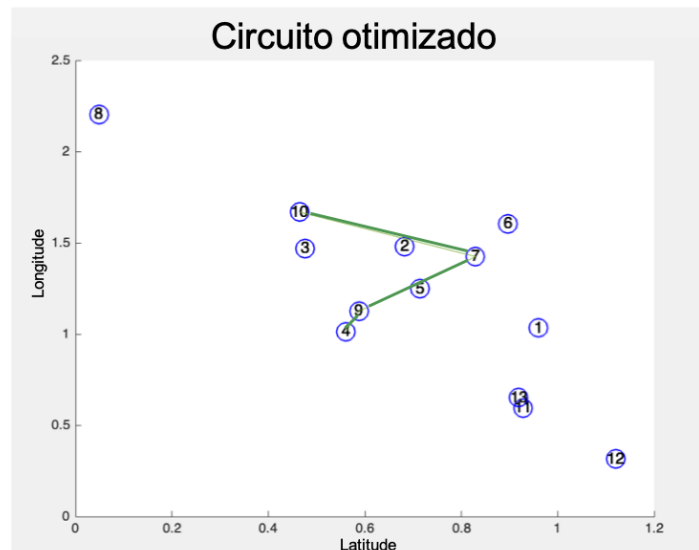


Figura 3.19 - Simulação da finalização da otimização.

Finalizado o desenvolvimento e simulação do algoritmo genético em ambiente de simulação, procedeu-se à implementação do algoritmo de otimização na aplicação móvel SHCity-Turista. A plataforma *Outsystems*, na qual está a ser desenvolvida a aplicação móvel permite o uso de programação através de blocos lógicos, o que dificulta a implementação do algoritmo genético desenvolvido.

Como referido anteriormente a utilização da API da *Google Maps* restringe a 8 o número de monumentos a incluir no circuito otimizado. Devido às limitações da plataforma *Outsystems* os tempos de espera utilizados no algoritmo da aplicação móvel são apenas aqueles a quando do início do circuito, não tendo em conta as previsões de tempo de espera futuras dos diferentes monumentos/lugares como acontecia no algoritmo genético desenvolvido em ambiente de simulação. Os blocos disponíveis para o desenvolvimento do algoritmo são os de condições *if/else* e os de ciclos *for/while*, os quais foram utilizados na implementação do algoritmo de otimização que se encontra explanado pelo pseudo-código do Algoritmo 3.7

Algoritmo 3.7 - Pseudo-código do algoritmo de otimização da aplicação móvel.

```

Para cada monumento/lugar
    Se Tempo de espera <= Tempo de espera dos restantes monumentos/lugares
        Este será o primeiro monumento/lugar do circuito
    Fim
Para cada monumento/lugar restante
    Verificar a (duração + tempo de espera + tempo de visita) entre o monumento/lugar
    e todos os restantes monumentos/lugares
    Escolher aquele que tem o menor tempo (duração + tempo de espera +
    tempo de visita)
    Assim sucessivamente até chegarmos ao último monumento/lugar
Fim

```

Esta implementação na plataforma *Outsystems* foi feita como exemplificado na Figura 3.20, neste caso para 3 monumentos/lugares escolhidos.

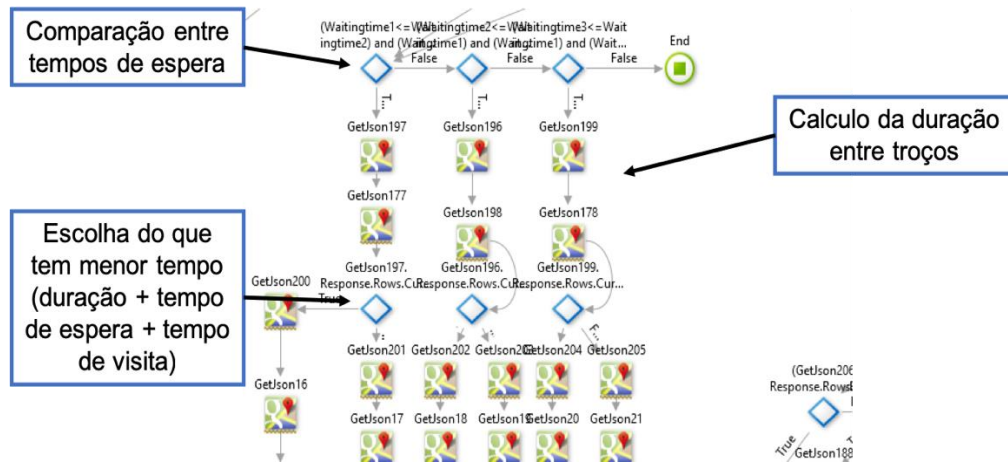


Figura 3.20 - Implementação do algoritmo de otimização em *Outsystems*.

3.4 Síntese

Em suma, neste capítulo foi feita uma descrição da arquitetura da aplicação e do seu algoritmo de otimização. Foi também feita uma descrição detalhada da implementação de todos os ecrãs da aplicação móvel SHCity-Turista, a qual foi feita utilizando a plataforma *Outsystems*, e da interligação entre todos os seus ecrãs. Foi apresentado o desenvolvimento em ambiente de simulação do algoritmo genético que traduz a formulação do problema analisado no âmbito desta dissertação. Devido às limitações da plataforma *Outsystems* a otimização de percursos teve que ser implementada recorrendo a blocos lógicos limitando assim as funcionalidades do mesmo.

Validação e Testes da Aplicação

Neste capítulo precede-se à apresentação das simulações e respetivos resultados do algoritmo genético desenvolvido em ambiente de simulação. São também apresentados os testes e resultados da aplicação móvel especificamente do algoritmo implementado. Por fim é exposta a validação da aplicação móvel SHCity-Turista por parte dos utilizadores através de um questionário anónimo.

4.1 Simulações em Ambiente de Simulação

Como já referido na fase de implementação de percursos fez-se uma divisão dos tempos de espera em 3 intervalos de tempo, sendo o primeiro das nove horas ao meio dia, o segundo do meio dia às dezasseis horas e o último das dezasseis às vinte horas, onde em cada um desses intervalos se encontra a média dos tempos de espera a cada hora até à data de criação do circuito otimizado. Pode-se observar no gráfico da Figura 4.1 a variação dos tempos médios de espera ao longo do dia para o segundo monumento, que corresponde à *Catedral y Plaza de la Catedral*, sendo que a distribuição dos tempos de espera é semelhante nos restantes monumentos/lugares.

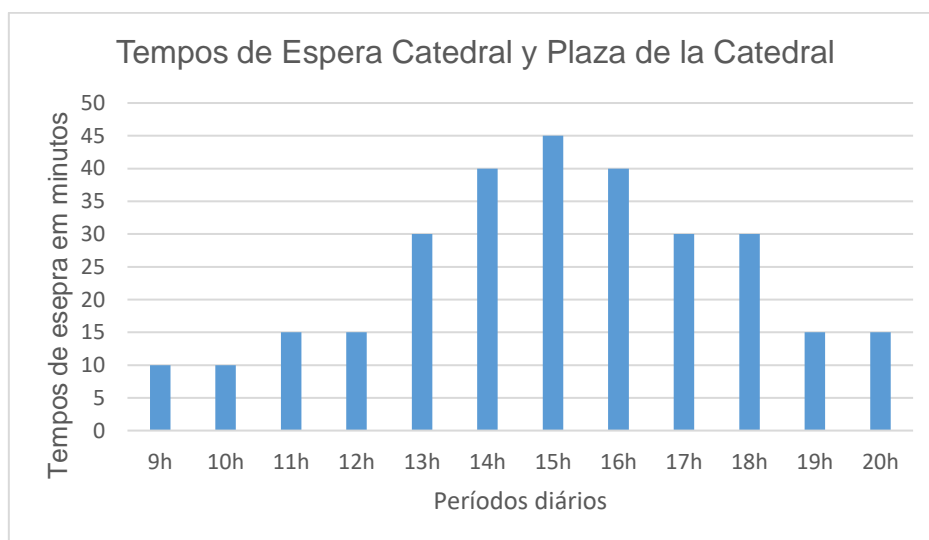


Figura 4.1 - Tempos médios de espera para a *Catedral y Plaza de la Catedral*.

Na Tabela 4.1 podemos observar os tempos médios de espera em minutos para cada intervalo de tempo definido para os monumentos 2, 8, 11 e 9, os quais vão ser usados para as simulações.

Tabela 4.1 - Tempos médios de espera dos monumentos 2, 8, 9 e 11 utilizados na simulação.

	9h-12h	12h-16h	16h-20h
<i>Catedral Y Plaza De La Catedral (2)</i>	14 minutos	38 minutos	26 minutos
<i>Real Monastero de Santo Tomás (8)</i>	12 minutos	50 minutos	20 minutos
<i>Iglesia de San Segundo (11)</i>	20 minutos	65 minutos	30 minutos
<i>Palacio Superunda (9)</i>	15 minutos	42 minutos	24 minutos

Estes monumentos têm como tempo médio de visita os valores apresentados na Tabela 4.2.

Tabela 4.2 - Tempos médios de visita dos monumentos 2, 8, 11 e 9.

Monumentos	Tempo Médio de Visita
<i>Catedral Y Plaza De La Catedral (2)</i>	50 minutos
<i>Real Monastero de Santo Tomás (8)</i>	70 minutos
<i>Iglesia de San Segundo (11)</i>	30 minutos
<i>Palacio Superunda (9)</i>	45 minutos

Começou-se por simular qual o tempo total de visita dos monumentos 2, 8, 11 e 9 sem qualquer tipo de otimização de percurso, ou seja, a ordem dos monumentos não sofre qualquer alteração obtendo se assim o percurso representado na Figura 4.2.

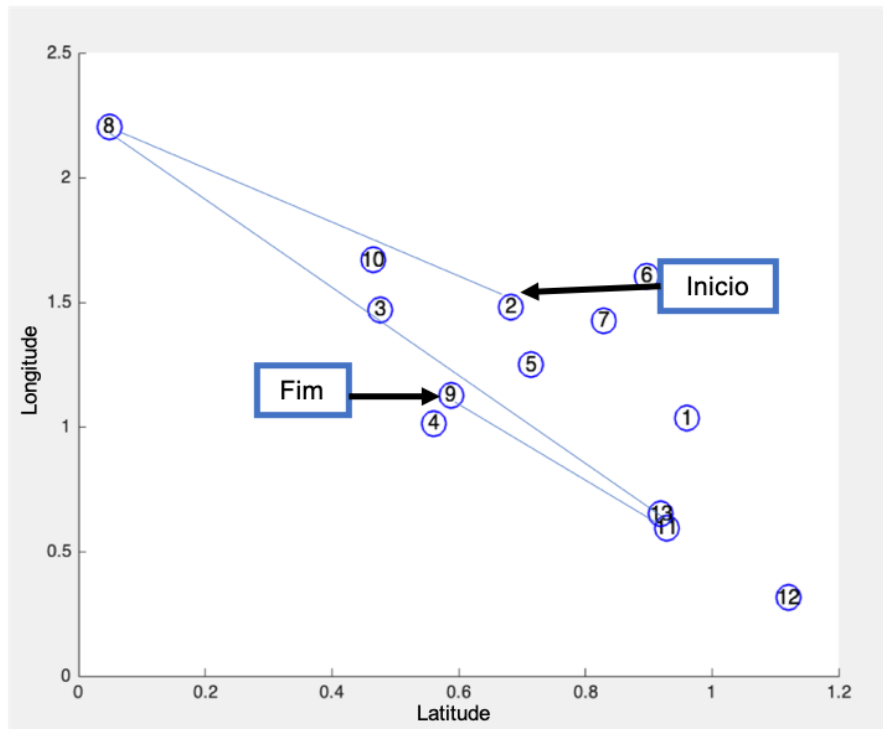


Figura 4.2- Circuito não otimizado obtido.

Foram realizadas 3 simulações, com início a horas diferentes, o primeiro às 10h, o segundo às 14h e o terceiro às 16h obtendo se assim as durações totais do circuito apresentados na Tabela 4.3.

Tabela 4.3 - Duração total obtida para o circuito não otimizado.

Hora de Início do Circuito	Duração do Circuito não Otimizado
10h	6h19m
13h	6h28m
16h	5h46m

Pela análise da tabela acima verificamos que iniciando o circuito as 14 horas obtemos uma duração total consideravelmente maior do que nos restantes casos o que é esperado visto este período corresponder aquele onde os tempos de espera são mais elevados.

Utilizando agora o algoritmo de otimização implementado em ambiente de simulação foram realizadas 3 simulações distintas. A primeira simulação com início às 10 horas e selecionando os monumentos 2, 8, 11 e 9 tem como circuito otimizado o representado na Figura 4.3.

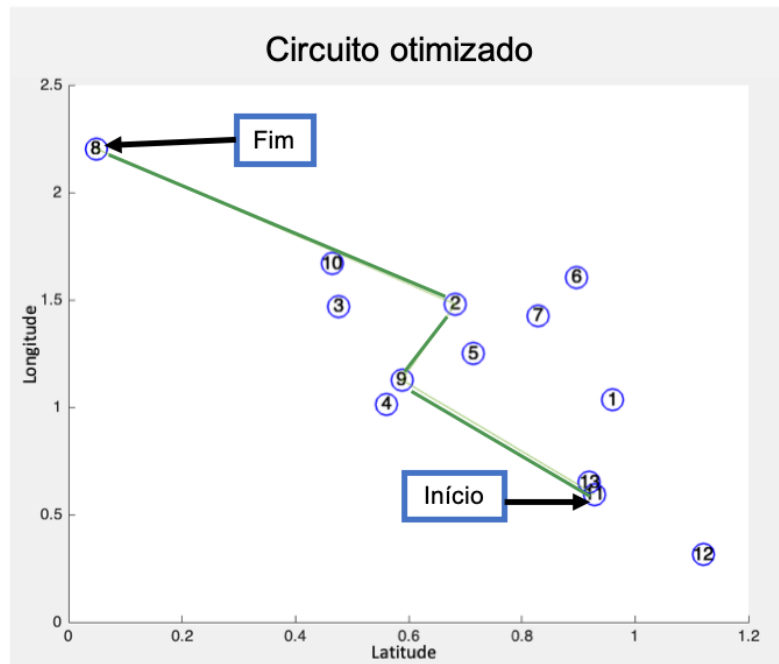


Figura 4.3- Circuito otimizado com início às 10 horas obtido.

A ordem obtida para a simulação da Figura 4.3 com início às 10 horas foi iniciar o circuito no monumento 11 seguido do 9, 2 e finalizar no 8 obtendo uma duração total de 5 horas e 46 minutos.

De seguida fez-se a simulação em que se inicia o circuito às 13 horas obtendo-se o circuito da Figura 4.4, com início no monumento 8, seguido pelo 2, 9 e a finalizar no 11 com uma duração total de 6 horas e 5 minutos.

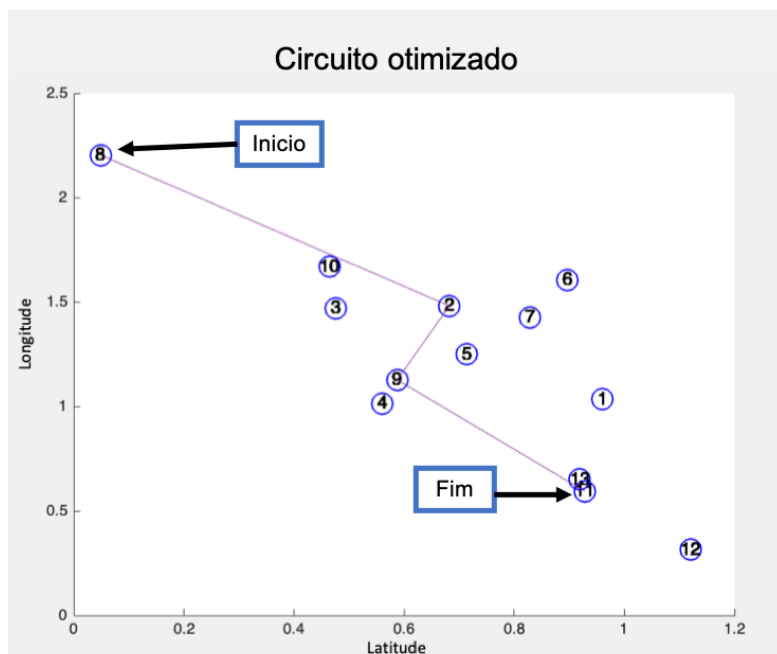


Figura 4.4- Circuito otimizado com início às 13 horas obtido.

Por último fez-se a simulação com início do circuito às 16 horas obtendo-se igualmente o circuito da Figura 4.3 com uma duração de 5 horas e 23 minutos.

Concluídas as simulações em ambiente de simulação podemos verificar na Tabela 4.4 a duração total dos circuitos otimizados para a seleção de monumentos 2, 8, 11 e 9 com início em horários diferentes.

Tabela 4.4 -Duração total obtida para os circuitos otimizados.

Hora de Início do Circuito	Duração do Circuito Otimizado
10h	5h46m
13h	6h05m
16h	5h23m

Verifica-se que tal como acontecia no circuito não otimizado, o período das 13 horas é aquele que tem o circuito com maior duração, o que é espectável visto os tempos de espera serem mais elevados neste período. Pode-se verificar pela análise do gráfico da Figura 4.5 que em todos os horários escolhidos verificou-se uma otimização da duração total do circuito tal como era pretendido.

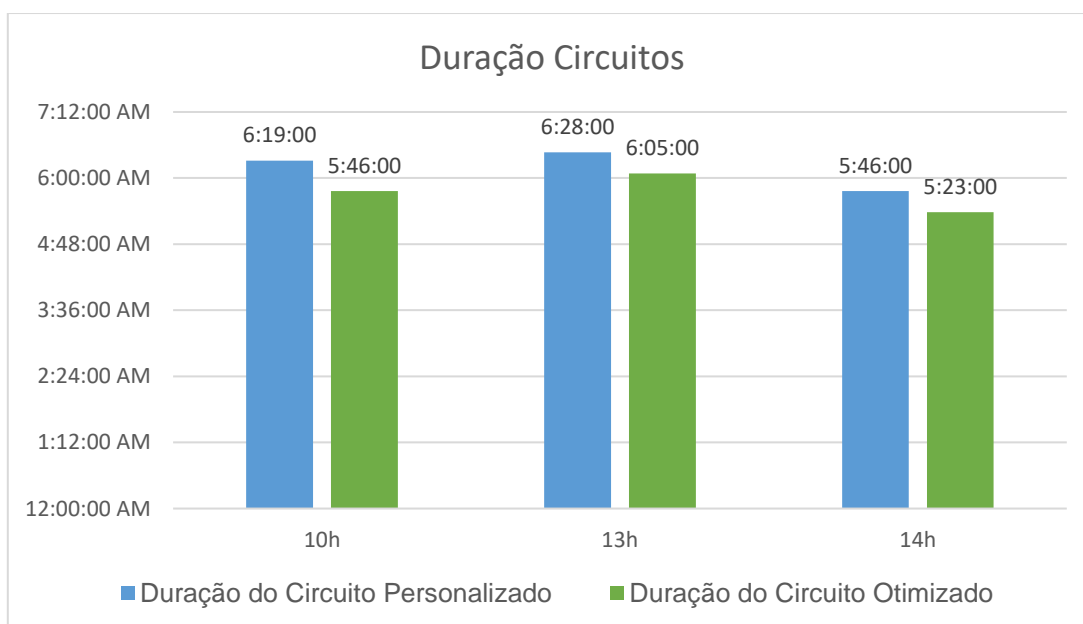


Figura 4.5 - Comparação entre a duração dos circuitos não otimizados e otimizados.

4.2 Testes da Aplicação

Como explanado na fase de metodologia e implementação, a aplicação móvel SHCity-Turista, tem como principal objetivo melhorar a experiência do utilizador na visita à cidade de Ávila, através de um conjunto de funcionalidade em que a principal é a otimização de circuitos tendo em conta o tempo de visita, duração e informação em tempo real acerca do tempo de espera em cada um dos monumentos/lugares.

Começou-se por fazer o teste sem otimização do percurso, ou seja, a ordem dos monumentos mantém-se a escolhida pelo utilizador apenas variando a hora de início do circuito em que neste caso para efeitos de simulação foram utilizadas tal como nos testes em ambiente de simulação 3 diferentes horas de início, sendo elas as 10 horas, as 13 horas e as 16 horas. Selecionaram-se os mesmos monumentos utilizados no teste em ambiente de simulação. Neste caso do teste da aplicação móvel, como já referido na fase de implementação devido às limitações da plataforma *Outsystems* os tempos de espera utilizados no algoritmo da aplicação móvel são apenas aqueles a quando do início do circuito, não tendo em conta as previsões de tempo de espera futuras dos diferentes monumentos/lugares. Os tempos de espera utilizados para cada monumento e para as diferentes horas de início do circuito encontram-se apresentados na Tabela 4.5. Os tempos médios de visita destes monumentos utilizados para o teste da aplicação móvel são os mesmos que os tempos utilizados na simulação em ambiente de simulação que se encontram apresentados na Tabela 4.2.

Tabela 4.5 - Tempos de espera dos monumentos utilizados no algoritmo de otimização da aplicação móvel.

	10h	13h	16h
<i>Catedral Y Plaza De La Catedral</i>	10 minutos	35 minutos	30 minutos
<i>Real Monastero de Santo Tomás</i>	15 minutos	45 minutos	18 minutos
<i>Iglesia de San Segundo</i>	24 minutos	60 minutos	23 minutos
<i>Palacio Superunda</i>	11 minutos	50 minutos	20 minutos

Para estes testes utilizou-se a opção de visita personalizada em que o utilizador escolhe os monumentos pela ordem de visita pretendida em que o circuito personalizado obtido foi o representado na Figura 4.6

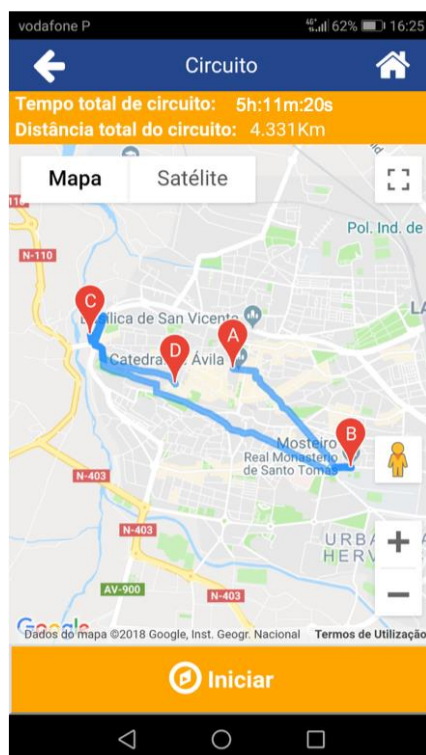


Figura 4.6 - Circuito personalizado obtido na aplicação móvel.

A duração deste circuito personalizado para os diferentes horários de início encontra-se apresentada na Tabela 4.6.

Tabela 4.6 - Durações obtidas do circuito personalizado da aplicação móvel.

Hora de Início do Circuito	Duração do Circuito Personalizado
10h	5h11m
13h	7h21m
16h	5h42m

Verifica-se uma grande variação entre iniciar o circuito às 10 horas ou às 16 horas e às 13 horas, esta variação na duração do circuito deve-se a grande afluência existente nos monumentos as 13 horas comparativamente aos restantes casos. O facto de não existir otimização na ordem dos monumentos, faz com que seja perdido uma grande quantidade de tempo no percurso entre monumentos como se pode observar pela análise da Figura 4.6 em que a ordem obtida não é a mais eficaz resultando assim numa distância total a percorrer de 4,3 quilómetros.

Concluídos os testes do circuito personalizado, testou-se o circuito otimizado em que foram escolhidos os mesmos 4 monumentos como apresentado na Figura 4.7 e feitas 3 simulações com

diferentes horas de início, tal como nas simulações anteriores com início às 10 horas, 13 horas e as 16 horas. Os tempos de espera e visita utilizados foram os mesmos que os utilizados na simulação do circuito personalizado que se encontram respetivamente na Tabela 4.5 e na Tabela 4.2.



Figura 4.7 - Seleção dos monumentos a incluir no circuito otimizado.

Para o primeiro teste com início as 10 horas o circuito otimizado obtido através do algoritmo de otimização implementado na aplicação móvel foi o representado na Figura 4.8, em que na parte superior da figura podemos verificar a duração obtida que foi de 4 horas e 49 minutos e uma distância total de 2,5 quilómetros. A ordem obtida para este circuito otimizado foi iniciar a visita na *Iglesia de San Segundo* de seguida visitar o *Palacio Superunda* e a *Catedral y Plaza de la Catedral* e por fim o *Real Monastero de Santo Tomás*.



Figura 4.8 - Circuito otimizado obtido na aplicação móvel.

Relativamente ao teste com início às 13 horas a ordem obtida para os monumentos foi a mesma que da simulação anterior que está representada na Figura 4.8, em que a distância total do circuito se manteve e a duração total obtida foi de 6 horas e 45 minutos. O teste das 16 horas, tal como no caso anterior manteve-se a ordem e a distância total sendo que neste caso a duração total obtida foi de 5 horas e 20 minutos. Os resultados destes testes encontram-se resumidos na Tabela 4.7

Tabela 4.7 - Durações obtidas do circuito otimizado da aplicação móvel.

Hora de Início do Circuito	Duração do Circuito Otimizado
10h	4h49m
13h	6h45m
16h	5h20m

Pela análise do gráfico da Figura 4.9 verifica-se que o algoritmo de otimização da aplicação móvel funcionou como esperado, otimizando a ordem de visita dos monumentos e consequentemente a duração e distância total do circuito, verificando-se uma diminuição de cerca de 2 quilómetros na distância total a percorrer entre o circuito personalizado e o otimizado.

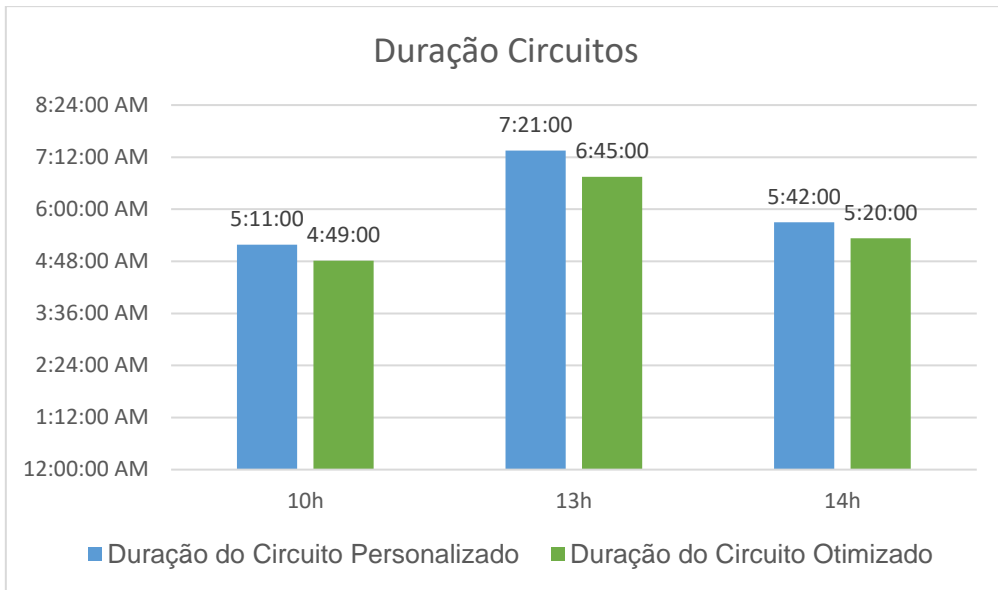



Figura 4.9 - Comparação entre a duração dos circuitos personalizados e otimizados obtidos na aplicação móvel.

4.3 Validação por Parte dos Utilizadores

De modo a saber qual a opinião dos utilizadores acerca da aplicação móvel SHCity-Turista foi realizado um formulário como o da Figura 4.10, em que primeiramente caso ainda não possuam a aplicação é possível descarregar a mesma através do código QR.

QR Code do download the App:



User Friendly *

1 2 3 4 5

Very complex to use Very easy to use

Number of Functionalities *

1 2 3 4 5

Misses a lot of functionalities It has all necessary functionalities

Visual Aspect *

1 2 3 4 5

Very Poor Very Good

Is the App Useful? *

Yes

No

Comments

A sua resposta

Figura 4.10 - Formulário de validação por parte dos utilizadores

Neste formulário são apresentadas uma série de questões acerca de diversos aspetos da aplicação móvel, nomeadamente o seu grau de complexidade em que 1 representa um elevado grau de complexidade e 5 uma aplicação bastante intuitiva e fácil de usar, as suas funcionalidades em que 1 representa uma elevada falta de funcionalidades e 5 uma aplicação completa em termos de funcionalidades e o aspeto visual dos diferentes ecrãs em que 1 representa uma fraca apresentação visual e 5 uma excelente apresentação visual. Uma outra questão colocada é relativamente à sua usabilidade, ou seja, se a aplicação é de facto útil ou não na visita à cidade de Ávila. Por fim podem ser inseridos comentários e sugestões relativos à aplicação móvel.

Os resultados deste formulário, ao qual um total de 20 utilizadores responderam, encontram-se apresentados nos gráficos da Figura 4.11 e da Figura 4.12.

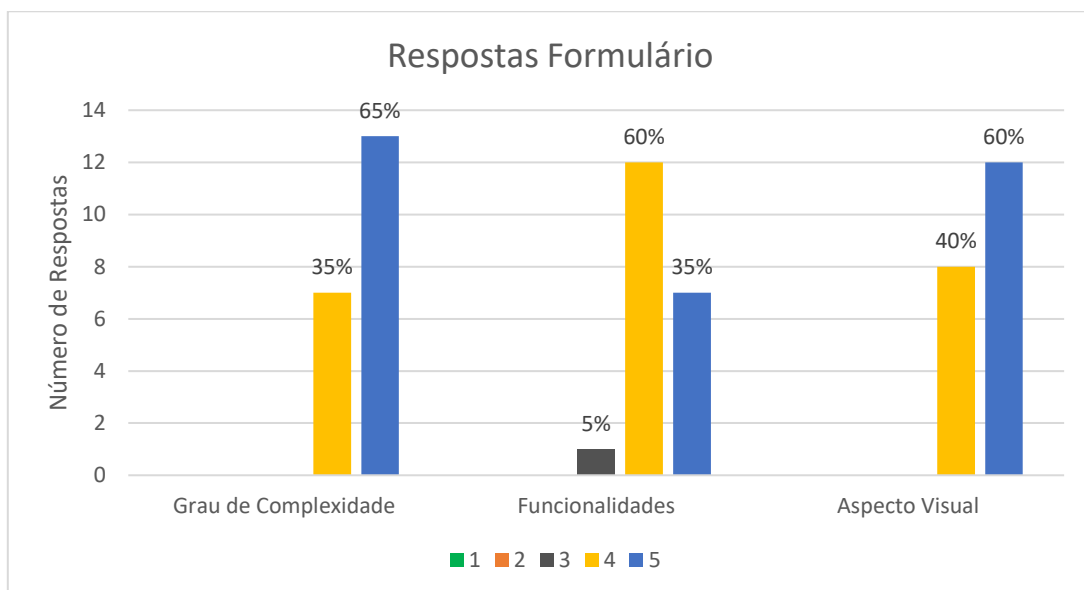


Figura 4.11 - Resultados das questões do formulário.



Figura 4.12 - Resultado da questão relativa à usabilidade.

Verifica-se pela análise da Figura 4.11 que de um modo geral os utilizadores consideram a aplicação móvel intuitiva, relativamente às funcionalidades da mesma consideram que é relativamente completa, sendo que existem algumas funcionalidades que podem ser adicionadas de modo a melhorar a aplicação móvel, como a partilha de circuitos com outros utilizadores e a inclusão na lista de circuitos de informação acerca da acessibilidade dos monumentos/lugares. As respostas relativas ao aspeto positivo foram bastante positivas, os utilizadores consideram a aplicação móvel SHCity-Turista visualmente atraente. Pela análise da Figura 4.12 conclui-se que todos os utilizadores consideram que é uma ferramenta útil na visita a Ávila, melhorando assim a experiência de visita do turista.

De seguida estão apresentados alguns comentários deixados pelos utilizadores:

- “Muito boa aplicação móvel, definitivamente vai ajudar os turistas na visita a Ávila.”
- “Uma aplicação muito útil que nos permite passear pela cidade de Ávila sempre com a informação de onde nos encontramos e para onde nos devemos deslocar. A parte da informação em tempo real permite uma melhor visita pela cidade num curto espaço de tempo.”
- “Em relação às funcionalidades, penso que seria positivo o utilizador poder guardar e partilhar os circuitos criados.”
- “Muito boa aplicação. Fácil e intuitiva de usar. Tem bastantes funcionalidades e está muito bem conseguida.”
- “Na opção criar circuito, podia ter o tempo de visita e o tempo de espera. Na lista de circuitos podia ter se era acessível a cadeira de rodas (com um símbolo) ao lado do número de monumentos. Outros serviços: podia ter opção de procura (começar a escrever e aparecer, tipo o nome do hospital ou assim....”

4.4 Síntese

Os resultados apresentados neste capítulo relativamente aos testes efetuados ao algoritmo de otimização implementado primeiramente em ambiente de simulação e posteriormente na aplicação móvel são satisfatórios, verificando-se uma otimização no tempo de visita em todos os casos testados relativamente aos circuitos não otimizados. A otimização do algoritmo implementado em ambiente de simulação tem em conta as previsões futuras do tempo de espera tornando o assim mais preciso, o que não acontece no algoritmo de otimização implementado na aplicação móvel que tem em conta o tempo de espera no momento de criação do percurso.

Relativamente aos resultados do formulário relativo à aplicação móvel todos os utilizadores consideram que é útil no auxílio da visita à cidade de Ávila, apreciam também o aspeto visual, a utilização de informação em tempo real e o fácil e intuitivo uso da mesma. Em termos de funcionalidades pelos resultados obtidos conclui-se que de um modo geral possui as essências, sendo que pode haver melhoria neste aspeto na implementação de novas.

Conclusões e Trabalho Futuro

Neste capítulo são feitas as conclusões finais desta dissertação, nomeadamente acerca da problemática, metodologia utilizada, da implementação feita e dos resultados obtidos no desenvolvimento da aplicação móvel SHCity-Turista.

Estando o desenvolvimento da aplicação móvel inserida num projeto inovador, *Smart Heritage City*, é necessário identificar trabalho futuro importante relacionado com o tema aqui apresentado.

5.1 Conclusões

O crescimento do turismo, tanto ao nível do número de turistas como de infraestruturas criou a necessidade de desenvolver novas ferramentas tecnológicas de auxílio nas diferentes fases que envolvem toda a atividade turística.

Tendo esta dissertação como principal objetivo o desenvolvimento de uma aplicação móvel que melhore a experiência do utilizador durante a visita à cidade de Avila, Espanha, foi então criada a aplicação móvel SHCity-Turista. Esta aplicação diferencia-se das restantes, pelo facto de utilizar informação sobre o monumento/lugar recolhida, em tempo real, de uma rede de sensores instalada na cidade de Ávila. Entre as suas 5 funcionalidades a principal é a possibilidade de o utilizador poder criar os seus próprios circuitos, estando incorporado na aplicação um algoritmo evolucionário que tem como objetivo otimizar o tempo de visita do turista.

Utilizou-se a plataforma *Outsystems* para a implementação da aplicação móvel, incluindo todos os seus ecrãs e a interação entre os mesmos. Quanto à implementação do algoritmo genético começou por ser implementado utilizando um ambiente de simulação sendo, posteriormente, implementado na aplicação através da plataforma *Outsystems*. Na implementação do algoritmo na aplicação móvel foram encontradas algumas dificuldades devido às limitações da plataforma *Outsystems*.

Num panorama geral os resultados permitiram perceber que a principal funcionalidade, a otimização de percursos através de algoritmos evolucionários funciona como esperado, otimizando o tempo de visita do utilizador. Ao nível dos resultados apresentados nos testes do algoritmo implementado em ambiente de simulação, a otimização utiliza previsões futuras dos tempos de espera dos monumentos/lugares, caso que não acontece no algoritmo da aplicação móvel, que utiliza apenas

os tempos de espera no momento da criação do circuito, tornando os resultados obtidos em ambiente de simulação mais precisos.

Relativamente aos resultados do formulário relativo à aplicação móvel, todos os utilizadores consideram que é útil no auxílio da visita à cidade de Ávila, apreciam também o aspeto visual, a utilização de informação em tempo real e o fácil e intuitivo uso da mesma.

5.2 Trabalho futuro

O trabalho desenvolvido permitiu o desenvolvimento de uma aplicação com algumas funcionalidades essenciais para auxiliar o turista durante a visita à cidade de Ávila. No entanto, foram encontradas algumas funcionalidades que devem ser implementadas futuramente, nomeadamente a capacidade de guardar e partilhar com outros utilizadores os circuitos criados e a avaliação dos monumentos/lugares e circuitos pré-definidos existentes na aplicação.

Tendo em conta a limitação da implementação do algoritmo de otimização na aplicação móvel, é importante melhorar esta implementação de modo a ter em conta previsões futuras de tempos de espera, tal como acontece no algoritmo desenvolvido em ambiente de simulação.

Por último, como a utilização da versão livre dos mapas da Google limita o número de monumentos a incluir em um único circuito a 8, seria importante utilizar uma versão em que não existisse esta limitação

Bibliografia

- [1] INE, “Estatísticas do Turismo 2016,” 2017.
- [2] C. Hidalgo and O. Maene, “The Nature of Spain’s International Cultural Tourism throughout the Economic Crisis (2008–2016): A Macroeconomic Analysis of Tourist Arrivals and Spending,” *Economies*, vol. 5, no. 3, p. 32, 2017.
- [3] A. Mar, F. Monteiro, P. Pereira, and J. Martins, “An Application to Improve Smart Heritage City Experience,” in *Advances in Digital Cultural Heritage*, 2018, pp. 89–103.
- [4] M. Christian, “Mobile Application Development in the Tourism Industry and its Impact on On-Site Travel Behavior,” Modul Vienna University, 2015.
- [5] The Statistics Portal, “Most popular Apple App Store categories in June 2015, by share of available apps,” 2012. [Online]. Available: <http://www.statista.com/statistics/270291/popular-categories-in-the-app-store/>. [Accessed: 16-Apr-2018].
- [6] Google, “The Five Stages of Travel.” [Online]. Available: <http://www.thinkwithgoogle.com/insights/featured/five-stages-of-travel/>.
- [7] P. Lombardi, S. Giordano, H. Farouh, and W. Yousef, “Modelling the smart city performance,” *Innovation*, vol. 25, no. 2, pp. 137–149, 2012.
- [8] J. Oliveira *et al.*, “FindTrip : Um Aplicativo Móvel para o Turismo Inteligente na Cidade do Natal,” *Univ. Fed. do Rio Gd. do Norte – UFRN*, 2014.
- [9] Google, “Google Play.” [Online]. Available: <https://play.google.com/store>.
- [10] Smartappcity, “SMARTAPPCITY.” [Online]. Available: <http://smartappcity.com/en/>.
- [11] Audio Trails 2018, “audiotrails.co.uk.” [Online]. Available: <http://audiotrails.co.uk/welcome-native-app-platform/>.
- [12] ETAG (Edinburgh Tourism Action Group), “Business Opportunities,” *Edinburgh Tour. Action Gr. / Bus. Oppor. Edinburgh – A World Herit. City*, vol. 1, no. 1, pp. 1–26, 2014.
- [13] Edinburgh World Heritage, “Edinburgh - World Heritage City App.” [Online]. Available: <https://www.ewht.org.uk/visit/edinburgh---world-heritage-city-app>.
- [14] V. Norwich, “heritagecity.” [Online]. Available: <http://www.heritagecity.org/about-us/norwich-heritage.htm>.

- [15] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A Survey on Evolutionary Computation Approaches to Feature Selection," *IEEE Trans. Evol. Comput.*, vol. 20, no. 4, pp. 606–626, 2016.
- [16] Q. Cai, L. Ma, M. Gong, and D. Tian, "A survey on network community detection based on evolutionary computation," *Int. J. Bio-Inspired Comput.*, vol. 8, no. 2, p. 84, 2016.
- [17] D. W. Boeringer and D. H. Werner, "Particle Swarm Optimization Versus Genetic Algorithms for Phased Array Synthesis," *IEEE Trans. Antennas Propag.*, vol. 52, no. 3, 2004.
- [18] E. Elbeltagi, T. Hegazy, and D. Grierson, "Comparison among five evolutionary-based optimization algorithms," *Adv. Eng. Informatics*, vol. 19, no. 1, pp. 43–53, 2005.
- [19] Felix Streichert, "Introduction to Evolutionary Algorithms," *Evol. Algorithms Mol. Des.*, vol. 8, pp. 1–13, 2008.
- [20] P. Gil and F. Januário, "Introdução às Técnicas de Computação Evolutiva," *Doc. apoio à Discip. Comput. Evol.*, 2015.
- [21] S. N. Sivanandam and S. N. Deepa, *Introduction to genetic algorithms*, 1st ed. Springer-Verlag Berlin Heidelberg, 2008.
- [22] T. Bäck and H.-P. Schwefel, "An Overview of Evolutionary Algorithms for Parameter Optimization," *Evol. Comput.*, vol. 1, no. 1, pp. 1–23, 1993.
- [23] I. Griffin and P. J. Fleming, "Multi-objective evolutionary computing solutions for control and system identification," in *Intelligent Control Systems using Computational Intelligence Techniques*, 2005, p. Chapter 3.

Anexos

Smart Tourism Routes Based on Real Time Data and Evolutionary Algorithms

Mário Amorim^{1*} ✉^[0000-0002-6337-2915], Adriana Mar¹ ^[0000-0003-4773-6819], Fernando Monteiro¹, Stella Sylaiou², Pedro Pereira¹ ^[0000-0002-0548-2091], João Martins¹

¹ CTS - UNINOVA
Department of Electrical Engineering
Faculty of Science and Technology
Universidade NOVA de Lisboa
2829-516 Caparica, Portugal

² School of Social Sciences,
Hellenic Open University, Greece

{m.amorim, am.jesus}@campus.fct.unl.pt,
fernando.j.c.c.monteiro@gmail.com, sylaiou@gmail.com,
{pmrp, jf.martins}@fct.unl.pt

Abstract. Tourism is an industry that has been growing rapidly in the last few years and it is expected that it will continue to grow. Due to the evolution of technology, mobile applications are being increasingly used in all kinds of industries, being one of them tourism. Presently there are already a few mobile applications used to increase the experience of the user when visiting a place, but these mobile applications lack some important features. This paper describes the development of a mobile application with integrated routing algorithms used to increase the experience of the tourists when visiting the city of Avila, Spain. The tourist will have at their disposal real time information about all the monuments available for visit, a full set of predefined circuits with different visit times and degrees of difficulty and also the possibility to create an optimized or personalized circuit combining the user preferences such as visiting time and number of monuments to visit.

Keywords: Tourism, Mobile applications, Routing algorithm, Real time data, Optimized route

1 Introduction

Tourism is one of the branches that has been growing more rapidly both in the number of tourists and in terms of tourist infrastructure. This growth has created the need for greater investment by cities not only in the restoration and preservation of monuments but also in the creation and improvement of support infrastructures for tourists [1].

Smart Heritage Management

The SHCity Project Approach

Fernando Monteiro
CTS-UNINOVA /FCT NOVA
Portugal
fernando.j.c.c.monteiro@gmail.com

Adriana Mar
CTS-UNINOVA /FCT NOVA
Portugal
am.jesus@campus.fct.unl.pt

Mário Amorim
FCT NOVA
Portugal
m.amorim@campus.fct.unl.pt

Rosa Ruiz Entrecanales
Ayuntamiento de Ávila
Unidade de Patrimonio de Ávila
Spain

Aurélien Henon
NOBATEK/INEF4
France

Susana San José Alonso
Centro Tecnológico CARTIF
Spain

Alessandra Gandini
Centro Tecnológico
TECNALIA
Spain

Rosa Pérez
Instituto Tecnológico AIDIMME
Spain

Miguel Ángel Abián
Instituto Tecnológico AIDIMME
Spain

José Carlos García
Fundación Santa María la Real del
Patrimonio Histórico
Spain

Daniel Basulto García-Risco
Fundación Santa María la Real del
Patrimonio Histórico
Spain

Abstract— Smart Heritage City (SHCity) is a European cooperation project which main challenge is to develop a technological solution aimed at improving the management of historic urban areas, aiding in their maintenance, conservation, energy saving, tourist use and decision making. SHCity implements a 4.0 tool in urban sets and an application focused on tourism management, which extends the work of the monitoring project SHBuildings. The technology is tested in Ávila, Spain, which is a historic World Heritage Site and whose example can be replicated to other historical sites in Europe.

Keywords— smart city, monitoring, 4.0 management, heritage buildings, rehabilitation.

I. INTRODUCTION

SHCity project is based on the installation of a network of sensors, deployed in different strategic points of the historical set of Ávila, which measure in real time different environmental, structural parameters, as well as others related to safety, energy consumption or the flow of visitors. The information provided will be used by two independent software tools: one for heritage managers, that will generate warnings, automatic responses to certain events and give recommendations for the decision making, and one for tourists, a mobile application focusing on smart tourism management.

As tourism is one of the fastest growing economic area both in the number of tourists and in terms of tourist infrastructure, this growth has created the need for greater investment by cities not only in the restoration and preservation of monuments but

also in the creation and improvement of support infrastructures for tourists [1].

In Europe, tourist activity is a source of considerable economic income that has been increasing in recent years. In Portugal, in 2016, there was a 1.4% increase in GDP (Gross Domestic Product) as shown in [2], representing one of the largest sources of employment in major European cities. In [3] it is shown that in Spain, between 2014 and 2015, there was an increase of around 13% in the number of jobs linked to tourism.

With the increase in the number of tourists and infrastructures, it is difficult for tourists to decide which tourist spots to visit and to collect information on these points due to the wide variety of sources. On the other hand, the available information about the main tourist attractions is not real-time information often finding itself outdated.

Also, municipalities have logistical problems in infrastructure control such as the number of daily visitors, occupancy rate and waiting times, among other factors, where real-time information is currently non-existent. Another problem encountered is the high number of platforms that allow the sharing of information about tourist attractions, making it increasingly difficult to concentrate the information in a single place accessible to all tourists [4].

The result of the project seeks to facilitate the decision-making process around the preservation of the historic area, its maintenance and sustainable tourism development, optimizing the work of the entities responsible for its management.