



Universidade do Minho

Escola de Engenharia

Licenciatura em Engenharia informática

Mestrado Integrado em Engenharia Informática

Unidade Curricular de Laboratórios de Informática IV

Ano Lectivo de 2021/2022

Where2Go

David Duarte A93253

Ema Dias A89518

Samuel Lira A94166

Sara Queirós A89491

Grupo 23

<<Novembro, 2021/2022>>

LI4

16

Data de Recepção	
Responsável	
Avaliação	
Observações	

Resumo

Na disciplina de Laboratórios de Informática IV, os alunos são incentivados a praticar a construção de software com metodologias e processos de Engenharia de Software, permitindo um maior planeamento e, conseqüentemente, validação de toda a estrutura que um projeto deverá conter. No caso, pretende-se aproximar os mesmos a desenvolver aplicações do mundo real, pois são estas que terão impacto na vida profissional que se aproxima do futuro. Desse modo, alerta-se para a necessidade de uma boa estruturação, esquematização e organização quando se pretende alcançar um projeto coeso e sem falhas.

Como alunos da Unidade Curricular, foram-nos providenciados alguns objetivos que devem ser atingidos com o desenvolvimento do projeto, tais como aprendizagem dos conceitos de Engenharia de Software e métodos de trabalho mais relevantes referentes a desenvolvimento de Software, orientar para a necessidade de estabelecer prazos de entrega e apresentação, e treinar o trabalho de equipa com acompanhamento de projeto, realçando a necessidade de pesquisa e escrita de relatórios completos e explicativos.

No caso, foi lançado o desafio de desenvolvermos um projeto de Software que permita a um utilizador saber quais os locais de interesse perto de si, sendo possível pedir um trajeto para um dos mesmos, concretizando desta forma um Guia de Locais de Interesse.

Área de Aplicação: << Arquitetura de bases de dados, Guia de Locais, construção de trajetos >>

Palavras-Chave: << Base de dados, Locais de Interesse, Aplicação do mundo real >>

Índice

Resumo	3
Índice	4
Índice de Figuras	6
Índice de Tabelas	7
Capítulo 1: Introdução	8
Contextualização	8
Apresentação do Caso de Estudo	9
Motivação e Objectivos	10
Justificação e utilidade do sistema	10
Estabelecimento da identidade do projeto	11
Identificação dos recursos necessários	11
Análise de Viabilidade	12
Modelo do Sistema a Implementar - Maqueta	13
Definição de Medidas de Sucesso	15
Plano de Desenvolvimento - Diagrama de Gantt	20
Capítulo 2: Levantamento e Análise de Requisitos	24
Apresentação da estratégia e método	24
Descrição geral dos requisitos levantados	25
Requisitos não funcionais	27
Validação dos requisitos estabelecidos	28
Capítulo 3: Especificação e Modelação do Software	28
Apresentação geral da especificação	28
Aspetos estruturais: Diagrama de Classes	28
Aspetos estruturais: Diagrama de Packages	30
Aspetos Comportamentais:	31
Modelo de Domínio	31
Diagrama de Use Cases	31
Especificação de Uses Cases.	33
Consultar Lista de Pontos de Interesse:	33
Pedir Sugestão de cidades:	33
Escolher local:	33
Pedir direções:	34
Cancelar pedido de direções:	34
Confirmar chegada:	34
Pedir ajuda:	35
Avaliar local.	35
Consultar histórico.	36
Capítulo 4: Conceção do Sistema de Dados	36
Apresentação geral da estrutura do sistema de dados	36
Descrição dos elementos de dados e seus relacionamentos	37

Modelo Lógico	37
Modelo Físico	38
Capítulo 5: Esboço das interfaces do Sistema	40
Estrutura Geral das Interfaces do Sistema	40
Caracterização das interfaces	40
Capítulo 6: Conclusões e Trabalho Futuro	43
Siglas e Acrónimos	44
Referências Bibliográficas	45

Índice de Figuras

Fig.1 - Fases de um projeto	10
Fig.2 - Matriz de avaliação de riscos de um projeto	14
Fig.3 - Maqueta do Sistema	14
Fig.4 - Interação Sistema - Base de Dados	15
Fig.5 - Aplicação Where2Go	16
Fig.6 - Questionário de utilização do 'TripAdvisor'	17
Fig.7 - Questionário da utilidade do 'TripAdvisor'	17
Fig.8 - Questionário de tipo de utilização do 'TripAdvisor' para procura de restaurantes	18
Fig.9 - Questionário de tipo de utilização do 'TripAdvisor' para procura de hotéis	18
Fig.10 - Questionário de tipo de utilização do 'TripAdvisor' para procura de museus	19
Fig.11 - Questionário de tipo de utilização do 'TripAdvisor' para procura de espaços de lazer	19
Fig.12 - Questionário de tipo de utilização do 'TripAdvisor' para pesquisa de locais	20
Fig.13 - Questionário da utilização de tecnologias para viajar	20
Fig.14 - Diagrama de Gantt	23
Fig.15 - Diagrama de Gantt - Especificação	23
Fig.16 - Diagrama de Gantt - Construção	24
Fig.17 - Diagrama de Classes	30
Fig.18 - Diagrama de Packages	31
Fig.19 - Modelo de Domínio	32
Fig.20 - Diagrama de Use Cases	33
Fig.21 - Modelo Lógico	38
Fig.22 - Base de Dados da aplicação em SQL	40
Fig.23 - Mockup Página Login	42
Fig.24 - Mockup Página Principal	42
Fig.25 - Mockup Página de Resultados relativos a pedidos de POI	43
Fig.26 - Mockup Pedido Direções para um POI	43

Índice de Tabelas

Tabela 1 - Identidade do Projeto	11
Tabela 2 - Distribuição de Tarefas - Especificação.	21
Tabela 3 - Distribuição de Tarefas - Construção.	21
Tabela 4 - Requisitos Funcionais	27
Tabela 5 - Requisitos Não Funcionais	28

Capítulo 1: Introdução

Contextualização

Laboratórios de Informática IV é uma Unidade Curricular que pretende que se planeie, estruture e organize o funcionamento de uma aplicação do mundo real, no caso um Guia de Locais de Interesse, para que no final da UC seja elaborado esta aplicação de forma eficiente e limitando os erros de gestão e delineação de tempo, ou até mesmo erros da construção da mesma.

Desta forma, a equipa decidiu que para este projeto a aplicação seria desenvolvida para o país de Malta. Malta é uma país da Europa com pequena área geográfica, situando-se no mar Mediterrâneo, ao sul de Itália. É um país conhecido pelo seu teor turístico, onde se encontram vários locais históricos devido à grande sucessão de governantes.

Após ser conhecido o âmbito do projeto, é necessário estudar cuidadosamente como desenvolver uma aplicação. Como é referido no livro de apoio bibliográfico de António Miguel [2], os projetos de Software possuem um ciclo de vida, onde se inclui as fase de analisar os requisitos do software, arquitetar, desenhar e construir e integrar os elementos constituintes do mesmo e, por fim, testar os eventuais erros que possam ter ocorrido. De facto, o que se pretende reter desta UC é a necessidade de um bom planeamento para a construção do projeto e, para tal, é necessário recorrer ao conceito para detetar aquilo que é necessário efetuar. Da mesma forma, com apoio bibliográfico[2] percebemos que planear deduz as seguintes tarefas: identificar os resultados e marcos do projeto, desenvolver um plano de projeto e comunicá-lo entre a equipa, assim como determinar as necessidades de recursos, estimar prazos e eventuais custos, atribuir responsabilidades no projeto, avaliando as capacidades de cada elemento do grupo e suas aptidões e, finalmente estabelecer critérios para a aceitação/ validação do projeto.

Além do que já foi referido, é imprescindível gerir o projeto com avaliações contínuas e usar métodos sistemáticos de registo da situação do mesmo.

Assim, tal como é representado no livro as fases que um projeto contém são:

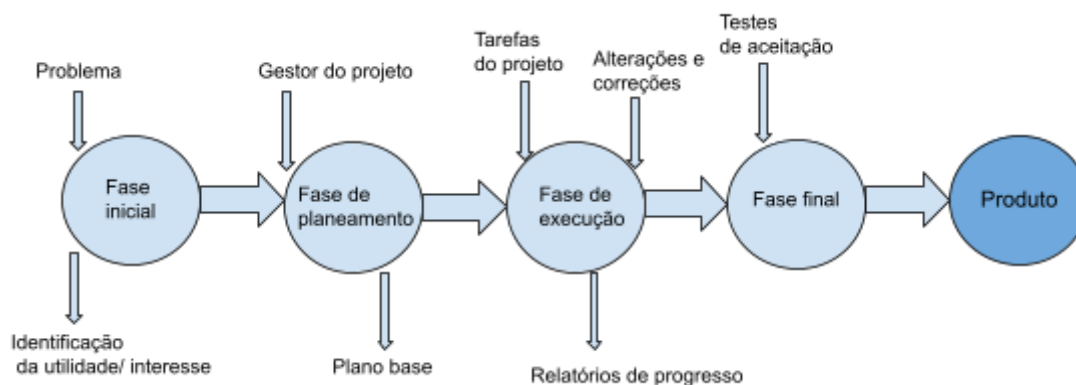


Fig. 1 - Fases de um projeto

Posto isto, é possível verificar e perceber quais são as etapas essenciais ao sucesso do desenvolvimento do projeto, permitindo, desta forma, concluir e apresentar um produto que seja um Guia de Locais de Interesse sem erros de planeamento e que seja válido no âmbito da Unidade Curricular.

Apresentação do Caso de Estudo

Como caso de estudo temos um Guia de Locais de Interesse em Malta, ou seja, pretende-se que se desenvolva um projeto onde seja possível um utilizador selecionar a cidade onde se encontra e lhe seja apresentada uma lista de locais de interesse, dentro do domínio de Malta.

Na realidade, é possível perceber que este Guia que vai ser desenvolvido é nada mais nada menos que uma lista de locais que são considerados interessantes para visita, tendo em conta o utilizador em causa.

Ao refletir, em grupo, o conceito de “Local de Interesse”, conclui-se que este pode ser dividido em várias categorias, variando de restaurantes e hotéis, até espaços de lazer e museus, sendo estes últimos, geralmente, os locais mais procurados em uma visita turística. Considera-se importante a existência das várias categorias, uma vez que produz uma maior utilidade da aplicação ao utilizador.

Para além disso, entendemos que o utilizador é aquele que seleciona o local que pretende visitar, de entre a lista de locais apresentados pela aplicação. Ele poderá selecionar a opção de calcular um trajeto que o leva do local onde se encontra ao local de interesse .

Motivação e Objectivos

Ao analisar a perspectiva de um turista comum, é notório que a falta de conhecimento geográfico básico sobre a cidade a ser visitada pode tornar-se um grande inconveniente. Como forma de resolver este problema, é criada esta aplicação, sendo que terá bastante utilidade no dia-a-dia de um turista mas também de uma pessoa comum. Isto deve-se ao facto de apresentar toda a variedade de restaurantes, hotéis e locais de lazer como jardins ou museus, existentes nas redondezas.

Assim, existe motivação para desenvolver um projeto que, no contexto da vida real, seria proveitoso, e que seria uma mais valia para o turismo, pois tornaria uma viagem mais segura, simples e com a possibilidade de calendarização de atividades, fazendo uma melhor gestão do tempo e de locais a visitar, sem ter de conhecer previamente a cidade. As visitas seriam menos complexas já que, a qualquer momento, estaria com o trajeto calculado para um local que seleccionasse, e poderia também perceber o que há de interesse numa dada cidade.

Pretendemos desse modo, atingir um bom planeamento que permita construir uma aplicação válida e estruturalmente correta no contexto em que se encontra inserida. Além disso, sabemos que a quantidade não é sinónimo de qualidade e, portanto, ter uma grande fonte de dados e uma má especificação não nos leva a um bom projeto. Assim, pretendemos alcançar uma boa estrutura que será aplicada a uma fonte de dados razoável, pois tendo uma boa fase de planeamento a fase de execução poderá ser mais tarde aplicada a qualquer fonte de dados.

Para além disso, Malta foi o país escolhido como objeto da nossa aplicação por se tratar de um local bastante procurado para o turismo devido à sua riqueza cultural, histórica e paisagística.

Justificação e utilidade do sistema

Este programa, tal como anteriormente referido, é realmente importante e útil nos dias atuais onde, raramente, são os momentos em que não se utiliza a tecnologia como recurso para a realização de uma viagem. De facto, já existem algumas aplicações que nos mostram e apresentam informações bastante similares às que temos como objetivo, como por exemplo o *Tripadvisor*, aplicação esta que tem um grande prestígio e sucesso.

Posto isto, sabendo que já existem aplicações do género que resultaram num grande êxito, concluímos, então, a evidente importância deste tipo de programas para o benefício e proveito da sociedade.

Se refletirmos sobre as últimas viagens que efetuamos, a probabilidade de ter sido feita uma breve pesquisa dos locais de interesse que a cidade tem é muito grande. Este tipo de pesquisa também poderia ter sido efetuada sobre a própria cidade onde vivemos, como forma de descobrir bons e/ou restaurantes e ainda espaços de lazer.

Por conseguinte, acredita-se que este projeto é bastante interessante no contexto diário de um utilizador e que possui uma forte importância nas deslocações e conhecimento de novos locais, rentabilizando e simplificando a experiência.

Estabelecimento da identidade do projeto

Nome	Where2Go
Categoria	Guia de Locais de Interesse
Idioma	Inglês
Faixa Etária	12+
Prazo	24 de Janeiro de 2022
Equipa	DESS - Software Development

Tabela 1 - Identidade do Projeto

Identificação dos recursos necessários

Relativamente aos recursos necessários iremos começar por abordar os humanos. Para realizar este projeto integramos uma equipa de 4 pessoas. Destas 4 pessoas, existem 4 tipos de atividades que as mesmas devem assumir. De facto, para o desenvolvimento do trabalho a que nos propusemos deverão ser necessário 2 engenheiros de Software, Sara e Samuel, que elaboram a arquitetura do mesmo, 2 programadores que irão executar o que foi planeado pelos engenheiros, Ema e David, dois analistas, Ema e Samuel, que verifica a viabilidade e os erros que poderão existir e, por fim, dois designers, Sara e David, que fica responsável pela parte exterior e visível ao utilizador. Sendo a equipa constituída por 4 elementos haverá elementos que terão que executar múltiplas tarefas.

Em termos materiais, recorreu-se a ferramentas tais como o GoogleDocs, GoogleSheets, GoogleForms, MicrosoftProject e ainda VisualParadigm. Além disso, foi utilizado o Adobe Illustrator e o art board studio, o primeiro para auxiliar o projeto visualmente através de imagens desenhadas pela equipa e o segundo para elaborar os mockups mais tarde realizados. Assim como, o MySQL para desenvolver e estruturar a base de dados.

Análise de Viabilidade

Mais uma vez, o livro de apoio bibliográfico oferece-nos uma visão bastante detalhada do que é necessário para efetuar um estudo de exequibilidade, explicando que nas aplicações do mundo real deve ser elaborado um relatório a apresentar à empresa ou cliente que requisitou o projeto.

Posto isto, aquilo que deve ser analisado para perceber se efetivamente um projeto é ou não viável, são os custos, os fatores de risco técnicos e avaliação de riscos, o modo como o sistema proposto leva à solução do problema apresentado, os riscos legais associados e a possibilidade de cumprir os prazos, tarefas e efetivamente existirem os recursos necessários.

De uma forma geral, é possível concluir que em termos de custos e riscos legais, sendo esta uma aplicação realizada em termos pedagógicos a justificação económica e análise legal não tem razão de ser elaborada, já que as ferramentas que serão utilizadas se encontram disponíveis de forma gratuita. Em relação aos riscos técnicos, acreditamos que com um planeamento bem executado, onde todas as hipóteses são analisadas e todos os problemas são ponderados, o risco técnico é reduzido de forma significativa. Já relativamente à resolução do projeto proposto, mais uma vez consideramos que cumprindo todas as fases que a Engenharia de Software nos apresenta e aproveitando o trabalho em equipa de forma benéfica, iremos conseguir atingir os objetivos a que nos propomos. Por fim, as datas de conclusão de projeto foram seriamente discutidas entre a equipa para que se pudesse estabelecer prazos reais e praticáveis, de maneira a realizar todas as tarefas com a duração necessária, sem conduzir a erros de má gestão de tempo.

Em suma, percebemos que a análise da exequibilidade de um projeto é de teor indispensável, pois permite-nos ponderar se todas as questões logísticas serão possíveis de realizar. No caso do projeto do Guia de Locais de Interesse, concluímos que possuímos todos os recursos necessários, incluindo um prazo de conclusão favorável, quando bem aproveitado e gerido o tempo de duração para todas as fases. Além disso, consideramos que todas as restantes cláusulas que permitem chegar ao sucesso do guia são seguras e exequíveis quando bem refletidas e quando bem esquematizado o software a desenvolver.

Por fim, tal como é apresentado no livro Gestão de Projetos de Software, apresentamos a seguinte tabela, que pensamos ser um bom auxílio visual, mostrando uma matriz de avaliação dos riscos de um projeto:

		Estrutura reduzida	Elevada Estrutura
Elevada familiaridade com a tecnologia ou área aplicacional	Grandes projetos	Risco médio	Risco baixo ou muito baixo
	Pequenos projetos	Risco médio	Risco baixo ou muito baixo
Reduzida familiaridade com a tecnologia ou área aplicacional	Grandes projetos	Risco elevado	Risco médio
	Pequenos projetos	Risco elevado	Risco médio

Risco baixo ou muito baixo

Risco médio

Risco elevado

Fig. 2 - Matriz de avaliação de riscos de um projeto

Ao analisar esta matriz, facilmente percebemos a elevada importância de uma estrutura num projeto, pois sendo eles pequenos ou grandes, mesmo não existindo uma familiarização prévia com a tecnologia e/ou área aplicacional, uma boa estruturação diminui o risco inerente ao mesmo.

Modelo do Sistema a Implementar - Maqueta

Reunindo as conceções de todos os elementos, percebeu-se rapidamente que a aplicação deveria ser constituída por 3 partes: utilizador, servidor, interface. Deste modo, o utilizador iria comunicar com a interface do servidor, fornecendo informações pedidas e seleccionando opções. Por sua vez, a interface devolvia respostas às ações do mesmo, como é visível no seguinte esquema:

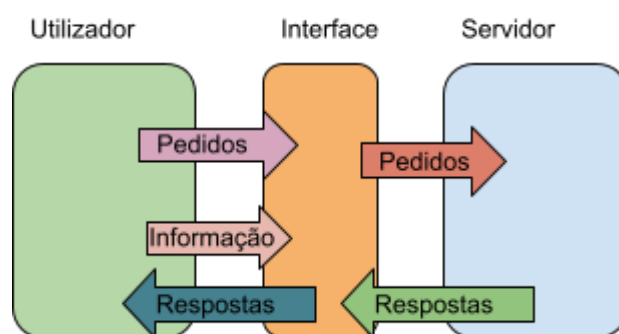


Fig. 3 - Maqueta do Sistema

No entanto, conforme foi elaborado o esquema conclui-se que o sistema teria de possuir uma fonte de dados que comunicasse com o servidor, de forma a ser possível responder aos pedidos que lhe foram enviados. Assim, a base de dados foi adicionada ao que foi estruturado anteriormente :

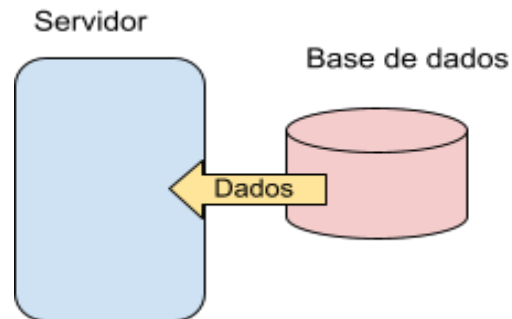


Fig. 4 - Interação Sistema - Base de Dados

Posto isto, é possível perceber as necessidades do projeto e aquilo que teria de ser elaborado e implementado. Começando pelas ações do utilizador, entendeu-se que o mesmo terá a possibilidade de se autenticar na aplicação, caso pretenda manter um histórico e/ou classificar o local visitado, quer seja com estrelas ou adicionando um comentário. Além disso, o utilizador será responsável por indicar a cidade onde se encontra e, seguidamente, deverá seleccionar o tipo de local que procura, entre as 4 possibilidades: restaurantes, hotéis, espaços de lazer ou museus. Com estas informações e acedendo à fonte de dados, a aplicação tem a capacidade de devolver uma lista, preferencialmente organizada segundo as classificações dos locais que cumprem os parâmetros pedidos. Caso um dos locais sugeridos seja do interesse do utilizador, é possível pedir um trajeto que o leve do local onde se encontra, através da inserção das devidas coordenadas, até ao local de interesse desejado.

Como tal, entende-se que o local deverá conter a sua localização em coordenadas, a indicação da cidade, uma descrição (onde poderá estar, por exemplo, um contacto, um horário, o que se pode encontrar no mesmo, entre outros), um tipo (restaurante, hotel, espaço de lazer ou museu) e o número de utilizadores visitantes(por utilizador visitante designa-se o usuário que após seleccionar o local na aplicação, confirmou a chegada ao destino), assim como as avaliações feitas pelos utilizadores relativamente ao local.

Em suma, acredita-se que estas funcionalidades respondem com clareza àquilo que nos propusemos a desenvolver.

Idealizamos que a utilização da aplicação seja semelhante à imagem seguinte:



Fig. 5 - Aplicação Where2Go

Definição de Medidas de Sucesso

De forma a averiguar o sucesso da aplicação, foi necessário avaliar a utilização de uma plataforma concorrente, o TripAdvisor, através de um questionário criado pela equipa e divulgado pela maior amostra de público alvo possível.

É de facto possível inferir que o TripAdvisor possui funcionalidades bastante semelhantes com as que se pretendem implementar.

Até este momento, foram obtidas 57 respostas :

Já utilizou o TripAdvisor?

57 respostas

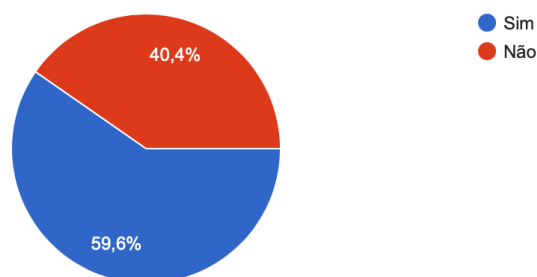


Fig.6 - Questionário de utilização do 'TripAdvisor'

Através do gráfico anterior, é possível perceber que apesar de 40% não terem utilizado a plataforma, reconhecem a utilidade da mesma, como é apresentado a seguir:

O que pensa sobre a utilidade do TripAdvisor?

57 respostas

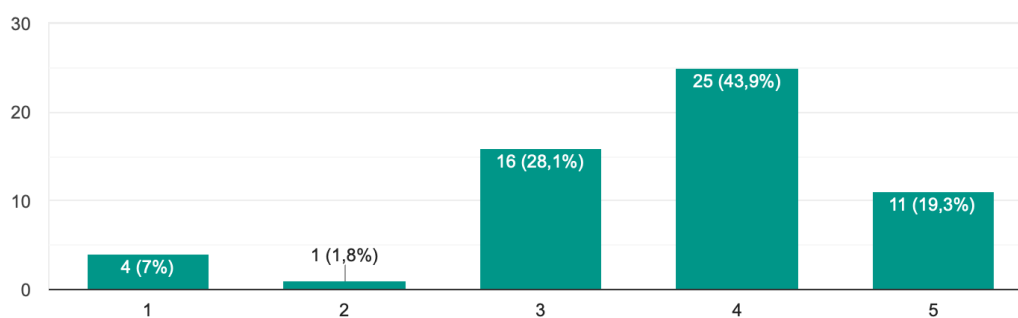


Fig.7 - Questionário da utilidade do 'TripAdvisor'

Para perceber as preferências dos utilizadores para os 4 tipos de espaços que anteriormente foram considerados, foi perguntado no questionário a frequência da procura dos mesmos, numa escala de raramente a regularmente. A equipa considera que é pertinente dar preferência aos locais com o maior número de respostas incidentes entre o 3 e 5.

Com que frequência procura restaurantes?

57 respostas

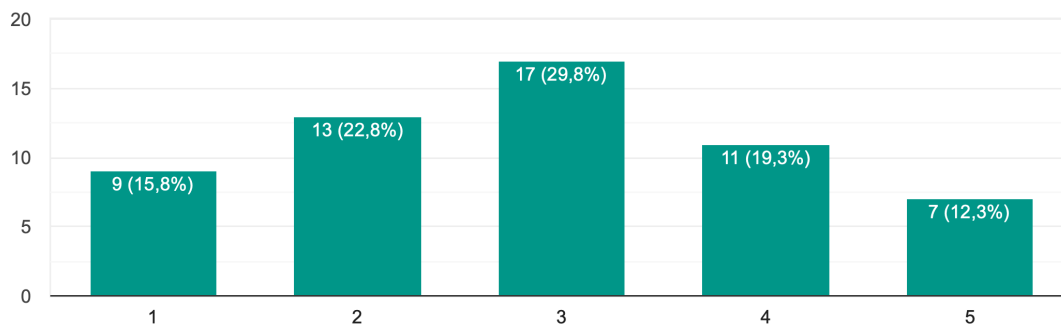


Fig.8 - Questionário de tipo de utilização do 'TripAdvisor' para procura de restaurantes

Relativamente aos restaurantes, a maioria, 35 pessoas, procura os mesmos.

Com que frequência procura hotéis?

57 respostas

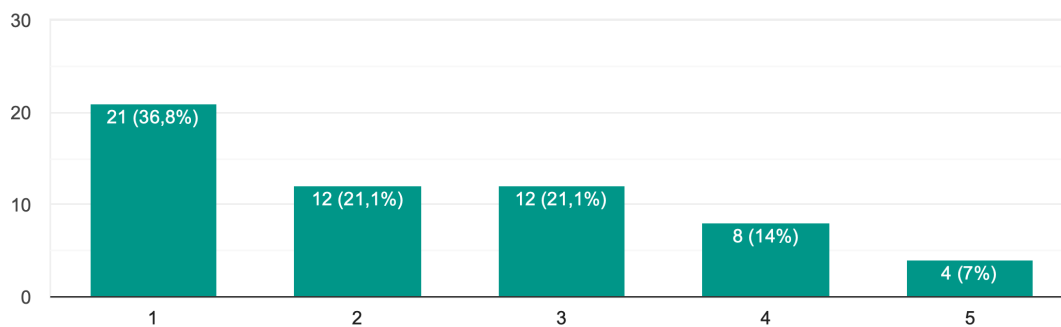


Fig.9 - Questionário de tipo de utilização do 'TripAdvisor' para procura de hotéis

No que toca aos hotéis, a maioria não os procura, 24 pessoas.

Com que frequência procura museus?

57 respostas

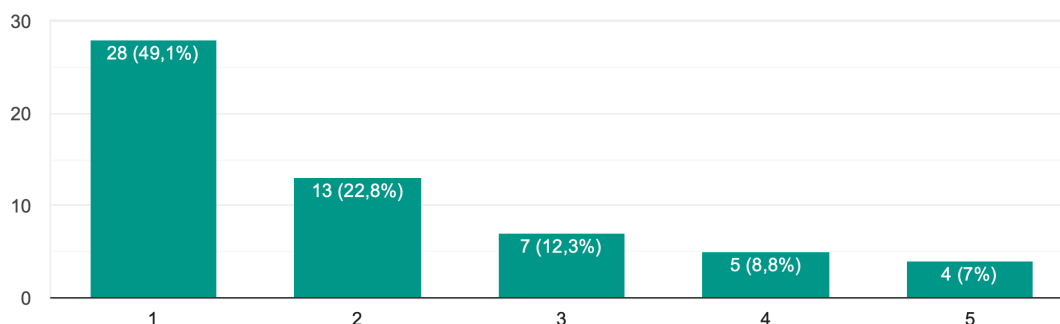


Fig.10 - Questionário de tipo de utilização do 'TripAdvisor' para procura de museus

Em relação aos museus, também a maioria não os procura, 41 pessoas

Com que frequência procura espaços de lazer?

57 respostas

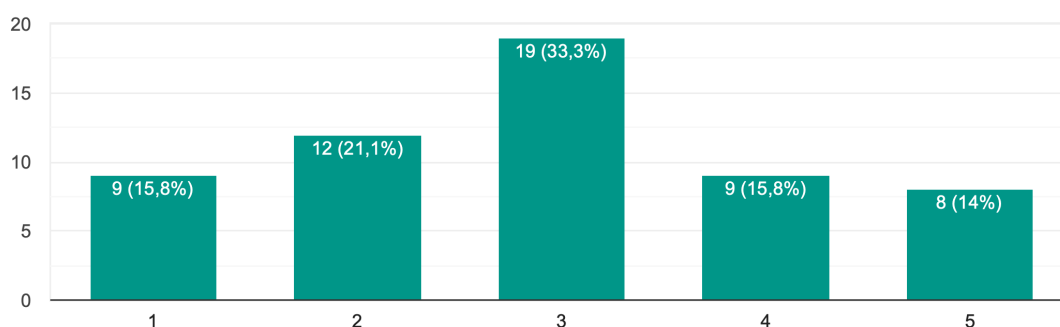


Fig.11 - Questionário de tipo de utilização do 'TripAdvisor' para procura de espaços de lazer

Já para os espaços de lazer, a maioria os procura, 36 pessoas.

Posto isto, percebemos que existem 2 tipos de locais que não fazem sentido serem incluídos no projeto, sendo estes os museus e os hotéis, já que não trariam sucesso a este. É de realçar que o sucesso de uma aplicação é avaliado conforme a utilização da mesma, portanto é necessário avaliar quais são as procuras do público alvo e construir a mesma com base nestes resultados.

Por fim, foram inseridas duas questões que permitem, mais uma vez, perceber a probabilidade de utilização da aplicação e percebemos que a maioria do público iria utilizar com base nas seguintes respostas:

Probabilidade de numa viagem pesquisarem sobre o local

57 respostas

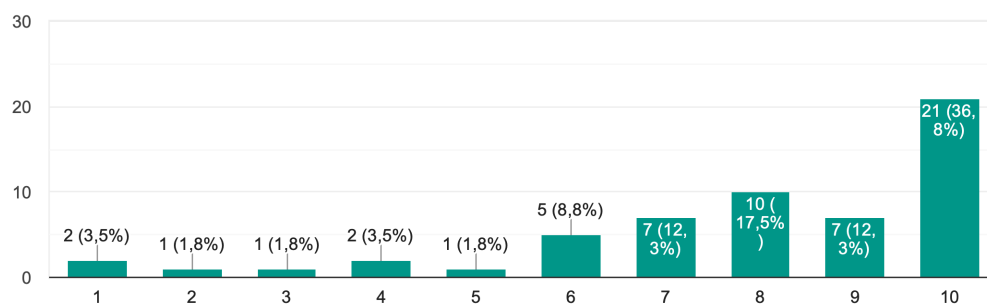


Fig.12 - Questionário de tipo de utilização do 'TripAdvisor' para pesquisa de locais

Já fez alguma viagem sem recurso às tecnologias para pesquisar os locais a visitar?

57 respostas

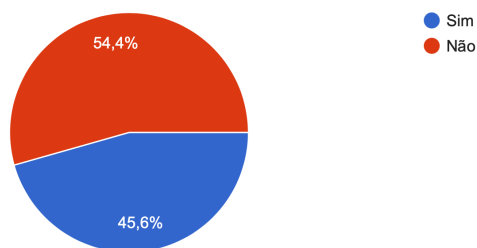


Fig.13 - Questionário da utilização de tecnologias para viajar

Plano de Desenvolvimento - Diagrama de Gantt

O diagrama de Gantt é um gráfico que é realizado com intuito de organizar e, de forma ilustrativa, controlar o cronograma de um projeto. Assim, este inclui as tarefas a serem realizadas, com as suas datas de início e duração. Este revela, ainda, se as mesmas são feitas paralelamente ou sequencialmente, e mostra qual a previsão para a conclusão das mesmas.

Este gráfico permite, então, que o projeto seja acompanhado e gerido de acordo com as etapas necessárias ao seu sucesso, facilitando a elaboração de um trabalho completo e preciso.

Tendo em consideração que a fase inicial onde é realizada a fundamentação e planeamento já foi previamente elaborada, este diagrama apenas incluirá as tarefas seguintes, que compreendem o desenvolvimento e construção.

No que diz respeito à fase de desenvolvimento, é dado especial ênfase ao levantamento de requisitos, pois esta tarefa é uma componente crucial para a elaboração do projeto. Desta forma, são reservados 10 dias para a mesma. Não menos importante, esta fase integra a construção de diagramas e modelação da base de dados.

Já relativamente à fase de construção, é essencial conter um momento para testar o funcionamento do programa e povoamento da base de dados previamente modelada.

Além disso, é importante referir que após a conclusão do trabalho são reservados alguns dias no final de cada fase para que a equipa tenha oportunidade de refletir sobre o trabalho elaborado, existindo margem para a correção de eventuais erros e problemas de maior impacto.

Desenvolvimento da aplicação	Início	Fim	Recursos
<input type="checkbox"/> Especificação	20/10/2021	29/11/2021	
<input type="checkbox"/> Levantamento de Requisitos	20/10/2021	26/10/2021	
<input type="checkbox"/> Análise de Estudos	20/10/2021	22/10/2021	Engenheiro Software, Programador
<input type="checkbox"/> Entrevista a utilizadores de aplicações semelhantes	22/10/2021	24/10/2021	Engenheiro Software
<input type="checkbox"/> Entrevista a pessoas que gostariam de utilizar uma aplicação do género	24/10/2021	26/10/2021	Engenheiro Software
<input type="checkbox"/> Análise de Requisitos	26/10/2021	28/10/2021	Engenheiro Software, Programador

❑ Modelação de Domínio	28/10/2021	30/10/2021	Programador
❑ Diagrama de Use Cases	30/10/2021	02/11/2021	Programador
❑ Diagrama de Packages	02/11/2021	04/11/2021	Programador
❑ Diagrama de Classes	04/11/2021	06/11/2021	Engenheiro Software
❑ Análise de Subsistemas	06/11/2021	08/11/2021	Programador
❑ Modelação da Base de Dados	08/11/2021	12/11/2021	Engenheiro Software
❑ Estruturação da Interface	12/11/2021	22/11/2021	Programador
❑ Reunião e Avaliação das Tarefas Desenvolvidas	22/11/2021	23/11/2021	Engenheiro Software, Programador, Cliente
❑ Correção de eventuais problemas e ajustes	23/11/2021	29/11/2021	Engenheiro Software, Programador

Tabela 2 - Distribuição de Tarefas - Especificação.

De forma intermédia, prevê-se que seja feita uma análise ao planeamento e especificação de outro grupo, de maneira que se reserva alguns dias até 20 de Dezembro para a entrega de um relatório e a partir desse momento é que se inicia a fase de construção, com base nesse relatório e aspetos referidos no mesmo. Por outras palavras, o grupo que executará a fase de construção do projeto, não será o mesmo das fases anteriores.

Desenvolvimento da aplicação	Início	Fim	Recursos
❑ Construção	20/12/2021	24/01/2022	
❑ Implementação da Base de Dados	20/12/2021	29/12/2021	Programador
❑ Povoamento da Base de Dados	29/12/2021	01/01/2022	Programador
❑ Implementação do Modelo	01/01/2022	06/01/2022	Programador
❑ Implementação da Interface	06/01/2022	12/01/2022	Engenheiro Software
❑ Desenvolvimento de Testes	12/01/2022	16/01/2022	Programador
❑ Correção de Eventuais Erros	16/01/2022	24/01/2022	Programador

Tabela 3 - Distribuição de Tarefas - Construção.

Apesar de ser apresentado um diagrama em que a distribuição de tarefas da fase de Construção é marcada como sendo realizada de forma sequencial, é necessário realçar que, sendo um trabalho em equipa, e não havendo dependência entre as mesmas tarefas, poderão

ser realizadas paralelamente, tais como implementação da base de dados, de interface e do modelo.

Diagrama de Gantt

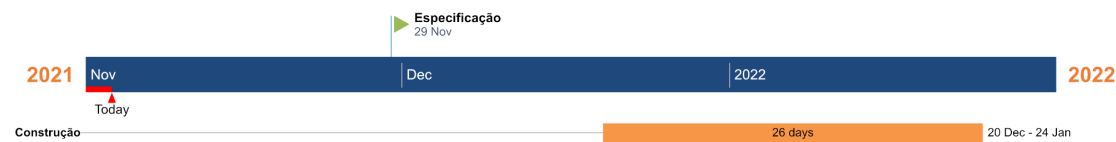


Fig.14 - Diagrama de Gantt



Fig.15 - Diagrama de Gantt - Especificação

Construção



Fig.16 - Diagrama de Gantt - Construção

Capítulo 2: Levantamento e Análise de Requisitos

Apresentação da estratégia e método

Tal como é possível observar no diagrama de Gantt, a fase de Especificação inicia-se com o levantamento de requisitos e compreende uma análise de Estudos, onde realmente é feita uma extensa procura das necessidades do programa. Para isso, são realizadas entrevistas tanto a utilizadores que já utilizam aplicações semelhantes como a utilizadores que não usam, mas compreendem a utilidade das mesmas e gostariam de começar a usar. Dessa forma, cada elemento do grupo ficou responsável de efetuar uma reunião com pelo menos 5 possíveis utilizadores do programa que estivessem dispostos a colaborar na resolução dos requisitos do projeto.

Como tal, foram apresentadas algumas questões tais como:

- “Gostaria de utilizar a aplicação apenas com sessão iniciada?” - Ao que responderam que com sessão iniciada poderiam usufruir de algumas funcionalidades privadas o que seria benéfico, no entanto alguns dos entrevistados disseram que de forma mais prática no dia a dia, gostariam de poder aceder como convidados.
- “O que considera essencial numa aplicação que lhe fornece os pontos de interesse de Malta?” - Disseram que seria extremamente útil ter a possibilidade de ser apresentado um mapa com as direções para esse mesmo local. Assim, como sugerir algumas cidades, ou até mesmo apresentar um histórico dos locais aos quais já se deslocaram.
- “Costuma avaliar os locais que visita?” - Alguns dos entrevistados disseram que não costumam avaliar locais, mas que gostam de ler as avaliações de outros utilizadores em plataformas do género. Já outras pessoas, responderam que geralmente deixam sempre algum comentário aos espaços que visitam, pois entendem a necessidade e a utilidade dessas mesmas avaliações para ajudar os outros utilizadores nas suas viagens.

Para além das perguntas acima colocadas, já tinha sido efetuado um questionário onde foi possível avaliar as preferências do público alvo, no que diz respeito aos locais que

costumam procurar, chegando-se à conclusão que essencialmente a maioria das pessoas têm interesse em investigar restaurantes e espaços de lazer.

Com estas informações, a equipa conseguiu reunir os requisitos do projeto e organizou-se os mesmos em dois tipos: requisitos funcionais e requisitos não funcionais.

Com recurso ao livro *Software Engineering* de Ian Sommerville [1] percebeu-se que os requisitos funcionais são aqueles que descrevem o que o sistema deve realizar, que dependem naturalmente do tipo de software e dos utilizadores. Já os requisitos não funcionais são aqueles que não retratam o serviço fornecido ao utilizador.

Dessa forma, é possível obter a seguinte tabela de requisitos:

Descrição geral dos requisitos levantados

Funcionais	
Iniciar sessão	
Utilizador	Para se autenticar, o utilizador deverá inserir o username e a password.
Sistema	O sistema deverá verificar os dados inseridos e verificar se os mesmos correspondem a algum utilizador. Caso contrário, não deverá validar o pedido de autenticação.
Alterar password	
Utilizador	O utilizador deverá conseguir alterar a password associada ao seu username.
Sistema	O sistema deverá atualizar a nova password do respetivo username.
Registar na aplicação	
Utilizador	Caso não possua sessão o utilizador deverá poder registar-se fornecendo o seu nome, username e password.
Sistema	O sistema deverá verificar se o username já se encontra registado, caso não esteja deverá armazenar os dados fornecidos pelo utilizador e registar uma nova conta.
Entrar como convidado	
Utilizador	O utilizador deverá conseguir aceder às

	funcionalidades básicas como convidado.
Sistema	O sistema deverá permitir pedidos de entrada como convidado e fornecer as funcionalidades básicas.
Consultar lista de locais de interesse	
Utilizador	O utilizador deverá ser capaz de aceder a uma lista de locais de interesse da cidade pretendida.
Sistema	O sistema deverá ser capaz de fornecer uma lista de locais de interesse na cidade pretendida.
Pedir sugestões de cidade	
Utilizador	O utilizador deverá conseguir pedir sugestões de cidades para visitar.
Sistema	O sistema deverá fornecer 3 cidades de sugestão.
Selecionar local	
Utilizador	O utilizador deverá poder selecionar um dos locais da lista que consultou.
Sistema	O sistema deverá permitir que o utilizador possa selecionar um dos locais.
Pedir direções para o local	
Utilizador	O utilizador deverá ser capaz de pedir direções e navegar num mapa.
Sistema	O sistema deverá calcular o trajeto e apresentar mapa navegável.
Cancelar de pedido de direções	
Utilizador	O utilizador deverá conseguir cancelar o pedido de direções.
Sistema	O sistema deverá eliminar mapa e deixar de apresentar as direções. Deverá também informar que o pedido foi cancelado.
Confirmar chegada	
Utilizador	O utilizador deverá conseguir confirmar a chegada ao local.
Sistema	O sistema deverá guardar o local de

	interesse, caso o utilizador esteja com sessão iniciada, no histórico.
Avaliar local	
Utilizador	O utilizador deverá ser capaz de avaliar o local que visitou através de estrelas, de 0 a 5, com a possibilidade de deixar um comentário.
Sistema	O sistema deverá ser capaz de armazenar as avaliações deixadas pelos utilizadores.
Consultar histórico	
Utilizador	O utilizador deverá conseguir consultar o seu histórico de locais visitados.
Sistema	O sistema deverá verificar se o utilizador tem a sessão iniciada, e caso tenha deverá fornecer uma lista com os locais visitados pelo mesmo.
Pedir ajuda	
Utilizador	O utilizador deverá conseguir pedir ajuda no momento inicial da aplicação.
Sistema	O sistema deverá fornecer uma breve descrição do que é o programa e de como se fará a autenticação, registo na aplicação ou entrada como convidado.
Terminar sessão	
Utilizador	O utilizador deverá poder cancelar a sessão que tenha iniciada.
Sistema	O sistema deverá remover a sessão iniciada, voltando à página inicial.

Tabela 4 - Requisitos Funcionais

Requisitos não funcionais

Requisitos não funcionais
A aplicação deverá estar disponível 24 horas durante os 7 dias da semana.
A aplicação deverá ser interativa e fácil de utilizar.
O programa deve ser flexível ao nível da dependência, pois deverá permitir utilizar diferentes fontes de dados, caso se pretenda o fazer.
O programa deverá ter o mínimo de segurança possível, essencialmente deverá validar os inícios de sessão.

Tabela 5 - Requisitos Não Funcionais

Validação dos requisitos estabelecidos

Após reunirmos novamente com os potenciais utilizadores da aplicação percebeu-se que os requisitos estabelecidos vão ao encontro das necessidades dos mesmos, seguindo aquilo que foi pedido. Além disso, também analisou-se que não existiam conflitos entre requisitos e que os mesmos foram descritos detalhadamente.

Conclui-se desse modo que os requisitos não são ambíguos, nem redundantes, e estão organizados e compreensíveis.

Capítulo 3: Especificação e Modelação do Software

Apresentação geral da especificação

Para desenvolver este projeto é necessário a colaboração de dois Engenheiros de Software e dois Programadores de forma a elaborar e definir os aspetos comportamentais e estruturais da aplicação. Assim, de forma a especificar o trabalho a desenvolver, foram elaborados diagramas de *Use Cases* que apresentam a parte comportamental que a aplicação deve possuir na interação com o utilizador. Por sua vez, o modelo de domínio e Diagrama de Classes elaborados estruturam a aplicação, já orientando para as classes, funções e interfaces que serão necessárias. O mesmo acontece como a estrutura da base de dados (representada pelo Modelo de Domínio), que foi elaborada de forma a suprir todas as necessidades de guardar informação de forma organizada, facilitando nos acessos e desempenho da aplicação.

Aspetos estruturais: Diagrama de Classes

O Diagrama de Classes retrata o mapeamento de um sistema ao modelar suas classes, operações, atributos e relações entre objetos, fornecendo assim uma forma visual e intuitiva do funcionamento do sistema. Este diagrama é por muitos considerado o mais importante diagrama UML, pois torna evidente e clarifica a estrutura do sistema que irá ser implementado, descrevendo o que deve estar presente na implementação posterior do projeto. Visualmente, após a construção deste mesmo diagrama é possível visualizar as necessidades do programa e também auxilia uma mais tarde modelação de dados, para um completo sistema de informação.

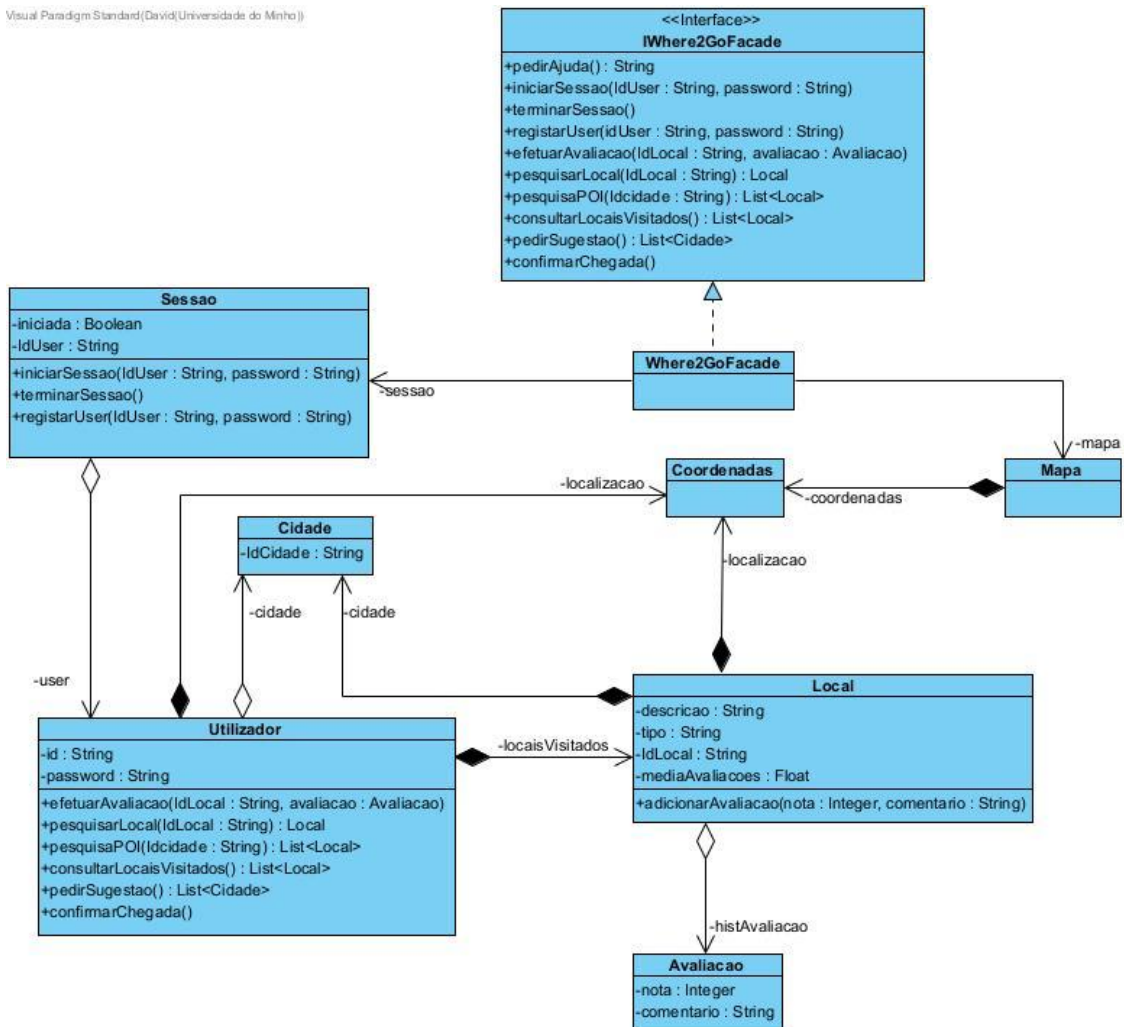


Fig.17 - Diagrama de Classes

De facto, inicialmente foi criada uma *Where2GoFacade* que permite gerir a lógica de negócio do sistema, associando-se a mesma à classe Sessão e Mapa. A classe Sessão contém métodos relacionados com a gestão da conta de um utilizador, utilizador este que também é uma classe com os atributos identificador e password. Além disso, a classe Sessão contém como atributos um identificador do utilizador a que pertence a sessão e um booleano que deverá indicar se esta está ou não iniciada. Posto isto, os métodos do utilizador regem-se pelas funcionalidades que o mesmo poderá realizar, seja efetuar avaliações, pesquisar locais, consultar histórico, pedir sugestões de cidades ou até mesmo confirmar chegada a um dado local. Com estas informações, percebemos que associado ao utilizador deverá estar a classe local que de forma comum ao anterior relaciona-se com a cidade, com identificador como atributo, pois é uma classe comum a ambos. Como atributos de um local tem-se a descrição do mesmo, o seu tipo, o seu identificador e a média de avaliações, possuindo um método de adicionar avaliação a si mesmo, o que denota a relação com a classe avaliação, cujos atributos

são a nota, e um comentário. Por fim temos a classe mapa que se associa à classe coordenadas, que está associada ao utilizador.

Aspetos estruturais: Diagrama de Packages

Um diagrama de packages tem como principal função descrever os pacotes ou pedaços do sistema, divididos de forma lógica, evidenciando as dependências entre eles. Um pacote representa um conjunto de classes e exibe relações de dependência com outros. Desta forma, a nossa aplicação tem como pacotes principais “Where2Go”, que gere todo o funcionamento da aplicação em termos de funcionalidades, nomeadamente, a efetiva sugestões e acompanhamento de POI, e “Gestão Contas” que permite gerir as informações, sessões e perfis de cada utilizador.

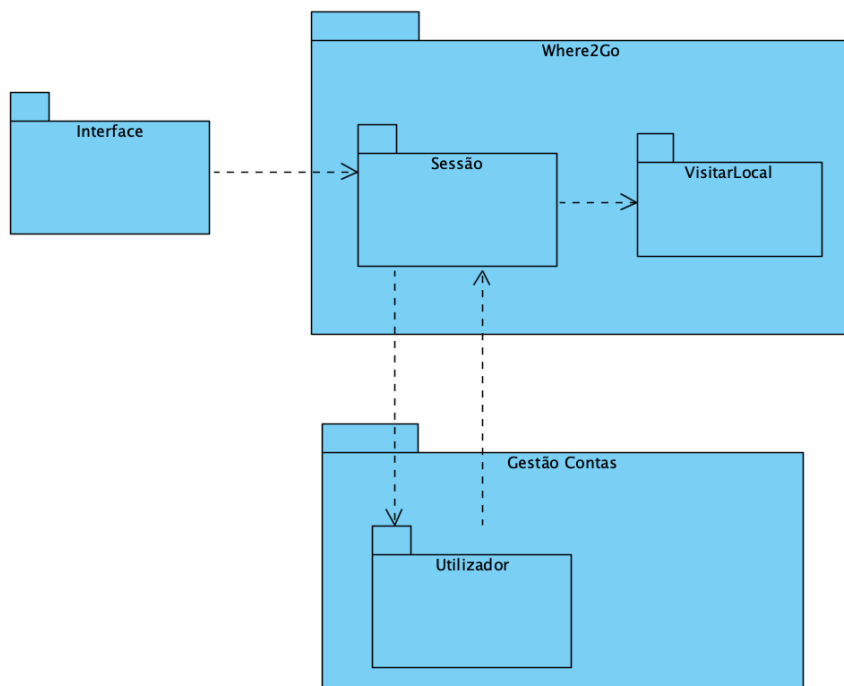


Fig.18 - Diagrama de Packages

Aspetos Comportamentais:

Modelo de Domínio

O modelo de domínio tem como objetivo capturar as informações relevantes sobre o universo de um projeto. Ao representar os vários conceitos do sistema e os relacionamentos entre os mesmos, este modelo oferece-nos uma visão comportamental e interativa do problema, servindo como base para perceber se de facto os requisitos poderão ser concretizados.

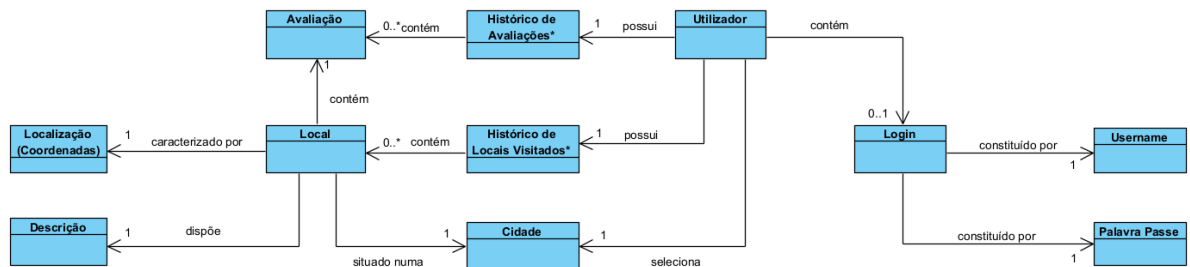


Fig.19 - Modelo de Domínio

Como é possível observar na imagem anterior, existem diversas interações entre os conceitos estipulados. Começando pelo login, deduzimos que este é constituído tanto por um nome de usuário como também uma palavra passe. No entanto, também sabemos que esse login pode ou não ser contido por um utilizador, pois temos conhecimento que o utilizador não necessita de forma obrigatória de um login para usufruir do sistema, ou pelo menos das componentes básicas do sistema. Todavia, aqueles que possuem um login, deverão possuir um histórico das avaliações por si inseridas, e trivialmente esse histórico poderá estar vazio ou conter várias avaliações. Mais uma vez, existe outro histórico associado ao utilizador que é o histórico de locais por si visitados, que nada mais poderá conter do que locais, ou também poderá estar vazio, caso ainda não tenha ainda visitado nenhum. Por sua vez, o local dispõe de uma descrição e está situado numa cidade, cidade esta que também é selecionada por um utilizador.

Diagrama de Use Cases

O diagrama de *Use Cases* explicita as funcionalidades do sistema do projeto, representando, dessa forma, os requisitos funcionais necessários.

No mesmo, definem-se dois atores possíveis do sistema, um deles tem a sessão iniciada e o outro não. Como tal, o primeiro contém um maior número de funcionalidades a si associadas e o segundo apenas é ligado a funcionalidades básicas e gerais do sistema.

Assim, apenas um caso de uso é restrito ao **ator sem registo**, que claramente é realizar o registo na aplicação.

As funcionalidades comuns a ambos são aquelas que não requerem o armazenamento de informações associadas a uma conta como é **avaliar um local** que implica ser um comentário escrito por um utilizador que esteja registado, ou **consultar histórico** que prevê ter uma lista de locais visitados de um dado usuário e, trivialmente, iniciar sessão, terminá-la e mudar a password. Posto isto, todos os casos de uso restantes são possíveis de realizar por qualquer um dos atores, incluindo o **entrar como convidado**, já que não se considera obrigatório que aqueles que contêm sessão a tenham que iniciar para usar a aplicação.

Em suma, este diagrama permite que a equipa tenha representados todos os cenários de interação do sistema com pessoas, modelando o fluxo básico de eventos.

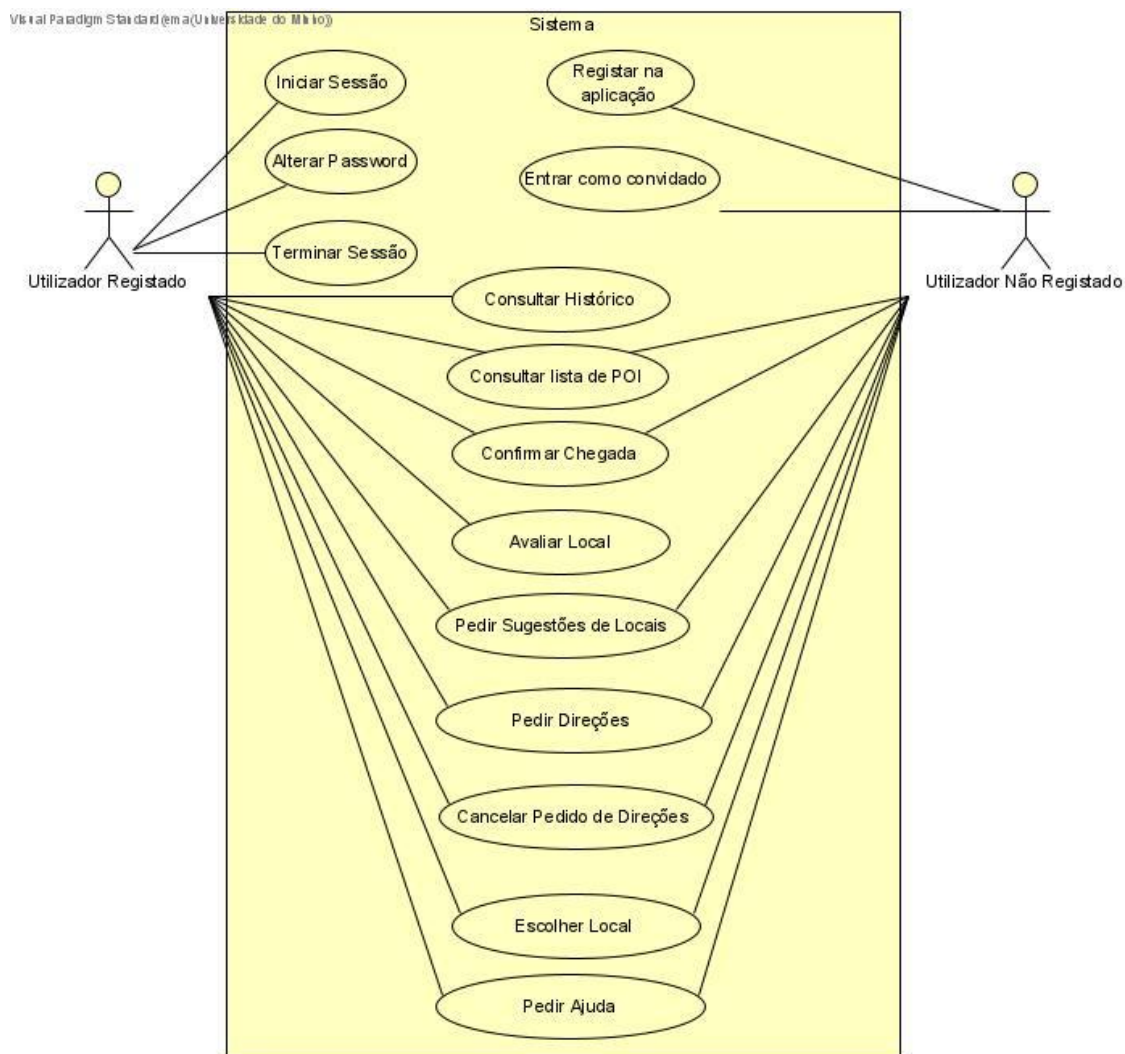


Fig.20 - Diagrama de Use Cases

Especificação de Uses Cases.

Dos Use Cases apresentados no diagrama, existem quatro que são comuns nas aplicações e, como tal, a equipa considera que não é necessário especificá-los. Estes são Iniciar Sessão, Terminar sessão, Entrar como Convidado, Registrar na Aplicação e Alterar Password, os quais compreendem um comportamento por todos conhecido e trivial.

Já os restantes requerem uma especificação por parte da equipa.

Consultar Lista de Pontos de Interesse:

Ator: Utilizador Registado e Utilizador Não Registado.

Pré- Condição: O Utilizador pediu para consultar pontos de interesse.

Pós- Condição: O Sistema apresenta a lista de POI.

Cenário Normal:

1. Ator: Insere a cidade onde se encontra.
2. Ator: Insere o tipo de local que pretende visitar.
3. Sistema: Apresenta a lista de Pontos de Interesse.

Cenário Alternativo 1 (passo 1):

- 1.1 Sistema: Informa que a cidade é inválida. Regressa ao passo 1

Pedir Sugestão de cidades:

Ator: Utilizador Registado e Utilizador Não Registado.

Pré- Condição: O utilizador pediu sugestões de cidades.

Pós- Condição: O Sistema apresenta as sugestões.

Cenário Normal:

1. Sistema: Apresenta três sugestões de cidades.

Escolher local:

Ator: Utilizador Registado e Utilizador Não Registado.

Pré-Condição: O Sistema possui na sua base de dados os locais da lista apresentada anteriormente.

Pós-Condição: O Sistema guarda a informação de qual a cidade selecionada.

Cenário Normal:

1. Ator: Seleciona Local.
2. Sistema: Valida local selecionado.
3. Sistema: Guarda local selecionado.

Cenário Alternativo : Local não é válido. [2]

- 2.1. Sistema: Informa que o local não é válido e pede para que selecionado outro.
- 2.2. Regressa ao ponto 2.

Cenário Exceção: O Utilizador não seleciona nenhum local [1].

- 1.1. Sistema: Informa que não foi selecionado nenhum local.
- 1.2. Regressa à página anterior.

Pedir direções:

Ator: Utilizador Registado e Utilizador Não Registado.

Pré-Condição: O Utilizador selecionou um local.

Pós-Condição: Visualiza-se o mapa com as direções.

Cenário Normal:

1. Ator: Seleciona opção de pedir direções para o local selecionado anteriormente.
2. Sistema: Apresenta o mapa com as direções.

Cancelar pedido de direções:

Ator: Utilizador Registado e Utilizador Não Registado.

Pré-Condição: O Utilizador efetuou um pedido de direções.

Pós-Condição: Não se visualiza o mapa com as direções.

Cenário Normal:

1. Ator: Seleciona opção de cancelar pedido de direções para o local selecionado anteriormente.
2. Sistema: Deixa de apresentar o mapa com as direções.

Confirmar chegada:

Ator: Utilizador Registado e Utilizador Não Registado.

Pré-Condição: O Utilizador selecionou um local anteriormente.

Pós-Condição: Caso se tenha pedido direções deixa-se de se visualizar o mapa com as direções.

Cenário Normal:

1. Ator: Seleciona opção de confirmar chegada ao local selecionado anteriormente.
2. Sistema: Deixa de apresentar o mapa com as direções.
3. Sistema: Caso o utilizador seja registado e tenha a sessão iniciada, guarda o local no histórico associado à conta.

Pedir ajuda:

Ator: Utilizador Registado e Utilizador Não Registado.

Pré-Condição: O Utilizador seleciona a opção de pedir ajuda .

Pós-Condição: O Sistema informa o utilizador através de um aviso.

Cenário Normal:

1. Sistema: Apresenta uma descrição de como se faz um registo, como se inicia sessão e explica o que é a aplicação.

Avaliar local.

Ator: Utilizador Registado.

Pré-Condição: O Utilizador tem sessão iniciada e possui no seu histórico o local.

Pós-Condição: A avaliação é adicionada ao local.

Cenário Normal:

1. Ator: Seleciona local.
2. Sistema: Valida que local existe e está no histórico do ator.
3. Ator: Insere número de estrelas.
4. Ator: Insere comentário.
5. Sistema: Guarda a avaliação.
6. Sistema: Regressa à página inicial.

Cenário Alternativo: Utilizador não insere comentário [4].

- 4.1. Prossegue para o ponto 6.

Cenário Exceção: Utilizador não contém local no histórico [1].

1.1. Sistema notifica que o local ainda não foi visitado e que não permite deixar comentário em locais ainda não visitados.

Consultar histórico.

Ator: Utilizador Registrado.

Pré-Condição: O Utilizador tem sessão iniciada e seleciona a opção de consultar histórico.

Pós-Condição: Histórico do utilizador é apresentado.

Cenário Normal:

1. Sistema: Consulta na base de dados as viagens confirmadas pelo utilizador.
2. Ator: Informa que já consultou o histórico.
3. Sistema: Regressa à página inicial.

Capítulo 4: Conceção do Sistema de Dados

Apresentação geral da estrutura do sistema de dados

Assim que a equipa estabeleceu os aspetos estruturais e comportamentais do projeto, rapidamente se apercebeu das necessidades requeridas na base de dados, já que a mesma deveria responder aos requisitos e acompanhar as especificações anteriormente definidas.

Inicialmente realizou-se uma pesquisa relativamente ao que consiste um Sistema de Gestão de Base de Dados. Com esta procura de informação entendemos que uma base de dados é importante para alcançar o sucesso de gestão de informação de qualquer empresa, já que aumenta a produtividade e melhora o tempo de resposta. Esta deve organizar e gerir as informações relativas a um dado domínio. Desta forma, diminui o risco das operações e aumenta a segurança, pois pode restringir o acesso às informações.

Posto isto, considerou-se que as entidades necessárias no sistema são utilizador registado, utilizador não registado, cidade, local, avaliação e uma relação entre o utilizador registado e o local denominada por histórico.

Estas entidades foram consideradas essenciais para o armazenamento de informações que o projeto necessita, assim como os seus atributos, que serão posteriormente especificados.

Posto isto, desenvolveu-se o modelo lógico e o modelo físico.

Descrição dos elementos de dados e seus relacionamentos

Modelo Lógico

O modelo lógico representa as estruturas que irão armazenar os dados do sistema, estruturas estas que são as entidades e seus respectivos atributos. Neste modelo, declara-se os tipos dos dados que serão armazenados, assim como as relações estabelecidas entre as entidades, e o tamanho que os dados irão ocupar. A seguinte imagem, ilustra o modelo realizado pela equipa.

Efetivamente é possível perceber de forma clara as relações existentes, no caso o Utilizador Registrado e o Utilizador Não Registrado têm ambos uma relação perante a Cidade, pois tanto um como o outro têm a possibilidade de indicar qual a cidade em que se encontram ou onde se irão encontrar. Além disso, o Utilizador Registrado poderá inserir avaliações a um dado local, o que envolve mais uma vez uma relação entre o próprio, a avaliação que insere que estará associada ao local em questão. Para além disso, também é crucial criar uma relação entre os locais e o Utilizador Registrado, já que este irá possuir um histórico.

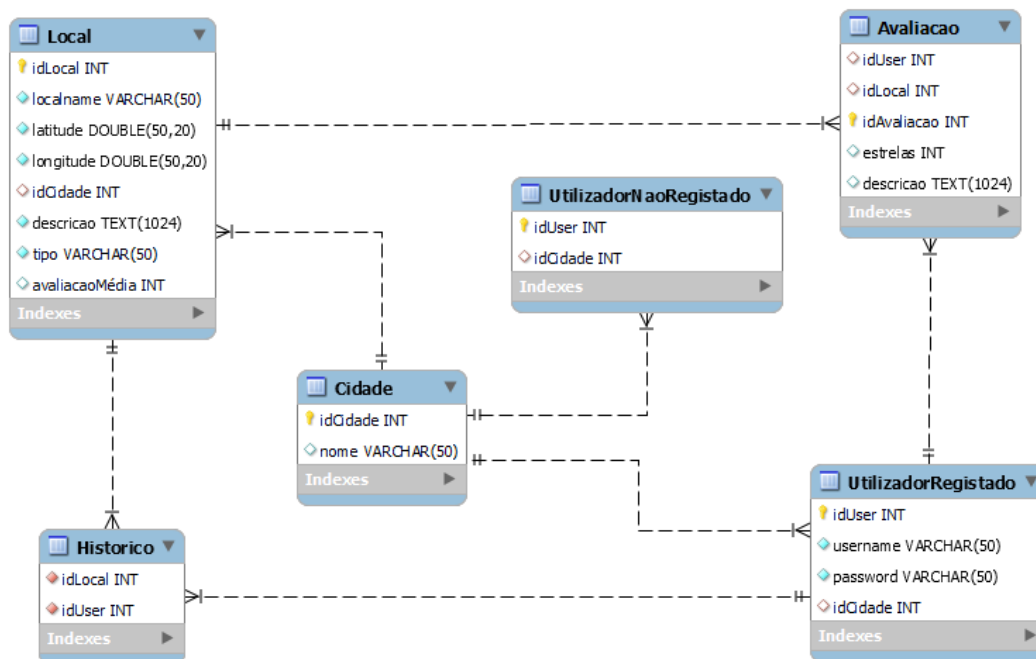


Fig.21 - Modelo Lógico

Modelo Físico

No modelo físico, através do MySQL, pode-se especificar as tabelas representativas das entidades e respetivos atributos, definindo também as chaves primárias e estrangeiras de cada uma. Como é possível observar na imagem seguinte é criada uma *database* onde:

- A primeira tabela criada é a **Cidade**, identificada através de um identificador que é a sua chave primária, além disso possui também como atributo o seu nome, cujo tipo é um VARCHAR(50).
- Seguidamente, é criada a tabela **UtilizadorRegistado** e **UtilizadorNaoRegistado**, em ambos mais uma vez a chave primária é um identificador e para o primeiro são adicionados os atributos *username* e *password*, para armazenar as credenciais dos mesmos, cujos tipos são VARCHAR(50), além do identificador da cidade para ambos que será guardado assim que o utilizador selecionar a cidade em que se encontra.
- Também é construída uma tabela para o **Local**, com a chave primária do identificador do mesmo, e com os atributos do seu nome, das coordenadas, descritas como latitude e longitude com tipos VARCHAR(50) para o primeiro e DOUBLE(50,20) para os seguintes. É adicionado também os atributos do identificador cidade tal como as tabelas anteriores, uma descrição de tipo TEXT(1024), uma avaliação média, dada como INT.

- Por fim, a última entidade é a **Avaliacao**, dada como uma tabela identificada pela chave primária do seu identificador, e com as chaves estrangeiras do identificador do utilizador que realizou a avaliação e do identificador do local em questão. Esta é constituída por um número de estrelas e opcionalmente uma descrição.

Além destas tabelas que identificam as entidades do sistema, é ainda criada uma relação entre os locais e os utilizadores registadas que é a tabela **Historico**.

Após uma análise detalhada, foi possível inferir que esta base de dados permite responder a todos os requisitos, pois as relações entre as tabelas assim o permitem. Posto isto, conclui-se que esta é a base de dados que é necessária ao projeto, que proporcionará o acesso aos dados armazenados para responder aos pedidos feitos pelos utilizadores.


```

1 • DROP DATABASE listadepontos;
2 • CREATE DATABASE listadepontos;
3 • USE listadepontos;
4
5 • CREATE TABLE Cidade(
6     idCidade int AUTO_INCREMENT,
7     nome varchar(50),
8     PRIMARY KEY (idCidade)
9 );
10
11 • CREATE TABLE UtilizadorRegistado (
12     idUser int AUTO_INCREMENT,
13     username varchar(50) NOT NULL,
14     password varchar(50) NOT NULL,
15     idCidade int,
16     PRIMARY KEY(idUser),
17     FOREIGN KEY(idCidade) REFERENCES Cidade(idCidade)
18 );
19
20 • CREATE TABLE UtilizadorNaoRegistado (
21     idUser int AUTO_INCREMENT,
22     idCidade int,
23     PRIMARY KEY(idUser),
24     FOREIGN KEY(idCidade) REFERENCES Cidade(idCidade)
25 );
26
27 • CREATE TABLE Local (
28     idLocal int AUTO_INCREMENT,
29     localname varchar(50) NOT NULL,
30     latitude double(50,20) NOT NULL,
31     longitude double(50,20) NOT NULL,
32     idCidade int,
33     descricao text(1024) NOT NULL,
34     tipo varchar(50) NOT NULL,
35     avaliacaoMedia int,
36     PRIMARY KEY(idLocal),
37     FOREIGN KEY(idCidade) REFERENCES Cidade(idCidade)
38 );
39
40 • CREATE TABLE Avaliacao(
41     idUser int,
42     idLocal int,
43     idAvaliacao int AUTO_INCREMENT,
44     estrelas int,
45     descricao text(1024),
46     PRIMARY KEY(idAvaliacao),
47     FOREIGN KEY(idUser) REFERENCES UtilizadorRegistado(idUser),
48     FOREIGN KEY(idLocal) REFERENCES Local(idLocal)
49 );
50
51
52 • CREATE TABLE Historico(
53     idLocal INT NOT NULL,
54     idUser INT NOT NULL,
55     FOREIGN KEY(idLocal) REFERENCES Local(idLocal),
56     FOREIGN KEY(idUser) REFERENCES UtilizadorRegistado(idUser)
57 );
58

```

Fig.22 - Base de Dados da aplicação em SQL

Capítulo 5: Esboço das interfaces do Sistema

Estrutura Geral das Interfaces do Sistema

As interfaces permitem exteriorizar o conteúdo de um programa aos olhos de um utilizador.

De facto, como previamente referido, o servidor está associado a uma interface que permite a interação utilizador-programa, como a realização de pedidos ao programa e a apresentação das respostas de tais pedidos ao utilizador, através de elementos gráficos e de forma flexível. Além disso, esta permite que se aceda a recursos sem que se comprometa a segurança do programa já que a mesma é implementada pelo programador.

Tendo em conta que é o utilizador que vai interagir com a interface, a mesma deve ser clara e objetiva, não alterando assim os padrões já utilizados por outras aplicações, a fim de gerar uma certa familiaridade do utilizador com a aplicação. Por exemplo, as paletes de cores, ou até mesmo o posicionamento onde os botões normalmente se encontram.

As interfaces deverão ser de fácil acesso e portanto antes de as desenvolver é necessário elaborar rascunhos (mockups) para as mesmas, onde se realizam estruturas que são consideradas e pensadas pela equipa, não sendo estas definitivas, mas que auxiliam o programador quando as conceber.

Caracterização das interfaces

Tal como referido anteriormente, para realizar as propostas de interface foi utilizada a ferramenta Artboard studio e Adobe Illustrator, de forma a completar estas interfaces com elementos criados pela equipa.

De forma inicial, criou-se um mockup, ou seja, um rascunho da futura interface, onde será possível iniciar sessão ou selecionar outras opções como pedir ajuda, mudar a palavra-passe, criar conta, ou entrar como convidado.

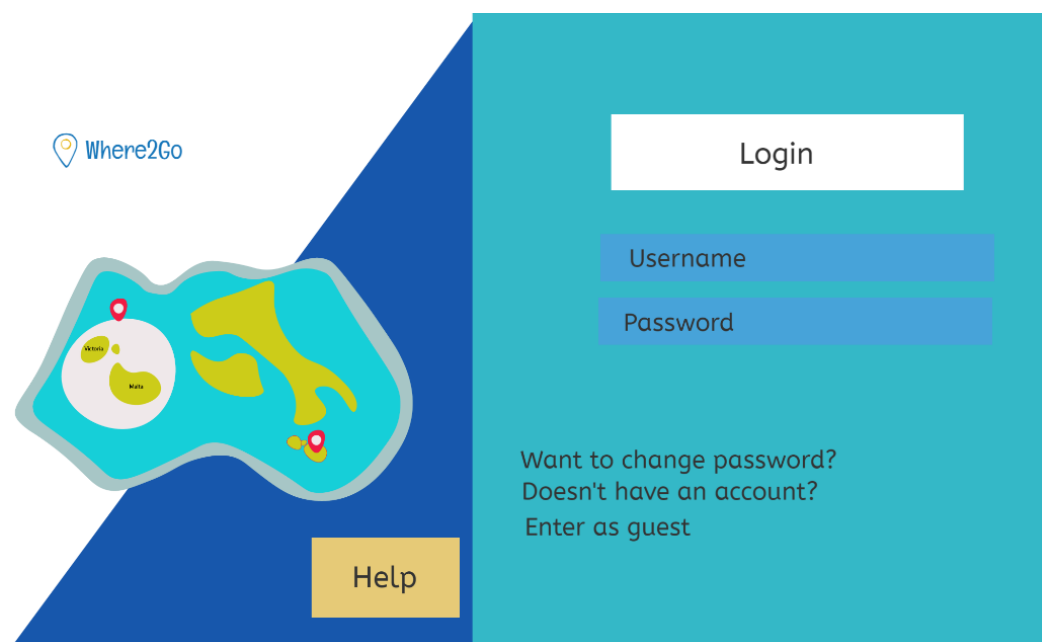


Fig.23 - Mockup Página Login

Seguidamente, propõem-se como interface que se possa inserir o nome da cidade onde se encontra, e o tipo de local que procura ou opcionalmente pedir sugestões de cidades.

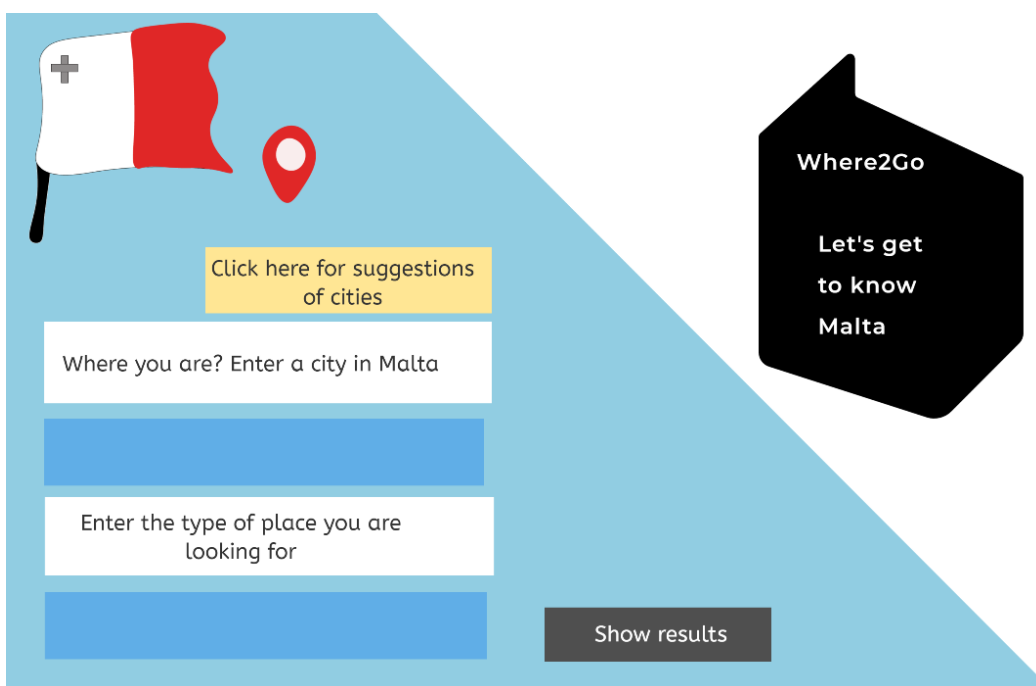


Fig.24 - Mockup Página Principal

Posteriormente, deverá aparecer a lista dos locais de interesse, podendo o utilizador seleccionar o local pretendido e, pedir direcções para o mesmo.

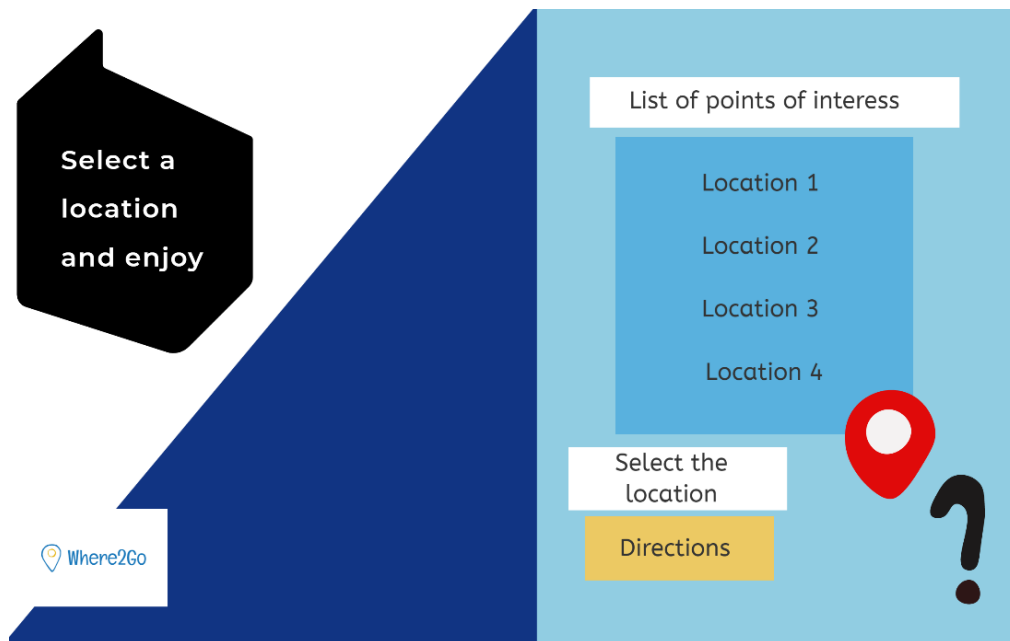


Fig.25 - Mockup Página de Resultados relativos a pedidos de POI

Por fim, realizou-se o mockup referente ao pedido de direcções, onde o utilizador deverá visualizar um mapa que o levará ao local escolhido.

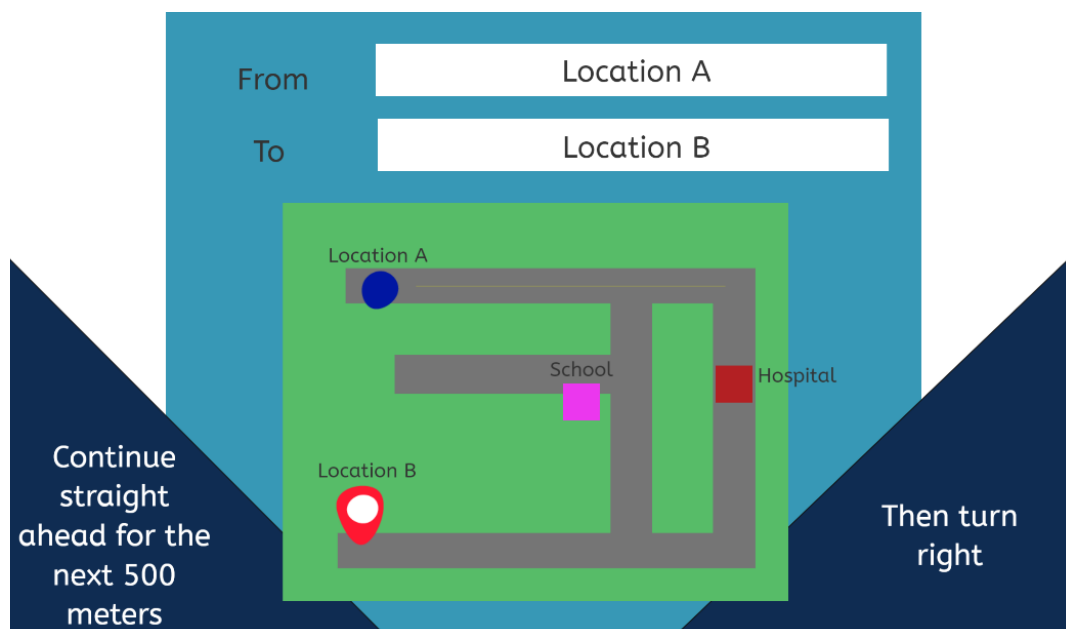


Fig.26 - Mockup Pedido Direcções para um POI

É de realçar que os mockups realizados são meramente representativos daquilo que idealizamos como interfaces.

Capítulo 6: Conclusões e Trabalho Futuro

Chegando ao último capítulo desta primeira fase é possível efetuar uma retrospectiva de todas as etapas do planeamento e especificação que foram realizadas pela equipa.

De facto, foram desenvolvidos vários tópicos com o objetivo de reduzir o risco de não atingir o sucesso no trabalho futuro que constitui a implementação do projeto.

Atendendo que a seguinte fase será realizada por outra equipa de trabalho, foi acrescida uma responsabilidade superior de detalhar exaustivamente tudo aquilo que se estipulou e organizou, sendo este um fator que aumenta a dificuldade pois aquilo que foi redigido tem que ser claro e objetivo não só para a própria equipa, mas também para outros colegas.

Apresentados estes factos, considera-se e acredita-se que todos os argumentos a que a equipa se propôs foram realizados de forma correta e acima de tudo elaborados minuciosamente. Além disso, crê-se que os diagramas UML realizados são os mais adequados e completos, na medida que ainda oferece-se alguma liberdade aos colegas dentro do nosso planeamento para que realizem a implementação sem que se deixe de exemplificar e detalhar aquilo que se pretende.

Por fim, considerando o exposto, a equipa julga ter abordado todos os tópicos apropriada e pertinentemente .

Siglas e Acrónimos

UC: Unidade Curricular

POI : *Points of Interest*

Referências Bibliográficas

- [1] Software Engineering, de Ian Sommerville, 10ª edição.
- [2] Gestão de Projetos De Software, de António Miguel, 5ª edição.

