



Universidade do Minho

Escola de Engenharia

Licenciatura em Engenharia informática

Mestrado Integrado em Engenharia Informática

Unidade Curricular de Laboratórios de Informática IV

Ano Letivo de 2021/2022

Gerês4U

Beatriz Rodrigues, Diogo Casal Novo

Francisco Neves, Gabriela Prata, João Carvalho

Janeiro, 2022

LI4

Data de Recepção	
Responsável	
Avaliação	
Observações	

Gerês4U

Beatriz Rodrigues, Diogo Casal Novo,

Francisco Neves, Gabriela Prata, João Carvalho

Janeiro, 2022

Resumo

Este relatório foi realizado no âmbito do desenvolvimento de uma aplicação que tem como objetivo auxiliar os seus utilizadores na planificação de uma viagem ao Parque Nacional da Peneda Gerês, indicando-lhes pontos de interesse das mais variadas categorias e fornecendo-lhes várias informações acerca de cada um desses pontos.

Nesta fase do projeto procedeu-se à implementação de uma especificação recebida. Inicialmente, aquando da receção da especificação, foi realizada uma apreciação dessa especificação, onde foram apresentadas várias críticas que se consideraram pertinentes e uma avaliação aos diferentes pontos da mesma, que foi entregue à equipa docente da disciplina.

Iniciando-se a fase de implementação, numa primeira etapa realizou-se um enquadramento geral do projeto, foi definido o trabalho que teria de ser desenvolvido tendo como base a especificação que nos foi fornecida, os recursos que seriam necessários à realização do mesmo e foi também realizado um planeamento da fase de implementação recorrendo a um diagrama de *Gantt*.

O passo final tratou-se da implementação da aplicação, onde se definiu a estratégia e o método que seriam adotados ao longo da conceção da aplicação. Foi então desenvolvida a aplicação tendo sempre em mente tudo o que foi referido, apresentando-se neste relatório os serviços que foram efetivamente implementados, uma breve explicação da arquitetura da aplicação e ainda a caracterização das componentes do sistema, apresentando-se também alguns exemplos do funcionamento da aplicação.

Área de Aplicação: Ecoturismo, Natureza, Desenvolvimento de Software, Engenharia de Software

Palavras-Chave: Base de Dados Relacionais, Software, C#, .NET, MySQL, HTML, CSS, Desenvolvimento *Backend*, Desenvolvimento *Frontend*, Peneda-Gerês

Índice

Resumo	i
Índice	ii
Índice de Figuras	iv
1. Introdução	1
1.1. Enquadramento	1
1.2. Trabalho a desenvolver	1
1.3. Recursos Utilizados	2
1.3.1. Plataforma de desenvolvimento	2
1.3.2. Software de Suporte	2
1.3.3. Visual Studio 2022	3
1.4. Plano de desenvolvimento	3
2. Desenvolvimento da aplicação	5
2.1. Estratégia e método	5
2.2. Identificação e caracterização da aplicação e dos seus serviços	5
2.3. Arquitetura geral da aplicação desenvolvida	6
2.3.1. Padrão <i>MVC</i>	6
2.3.2. Camada <i>Model</i>	7
2.3.3. Camada <i>View</i>	8
2.3.4. Camada <i>Controller</i>	8
2.3.5. Camada <i>Data</i>	9
2.4. Definição e caracterização de cada um dos seus componentes	10
2.4.1. Registo	10
2.4.2. Autenticação	11
2.4.3. Ecrã Inicial do Cliente	11
2.4.4. Ecrã Inicial do Revisor	12
2.4.5. Consulta dos Locais a Visitar	12
2.4.6. Informação de um Ponto	13
2.4.7. Consulta de Histórico	13
2.4.8. Sugerir um Ponto	14
2.4.9. Consulta de Sugestões	14
2.4.10. Informação de uma Sugestão	15
2.4.11. Remoção de um Ponto	15
2.4.12. Inserção de um Ponto	16
2.4.12. Atualizar um Ponto	16

3. Conclusões e Trabalho Futuro	17
3.1. Conclusões	17
3.2. Trabalho futuro	18
Referências	19
Lista de Siglas e Acrónimos	20
Anexos	21

Índice de Figuras

Figura 1 : Especificações das tarefas	3
Figura 2 : Diagrama de Gantt	4
Figura 3 - Modelo Lógico da Base de Dados	9
Figura 4 - Registo	10
Figura 5 - Autenticação	11
Figura 6 - Ecrã Inicial do Cliente	11
Figura 7 - Ecrã Inicial do Revisor	12
Figura 8 - Consulta dos locais a visitar	12
Figura 9 - Informação de um Ponto	13
Figura 10 - Consulta de Histórico	13
Figura 11 - Sugerir um Ponto	14
Figura 12 - Consulta de Sugestões	14
Figura 13 - Informação de uma Sugestão	15
Figura 14 - Remoção de um Ponto	15
Figura 15 - Inserção de um Ponto	16
Figura 16 - Atualizar um Ponto	16
Figura 17 - Apreciação à Especificação Recebida	21
Figura 18 - Continuação da Apreciação à Especificação Recebida	22

1. Introdução

1.1. Enquadramento

Para a realização desta fase, que consiste na implementação de uma especificação, foi-nos entregue um relatório de especificação que contém a informação da aplicação a ser desenvolvida. Posto isto, pretende-se desenvolver um software que seja capaz de realizar todas as funcionalidades pretendidas e que cumpra os requisitos pedidos. Devem também ser seguidos, dentro dos possíveis, os diversos diagramas fornecidos na especificação recebida.

Antes de iniciar a fase de implementação foi ainda realizada uma apreciação e avaliação da especificação recebida, de forma a tornar perceptível as dificuldades, bem como os pontos fortes e, até mesmo, as impossibilidades na futura implementação da especificação.

1.2. Trabalho a desenvolver

Tendo em conta a especificação recebida, é possível compreender que a aplicação pretendida deverá ser centrada no Ecoturismo presente no Parque Nacional da Peneda-Gerês e na preparação das viagens dos turistas que pretendam frequentar o parque usufruindo dos seus recantos, até mesmo aqueles mais desconhecidos.

O seu principal objetivo é o auxílio ao utilizador na descoberta de novos locais e no planeamento de percursos para visitas ao Parque Nacional da Peneda-Gerês, utilizando como base os diversos pontos de interesse espalhados pelo parque. Desta forma, este poderá descobrir os mais diversos locais do Parque, desde os mais conhecidos até a alguns mais remotos, bem como obter informações relevantes sobre os mesmos, e por fim definir os percursos que melhor se adequem aos seus objetivos.

Esta aplicação deverá também disponibilizar um histórico de pontos já visitados pelo utilizador para que, no caso deste possuir alguma dúvida sobre os locais já visitados, possa aceder ao seu histórico

de visitas presente na aplicação. Os pontos de interesse podem ser adicionados pelo próprio utilizador ao seu histórico sempre que visitar um novo ponto.

Será ainda disponibilizada aos utilizadores a possibilidade de sugerir novos pontos de interesse que considerem relevantes para que, posteriormente, estes sejam integrados na base de dados da aplicação podendo, desta forma, auxiliar novos e diferentes turistas a obter um maior conhecimento dos pontos de interesse disponíveis no Parque Nacional da Peneda-Gerês.

Além de tudo isto, a aplicação deverá ainda conter diversos revisores responsáveis pela manutenção da informação disponibilizada pela aplicação, permitindo assim a evolução da mesma com base nas propostas dos utilizadores e, até mesmo, com o conhecimento acerca do Parque destes elementos. Assim sendo, estes serão responsáveis pela aceitação, ou não, das sugestões dos diversos utilizadores da aplicação, pela inserção de novos pontos, remoção de pontos já inexistentes e, até mesmo, pela atualização dos pontos presentes.

1.3. Recursos Utilizados

1.3.1. Plataforma de desenvolvimento

Apesar de na especificação recebida existirem *mockups* que poderiam sugerir o desenvolvimento de uma aplicação mobile, foi optado pelo desenvolvimento de uma aplicação *web* uma vez que, a plataforma pretendida não se encontra explícita na especificação recebida e que, no ponto de vista do grupo de desenvolvimento da aplicação, sendo esta orientada para planejar viagens ao Gerês, faz mais sentido que o planeamento seja realizado antes de iniciar a viagem e não durante a mesma. Além disso, tendo em conta os diversos requisitos explicitados na especificação recebida, nunca se demonstra necessária a obtenção da localização do utilizador, pelo que uma aplicação *mobile* não traria qualquer vantagem à aplicação pretendida.

1.3.2. Software de Suporte

Uma vez que foi optado pela implementação de uma aplicação em ambiente *web* foi utilizada a *framework ASP.NET*, utilizando a funcionalidade *MVC*, para o desenvolvimento do projeto, já que esta apresenta uma utilização extensa em contexto real e ainda se revela apropriada para o desenvolvimento de aplicações *web*, devido à sua velocidade e suporte fornecido à equipa de desenvolvimento.

Para o *Back-End* foi utilizada a linguagem de programação *C#*, dada a facilidade de integrar e manipular a base de dados, bem como, à semelhança da *framework* anteriormente mencionada, esta revelar uma utilização extensa em contexto real no que toca ao desenvolvimento de aplicações *web*.

Para o desenvolvimento da Base de Dados foi utilizado *MySQL*, dada a facilidade da sua utilização e integração no programa, a sua utilização no mercado em contexto real, sendo mesmo, a 2ª mais utilizada, e o facto de apresentar um baixo consumo de recursos.

Por fim, visto que se trata do desenvolvimento de uma aplicação *web*, para o *Front-End* foram utilizados *HTML* e *CSS*.

1.3.3. Visual Studio 2022

De forma a apresentar uma implementação correta e atempada do *software* pretendido, foi escolhido para o desenvolvimento do projeto a utilização do *IDE*, *Visual Studio 2022*, já que, além de ser uma das ferramentas mais utilizadas no mercado dos dias de hoje e suportar as diversas linguagens requeridas, conforme explicitado anteriormente, para o desenvolvimento desta aplicação, este *IDE* oferece ainda uma grande variedade de funcionalidades que se revelaram muito úteis para as diversas fases necessárias para o desenvolvimento deste projeto.

1.4. Plano de desenvolvimento


		Nome	Duração	Início	Fim	Antecessores	Nomes dos Recursos
1		Construção de Geres4U	32 dias?	10-12-2021 8:00	24-01-2022 17:00		
2		Arquitetura do Sistema	2 dias?	10-12-2021 8:00	13-12-2021 17:00		
3		Construção da Arquitetura do Sistema	1 dia?	10-12-2021 8:00	10-12-2021 17:00		Programador
4		Validação da Arquitetura do Sistema	1 dia?	13-12-2021 8:00	13-12-2021 17:00	3	Programador
5		Conceção e criação da Base de Dados	7 dias?	14-12-2021 8:00	22-12-2021 17:00		
6		Elaboração do modelo lógico	1 dia?	14-12-2021 8:00	14-12-2021 17:00	4	Engenheiro de Software
7		Implementação da Base de Dados	5 dias?	15-12-2021 8:00	21-12-2021 17:00	6	Programador
8		Povoação da Base de Dados	1 dia?	22-12-2021 8:00	22-12-2021 17:00	7	Programador
9		Programação da Camada Model	5 dias?	23-12-2021 8:00	29-12-2021 17:00	8	Programador
10		Programação da Camada Controller	5 dias?	30-12-2021 8:00	05-01-2022 17:00	9	Programador
11		Programação da Camada View	7 dias?	06-01-2022 8:00	14-01-2022 17:00	10	Programador
12		Validação	5 dias?	17-01-2022 8:00	21-01-2022 17:00	11	
13		Correção de Erros	5 dias?	17-01-2022 8:00	21-01-2022 17:00		Programador
14		Realização de Testes	5 dias?	17-01-2022 8:00	21-01-2022 17:00		Programador
15		Documentação	32 dias?	10-12-2021 8:00	24-01-2022 17:00		Programador

Figura 1 : Especificações das tarefas

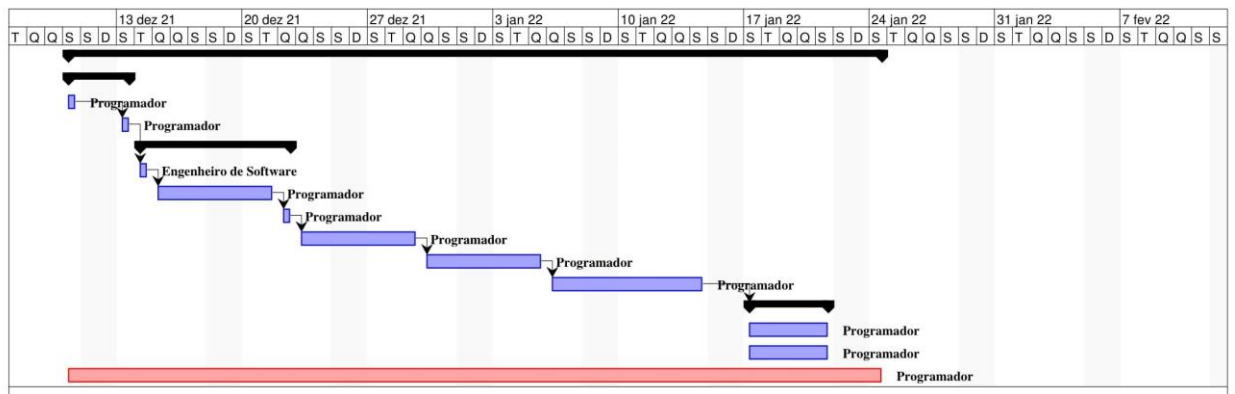


Figura 2 : Diagrama de Gantt

2. Desenvolvimento da aplicação

2.1. Estratégia e método

Esta aplicação foi desenvolvida de forma a cumprir com os requisitos que estavam presentes no relatório de especificação recebido. Para isso, foi optado pelo desenvolvimento de uma aplicação para ser acedida pelo cliente via *web*. Desta forma o cliente poderá aceder ao servidor que por sua vez usa a informação que está guardada numa base de dados.

A Base de Dados apresentada na especificação revelava impossibilidades de implementação devido à existência de ambiguidades e erros, bem como à falta de informações necessárias para o seu desenvolvimento. Tendo isto em conta, após reunião com a equipa docente, na qual foi sugerido ao grupo de desenvolvimento o envio de um *email* aos responsáveis pelo desenvolvimento da especificação recebida solicitando um modelo lógico da base de dados idealizada, estes não se mostraram disponíveis em conceder esse pedido. Como tal, foi necessário criar, durante esta fase, uma nova base de dados de raiz, que permitisse a implementação idealizada de uma forma adequada.

2.2. Identificação e caracterização da aplicação e dos seus serviços

Nesta fase foram implementadas todas as funcionalidades presentes nos *Use Cases* e nos requisitos funcionais da especificação recebida. No entanto, na especificação recebida, surgem ainda diversas funcionalidades apenas nos *mockups* sem nunca terem sido antes referidas (p.e. *Editar Perfil*), o que levou o grupo de desenvolvimento a descartar a implementação destas funcionalidades, visto que, a função dos *mockups* não é a apresentação de funcionalidades da aplicação, mas sim o aspeto estético dela e que estas não contavam na especificação de requisitos funcionais da aplicação.

Deste modo, um cliente da aplicação será capaz de:

- Registrar-se;
- Iniciar Sessão;
- Terminar Sessão;
- Consultar os pontos de interesse presentes;

- Consultar a informação relativa a um ponto de interesse;
- Consultar uma lista de pontos de interesse filtrada pela categoria em que estes se enquadram;
- Adicionar pontos de interesse ao seu histórico de visitas;
- Remover pontos de interesse do seu histórico de visitas;
- Consultar o seu histórico de visitas;
- Sugerir um ponto de interesse para ser adicionado à base de dados.

Por outro lado, um revisor será capaz de:

- Iniciar Sessão;
- Terminar Sessão;
- Aceder às sugestões de pontos de interesse;
- Aceitar uma sugestão de ponto de interesse;
- Rejeitar uma sugestão de ponto de interesse;
- Adicionar um ponto de interesse à base de dados;
- Remover um ponto de interesse da base de dados;
- Atualizar um ponto de interesse da base de dados.

2.3. Arquitetura geral da aplicação desenvolvida

2.3.1. Padrão MVC

Tendo em conta a especificação recebida onde não era explícita a arquitetura pretendida, sendo que, em diversas partes surgiam contradições em relação a este ponto, e a *framework* escolhida pelo grupo de desenvolvimento, foi escolhida a utilização de um padrão MVC para a conceção da aplicação pretendida.

Utilizando este padrão, foi possível proceder à separação explícita dos conceitos das diversas camadas necessárias à implementação do projeto.

É ainda importante realçar que os métodos presentes nos diagramas de sequência da especificação recebida não contemplavam qualquer acesso a uma base de dados, tratando os dados como se estes se encontrassem em estruturas específicas da linguagem, tais como *Maps* em *JAVA* ou *Dictionaries* em *C#* ou *Python*, pelo que, o grupo de desenvolvimento se viu na obrigação de adaptar estes diagramas de sequência ao contexto real em que a aplicação foi desenvolvida.

Por fim, algumas classes foram ainda passíveis de alterações, devido a incongruências apresentadas ao longo da especificação recebida ou, até mesmo, a inutilizações de campos mencionados e impossibilidades de implementação tendo em conta o objetivo principal da aplicação,

conforme explicitado na folha de cálculo em anexo referente à apreciação sobre a especificação recebida.

2.3.2. Camada *Model*

Nesta camada encontram-se presentes as diversas classes que servem de modelos quer para receção de dados por parte dos utilizadores, quer para a apresentação de dados aos utilizadores. De forma mais detalhada, pode-se considerar as seguintes funções para cada classe:

- *Authentication* – Fornece a possibilidade de autenticação a um cliente/revisor, tornando obrigatório o preenchimento dos campos necessários no formulário de autenticação.
- *Category* – Permite que seja associado a um nome de categoria o seu *ID* (inteiro) presente na Base de Dados. Caso o nome da categoria não seja reconhecido, o *ID* é tratado como -1, permitindo assim que a classe responsável pela validação da categoria possa validar ou não uma categoria sem necessitar de aceder à base de dados.
- *Category Validation* – Permite a validação do nome de uma categoria inserida pelo utilizador.
- *Client* – Permite o registo de um cliente na aplicação garantindo o preenchimento dos diversos campos necessários e com as características indicadas na especificação recebida.
- *ErrorViewModel* – Classe necessária para a utilização da *framework* utilizada tendo em vista a arquitetura escolhida. Permite a deteção de um erro.
- *PointOfInterest* – Permite a visualização de um ponto de interesse por parte de um utilizador.
- *PointOfInterestByReviser* – Fornece a possibilidade de inserção de um ponto de interesse por parte de um revisor, tornando obrigatório o preenchimento dos campos necessários e com as características indicadas na especificação recebida.
- *PointOfInterestRemoval* – Permite a remoção de um ponto de interesse indicando o seu *ID*, torna-se necessária esta classe, visto que, na especificação recebida, é explicitado nos *Use Cases* que para a remoção de um ponto de interesse, o revisor deverá solicitar a remoção de um ponto de interesse e o sistema apresentar os campos necessários para a remoção.
- *PointOfInterestSuggestion* – Permite o preenchimento do formulário para a sugestão de um ponto de interesse, tornando obrigatório o preenchimento dos campos necessários e com as características indicadas na especificação recebida.

- *PointOfInterestToUpdate* – Permite o preenchimento do formulário para a atualização de um ponto de interesse, tornando obrigatório o preenchimento dos campos necessários e com as características indicadas na especificação recebida.

2.3.3. Camada *View*

Esta camada é a responsável pela representação dos dados, permitindo assim que o utilizador possa visualizar os itens pretendidos. Esta é constituída por 4 secções relevantes: *Client*, *Home*, *Reviser* e *Shared*.

Na pasta *Home*, foram colocados os ficheiros *.cshtml* responsáveis pela disponibilização dos ecrãs de registo e de autenticação.

Por outro lado, na pasta *Client*, foram posicionados os ficheiros responsáveis pelo menu principal do cliente (*"Index.cshtml"*), pelo menu de pontos de interesse disponíveis (*"GetPointsOfInterest.cshtml"*), pelo menu de histórico de visitas (*"GetPointsVisited.cshtml"*), pelo menu que permite ler mais informação acerca de um ponto em específico (*"GetSpecificPointOfInterest.cshtml"*) e pelo menu para sugerir novos pontos de interesse (*"SuggestPointOfInterest.cshtml"*).

Para além disso, na pasta *Reviser* existem ficheiros semelhantes aos do cliente (relativamente ao menu principal, de pontos de interesse e de informação extra acerca de um ponto), no entanto estes têm algumas diferenças no seu conteúdo porque este tipo de utilizador tem acesso a funcionalidades específicas (p.e. o acesso ao menu de sugestões realizadas pelos clientes, representado no ficheiro *"GetSugestions.cshtml"*). Um revisor pode ainda inserir, remover ou atualizar manualmente um ponto de interesse, respetivamente através dos ficheiros *"AddPointOfInterest.cshtml"*, *"RemovePointOfInterest.cshtml"* e *"UpdatePointOfInterest.cshtml"*.

Por fim, na pasta *Shared* foram colocados *layouts* que permitem a presença de uma barra lateral e o logótipo do programa em todas as páginas.

2.3.4. Camada *Controller*

Esta camada é responsável pela mediação entre a entrada e a saída dos diversos dados necessários para o funcionamento da aplicação. Desta forma, é a responsável por controlar as chamadas à base de dados, quando estas forem necessárias, e, finalmente, pela chamada das potencialidades presentes na *View* para a apresentação dos dados aos utilizadores. De forma mais detalhada, podem-se considerar as seguintes funções para as classes aqui contidas:

- *HomeController* – Responsável pelos métodos alcançáveis a partir da página inicial da aplicação antes de qualquer utilizador se ter autenticado.

- *ClientController* – Responsável pelos métodos alcançáveis após um cliente se ter autenticado. É através dele que se tornam possíveis a implementação dos requisitos funcionais da aplicação para um cliente.
- *ReviserController* - Responsável pelos métodos alcançáveis após um revisor se ter autenticado. É através dele que se tornam possíveis a implementação dos requisitos funcionais da aplicação para um cliente.

2.3.5. Camada *Data*

Tendo em conta que a base de dados da especificação fornecida apresentava contradições e se revelava implementável por diversos motivos, tais como a presença de entidades sem qualquer chave primária ou sem qualquer atributo, o grupo de desenvolvimento necessitou de desenvolver a sua própria base de dados relacional de raiz.

Visto isto, apresenta-se em seguida o modelo lógico da base de dados implementada.

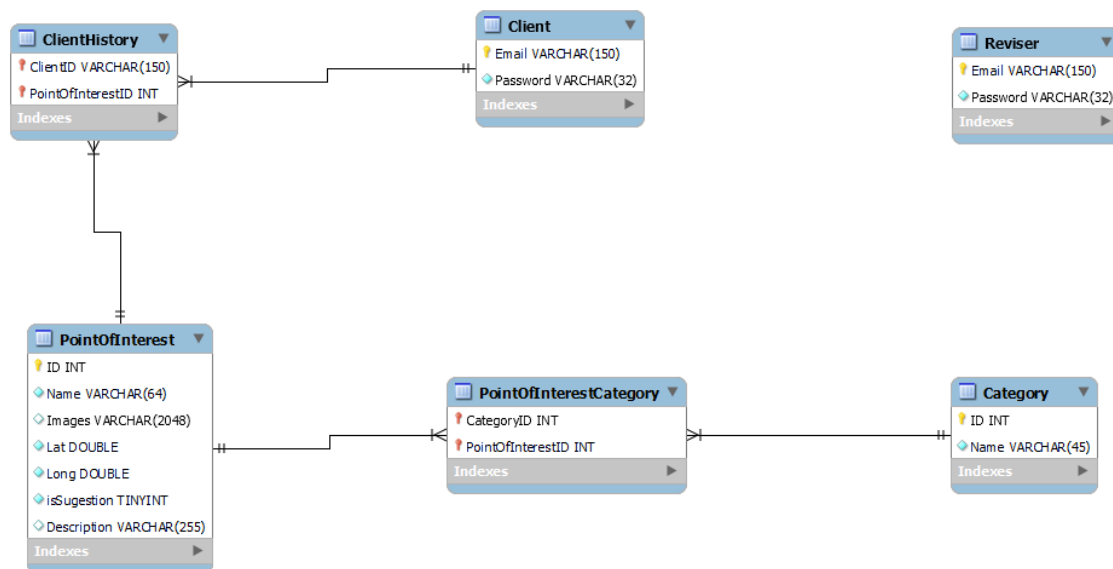


Figura 3 - Modelo Lógico da Base de Dados

Esta camada é a responsável pelos acessos à base de dados, sendo que, desta forma, é através dela que se obtém os dados necessários para o correto funcionamento da aplicação, bem como, tendo em conta a especificação recebida, se inserem novos dados e atualizam ou removem os já presentes.

Nesta camada encontram-se presentes os *DataModels* que permitem criar modelos das tabelas presentes na base de dados. Além disso, as funções das restantes classes são as seguintes:

- *DataAccess* – Permite a definição dos métodos gerais para o carregamento e inserção de dados.
- *CategoryData* – Permite os acessos à tabela *Category*.
- *ClientData* – Permite os acessos à tabela *Client*.

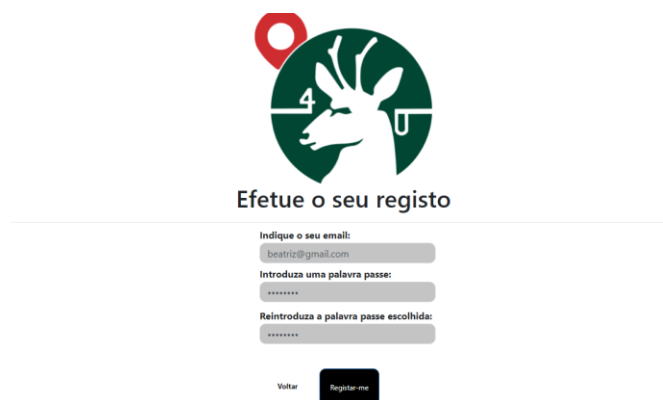
- *ClientHistoryData* – Permite os acessos à tabela *ClientHistory*.
- *PointOfInterestCategoryData* – Permite os acessos à tabela *PointOfInterestCategory*.
- *PointOfInterestData* – Permite os acessos à tabela *PointOfInterest*.
- *ReviserData* – Permite os acessos à tabela *Reviser*.

2.4. Definição e caracterização de cada um dos seus componentes

De forma a cumprir os requisitos da especificação recebida, a aplicação desenvolvida apresenta diversos componentes, sendo eles os seguintes:

2.4.1. Registo

Para um cliente poder utilizar a aplicação desenvolvida, este deverá ser o primeiro passo a efetuar, pois só clientes registados terão acesso à aplicação após efetuarem o seu *login*.



The image shows a registration form for an application. At the top is a logo featuring a white deer head inside a green circle, with a red location pin icon above it. Below the logo is the text 'Efetue o seu registo'. The form contains three input fields: 'Indique o seu email:' with the value 'beatriz@gmail.com', 'Introduza uma palavra passe:' with masked characters, and 'Reintroduza a palavra passe escolhida:' with masked characters. At the bottom left is a 'Voltar' button, and at the bottom right is a 'Registe-me' button.

Figura 4 - Registo

2.4.2. Autenticação

Para um utilizador ter acesso às funcionalidades da aplicação, este deverá, inicialmente, autenticar-se, providenciando o seu *email* e *password* de registo.



Figura 5 - Autenticação

2.4.3. Ecrã Inicial do Cliente

Após se ter autenticado, o cliente terá acesso ao ecrã inicial da aplicação, a partir dele, poderá começar a usufruir das diversas funcionalidades oferecidas pela aplicação.

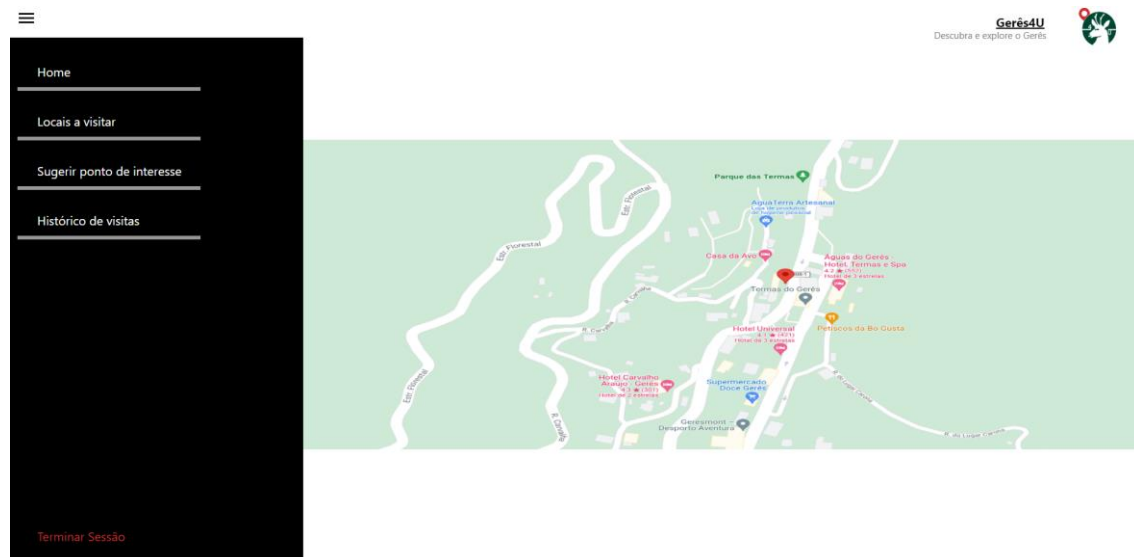


Figura 6 - Ecrã Inicial do Cliente

2.4.4. Ecrã Inicial do Revisor

Caso o utilizador autenticado se encontre registado como revisor na Base de Dados do programa, este terá acesso a um ecrã inicial ligeiramente diferente, visto que, as funcionalidades oferecidas pela aplicação serão diferentes.



Figura 7 - Ecrã Inicial do Revisor

2.4.5. Consulta dos Locais a Visitar

Uma das funcionalidades apresentadas pela aplicação é a consulta dos locais existentes na sua base de dados que já foram aceites por um revisor.

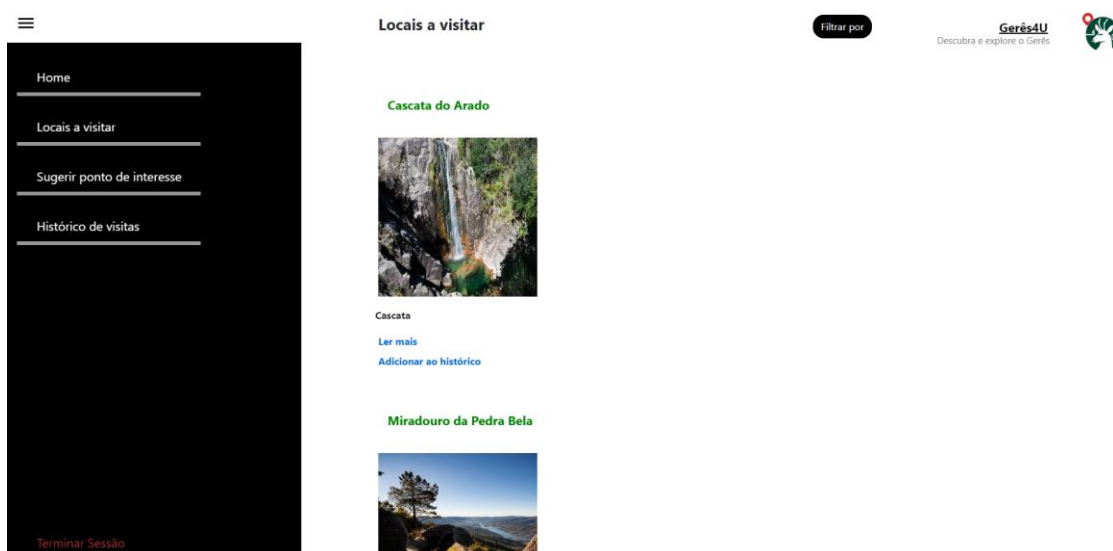


Figura 8 - Consulta dos locais a visitar

2.4.6. Informação de um Ponto

Pode também aceder-se aos detalhes de um ponto após seleccionar a opção “Ler mais” do ecrã apresentado acima.

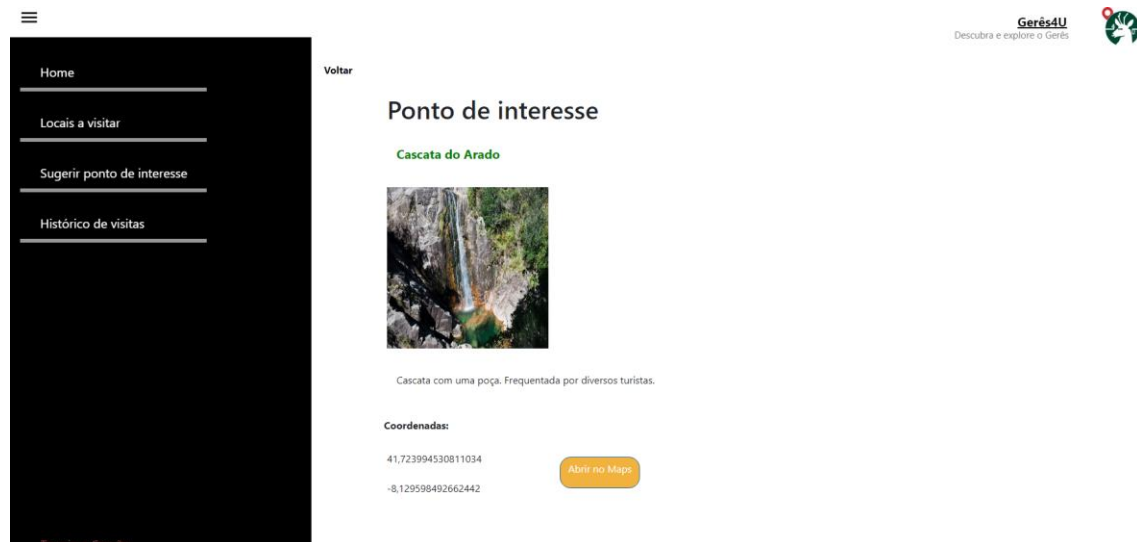


Figura 9 - Informação de um Ponto

2.4.7. Consulta de Histórico

É também possível consultar o histórico de um cliente, sendo que, este poderá adicionar itens e remover do seu próprio histórico. A inserção é feita através da consulta de locais a visitar seleccionando a opção “Adicionar ao histórico”, já a remoção é feita neste ecrã seleccionando a opção “Remover do histórico”.

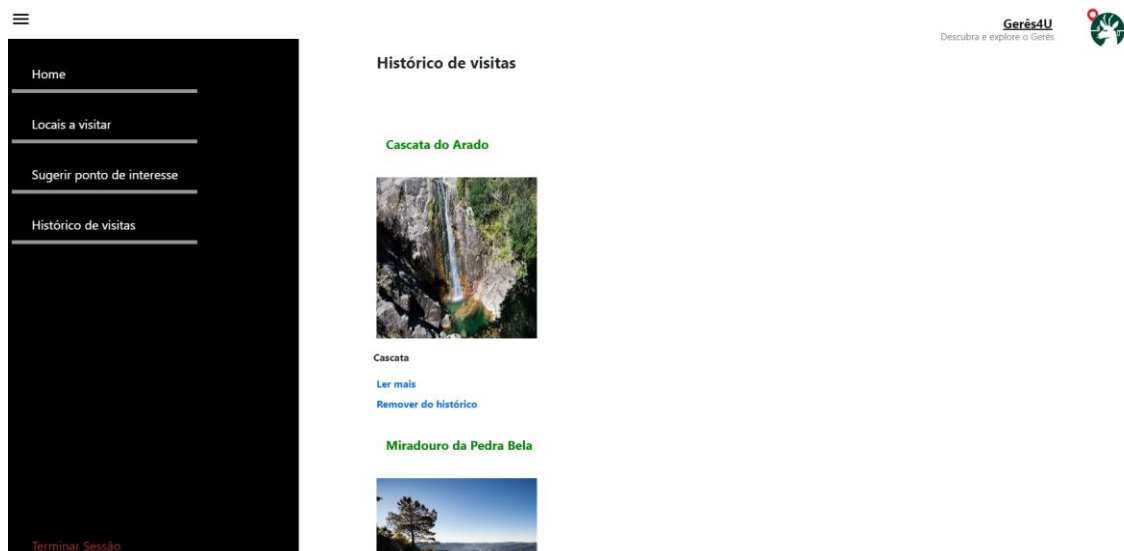


Figura 10 - Consulta de Histórico

2.4.8. Sugerir um Ponto

Por fim, um cliente poderá ainda sugerir um ponto de interesse que considere que deveria estar na base de dados do programa. A inserção do ponto fica pendente da aprovação de um revisor.

The screenshot shows the 'Sugerir um novo ponto de interesse' (Suggest a new point of interest) form. On the left is a dark sidebar with a menu containing: Home, Locais a visitar, Sugerir ponto de interesse (highlighted), Histórico de visitas, and Terminar Sessão. The main content area has a title 'Sugerir um novo ponto de interesse:' followed by input fields for 'Nome' (filled with 'Cascata das 7 Lagoas'), 'Latitude' (filled with '41.750111'), 'Longitude' (filled with '8.075044'), and 'Categoria' (filled with 'Lagoa'). A larger 'Descrição' field contains the text 'Local de perfeita convívio com a natureza e existindo ainda uma d...'. A 'Sugerir' button is at the bottom right. A 'Voltar' link is at the bottom left of the main area. The top right corner features the 'Gerês4U' logo and tagline 'Descubra e explore o Gerês'.

Figura 11 - Sugerir um Ponto

2.4.9. Consulta de Sugestões

De forma a poder aceitar ou rejeitar sugestões dos clientes da aplicação, o revisor pode consultar as sugestões inseridas.

The screenshot shows the 'Sugestões' (Suggestions) page. The dark sidebar menu on the left includes: Home, Locais a visitar, Sugestões dos utilizadores (highlighted), Adicionar ponto de interesse, Remover ponto de interesse, and Atualizar ponto de interesse. The main content area has a title 'Sugestões' and a green heading 'Cascata das 7 Lagoas'. Below this is a placeholder image with the text 'Sem Imagem' and a broken image icon. Underneath the image, the word 'Cascata' is displayed, followed by a blue link 'Ler mais'. The top right corner shows the 'Gerês4U' logo and tagline 'Descubra e explore o Gerês'.

Figura 12 - Consulta de Sugestões

2.4.10. Informação de uma Sugestão

Para poder aceitar ou rejeitar uma sugestão, o revisor deverá selecionar a opção “Ler mais” do ecrã anterior da sugestão pretendida, sendo-lhe depois apresentados os detalhes da sugestão e as opções de aprovação ou rejeição da sugestão.

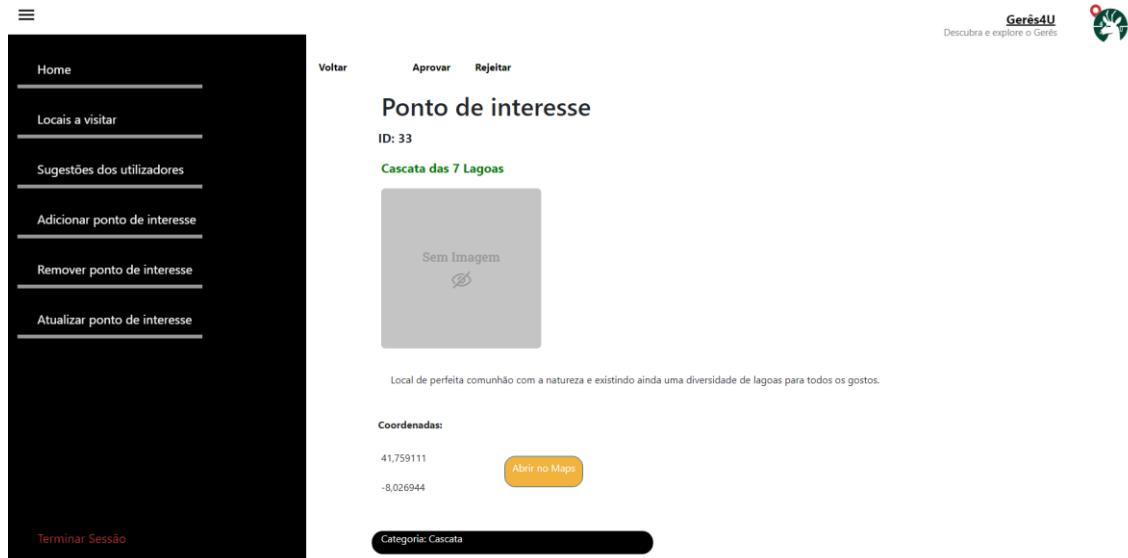


Figura 13 - Informação de uma Sugestão

2.4.11. Remoção de um Ponto

Um revisor pode também remover um ponto da base de dados fornecendo o seu ID.

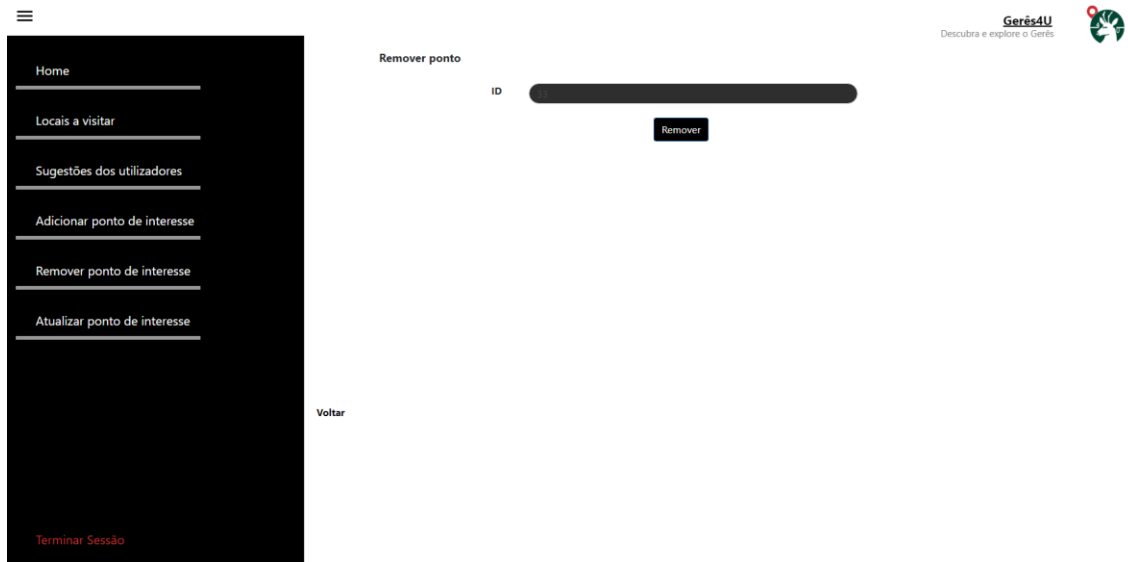


Figura 14 - Remoção de um Ponto

2.4.12. Inserção de um Ponto

Um revisor pode também adicionar um ponto à base de dados fornecendo os dados necessários.

The screenshot displays the 'Adicionar ponto' (Add point) interface of the Gerês4U application. On the left is a dark sidebar with a menu containing: Home, Locais a visitar, Sugestões dos utilizadores, Adicionar ponto de interesse, Remover ponto de interesse, Atualizar ponto de interesse, and Terminar Sessão. The main content area is titled 'Adicionar ponto' and contains the following form fields: 'Nome' (filled with 'Cascata dos C. Vazas'), 'Imagem' (filled with 'Cascata dos C. Vazas'), 'Latitude' (filled with '41.700311'), 'Longitude' (filled with '10.00044'), 'Descrição' (filled with 'Local de perfeita convivência com a natureza e existindo ainda uma dize'), and 'Categoria' (filled with 'Cascata'). A 'Voltar' button is located at the bottom left of the form, and an 'Adicionar' button is at the bottom right. The top right corner shows the Gerês4U logo and tagline 'Descubra e explore o Gerês'.

Figura 15 - Inserção de um Ponto

2.4.12. Atualizar um Ponto

Por fim, um revisor pode ainda atualizar um ponto da base de dados fornecendo os dados necessários.

The screenshot displays the 'Atualizar ponto' (Update point) interface of the Gerês4U application. The sidebar is identical to the previous figure. The main content area is titled 'Atualizar ponto' and contains the following form fields: 'Id' (filled with '1'), 'Nome' (filled with 'Cascata dos C. Vazas'), 'Imagem' (filled with 'Cascata dos C. Vazas'), 'Latitude' (filled with '41.700311'), 'Longitude' (filled with '10.00044'), and 'Descrição' (filled with 'Local de perfeita convivência com a natureza e existindo ainda uma dize'). A 'Voltar' button is located at the bottom left of the form, and an 'Atualizar' button is at the bottom right. The top right corner shows the Gerês4U logo and tagline 'Descubra e explore o Gerês'.

Figura 16 - Atualizar um Ponto

3. Conclusões e Trabalho Futuro

3.1. Conclusões

Ao desenvolver esta fase do projeto foi possível entender a importância do correto planeamento de uma aplicação para o sucesso da sua implementação.

Tendo em conta a especificação desta aplicação consideramos que as funcionalidades mais relevantes pretendidas pelo cliente foram implementadas com sucesso, embora a estrutura proposta na especificação não seja seguida à risca pois, como já foi referido, consideramos que esta apresentava graves falhas em diversas ocasiões.

Além disso, esta fase revelou-se particularmente difícil e complexa já que nos deparamos com um paradigma de desenvolvimento em que não tínhamos qualquer experiência, além das linguagens de programação que foram utilizadas, com as quais também nunca tínhamos contactado antes. Também a conceção, implementação e os diversos acessos necessários à Base de Dados se revelaram uma dificuldade, visto que, tal como os pontos anteriormente referidos, nunca trabalhamos com bases de dados antes.

Assim sendo, apesar de todas as dificuldades enumeradas, consideramos que estas foram superadas com sucesso e que o desenvolvimento deste projeto foi bastante positivo, tendo este contribuído para que nos familiarizássemos com um novo ambiente de desenvolvimento, ajudando-nos a perceber melhor o que se espera de uma equipa de desenvolvimento e, sobretudo, o quão importante as fases de planeamento podem ser para o desenvolvimento de um projeto com sucesso.

3.2. Trabalho futuro

Tendo em conta a especificação recebida, pretende-se que, ao longo do tempo, a aplicação desenvolvida continue a evoluir chegando a cada vez mais utilizadores através de novas funcionalidades que a tornem mais atrativa e útil.

Posto isto pretende-se que, no futuro, a aplicação seja capaz de aceder à localização atual do cliente dando-lhe indicações para este conseguir chegar a um ponto de interesse selecionado ou sugerindo-lhe pontos de interesse que estejam próximos de si. Esta funcionalidade levaria a que o conceito desenvolvido fosse expandido para além de uma aplicação *web*, sendo assim necessário o desenvolvimento de uma aplicação *mobile*.

Além disso consideramos que seria desejável também que a aplicação oferecesse ao utilizador a possibilidade de avaliar e um ponto de interesse, assim como a de consultar mais detalhes de cada ponto de interesse, como a tabela de preços e o horário de alguns serviços.

Por fim, consideramos também que o cliente poderia ter a possibilidade de editar o seu perfil ou, até mesmo, de recuperar a sua *password* através do *email* fornecido para registo na aplicação.

Referências

Sommerville, I., 2011. Software engineering. 10th ed. Boston: Addison-Wesley.

Belo, O., 2021. Bases de Dados Relacionais: Implementação com MySQL. 1ª ed. Lisboa: FCA – Editora de Informática.

Microsoft. [online] Disponível em: <<https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/>>[Consultado a 24 de janeiro de 2022].

MySQL. [online] Disponível em: <<https://dev.mysql.com/doc/>>[Consultado a 24 de janeiro de 2022]

Microsoft. [online] Disponível em: <<https://docs.microsoft.com/en-us/aspnet/core/mvc/views/razor?view=aspnetcore-5.0>> [Consultado a 24 de janeiro de 2022]

Lista de Siglas e Acrónimos

LI4 Laboratórios de Informática IV

Anexos

Universidade do Minho
Licenciatura em Engenharia Informática
Mestrado Integrado em Engenharia Informática
Laboratórios de Informática IV
2021/2022 - 1º Semestre

Avaliação da Fundamentação e Especificação do Sistema de Software

Grupo	30
-------	----

Relatório Recebido

Grupo Nr:	31
Título:	GerB&U

Grelha de Avaliação

Elemento	Descrição	Pontuação (0,5)	Justificação
A1	Definição e Fundamentação	3	<p>Na nossa opinião, o sistema foi decentemente contextualizado, apesar de acharmos que deveria descrever melhor a zona selecionada, o Parque Nacional Peneda-Gerês e as suas características rurais, para futuramente demonstrar a utilidade da aplicação neste caso em específico.</p> <p>Foram apresentados elementos suficientes para se conhecer a motivação e objetivos. No entanto, na justificação e utilidade do sistema há uma pequena contradição: previamente indicam que a aplicação facilita o planeamento e nesta secção referem que esta substitui o planeamento, tornando-o desnecessário. Para além disso, na identidade do projeto consideramos que dizer que o Parque Nacional Peneda-Gerês é o "principal foco" não é correto, visto que até agora é indicado como o único foco. Por outro lado, a maior parte das medidas de sucesso indicadas não são medidas de sucesso (por exemplo: entregar o projeto a tempo, entregar e discussão de ideias, entre outras).</p> <p>Relativamente aos recursos necessários, notamos que não foram especificados os recursos tecnológicos de todo. Quanto aos recursos humanos, a terminologia usada não é objetiva, dando origem a ambiguidades. Por exemplo, "Pessoas responsáveis por desenvolverem os diversos diagramas para planejar o projeto" deveria ser Engenheiro de Software. Para além disso, "Servidor para guardar informação sobre utilizadores e locais de interesse" corresponde simplesmente à base de dados.</p> <p>Por outro lado, a maquete de sistema corresponde na verdade a uma arquitetura do sistema que sugere uma arquitetura Data/Business/UI. No entanto, o texto que a acompanha deixa explícito que é pretendida uma arquitetura MVC, o que é contraditório.</p> <p>Por fim, o plano de desenvolvimento parece bem construído em geral, no entanto os recursos deveriam ser os trabalhadores de um projeto numa realidade de trabalho (programador, engenheiro de software, entre outros) ao invés do número de aluno de responsáveis pelo projeto. O plano beneficiava também se especificasse de melhor forma as tarefas em subtarefas.</p>
	<i>O sistema foi bem contextualizado? Foram apresentados elementos suficientes para se conhecer a motivação, objetivos, medidas de sucesso e a utilidade do sistema? O plano de desenvolvimento foi bem organizado e os recursos necessários identificados e explicados?</i>		
A2	Levantamento e Análise de Requisitos	3	<p>O processo de requisitos está incompleto, apesar dos requisitos estarem, em geral, bem organizados e detalhados. Em primeiro lugar, não foi revelada a forma de levantamento dos requisitos. Para além disso, os requisitos funcionais estão incompletos. A situação que se destaca mais é o facto de nunca mencionarem as interações exclusivas do revisor com o sistema ou sequer da sua existência, apesar de o mencionarem na maquete de sistema e de o incluírem na especificação UML. Todos os requisitos são descritos para um utilizador genérico, até para as interações que apenas um cliente é capaz de efetuar, o que é enganoso. Quanto aos requisitos não funcionais, o referente à aplicação não demorar mais do que 15 segundos a responder é problemático, uma vez que 15 segundos é um tempo demasiado longo e uma vez que esta ideia foi abandonada no resto da especificação do projeto. Nestes requisitos, os revisores são referidos, para indicar que estes devem estar registados na base de dados. Aproveitamos para referir que a ideia de um revisor que avalia todas as sugestões efetuadas pelo cliente não é sustentável a longo prazo, quando se tem em vista o aumento do alcance da aplicação. Consideramos que deveriam existir condições para uma sugestão de um cliente chegar ao revisor como, por exemplo, o mesmo ponto de interesse ser sugerido um determinado número de vezes. Por fim, em nenhum momento é explicada como é obtida a localização do cliente quando, na especificação UML, ela é claramente necessária.</p>
	<i>O processo de requisitos foi bem desenvolvido? Foi revelada a forma como foi realizado o seu levantamento? Os requisitos estão claros, bem organizados e detalhados?</i>		

Figura 17 - Apreciação à Especificação Recebida

A3	Especificação UML		<p>A nosso ver, a especificação UML não está organizada nem completa. Existem várias contradições e vários elementos em falta.</p> <p>Em primeiro lugar, o diagrama de domínio não refere que o ponto de interesse é dotado de um nome. Para além disso tem entidades desnecessárias que até são abandonadas na restante especificação UML: Informação (a Localização, Categoria e Fotografia) podem estar ligadas diretamente à entidade Ponto de Interesse, localização que passa a ser no diagrama de classes uma classe designada por Coordenada.</p> <p>Os principais casos de uso, em geral estão bem identificados, apesar de que as tabelas de especificação deveriam ter uma linha dedicada à identificação clara dos (atores) envolvidos. Para além disso, ao longo das tabelas de especificação, são usadas nas pré-condições "Utilizador/Cliente/Revisor registado e autenticado", o que nos parece redundante uma vez que utilizadores não registados não devem conseguir iniciar sessão. Nos casos de uso "Registar cliente" e "Iniciar sessão", o primeiro evento deveria ser a sua seleção. Sem esta opção, ao lançar a aplicação o utilizador não terá nenhuma ligação entre o ecrã inicial e estas duas opções. Quanto ao caso de uso, "Adicionar pontos de interesse ao histórico de visitas", achamos ineficiente a longo prazo com o crescimento da aplicação ser necessário mostrar a lista total dos pontos de interesse para o utilizador procurar manualmente o ponto de interesse que pretende. Deveria haver um requisito que permitisse fazer pesquisa. Por outro lado, nos casos "Aceitar sugestão de um ponto de interesse" e "Rejeitar sugestão de um ponto de interesse", o evento 2 é desnecessário, podendo ser simplesmente um caso de exceção do primeiro. O "Rejeitar sugestão de um ponto de interesse" tem um problema na descrição: ao invés de dizer "Revisor rejeita Sugestão de Ponto do Interesse para poder depois registá-lo" deveria dizer "Revisor rejeita Sugestão de Ponto do Interesse para poder depois removê-lo". Por fim, no caso "Remover Ponto de Interesse" existe como pré-condição haverem pontos de interesse na base de dados mas depois indicam que esta pode ser uma exceção.</p> <p>Quanto ao diagrama de classes apresentado, este está extremamente incompleto, com métodos, classes e variáveis de instância em falta. Um exemplo em particular é a inexistência de uma superclasse Utilizador, subclasses Cliente e Revisor associadas, visto que "é um" (no diagrama de domínio) sugere uma relação de herança. Para além disso, no Dados/facade está em falta, pelo menos, um Map com revisores, um Map com clientes, um Map com pontos de interesse e um Map com sugestões de pontos de interesse). É também inconsistente com a arquitetura MVC e com a arquitetura Data/Business/UI em simultâneo. Por fim, a designação Dados/facade é duplamente enganosa: em primeiro lugar, numa arquitetura Data/Business/UI visto seria na verdade o Business; por outro lado, Facade sugere uma interface, o que não é o caso. Visto que esta implementa Dados/facade.</p> <p>Alguns diagramas de sequência apresentam ineficiências. No caso do getCoordenadasPorCategoria, o Map<Coordinate,Ponto de Interesse> sugere que a coordenada do ponto de interesse pode ser usada como uma Key e que posteriormente na base de dados seria uma chave primária, o que não faz sentido uma vez que numa dada coordenada podem existir múltiplos pontos de interesse. Assim, sugere-se que o ponto de interesse tenha um ID único específico, no formato String. Para além disso, o avaliasugestao tem um nome muito pouco sugestivo, uma vez que não faz nenhuma avaliação, mas aplica o resultado da avaliação em si. No addHistorico não faz sentido o histórico ser representado por uma List<Coordinate> e fazer uma verificação para garantir que não há repetições do mesmo ponto de interesse. Aconselha-se a substituição por um Set<Coordinate> (na verdade um Set<String> com a alteração anterior à chave que identifica um ponto de interesse). Este caso beneficiava ainda de receber o Cliente em si e chamar um método que fizesse esta adição ao nível da classe Cliente, dispensando a validação de se o email recebido corresponde a um cliente existente.</p>	1					
	<p>A especificação UML está organizada e completa? Acompanha de perto os requisitos previamente estabelecidos?</p> <p>Os principais casos de uso estão bem identificados e detalhados, bem como os atores envolvidos? Os objectos do sistema a desenvolver e os seus relacionamentos estão bem descritos?</p>								
A4	Sistema de Dados		<p>Os sistemas de dados estão apresentados de forma clara e sustentada? Os modelos apresentados foram devidamente enquadrados na especificação UML realizada?</p>	2	<p>Os sistema de dados foram apresentados em tabelas que especificam algumas entidades, os seus atributos e o tipo de dados utilizados para os representar, para além de alguns relacionamentos entre elas. No entanto, não especificam chaves primárias em ocasião nenhuma nem apresentam o diagrama de modelo lógico da base de dados.</p>				
A5	Interfaces		<p>Os interfaces (mockups) do sistema são claros e foram devidamente explicados? É possível identificar a sua utilização no processo de especificação UML que foi realizado?</p>	3	<p>Os mockups são claros e foram explicados devidamente. No entanto, refletem várias funcionalidades que nunca foram referidas nos requisitos, nem no processo de especificação UML (por exemplo, editar perfil, parâmetros novos para cada local como existência de wc e condições, entre outros).</p>				
A6	Implementação		<p>Com base na especificação realizada e na documentação disponibilizada é possível fazer a implementação do sistema requerido?</p>	2	<p>Consideramos que a especificação e documentação realizada deixa a desejar para a implementação. A existência de muitas inconsistências e de falta de informação em diversas ocasiões leva-nos a ter de decidir quais as partes da especificação apresentada que iremos implementar e que partes teremos que descartar completamente por estarem erradas. Não houve também qualquer especificação quanto ao facto da app ser mobile ou web.</p> <p>Para além das contradições, apresenta uma linguagem por vezes confusa e várias intervenções acerca do processo de avaliação e dos prazos da UC que são totalmente dispensáveis tendo em conta o objetivo do projeto.</p>				
					<table><tr><td>Total:</td><td>14,00</td></tr><tr><td>Média:</td><td>2,33</td></tr></table>	Total:	14,00	Média:	2,33
Total:	14,00								
Média:	2,33								

Figura 18 - Continuação da Avaliação à Especificação Recebida