



Universidade do Minho
Escola de Engenharia
Mestrado Integrado em Engenharia Informática

Unidade Curricular de Laboratórios de Informática IV

Ano Letivo de 2015/2016

JustTraillt

Beatriz Aarão	A71015
João Silva	A72023
Octávio Maia	A71369
Rui Freitas	A72399

Março, 2016

LI4

Data de Recepção	
Responsável	
Avaliação	
Observações	

JustTraillt

Beatriz Aarão **A71015**
Octávio Maia **A71369**
João Silva **A72023**
Rui Freitas **A72399**

Março, 2016

Resumo

Este projeto tem como objetivo o desenvolvimento de uma aplicação (*JustTraillt*) capaz de reproduzir um assistente de campo. A nossa empresa decidiu portanto implementar um assistente capaz de auxiliar um batedor no reconhecimento dos percursos a efetuar por uma dada equipa de Rally.

Na primeira fase, foi projetada a aplicação, através da discussão com a empresa RallyScouts dos requisitos a cumprir. Foram ainda divididas as tarefas por todos os membros da equipa de desenvolvimento do projeto, recorrendo a um diagrama de Gantt, permitindo assim uma rápida e eficaz elaboração do mesmo.

Nesta segunda fase,

Área de Aplicação: Qualquer deporto envolvendo análise de percurso.

Palavras-Chave: Microsoft SQL Server, Microsoft Visual Studio, Bases de Dados Relacionais, Gestão de Índices, C#, Protocolos de Comunicação, Microsoft Project, Rally, Batedor, JustTraillt.

Índice

Resumo	i
Índice	ii
Índice de Figuras	iv
Índice de Tabelas	v
1. Introdução	1
1.1. Contextualização	1
1.2. Apresentação do Caso de Estudo	2
1.3. Motivação e Objectivos	2
1.4. Estrutura do Relatório	4
2. Maqueta	5
3. Diagrama de Gantt	6
4. Especificação	8
4.1. Análise de Requisitos Funcionais	8
Administração	8
Aplicação Android	9
4.2. Análise de Requisitos Não Funcionais	9
Administração	9
Aplicação Android	10
4.3. Diagramas em UML	11
4.3.1 Modelo de Domínio	11
4.3.2 Diagramas de Classes	12
4.3.3 Diagrama Use Cases	14
4.3.4 Protocolo comunicação	23
4.3.5 Estados de batedores e atividades no BackOffice	27
4.4. Construção da Base de Dados do BackOffice	29

4.4.1 Construção do modelo Conceptual de dados	29
4.4.2 Construir e validar o modelo lógico de dados	37
4.5. <i>Construção da Base de Dados da Aplicação Mobile</i>	44
4.6. <i>Gramática utilizada pelo batedor</i>	45
4.7. <i>Inicialização da Aplicação</i>	47
5. Diagrama de Gantt (2ª Fase)	48
6. Conclusões e Trabalho Futuro	49
Referências	50
Lista de Siglas e Acrónimos	51
Anexos	52

Índice de Figuras

Figura 1 - Maqueta da aplicação (JustTraillt)	5
Figura 2 - Maqueta da aplicação (BackOffice)	5
Figura 3 - Tarefas do projeto	6
Figura 4 - Diagrama de Gantt	7
Figura 5 - Modelo Domínio do Sistema	11
Figura 6 - Diagrama de Classes BackOffice	13
Figura 7 - Diagrama de Use Cases	14
Figura 8 - Use Case Tabular Regista Nota	15
Figura 9 - Diagrama de Atividade Efetua Nota	16
Figura 10 - Use Case Tabular Inserir Percurso	18
Figura 11 - Use Case Tabular Registo de Atividade	18
Figura 12 - Diagrama de Atividade Regista Atividade	19
Figura 13 - Use Case Tabular Consulta Atividade Concluida	20
Figura 14 - Enviar Relatório Email	21
Figura 15 - Use Case Tabular Gerar Relatório	21
Figura 16 - Diagrama de Atividade ilustrativo de pedido de nova atividade	25
Figura 17 - Enviar Atividade Concluida	26
Figura 18 - Máquinas de Estado para Batedor e Atividade	27
Figura 19 - Modelo Conceptual da Base de Dados do BackOffice	36
Figura 20 - Modelo Lógico da Base de Dados do BackOffice	38
Figura 21 - Mapa validador da Query Consultar Informação de um Batedor	40
Figura 22 - Mapa validador da Query Inserção de um Batedor	40
Figura 23 - Mapa validador da Query Receber Atividade Terminada	41
Figura 24 - Mapa validador da Query Consultar Atividade	41
Figura 25 - Mapa validador da Query Inserir Atividade	42
Figura 26 - Modelo Conceptual da Base de Dados <i>JustTraillt</i>	44
Figura 27 - Excerto do ficheiro de configuração	47
Figura 28 – Diagrama de Gantt	48
Figura 29 - Lista de tarefas da 2 ^a Fase	48

Índice de Tabelas

Tabela 1 - Atributos da entidade Batedor	29
Tabela 2 - Atributos da entidade Veiculo	29
Tabela 3 - Atributos da entidade Mapa	30
Tabela 4 - Atributos da entidade Nota	30
Tabela 5 - Atributos da entidade Atividade	31
Tabela 6 - Entidade Atividade	32
Tabela 7 - Entidade Nota	32
Tabela 8 - Entidade Batedor	32
Tabela 9 - Entidade Veiculo	32
Tabela 10 - Entidade Mapa	33
Tabela 11 - Identificação das relações	33
Tabela 12 - Gramática utilizada para definir o termo Grade	45
Tabela 13 - Gramática utilizada para definir o termo Duration	45
Tabela 14 - Gramática utilizada para definir o termo Further	46
Tabela 15 - Gramática utilizada para definir o termo Road	46

1. Introdução

1.1. Contextualização

A aplicação a desenvolver deverá comportar-se como um assistente de campo.

Esta deverá atuar como um auxiliar para alguém que esteja a fazer determinado trabalho no exterior, permitindo o registo de informações relativas à atividade que está a ser efetuada. Um assistente de campo deve ainda fornecer informações sobre qualquer assunto relacionado com o trabalho de campo quando solicitado.

Neste trabalho iremos então desenvolver um projeto enquadrado num contexto de provas desportivas de rally, provas estas realizadas em terrenos na sua maioria *offroad* por veículos preparados especialmente para o efeito.

Numa prova de rally o veículo leva duas pessoas, o piloto e copiloto. Estes trabalham em equipa de forma a conseguirem efetuar o percurso da prova no menor tempo possível e a causar o menor dano no veículo de modo a serem os vencedores da mesma.

Para facilitar a tarefa do piloto, durante a condução, o copiloto vai indicando ao longo da prova as características do trajeto que se aproxima. De modo a conseguir tal feito, este faz-se acompanhar por um meio de suporte com as notas relativas ao percurso, indicando estas as condições e características do mesmo. As notas devem ser recolhidas anteriormente à realização da prova por batedores que efetuam o reconhecimento do percurso tomando notas relativas a todas as suas características de forma a que a sua "tradução" seja possível efetuar para a linguagem que o copiloto utiliza no seu meio de suporte, segundo a informação referenciada nesta [página](#). A nossa aplicação será assim responsável por auxiliar os batedores no reconhecimento do percurso.

O desenvolvimento do projeto será conseguido utilizando engenharia de software. Esta é hoje uma das áreas da engenharia com maior importância, uma vez que não é possível efetuar o desenvolvimento de uma aplicação sem que esta envolva qualquer estruturação do projeto de desenvolvimento.

A necessidade de estruturação do desenvolvimento deve-se não só à complexidade das aplicações, mas também ao facto de cada vez mais estas serem responsáveis pela gestão, bem como controlo de áreas críticas. Posto isto, não é possível encarar o seu desenvolvimento como algo banal, este processo deverá ser feito de forma metódica e fundamentada seguindo padrões de desenvolvimento reconhecidos.

1.2. Apresentação do Caso de Estudo

Para a realização deste projeto, o grupo decidiu proceder à elaboração de uma aplicação que auxilia um batedor da empresa RallyScouts nas suas atividades de scouting.

Inicialmente o batedor ou alguém responsável pela organização de atividades da empresa, deverá preparar a atividade a realizar utilizando para tal efeito, o *BackOffice*, registando toda a informação que o batedor irá necessitar no processo de scouting.

Descarregar a informação para a aplicação *JustTraillt*, antes de iniciar a sua operação é de extrema importância, para que seja realizada uma boa preparação, pois as atividades podem ser realizadas em locais onde não exista acesso a redes móveis, o que tornaria a atividade para o batedor de elevada dificuldade devido à escassa informação que este poderá ter.

Todos os passos realizados e registados pelo batedor devem ser acompanhados pela sua localização geográfica, mais concretamente as suas coordenadas geográficas. Após a chegada do batedor ao campo de reconhecimento, este deverá recolher o máximo de informação, por voz, texto e imagens, percorrendo todos os pontos de interesse anteriormente registados, de forma a realizar um relatório com a maior qualidade possível.

Após a conclusão da atividade, o batedor deve colocar toda a informação no *BackOffice* da empresa, para que esta seja armazenada, processada e entregue à empresa que a solicitou.

1.3. Motivação e Objectivos

A iniciativa para o desenvolvimento da aplicação surgiu após um breve contacto por parte da empresa *RallyScout*, com serviços de marketing da nossa empresa XSolutionsDevelopment.

A *RallyScout* é uma empresa que presta serviços de reconhecimento de percursos para equipas de desportos motorizados, tendo como especialidade o rally. Esta caracteriza-se pela elevada qualidade nas suas análises e relatórios produzidos, tendo estes obtido enorme credibilidade por todo o mundo.

Embora a empresa seja respeitada por todo o mundo, tem a ambição de se superar e por isso, após uma auditoria interna realizada, depararam-se com o facto de alguns dos seus batedores serem desorganizados com os dados que retiram dos reconhecimentos aos percursos.

Uma má organização por parte dos batedores leva a maus relatórios, que por sua vez podem prejudicar as equipas (pilotos) nas suas competições fazendo com que estas percam segundos essenciais para as classificações. A *RallyScout* considera isto inadmissível, o que levou a administração da empresa a contratar os nossos serviços, com o objetivo de desenvolver uma aplicação que acompanhe o batedor no seu processo de análise do terreno.

Estas análises irão posteriormente ser catalogadas e organizadas de modo a permitir o seu acesso e reutilização em qualquer altura , permitindo assim uma otimização de recursos, mantendo um padrão elevado de qualidade.

No primeiro contacto foi-nos incutido que a qualidade era um fator fundamental e que a facilidade de utilização da aplicação seria importante, devido ao facto de a empresa ser constituída por pessoas de idade avançada que não teriam muita experiência com dispositivos tecnológicos.

Decidimos então, fazer duas aplicações, uma chamada *JustTraillt* cujo objetivo é ser utilizada em dispositivos *Android* e outra denominada de *BackOffice*, responsável por gerir os dados carregados e descarregados da mesma.

A primeira aplicação (*JustTraillt*), deverá ter a capacidade de permitir o *upload* de uma configuração inicial elaborada antes do trabalho de campo. Esta configuração inicial, deverá dizer respeito ao mapa a ser explorado, bem como à informação relativa ao veículo a utilizar nessa exploração.

Durante a exploração do percurso, esta aplicação deverá permitir o registo de informações relevantes através de uma linguagem oral e escrita específica orientada por *tags*, tais como: “*Curva; Terreno; Obstáculo; Piso; etc*”, bem como suporte para registo via fotografias.

Outra funcionalidade da aplicação diz respeito ao uso de GPS. Este suportará o registo do percurso efetuado, permitindo deste modo associar as notas anteriormente descritas a localizações.

A pesquisa de informação utilizando para isso a rede, será facultada a partir da aplicação, não sendo por isso necessário fechar a mesma. Desta forma será possível ao batedor pesquisar por exemplo a meteorologia, ou ainda a pesquisa por imagem ou texto.

Por fim, esta aplicação deverá permitir efetuar o *upload* de toda a informação adquirida para o *BackOffice*.

A nossa segunda aplicação (*BackOffice*), deverá suportar o download de um certo mapa da rede, permitindo a conversão do mesmo para um ficheiro legível por parte da primeira aplicação. Deverá ainda permitir a gestão das horas de trabalho de cada batedor e dos locais a analisar pelo mesmo.

Outra funcionalidade da aplicação consiste na realização de relatórios através do registo de voz gravado na aplicação *JustTraillt*, bem como no armazenamento de uma base de dados responsável pela gestão de todos os dados.

Por ultimo deverá suportar o envio de relatórios por email em formato PDF para outros batedores.

1.4. Estrutura do Relatório

Esta primeira fase do projeto visa apresentar e estruturar o mesmo. Desta forma, nos próximos capítulos, este relatório será composto por uma maqueta do projeto a desenvolver, bem como por um diagrama de Gantt no qual serão apresentadas e divididas as tarefas a realizar ao longo da construção das aplicações pelos diversos membros do grupo.

2. Maqueta

Com a maqueta apresentada à direita, conseguimos ver como exemplo alguns comandos utilizados na linguagem de um copiloto. É ainda possível observar o mapa de um possível percurso a ser analisado pelo batedor.

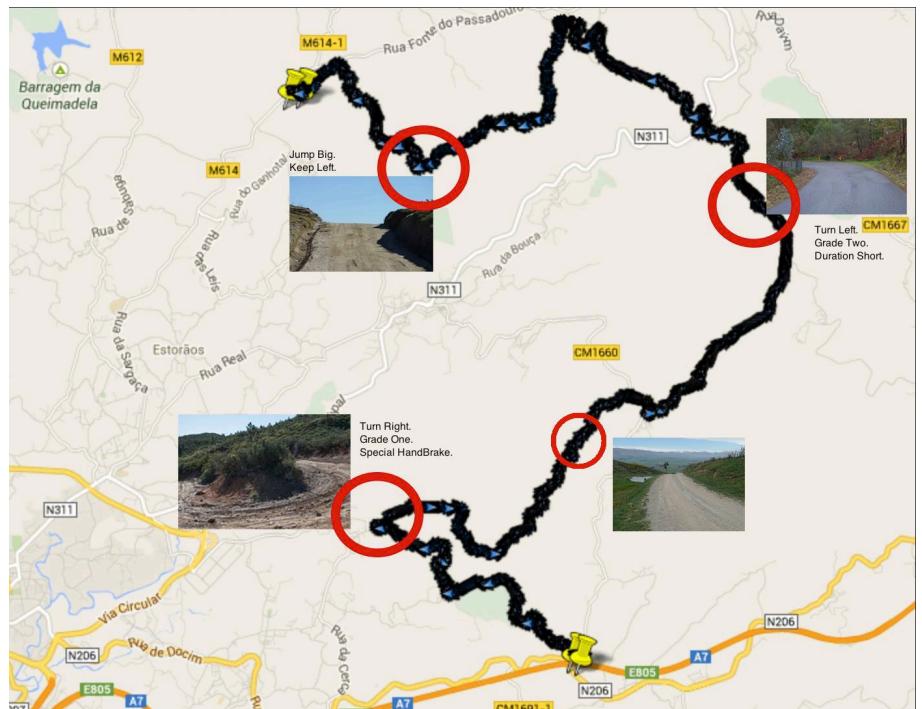


Figura 1 - Maqueta da aplicação (JustTraillt)

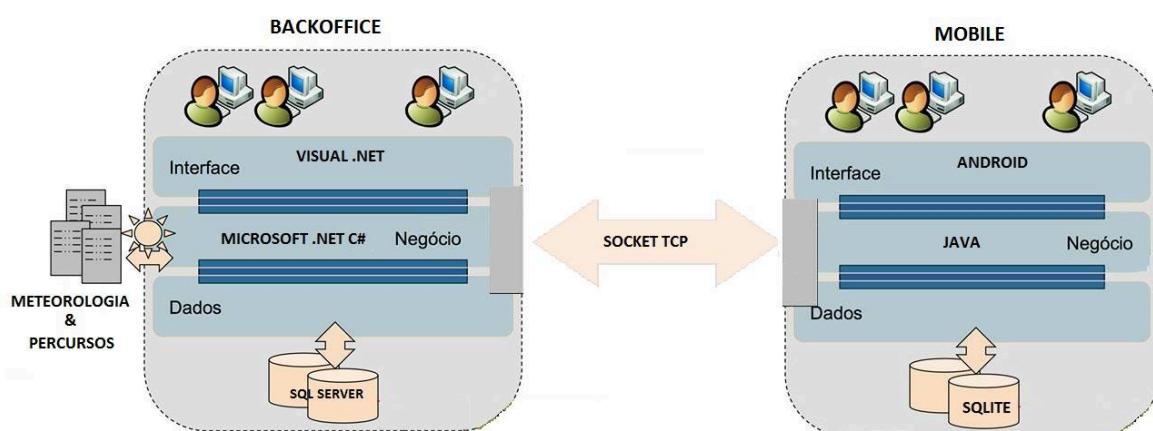


Figura 2 - Maqueta da aplicação (BackOffice)

A maqueta apresentada acima tem como objetivo demostrar a conexão existente entre a aplicação *BackOffice* e a *JustTraillt*.

3. Diagrama de Gantt

	Nome da Tarefa	Duração	Início	Conclusão	Predecessoras	Nomes de Recursos	Ação
0	« 1 GANTT	53,33 dias	Seg 22/02/16	Qui 09/06/16			
1	« 1 Fundamentação	14 dias	Seg 22/02/16	Sáb 19/03/16			
2	1.1 Escolha do tema para o projeto e análise de requisitos	4 dias	Seg 22/02/16	Seg 29/02/16		Beatriz;Joao;Octavio;Rui	
3	1.2 Escrita do Resumo	1 dia	Ter 01/03/16	Qua 02/03/16	2	Octavio	
4	1.3 Escrita da Contextualização	2 dias	Ter 01/03/16	Qui 03/03/16	2	Joao	
5	1.4 Escrita da Apresentação do Caso de Estudo	1 dia	Ter 01/03/16	Qua 02/03/16	2	Rui	
6	1.5 Escrita da Motivação	1 dia	Qua 02/03/16	Qui 03/03/16	5	Rui	
7	1.6 Escrita dos Objetivos	1 dia	Qua 02/03/16	Qui 03/03/16	3	Beatriz;Octavio	
8	1.7 Distribuição das Tarefas e Construção do Diagrama de Gantt	2 dias	Sex 04/03/16	Ter 08/03/16	7	Beatriz;Joao;Octavio;Rui	
9	1.8 Construção das Maquetas	2 dias	Qua 09/03/16	Sex 11/03/16	8	Beatriz;Joao;Octavio;Rui	
10	1.9 Escrita da Estrutura do Relatório	1 dia	Seg 14/03/16	Ter 15/03/16	9	Beatriz	
11	1.10 Escrita da Conclusão	1 dia	Ter 15/03/16	Qua 16/03/16	10	Joao	
12	« 2 Especificação	24,33 dias	Dom 20/03/16	Sex 06/05/16	1		
13	2.1 Discussão, de uma forma detalhada sobre o software a desenvolver	2 dias	Seg 21/03/16	Qua 23/03/16		Beatriz;Joao;Octavio;Rui	
14	2.2 Construção do Modelo de Domínio	3 dias	Qui 24/03/16	Qua 30/03/16	13	Beatriz;Joao	
15	2.3 Construção de Use Cases	4 dias	Seg 28/03/16	Ter 05/04/16		Octavio;Rui	
16	2.4 Construção dos Diagramas de Sequência	5 dias	Ter 29/03/16	Qui 07/04/16		Joao;Beatriz	
17	2.5 Construção do Diagrama de Classes	3 dias	Sex 08/04/16	Qui 14/04/16		Rui;Beatriz	
18	2.6 Especificação da linguagem para o reconhecimento da voz	2 dias	Qui 14/04/16	Ter 19/04/16	17	Joao;Octavio	
19	« 2.7 Base de Dados	4 dias	Ter 19/04/16	Qua 27/04/16	18		
20	2.7.1 Criação do modelo Conceptual	1 dia	Ter 19/04/16	Qui 21/04/16		Octavio;Joao	
21	2.7.2 Documentação do modelo Conceptual	2 dias	Ter 19/04/16	Sex 22/04/16		Octavio	
22	2.7.3 Criação do modelo Lógico	1 dia	Qui 21/04/16	Sex 22/04/16	20	Octavio	
23	2.7.4 Documentação do modelo Lógico	2 dias	Sex 22/04/16	Qua 27/04/16	21	Beatriz;Joao	
24	2.8 Geração da Documentação	16 dias	Qui 31/03/16	Ter 03/05/16		Octavio	
25	2.9 Realização de mockups para BackOffice	3 dias	Qui 24/03/16	Qua 30/03/16		Beatriz	
26	2.10 Realização de mockups para a app mobile	3 dias	Seg 21/03/16	Sex 25/03/16		Rui	
27	2.11 Construção do Relatório	16 dias	Seg 04/04/16	Qui 05/05/16		Beatriz;Joao;Octavio;Rui	
28	« 3 Desenvolvimento	16 dias	Sáb 07/05/16	Qui 09/06/16	12		
29	« 3.1 Desenvolvimento do BackOffice	15 dias	Sáb 07/05/16	Ter 07/06/16		Joao;Octavio	
30	3.1.1 Procura de uma biblioteca para implementação de mapas	1 dia	Seg 09/05/16	Ter 10/05/16		Octavio;Joao	
31	3.1.2 Leitura do ficheiro de mapa da prova	2 dias	Ter 10/05/16	Sex 13/05/16		Octavio	
32	3.1.3 Desenvolvimento Físico da Base de Dados	3 dias	Ter 10/05/16	Seg 16/05/16	22	Joao	
33	3.1.4 Ler coordenadas Geográficas	2 dias	Sex 13/05/16	Qua 18/05/16	31	Octavio	
34	3.1.5 Registar coordenadas geográficas	4 dias	Qua 18/05/16	Qui 26/05/16	33	Octavio	
35	3.1.6 Introdução do registo das atividades	5 dias	Qua 18/05/16	Sex 27/05/16		Joao	
36	3.1.7 Investigação de tecnologias de parsing de texto	3 dias	Qua 25/05/16	Ter 31/05/16	43	Joao	
37	3.1.8 Gerar relatórios em pdf's	4 dias	Ter 17/05/16	Qua 25/05/16		Octavio	
38	3.1.9 Enviar emails com relatórios em anexo	3 dias	Seg 30/05/16	Sex 03/06/16		Joao	
39	« 3.2 Desenvolvimento Aplicação em android	16,33 dias	Sex 06/05/16	Qua 08/06/16		Beatriz;Rui	
40	3.2.1 Exploração das tecnologias da Google	1 dia	Sex 06/05/16	Seg 09/05/16		Beatriz;Rui	
41	3.2.2 Registar coordenadas locais	1 dia	Seg 09/05/16	Qua 11/05/16	40	Rui	
42	3.2.3 Registar periodicamente o percurso do batedor	2 dias	Qua 11/05/16	Seg 16/05/16	41	Rui	
43	3.2.4 Procura de biblioteca para gravação de voz	1 dia	Seg 09/05/16	Qua 11/05/16	40	Beatriz	
44	3.2.5 Gravação de ficheiros de voz	3 dias	Qua 11/05/16	Ter 17/05/16	43	Beatriz	
45	3.2.6 Ter acesso a meteorologia passada pelo BackOffice	2 dias	Ter 17/05/16	Sex 20/05/16	44	Beatriz	
46	3.2.7 Tirar fotos	2 dias	Seg 16/05/16	Qui 19/05/16	42	Rui	
47	3.2.8 Realizar anotações de texto	1 dia	Sex 20/05/16	Ter 24/05/16	45	Beatriz	
48	3.2.9 Carregar dados do BackOffice	3 dias	Ter 24/05/16	Seg 30/05/16		Beatriz;Rui	
49	3.2.10 Carregar dados para BackOffice do recolhecimento realizado	3 dias	Seg 30/05/16	Seg 06/06/16	48	Beatriz;Rui	
50	4 Entrega	0 dias	Qui 09/06/16	Qui 09/06/16			

Figura 3 - Tarefas do projeto



Figura 4 - Diagrama de Gantt

4. Especificação

4.1. Análise de Requisitos Funcionais

Administração

1. Uma atividade deve ser planeada anteriormente à sua realização e introduzida no sistema por um gestor.

A atividade deve ser planeada e introduzida no BackOffice até ao final do dia anterior à sua realização, sendo que todas as atividades a realizar serão "processadas" no inicio do próprio dia e posteriormente transferidas para as diversas aplicações móveis.

Cada atividade possui um percurso (mapa GPS) fornecido pelo cliente que deseja o reconhecimento do mesmo. Uma atividade contém ainda o nome da equipa que a solicitou o serviço e as características anormais do veículo (considerações a ter que podem alterar o comportamento do veículo no percurso).

2. No final de uma atividade deve ser possível gerar um relatório com as notas tomadas durante a mesma, na linguagem utilizada pelos copilotos disponível neste [endereço](#).

A informação descarregada para o BackOffice pelos batedores, permite gerar dois relatórios, sendo o primeiro apenas para os copilotos, dizendo respeito às notas tiradas por voz. O segundo relatório, conterá toda a informação possível de obter, incluído imagens e os textos gerados pelo batedor, bem como as coordenadas geográficas do local onde a nota foi retirada.

Ambos serão gerados aquando do inicio do dia seguinte ao seu carregamento .

3. Permitir a gestão dos batedores

Cada batedor é identificado por um código único de batedor na empresa BTxxxxxx.

A gestão dos mesmos permite alocar os batedores de forma a estes não terem tarefas sobrepostas. Deverá ainda permitir gerir as horas de trabalho dos mesmos bem como os locais a analisar.

Aplicação Android

4. Permitir a transferência para o BackOffice dos dados relativos a uma atividade.

Após a conclusão da atividade em causa, a sua informação é transferida para o BackOffice mal a aplicação mobile e este estejam na mesma rede. Após a transferência com sucesso dos dados em causa, estes são eliminados da aplicação mobile ficando apenas guardados no BackOffice.

A aplicação permite definir o endereço do servidor em causa para a transferência de dados, embora a porta pela qual a transferência ocorre seja pré-definida.

5. Permitir gravação de voz, fotografias e texto

De modo a permitir a gravação de voz, foi implementado o reconhecimento de linguagem específica.

De modo a criar um registo, este tem de estar associado ao local onde a mesma se encontra de modo a ser gravada (apenas as notas gravadas servem para gerar o relatório que acompanha o copiloto no trajeto).

Os registos recolhidos ficam agrupados numa nota que pode possuir qualquer combinação dos três tipos de registo em cima mencionados (voz, fotografias e texto).

4.2. Análise de Requisitos Não Funcionais

Administração

6. É necessário transferir a informação presente no BackOffice relativa a uma atividade para aplicação.

A transferência da informação é possível apenas quando os dispositivos se encontram na mesma rede. Isto permite uma maior segurança uma vez que temos a garantia de que o tráfego não sai de controlo.

7. Uma atividade deve ser planeada anteriormente à sua realização e introduzida no sistema por um gestor.

Os percursos fornecidos pelo cliente devem ser de formato *kml* ou *gpx*.

8. No final de uma atividade deve ser possível gerar um relatório com as notas tomadas durante a mesma, na linguagem utilizada pelos copilotos disponível neste [endereço](#).

Ambos os relatórios devem ser gerados em *PDF*.

9. Portabilidade

A aplicação será desenvolvida para PC.

10. Comunicação

A conexão entre o BackOffice e a aplicação JustTraillt, será realizada utilizando socket TCP com segurança.

Aplicação Android

11. Portabilidade

A aplicação será desenvolvida para a plataforma Android.

12. Consulta de informações

Existe a possibilidade de consultar informações que o batedor considere importantes, desde que haja uma ligação à Web, quer seja por Wi-Fi ou rede móvel.

4.3. Diagramas em UML

Neste capítulo serão apresentados os diagramas descritos em UML, cujo servirão de base para a próxima etapa que será o desenvolvimento concreto das aplicações descritas anteriormente.

Tais diagramas serão apresentados seguindo a ordem aconselhada na UC de Desenvolvimento de Sistemas de Software, sendo apresentados diagramas estruturais, o Modelo de Domínio e diagramas de classes presentes em cada uma das diferentes do sistema global.

De seguida apresentaremos os diagramas funcionais tais como diagramas de máquinas de estado. O diagrama de Use Case que devem representar os requisitos considerados para a aplicação. Por fim para os Use Cases considerados serão apresentados os Diagramas de Sequencia e Diagramas de atividades que pretendem representar os “passos” que o sistema deverá efetuar de modo a que tal Use Case seja executado.

4.3.1 Modelo de Domínio

Com este diagrama é pretendido representar a parte estática dos Sistemas considerados numa fase antes do diagrama de classes, apresentando assim um digrama legível tanto para quem irá desenvolver o sistema como os seus clientes. Tal passo intermédio serve como ponto de controlo entre ambas as partes envolvidas.

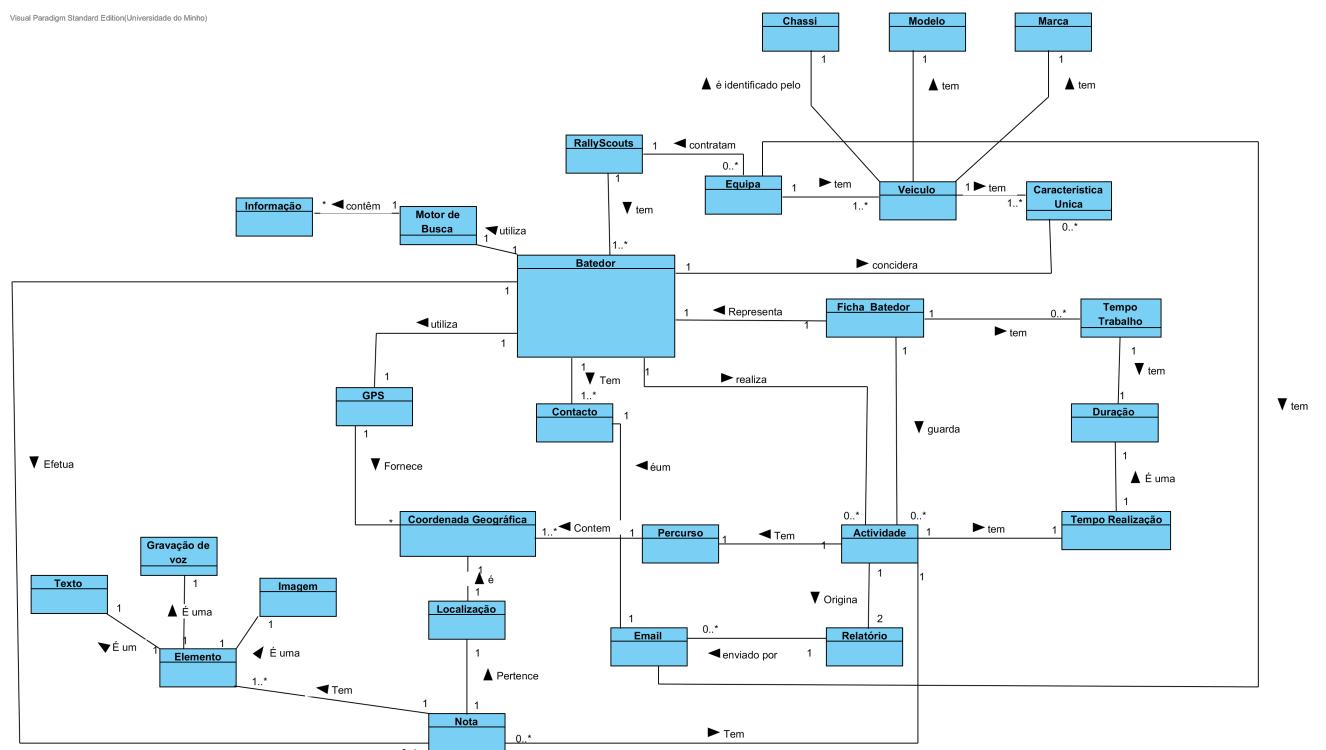


Figura 5 - Modelo Domínio do Sistema

As entidades principais consideradas foram as descritas nos requisitos, representando também as relações entre as mesmas. De destacar o batedor como o ponto principal do sistema assim como as atividades. Pelo diagrama é possível inferir que a atividade é realizada por apenas um batedor, dá origem a relatório relatórios, que poderão ser posteriormente enviados por correio eletrónico para clientes, e outros batedores.

Todas as atividades são compostas por varias notas tomadas durante a execução da mesma no terreno, tais notas podem ser de três tipos, imagem, texto ou clip de voz. Uma nota é também acompanhada das coordenadas geográficas onde foi registada.

A quando da realização caso seja necessário o batedor tem ainda a possibilidade de utilizar um motor de busca de modo a obter alguma informação necessária no momento, estando condicionado pela existência de acesso à rede.

Quando ocorre o término de uma atividade a ficha do batedor responsável pela mesma é atualizada contabilizando assim o total de tempo de trabalho com o tempo despendido na atividade em questão.

4.3.2 Diagramas de Classes

O passo seguinte à construção do modelo de domínio de suporte apresentado, este foi refinado e transformado no respetivo diagrama de classes.

Como se trata de uma aplicação que irá executar em dois sistemas distintos houve a necessidade para uma melhor percepção de fazer a separação em dois diagramas de classes. Um destes diagramas mais completo, com a representação de todas as classes consideradas para o sistema global que estará presente no sistema do Backoffice.

Representamos também o diagrama de classes que funcionará na aplicação JustTraillt, este pode ser considerado uma “vista” do que esta representado no Backoffice. Existem classes com representação parcial das do Backoffice e outras não têm qualquer representação no diagrama de classes da aplicação JustTraillt devido a não haver necessidade da sua representação para as funcionalidades consideradas.

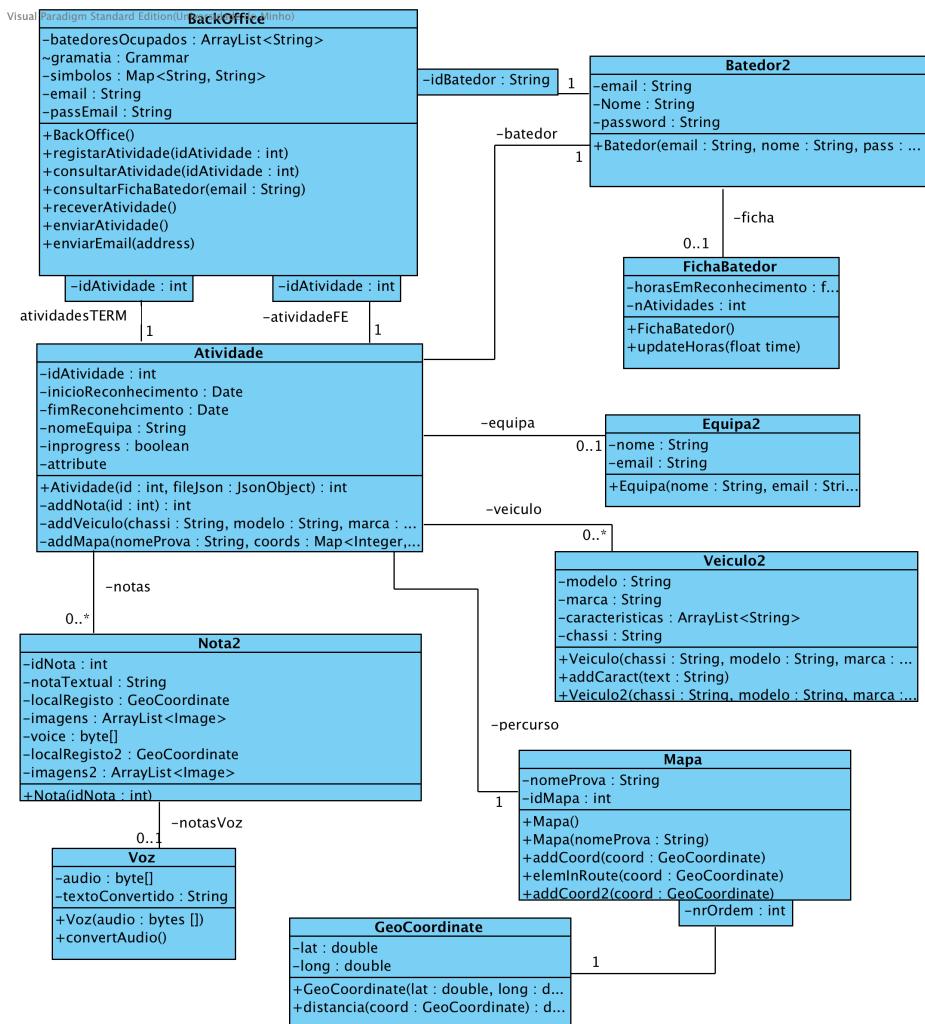


Figura 6 - Diagrama de Classes BackOffice

Com a apresentação dos diagramas de classes concluímos a então os diagramas estruturais representativos dos sistemas considerados, ficando assim a faltar apresentar os diagramas comportamentais que servirão para descrever o comportamento de todo o sistema.

[Diagrama de Classes JustTraillt](#)

4.3.3 Diagrama Use Cases

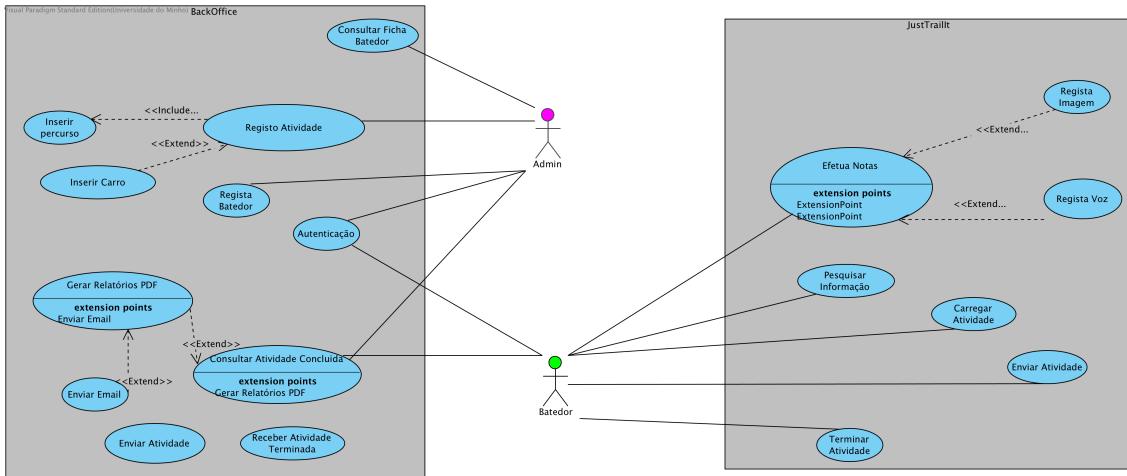


Figura 7 - Diagrama de Use Cases

A apresentação de um diagrama como este tem como propósito tal como o diagrama de modelo de domínio servir como um ponto de controlo se seja entendido facilmente por todas as partes que intervêm na génese do *software*, ou seja os engenheiros e técnicos responsáveis pelo desenvolvimento assim como os clientes.

Com este diagrama apresentamos a representação em forma gráfica das funcionalidades que o sistema deverá suportar descritas a quando do levamento de requisitos.

Tal como os diagramas anteriores este foi dividido em duas partes que constituem o sistema global. Para a aplicação JustTraillt foram considerados os Use Cases que permitem a procura de informação acerca de um assunto na Web, representado por “Pesquisar Informação”, e o conjunto de Use Cases que permitem a gestão e comunicação entre os dois sistemas sendo estes “Carregar Atividade”, “Enviar Atividade” que pretendem representar a troca de mensagens entre os sistemas. Existe também o Use Case “Terminar Atividade” que deverá ser executado quando um batedor termina o reconhecimento da atividade que se encontra a fazer.

Como a funcionalidade principal da aplicação mobile será o registo de notas quando o batedor se encontra em campo a trabalhar, o Use Case que descreve a processo de efetuar o registo de tais notas será apresentado em detalhe de seguida.

Super Use Case	Xsolutionsdevelopment	
Author	28/Abr/2016 16:21:19	
Date		
Brief Description	Use case que descreve o processo de realizar uma nota	
Preconditions	Atividade recebida do backoffice	
Post-conditions	Nota efetuada	
Flow of Events	Actor Input	System Response
	1 Solicita registo de nota	
	2	Guarda localização atual
	3	Solicita anotação textual
	4 Insere anotação textual	<<extend>> Regista Imagem
	5	<<extend>> Regista Voz
	6	
	7	Regista atividade com a localização guardada
	8	Informa que nota foi guardada com sucesso
Exceptions 1 [GPS indisponivel] (passo 2)	Actor Input	System Response
	1	Informa que o gps está indisponivel.
Alternative 2 [não inserir nota textual] (passo 3)	Actor Input	System Response
	1	Regressa a 4
Alternative 1 [mais que uma imagem] (passo 6)	Actor Input	System Response
	1	Regressa a 4
Exception 2 [nao existe um tipo de nota] (passo 7)	Actor Input	System Response
	1	Informa que tem de inserir pelo menos um tipo de nota

Figura 8 - Use Case Tabular Regista Nota

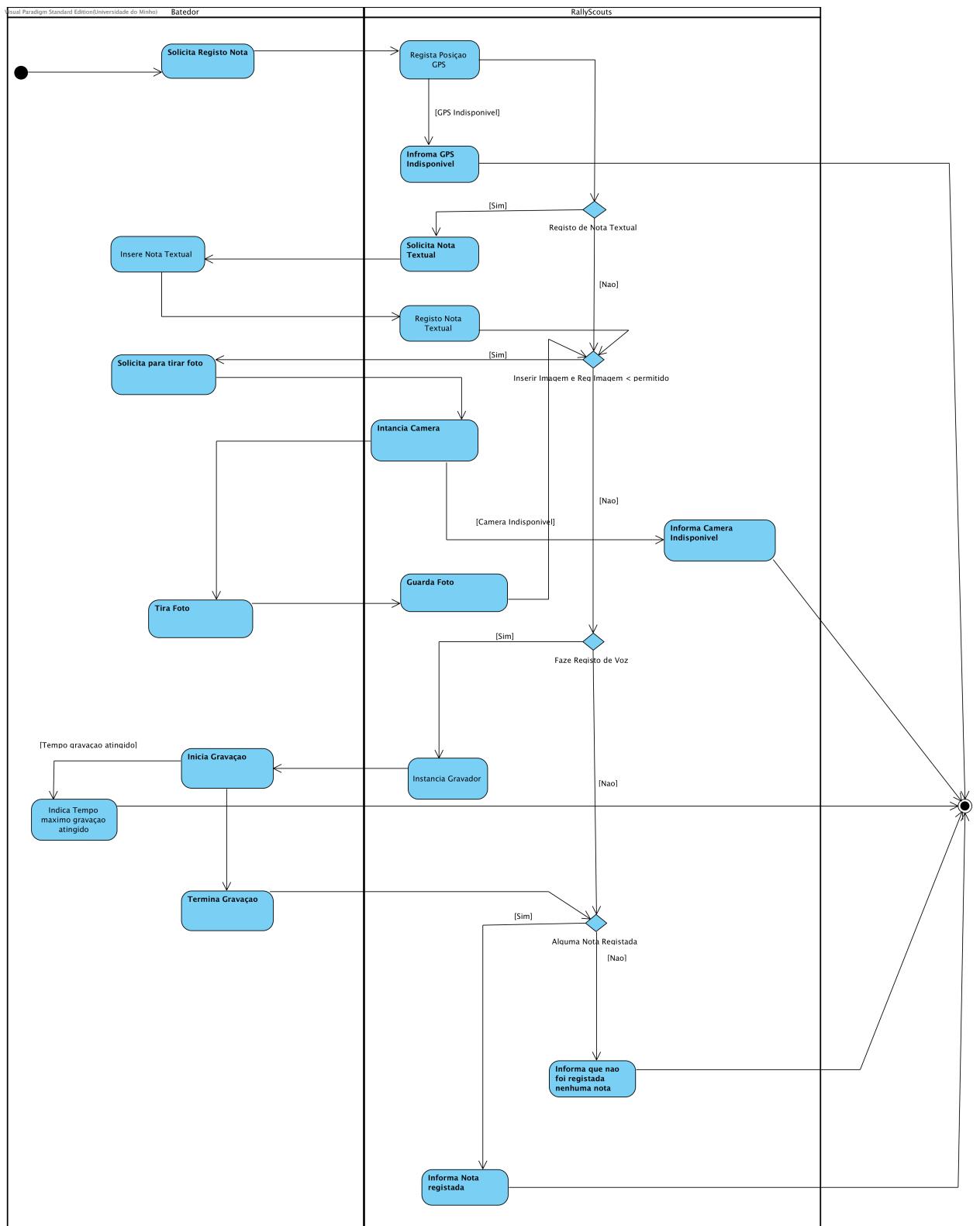


Figura 9 - Diagrama de Atividade Efetua Nota

[Diagrama de Sequência Efetua Nota](#)

[Diagrama de Sequência Regista Imagem](#)

[Diagrama de Sequência Regista Voz](#)

Com a apresentação de estes diagramas para a descrição de como uma nota deverá ser efetuada em vários pontos de vista de vista com diferentes nível de detalhe. Tais níveis de detalhe são destinados aos diferentes intervenientes no processo de criação.

Posto isto damos por concluída a documentação do sistema da aplicação JustTraillt.

- **BackOffice**

Focando-nos agora no sistema do Backoffice e nos Use Cases a ele associados verificamos a existência de mais Use Cases do que na aplicação JustTraillt isto é devido por ser a aplicação principal do sistema.

Para o Backoffice e as funcionalidades que este deverá suportar tais como a gestão dos batedores especificamos dois Use Cases distintos para esta tarefa sendo estes “Registar um batedor” e “Consultar Ficha Batedor”.

O Backoffice deve também como o JustTraillt e trocar informações com a outra aplicação do sistema, JustTraillt esta comunicação é assegurada através dos Use Cases de “Enviar Atividade” e “Receber Atividade Terminada”.

Para o Backoffice como atividades mais importantes e com descrição a ter em consideração decidimos que o registo de uma atividade e a consulta de uma atividade seriam as mais importantes a ter em consideração.

Para a ação efetuada pelo Admin de registar uma atividade para posteriormente ser efetuada é executado o Use Case “Registo de Atividade”. Nesta altura são inseridas no sistema do BackOffice todas as informações relacionadas com uma atividade como o nome da equipa e email que ordenou a atividade, o mapa fornecido que representa o percurso a fazer reconhecimento e também caso existam veículos associados à atividade estes também serão inseridos.

Super Use Case		
Author	Xsolutionsdevelopment	
Date	26/Abr/2016 21:09:28	
Brief Description	Use Case que descreve o registo de uma atividade para posteriormente ser executada por um batedor	
Preconditions		
Post-conditions	Atividade registada	
Flow of Events	Actor Input	System Response
	1 Solicita registo de atividade	
	2	Solicita nome e email da equipa que pediu relatorio
	3 Fornece o nome e o email	
	4	<<include>> Inserir Percurso
	5	<<extend>> Inserir Carro
	6	Fornece lista de batedores e solicita a escolha de um batedor
	7 Escolhe batedor	
	8	Regista atividade
Exception 1 [lista vazia] (passo 6)	Actor Input	System Response
	1	Informa que é necessário registar um batedor previamente.
Alternative 1 [mais que um carro] (passo 6)	Actor Input	System Response
	1	Regressa 5

Figura 11 - Use Case Tabular Registo de Atividade

Super Use Case	Registrar Atividade	
Author	Xsolutionsdevelopment	
Date	27/Abr/2016 13:24:01	
Brief Description	Use Case que descreve o processo de inserção de um percurso numa atividade	
Preconditions		
Post-conditions		
Flow of Events	Actor Input	System Response
	1	Solicita nome da prova
	2 Insere nome da prova	
	3	Solicita ficheiro com as coordenadas
	4 Insere path para o ficheiro	
	5	Processa ficheiro
	6	Regista coordenadas
Exception 1 [extensão incorreta] (passo 3)	Actor Input	System Response
	1	Informa que o ficheiro inserido tem uma extensão inapropriada
Exception 2 [erro aceder ficheiro] (passo 3)	Actor Input	System Response
	1	Informa que ocorreu um erro ao aceder ao ficheiro
Exception 3 [erro ler ficheiro] (passo 3)	Actor Input	System Response
	1	Informa que ocorreu um erro ao ler ao ficheiro

Figura 10 - Use Case Tabular Inserir Percurso

Use Case Tabular Inserir Carro

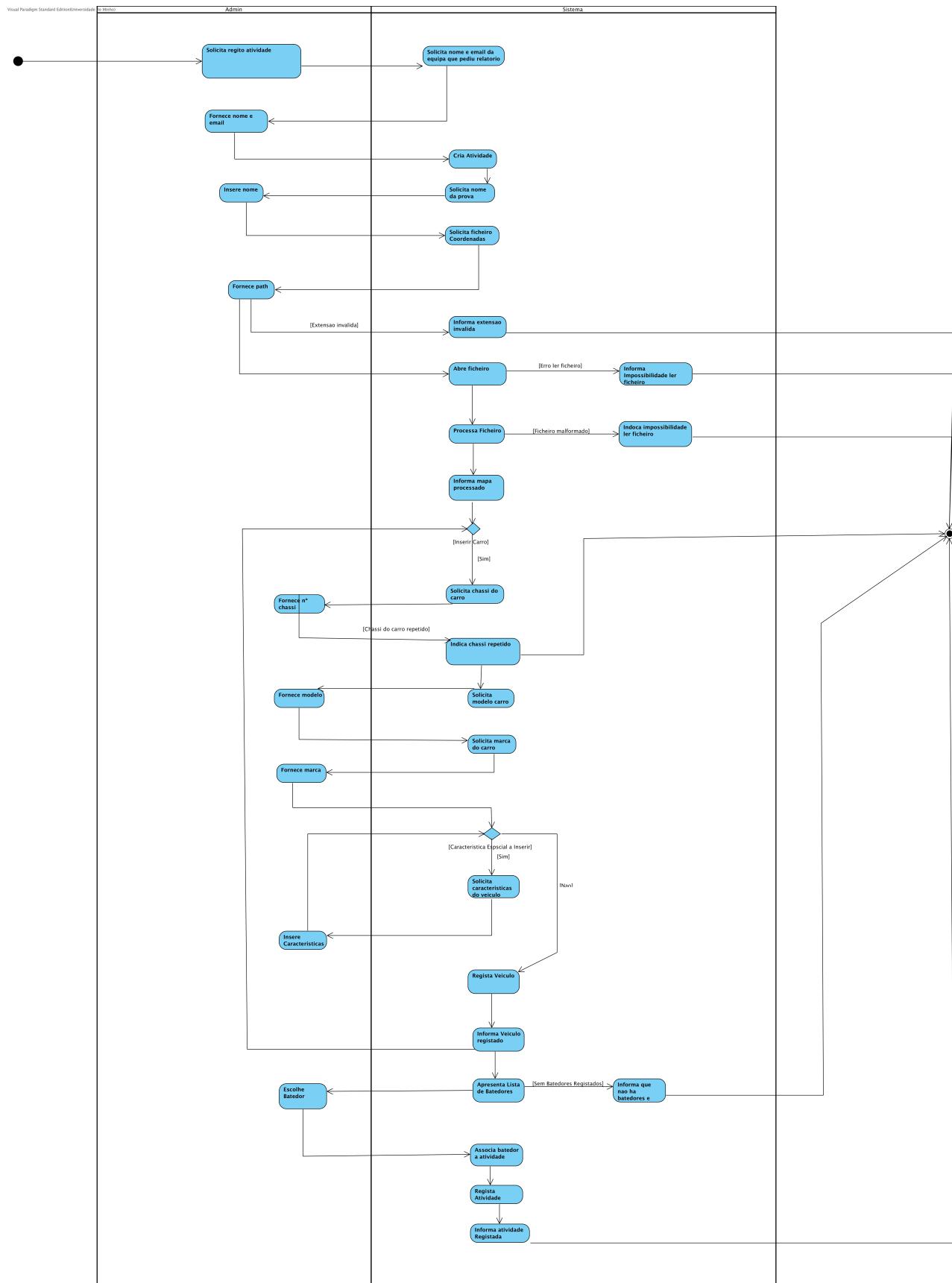


Figura 12 - Diagrama de Atividade Regista Atividade

[Diagrama de Sequência Regista Atividade](#)

[Diagrama de Sequência Insere Percurso](#)

[Diagrama de Sequência Insere Carro](#)

Outra importante função do BackOffice será depois de um atividade estar concluída, o sistema deve permitir a consulta dos detalhes de uma atividade. Caso o ator também pretenda relatórios em PDF estes deverão ser gerados pelo sistema para uma localização indicada por quem ordenou a criação dos mesmos.

Quando os relatórios estiverem gerados existe a possibilidade de estes serem enviados por *email*, para outro batedor ou para a equipa que solicitou a atividade.

Super Use Case		
Author	Xsolutionsdevelopment	
Date	27/Abr/2016 16:20:07	
Brief Description	Use Case que descreve o processo de consulta de toda a informação de uma atividade	
Preconditions	Existem atividades concluidas	
Post-conditions		
Flow of Events	Actor Input	System Response
	1 Indica intenção de consultar atividade concluida	
	2	Apresenta lista de atividades concluidas
	3 Seleciona atividade	
	4	Apresenta informações da atividade
	5 Seleciona nota	
	6	Apresenta detalhes da nota
	7	<<extend>> Gerar Relatorios
Alternative 1 [nao selecionar nota] (passo 5)	Actor Input	System Response
	1	Regressa a 7

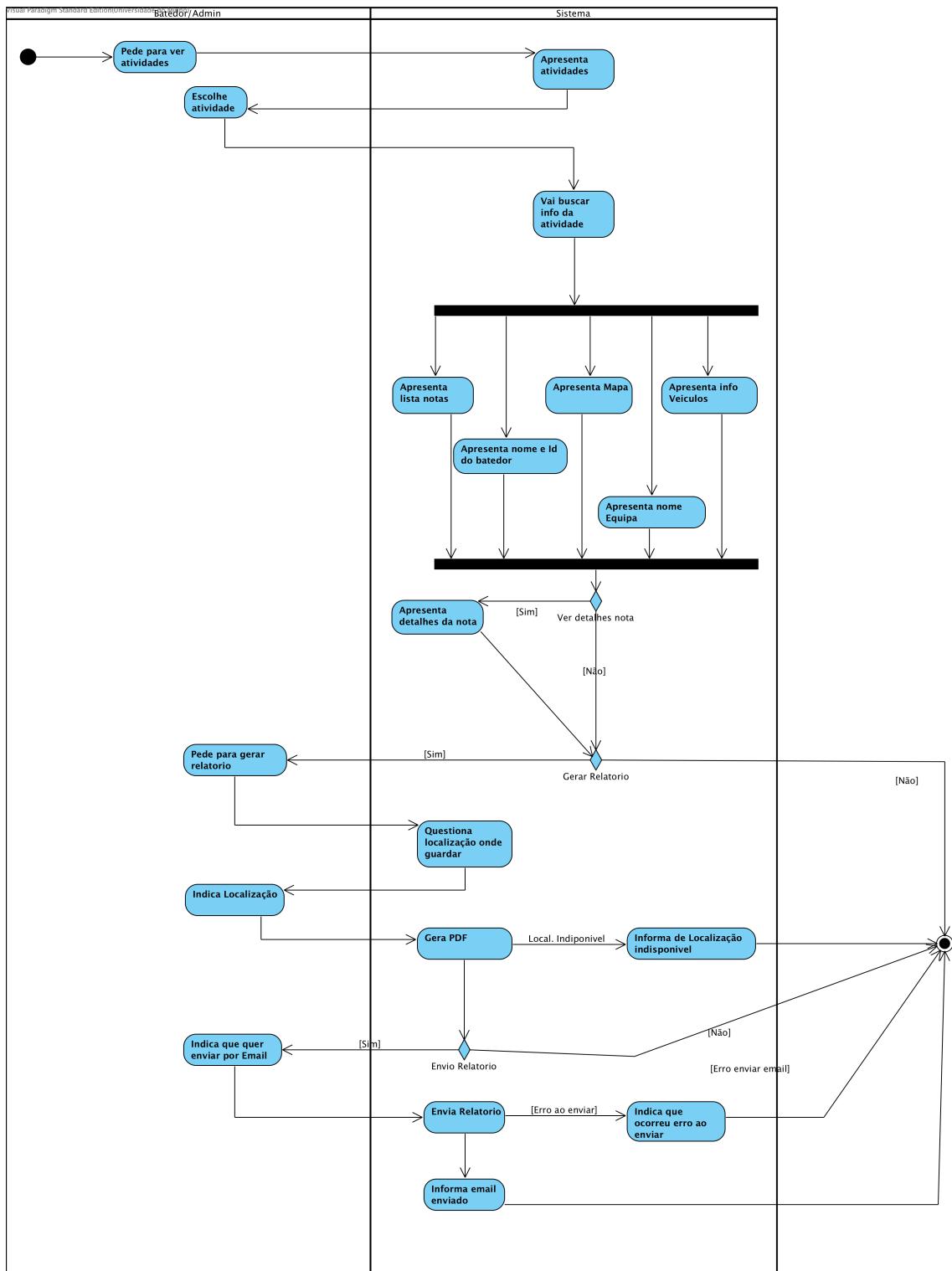
Figura 13 - Use Case Tabular Consulta Atividade Concluida

Super Use Case	Consultar Atividade Concluida	
Author	Xsolutionsdevelopment	
Date	27/Abr/2016 16:58:00	
Brief Description	Use Case que descreve o processo de geração do relatorio da atividade	
Preconditions	Atividade Concluida	
Post-conditions		
Flow of Events	Actor Input	System Response
	1	Solicita a localização na qual deve guardar os ficheiros
	2	Indica a localização
	3	Gera relatórios
	4	Indica os path para os ficheiros criados
	5	<<extend>> EnviarEmail
Exception 1 [erro localização] (passo 3)	Actor Input	System Response
	1	Comunica que a localização fornecida é inválida

Figura 15 - Use Case Tabular Gerar Relatório

Super Use Case	Gerar Relatorios	
Author	Xsolutionsdevelopment	
Date	27/Abr/2016 17:20:18	
Brief Description	Use case que descreve o processo de enviar por email os relatórios gerados	
Preconditions	Relatorios Gerados e email do remetente e do destinatário validos	
Post-conditions		
Flow of Events	Actor Input	System Response
	1 Solicita o envio do relatorio por email	
	2	Enviar email
	3	Informa que foi enviado o email
Exception 1 [erro enviar email] (passo 2)	Actor Input	System Response
	1	Indica que ocorreu um erro no envio do email

Figura 14 - Enviar Relatório Email



[Diagrama de Sequência Consultar Atividade Concluida](#)

[Diagrama de Sequência Gerar Relatório](#)

[Diagrama de Sequência Enviar Relatório](#)

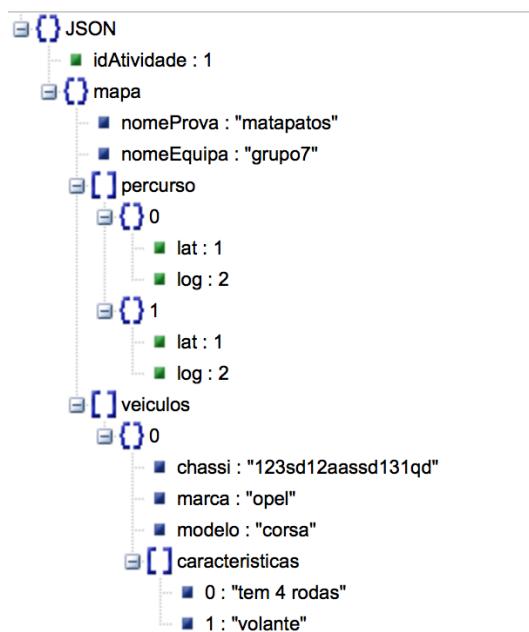
4.3.4 Protocolo comunicação

Como referido anteriormente o sistema a ser desenvolvido tratasse de um sistema distribuído em duas aplicações.

Posto isto no momento de sincronização e troca de dados entre as duas aplicações desenvolvidas é necessário a existência de um protocolo de comunicação, este protocolo deverá ter em consideração a ordem da troca de mensagens, bem como o formato das mensagens que deverem ser reconhecidas entre as duas aplicações intervenientes na comunicação.

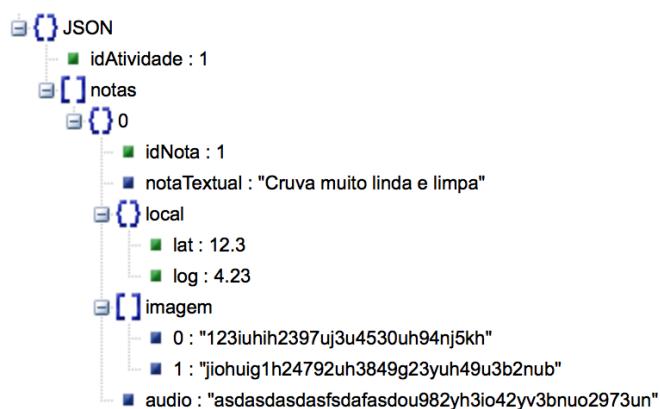
A escolha do formato das mensagens recaiu na utilização de ficheiros [JSON](#) devido a ser um formato com grande utilização e de fácil utilização.

Existem dois formatos de ficheiro JSON, um para cada sentido da comunicação, que serão apresentados em seguida. Estes ficheiros permitem também o envio de códigos de erro quando o campo “idAtividade” for composto por um número negativo.



O formato do ficheiro JSON enviado do BackOffice para a aplicação JustTraillt será formado pelo campo “idAtividade” que em condições normais é o identificador único da atividade em todo o sistema. Caso quando seja requisitada uma atividade ao BackOffice por parte de um batedor na sua instância de JustTraillt o este campo deverá ser atribuído o valor -1, e os restantes campos devem ser ignorados. De seguida é enviado o mapa e todas as restantes informações necessárias para o batedor, como a identificação da equipa e da prova, o conjunto das coordenadas que constituem o mapa do percurso a realizar. Por fim é também enviado a informação dos veículos que estão associados ao reconhecimento.

idAtividade	Significado
>0	Conteúdo normal
-1	Sem atividade disponível para reconhecimento
-2	Atividade recebida OK
-3	Erro Atividade Recebida



Para as comunicações destinadas ao BackOffice por parte da aplicação JustTraillt o existem dois tipos de JSON que podem ser enviados.

O JSON representativo de uma atividade concluída cujo a informação transportada indicará qual a atividade a que se refere pelo campo “idAtividade”, após isso são enviadas todas as notas tomadas durante o reconhecimento da atividade.

Quando algum batedor deseja uma nova atividade para ser realizada, o pedido enviado ao BackOffice deverá também ser feito em formato JSON onde apenas existe um campo “email” que transportará o email do batedor que pediu por nova atividade.

Estando concluído assim o formato das mensagens a serem envidas será apresentado o protocolo de troca de mensagens entre os dois sistemas envolvidos.

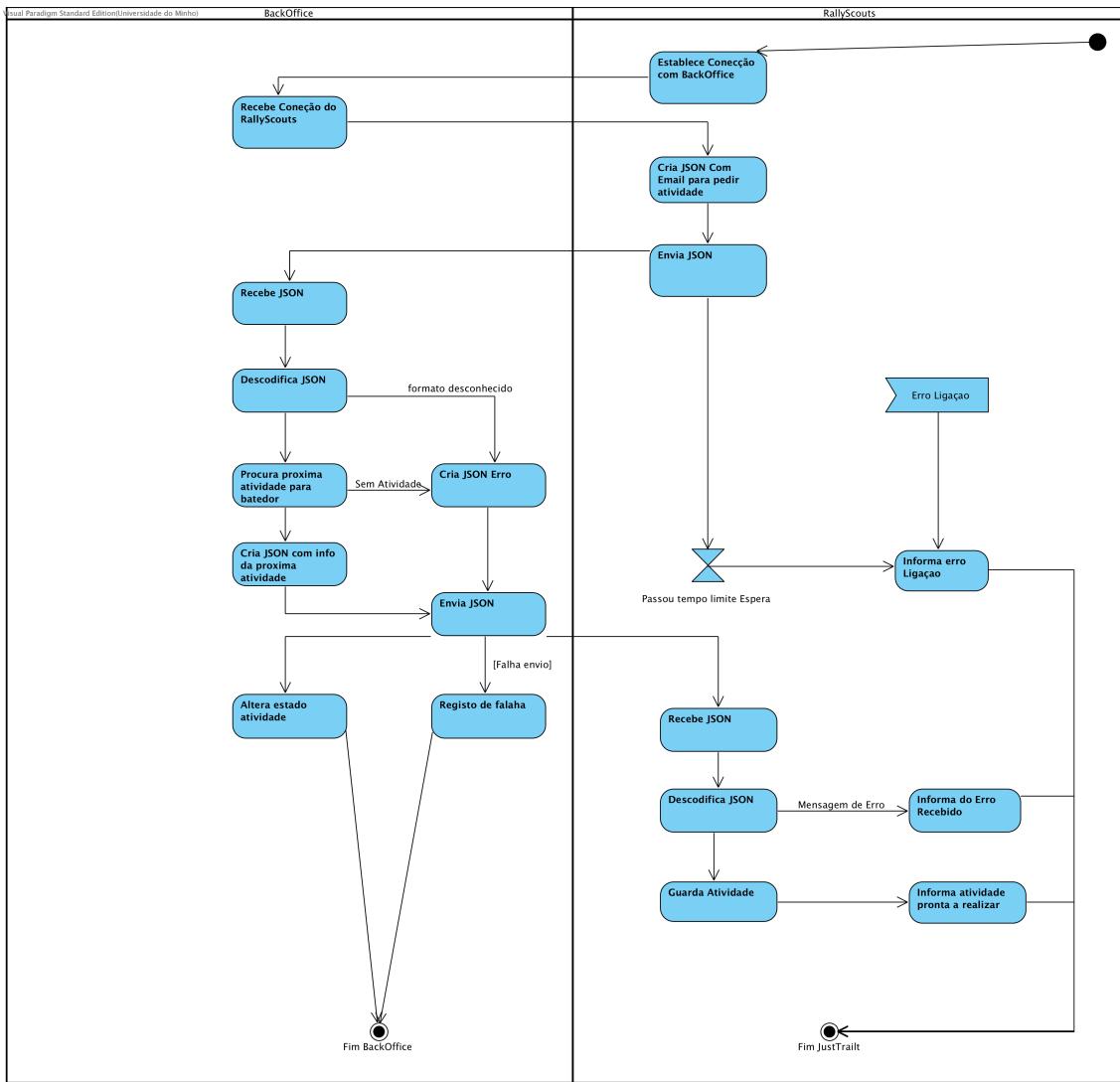


Figura 16 - Diagrama de Atividade ilustrativo de pedido de nova atividade

[Diagrama de Sequência Backoffice associado à comunicação](#)

[Diagrama de Sequência JustTraill associado à comunicação](#)

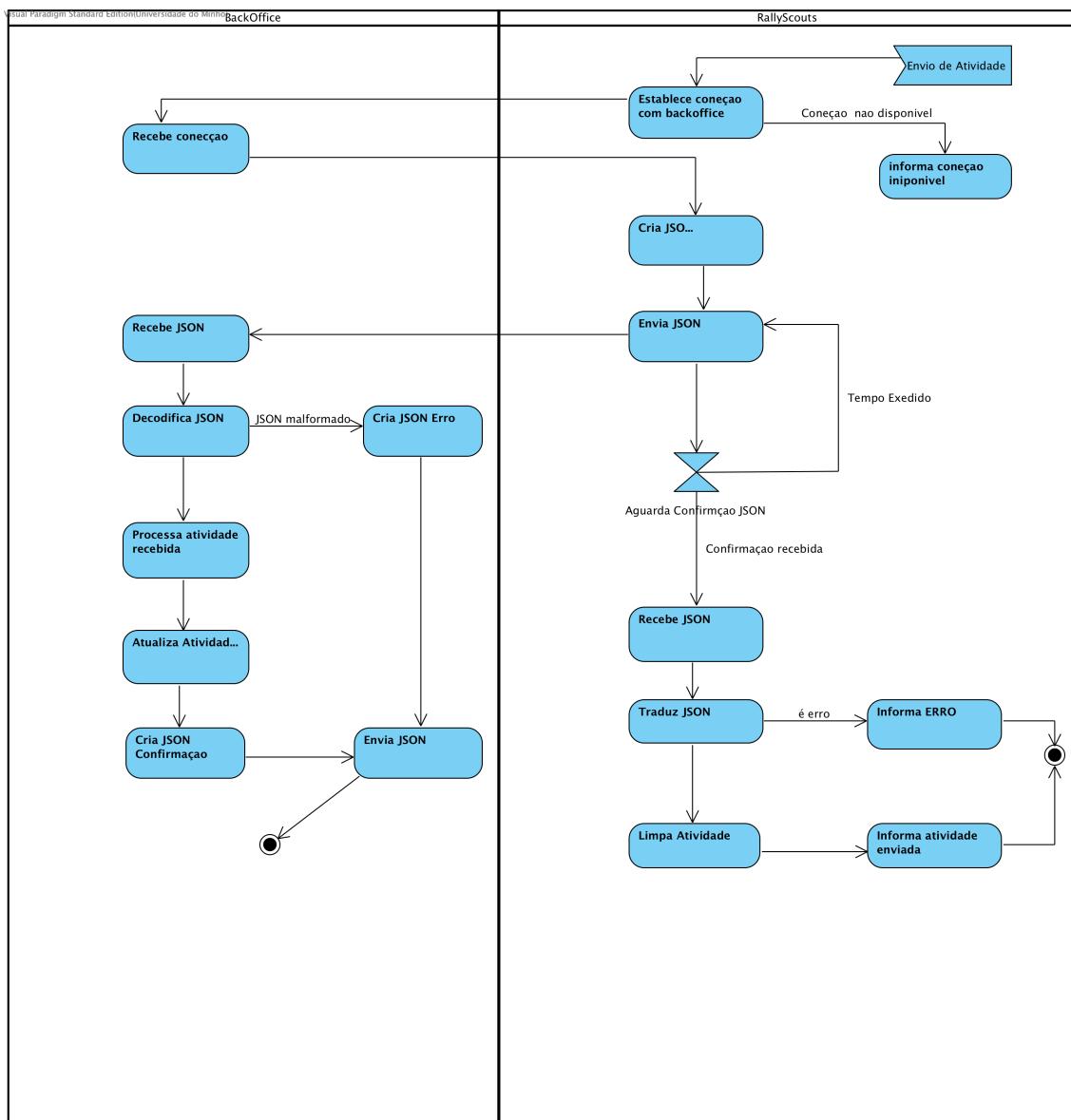


Figura 17 - Enviar Atividade Concluída

[Diagrama de Sequência Envio dados p/ BackOffice](#)

[Diagrama de Sequência Receber atividade terminada](#)

Com a apresentação dos diagramas de atividade que visam a representar a sequência de trocas de mensagens durante o processo de comunicação conclui-se assim o que há a referir acerca de protocolos de comunicação entre os sistemas.

4.3.5 Estados de batedores e atividades no BackOffice

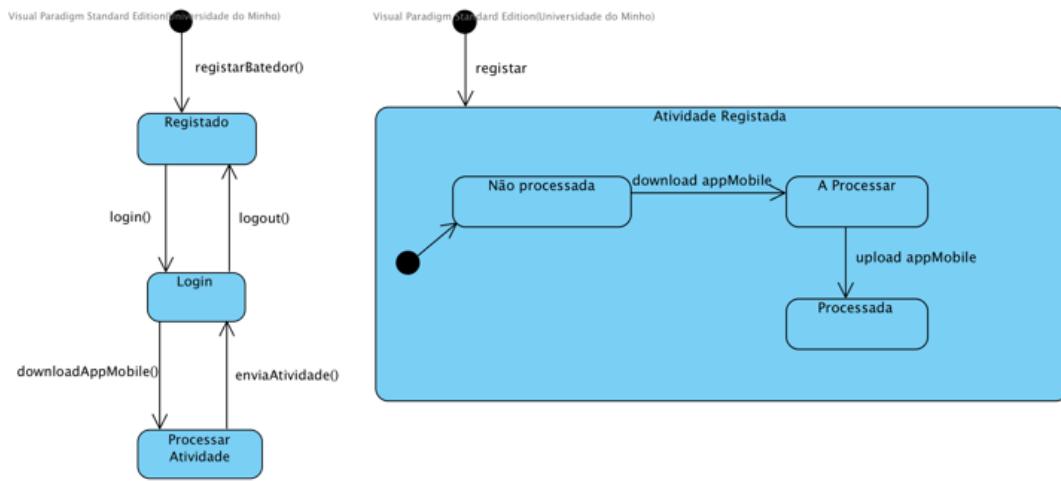


Figura 18 - Máquinas de Estado para Batedor e Atividade

Considerando a existência das entidades principais do sistema de BackOffice como sendo os Batedores e as Atividades nele registadas, torna-se assim necessário efetuar a representação dos estados que estes podem assumir durante a sua vida.

Acima encontram-se apresentados os dois diagramas de máquinas de estado que representam todos os estados possíveis das entidades referidas.

Os estados admissíveis para um batedor são o estado de registado, o seu estado *default* que é atingido quando acontece o seu registo. Uma vez registado um batedor apenas tem a possibilidade de passarão estado de login indicando que este de encontra a trabalhar com o sistema. Ao fazer o *download* de uma atividade para a sua aplicação JustTraillt o seu estado passa a ser que este se encontra a processar uma atividade. Quando um batedor retorna ao BackOffice com a atividade terminada e faz o *upload*, o seu estado passa novamente a login, uma vez terminados os trabalhos este efetua *logout* e volta ao estado de registado.

Como consideração importante visto não ser permitido a remoção de utilizadores do sistema estes não possuem nenhuma saída possível do sistema, estando sempre em algum dos estados referidos.

Relativamente as entidades e aos seus estados admissíveis tal como os batedores, as atividades uma vez registadas não podem ser removidas do sistema. Uma atividade passa a ser considerada no sistema como “Registada” através do seu registo por parte do administrador. Ao entrar no estado de registada a atividade é automaticamente inserida no estado de não processada pois de encontra-se à espera que o batedor responsável pelo seu reconhecimento a descarregue para a aplicação JustTraillt. Quando tal acontecimento ocorre a atividade passa para o estado “A processar”, representativo que alguém está a trabalhar nela. Quando o termo de reconhecimento de uma atividade é feito e esta é enviada novamente

para o Backoffice, é efetuado o seu processamento final. No final do processamento por parte do Backoffice a atividade é arquivada no sistema passando assim ao estado de “Processada”.

4.4. Construção da Base de Dados do BackOffice

4.4.1 Construção do modelo Conceptual de dados

I. Identificação dos atributos

- Batedor

Atributos	Descrição	Sinónimos	Tipo de Dados e Tamanho	Nulo	Multi-Valorado	Derivado	Composto	Valor por Defeito
Email	Identificação única de um Batedor	Endereço Eletrónico	50 caracteres variáveis	Não	Não	Não	Não	-
Nome	Indica o nome do batedor	Identificação	50 caracteres variáveis	Não	Não	Não	Não	-
Password	Indica a password do batedor	-	50 caracteres variáveis	Não	Não	Não	Não	-
FichaBatedor	Indica os dados de um batedor	-	-	Não	Não	Não	Sim	-
HorasDeReconhecimento	Indica as horas de reconhecimento de um batedor	Horário efetuado	Float	Não	Não	Não	Não	0
N_Atividades	Indica o nº de atividades efetuadas por um batedor	-	Int	Não	Não	Não	Não	0

Tabela 1 - Atributos da entidade Batedor

- Veiculo

Atributos	Descrição	Sinónimos	Tipo de Dados e Tamanho	Nulo	Multi-Valorado	Derivado	Composto	Valor por Defeito
Chassi	Identifica única de um veiculo	-	50 caracteres variáveis	Não	Não	Não	Não	-
Marca	Indica a marca de um veiculo	-	50 caracteres variáveis	Sim	Não	Não	Não	-
Modelo	Identifica o modelo de um veículo	-	50 caracteres variáveis	Sim	Não	Não	Não	-
Características	Identifica as características de um veiculo	Informação	400 caracteres variáveis	Sim	Sim	Não	Não	-

Tabela 2 - Atributos da entidade Veiculo

- **Mapa**

Atributos	Descrição	Sinónimos	Tipo de Dados e Tamanho	Nulo	Multi-Valorado	Derivado	Composto	Valor por Defeito
Id_Mapas	Identifica única de um mapa	-	Int	Não	Não	Não	Não	-
NomeProva	Indica o nome de uma prova	-	50 caracteres variáveis	Não	Não	Não	Não	-
Coordenadas	Representa as coordenadas de um mapa	-	-	Sim	Sim	-	Sim	-
NrCoordenada	Identificação única de uma coordenada	-	Int	Não	Não	Não	Não	-
Longitude	Indica a longitude de uma coordenada	-	Float	Não	Não	Não	Não	-
Latitude	Indica a latitude de uma coordenada	-	Float	Não	Não	Não	Não	-

Tabela 3 - Atributos da entidade Mapa

- **Nota**

Atributos	Descrição	Sinónimos	Tipo de Dados e Tamanho	Nulo	Multi-Valorado	Derivado	Composto	Valor por Defeito
Id_Nota	Identifica um dicionário	-	Int	Não	Não	Não	Não	-
NotaTextual	Representa as notas tiradas pelo batedor	Descrição	Text	Sim	Não	Não	Não	-
Imagens	Representa as imagens tiradas pelo batedor	fotos	Varbinary	Sim	Sim	Não	Não	-
Audio	Representa o áudio gravado pelo batedor	Som	VarBinary	Sim	Não	Não	Não	-
TextoConvertido	Representa a nota traduzida para texto	Texto	Text	Sim	Não	Não	Não	-
Coordenadas	Indica as coordenadas de uma nota	Posição	-	Não	Não	Não	Sim	-
Latitude	Indica a latitude de uma nota	-	Float	Não	Não	Não	Não	-
Longitude	Indica a longitude de uma nota	-	Float	Não	Não	Não	Não	-

Tabela 4 - Atributos da entidade Nota

- **Atividade**

Atributos	Descrição	Sinónimos	Tipo de Dados e Tamanho	Nulo	Multi-Valorado	Derivado	Comp osto	Valor por Defeito
Id_Atividade	Identificação única de uma atividade	-	Int	Não	Não	Não	Não	-
InicioReconhecimento	Indica o inicio do reconhecimento	-	Datetime	Sim	Não	Não	Não	-
FimReconhecimento	Indica o fim do reconhecimento	-	Datetime	Sim	Não	Não	Não	-
inProgress	Verifica de uma atividade se encontra em progresso	-	Bit	Não	Não	Não	Não	-
Equipa_Email	Identifica o email da equipa	Endereço Eletrónico	50 caracteres variáveis	Não	Não	Não	Não	-
Equipa_Nome	Identifica o nome da equipa	-	50 caracteres variáveis	Não	Não	Não	Não	-

Tabela 5 - Atributos da entidade Atividade

II. Identificação das entidades

De modo a demonstrar todas as entidades em causa neste projeto, vamos focar na entidade "principal" do projeto, a Atividade. Esta será a nossa entidade principal, e todas as outras entidades irão relacionar-se com esta, quer seja de forma direta ou indireta.

- **Atividade**

Descrição	Sinónimo	Ocorrência
Tabela utilizada para descrever uma Atividade presente no sistema.	-	Uma atividade é representativa de uma única atividade do sistema.

Tabela 6 - Entidade Atividade

- **Nota**

Descrição	Sinónimo	Ocorrência
Tabela utilizada para descrever uma Nota presente no sistema.	Comentário	Uma nota é usada para descrever uma única Atividade.

Tabela 7 - Entidade Nota

- **Batedor**

Descrição	Sinónimo	Ocorrência
Tabela utilizada para descrever um Batedor presente no sistema.	-	Um Batedor pode reconhecer vários percursos, um de cada vez.

Tabela 8 - Entidade Batedor

- **Veiculo**

Descrição	Sinónimo	Ocorrência
Tabela utilizada para descrever um Veículo presente no sistema.	-	Um veículo é utilizado numa Atividade.

Tabela 9 - Entidade Veiculo

- **Mapa**

Descrição	Sinónimo	Ocorrência
Tabela utilizada para descrever um Mapa presente no sistema.	-	Um mapa é utilizado por um Batedor numa Atividade.

Tabela 10 - Entidade Mapa

III. Identificação das Relações

Nome da Entidade	Multiplicidade	Relação	Multiplicidade	Nome da Entidade
Atividade	(0,n)	Efetua	(0,1)	Batedor
Atividade	(1,1)	Tem	(0,n)	Nota
Atividade	(1,1)	Tem	(0,n)	Veiculo
Atividade	(0,n)	Tem	(1,1)	Mapa

Tabela 11 - Identificação das relações

IV. Determinar atributos candidatos chave, chaves primárias e alternativas

Cada entidade deverá ter um atributo que a identifique com unicidade. Para isso é necessário identificar chaves candidatas, ou seja, conjuntos de atributos que poderão identificar uma ocorrência da entidade respetiva de modo único. A chave candidata que melhor se adequa para o efeito é escolhida como chave primária. As chaves candidatas restantes serão as chaves alternativas.

No processo de escolha de uma chave primária entre as chaves candidatas é preciso ter em conta vários fatores, como por exemplo, a probabilidade de os seus valores serem alterados, o seu número de caracteres, entre outros.

A seguir representam-se todas as entidades, respetivas chaves primárias (explicitando a razão de sua escolha) e as chaves alternativas, se existentes.

Atividade

A chave primária escolhida para representar a entidade Atividade, foi o atributo 'id_Atividade'. Esta é a única chave candidata uma vez que, apesar do atributo 'Email' ser único tal como o 'id_atividade', este é anulável.

Mapa

A chave primária escolhida para representar a entidade Mapa, foi o atributo 'id_Map', uma vez que é o único atributo que representa unicamente a mesma.

Nota

A chave primária escolhida para representar a entidade Nota, foi o atributo 'id_Nota', uma vez que é o único atributo que representa unicamente a mesma.

Batedor

A chave primária escolhida para representar a entidade Batedor, foi o atributo 'Email', uma vez que é o único atributo que representa unicamente a mesma.

Veiculo

A chave primária escolhida para representar a entidade Veiculo, foi o atributo 'Chassi', uma vez que, uma vez mais, é o único atributo que representa unicamente a mesma.

V. Validar modelo Conceptual

Para validar o modelo conceptual é necessário verificar se todas as transações previstas pelos requisitos anteriormente definidos são possíveis no modelo conceptual projetado.

- **Consultar informação batedor**

As informações de uma UC estão contidas na entidade ‘UC’, logo é possível obter a lista dessas informações.

- **Inserir Batedor**

A inserção de um batedor, torna-se possível através da entidade batedor.

- **Receber atividade terminada**

Fazer upload de uma atividade para o Backoffice, é possível através da consulta de uma atividade e posterior inserção das notas da atividade em questão. A inserção das notas é conseguida através da entidade Nota.

- **Consultar atividade**

A consulta de uma atividade é possível através da consulta de todas as entidades, uma vez que uma atividade tem um Veiculo, Mapa, Nota e ainda está associada a um Batedor.

- **Inserção de uma atividade**

A inserção de uma atividade torna-se possível através da entidade atividade.

Ao inserir uma atividade, as entidades alteradas são, respetivamente, a entidade Atividade, Veiculo e Mapa. A ordem de alteração destas entidades, deve-se ao facto de uma atividade ter de existir para ser possível a inserção de um Veiculo ou de um Mapa.

VI. Desenho do diagrama entidade-relacionamento

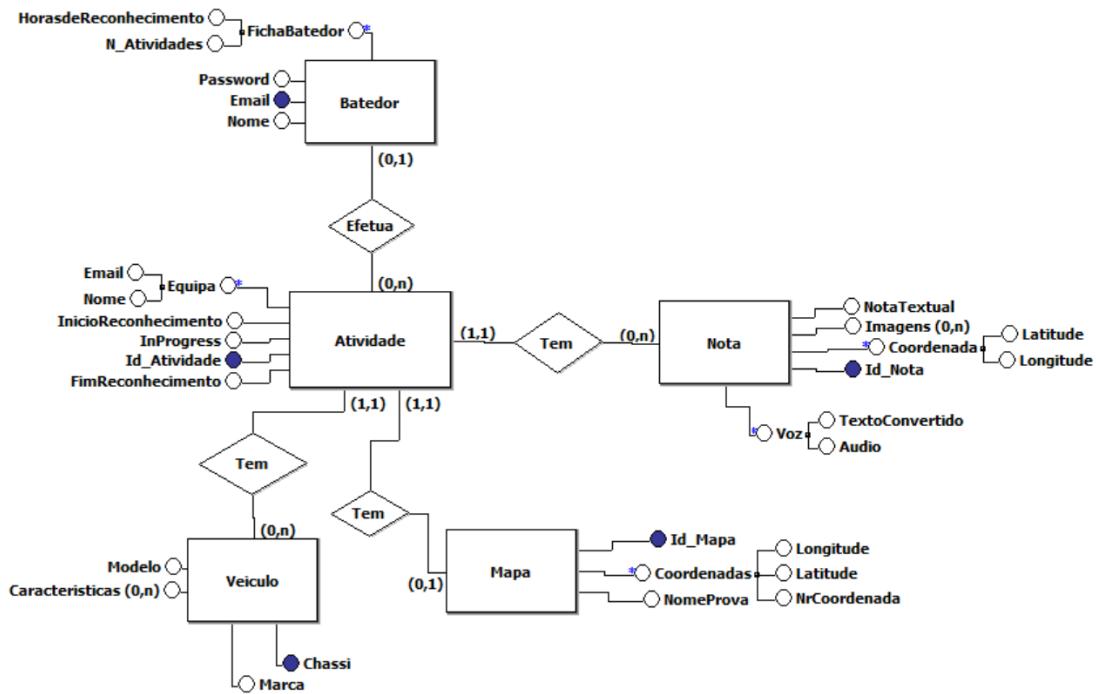


Figura 19 - Modelo Conceptual da Base de Dados do BackOffice

4.4.2 Construir e validar o modelo lógico de dados

I. Derivar relações para o modelo lógico de dados

Entidades fortes

No nosso projeto, podemos considerar como entidades fortes aquelas que possuem existência própria, ou seja, a entidade *Atividade* e *Batedor*.

Entidades fracas

No nosso projeto, podemos considerar como entidades fracas aquelas que não possuem existência própria, ou seja, a entidade *Nota*, *Mapa* e *Veiculo*.

Relacionamentos binários 1:N

Para cada um relacionamento binário 1:N, a entidade no "lado 1" da relação é designada como entidade-pai e a entidade no "lado N" é designado como a entidade filho. Para representar essa relação colocamos uma cópia da chave primária da entidade-pai na relação entidade-filho, para atuar como uma chave estrangeira. Esta chave é representada com o nome da entidade-pai.

Relacionamentos binários 1:1

No nosso projeto não existem relacionamentos binários 1:1.

Relacionamentos recursivos 1:1

No nosso projeto não existem relacionamentos recursivos 1:1.

Relacionamentos Superclasse/Subclasse

No nosso projeto não existem relacionamentos Superclasse/Subclasse.

Relacionamentos Binários N:M

Para cada relacionamento N:M foi criada uma relação que representa o relacionamento contendo os atributos que são parte dessa mesma relação. De seguida foi colocada uma cópia da chave primária das entidades que são parte da relação, que irão atuar tanto como chave primaria como chave estrangeira da nova relação.

Relacionamentos Complexos

No nosso projeto não existem relacionamentos complexos.

Atributos Multivalor

De modo a representar os atributos multivalor presentes no projeto, criamos chaves estrangeiras que são cópias das chaves primárias dos atributos em causa. As chaves estrangeiras em causa irão atuar como chave primária do relacionamento em causa.

Esquema lógico de dados

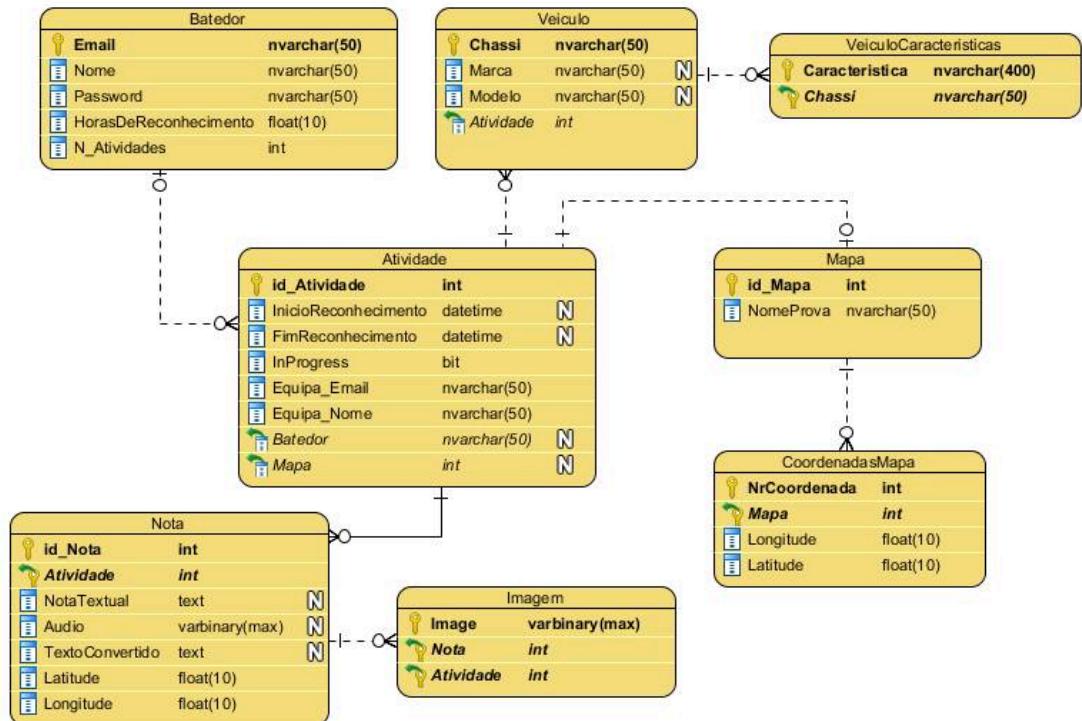


Figura 20 - Modelo Lógico da Base de Dados do BackOffice

[Modelo Lógico da Base de Dados Aplicação JustTraillt](#)

II. Validar Relações usando normalização

Uma parte importante no desenvolvimento de uma base de dados é verificar se está normalizada de modo a evitar redundância dos dados, ocupar espaço desnecessário e erros na sua manipulação. O mais importante na normalização de uma base de dados é testar até à 3^a forma normal (3FN).

Análise do modelo:

Nas dependências a baixo usa-se o sublinhado preenchido e o tracejado para identificar as chaves primárias e as chaves estrangeiras respetivamente. De modo a ser possível representar as chaves estrangeiras que ao mesmo tempo são chaves primárias, foi usado o duplo sublinhado. Estas dependências funcionais ajudam-nos a verificar se o modelo respeita a 3FN.

- **Tabela Batedor**

Email -> Nome, Password, HorasDeReconhecimento, N_Atividades;

- **Tabela Nota**

id_Nota, Atividade -> NotaTextual, Audio, TextoConvertido, Latitude, Longitude;

- **Tabela Imagem**

Image, Nota, Atividade;

- **Tabela Atividade**

id_Atividade -> InicioReconhecimento, FimReconhecimento, inProgress, Equipa_Email, Equipa_Nome, Batedor, Mapa;

- **Tabela Mapa**

id_Mapas -> NomeProva;

- **Tabela CoordenadasMapa**

NrCoordenada, Mapa -> Longitude, Latitude;

- **Tabela Veiculo**

Chassi -> Marca, Modelo, Atividade;

- **Tabela VeiculoCaracteristicas**

Caracteristica, Chassi:

Como em nenhuma tabela se verifica uma relação de transitividade de um atributo em relação à sua chave primária, verifica-se que a nossa base de dados está normalizada até a 3FN.

III. Validar relações contra Queries do utilizador

Após a passagem do modelo conceptual para modelo lógico e a correspondente

normalização dos dados, iremos validar o modelo com as possíveis *queries* do utilizador. Enunciaremos cada transação que achamos pertinente e a sua correspondente validação no modelo.

- **Será possível consultar a informação de um batedor?**

Sim, uma vez que todas as informações de um ‘Batedor’ estão contidas na tabela ‘Batedor’.

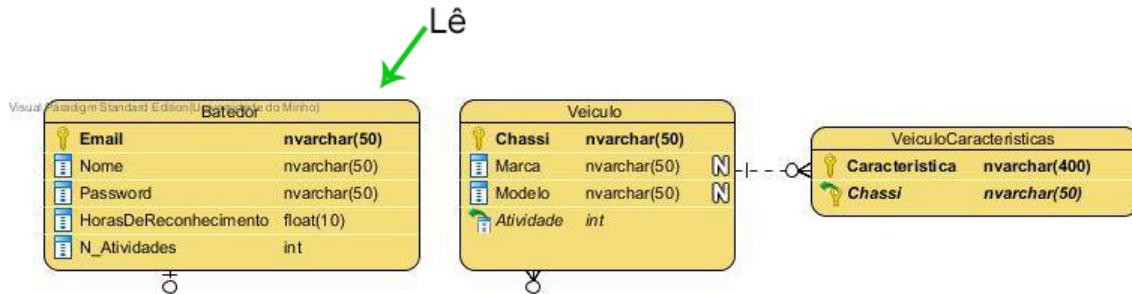


Figura 21 - Mapa validador da Query Consultar Informação de um Batedor

- **Será possível inserir um batedor?**

Sim, através da tabela ‘Batedor’.

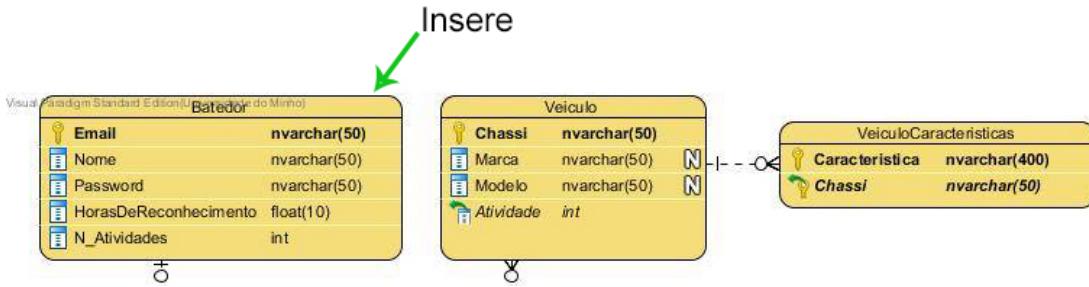


Figura 22 - Mapa validador da Query Inserção de um Batedor

- **Será possível receber uma atividade terminada?**

Sim, através da consulta de uma atividade e posterior inserção das notas da atividade em questão. A inserção das notas é conseguida através da tabela Nota.

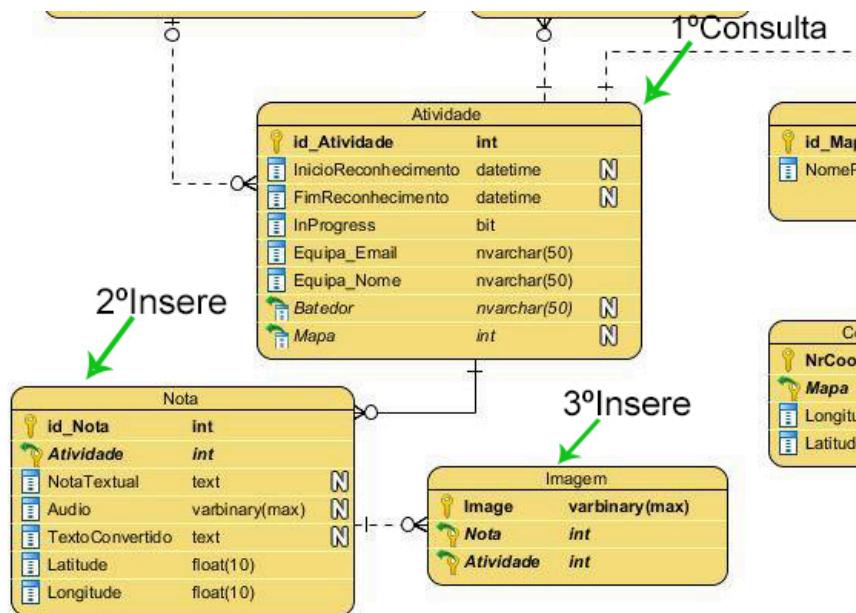


Figura 23 - Mapa validador da Query Receber Atividade Terminada

- **Será possível consultar uma atividade?**

Sim através da consulta de todas as tabelas, uma vez que uma atividade tem um Veiculo, Mapa, Nota e ainda está associada a um Batedor.

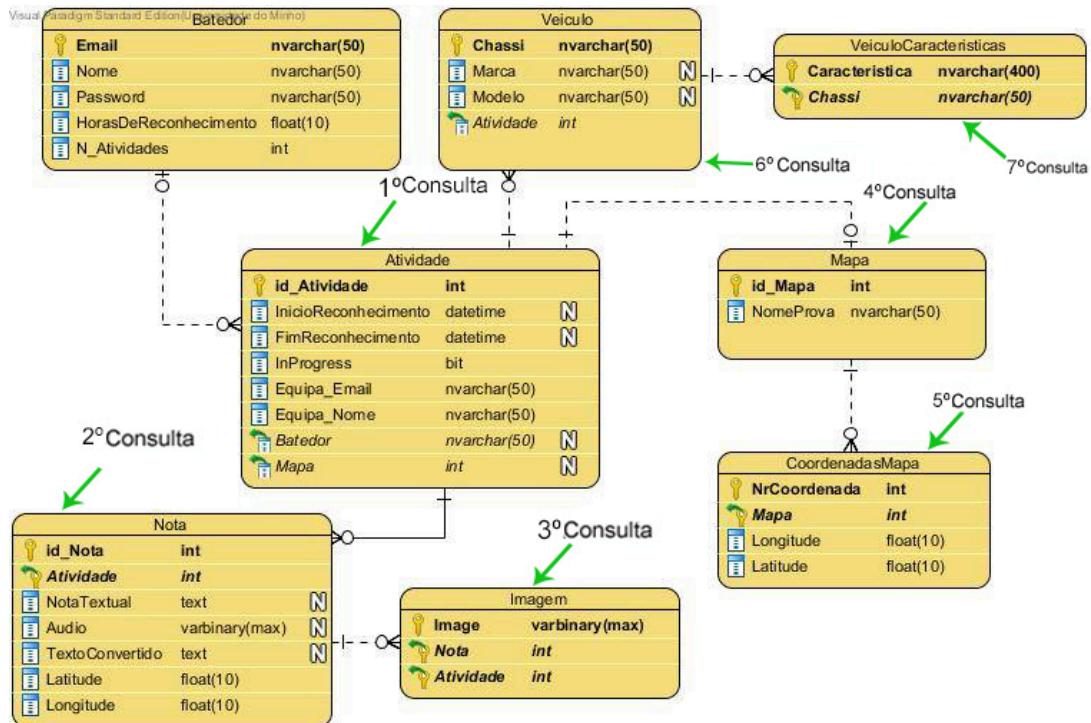


Figura 24 - Mapa validador da Query Consultar Atividade

- **Será possível inserir uma atividade?**

A inserção de uma atividade torna-se possível através da tabela atividade.

Ao inserir uma atividade, as tabelas alteradas são, respetivamente, as tabelas Atividade, Veiculo, Mapa e Nota. A ordem de alteração destas tabelas, deve-se ao facto de uma atividade ter de existir para ser possível a inserção de um Veiculo, Mapa ou Nota.

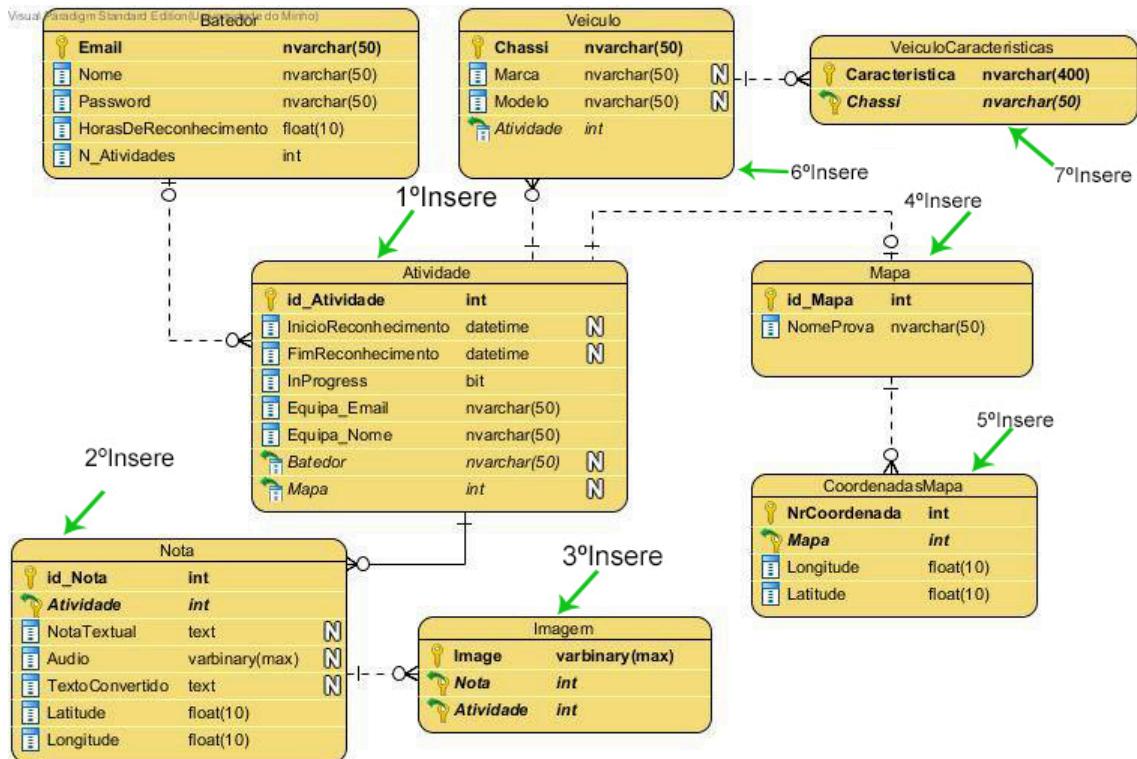


Figura 25 - Mapa validador da Query Inserir Atividade

IV. Definição do tamanho inicial da base de dados e análise do seu crescimento futuro

No que diz respeito ao crescimento da Base de Dados, é nosso objetivo que esta acompanhe o crescimento das diversas Atividades, ou seja, que possam ser inseridos novos Batedores, Veículos, entre outros. Acreditamos que a nossa Base de Dados têm a capacidade de ser estendida e de evoluir de forma a suportar novos requerimentos, sem que os utilizadores existentes na Base de Dados aquando do crescimento sejam significativamente afetados.

V. Revisão do modelo lógico final com os futuros utilizadores do sistema da base de dados

Após a elaboração do modelo lógico da base de dados, voltamos a reunir com alguns dos futuros utilizadores do sistema. Estes, depois de terem com a nossa ajuda analisado e percebido o diagrama, validaram-no. Assim sendo, esta validação permitiu o avanço para o modelo físico.

4.5. Construção da Base de Dados da Aplicação Mobile

A base de dados instalada na aplicação Mobile, trata-se de uma vista da base de dados do BackOffice. Esta vista, diz respeito a todas as entidades da base de dados do Backoffice, estando no entanto limitada a apenas alguns atributos, visto que estes deixam de ter importância no contexto da aplicação. De reparar que a aplicação diz respeito a apenas uma atividade. Devido ao facto de ser uma vista, esta respeita todas as regras de normalização referidas anteriormente.

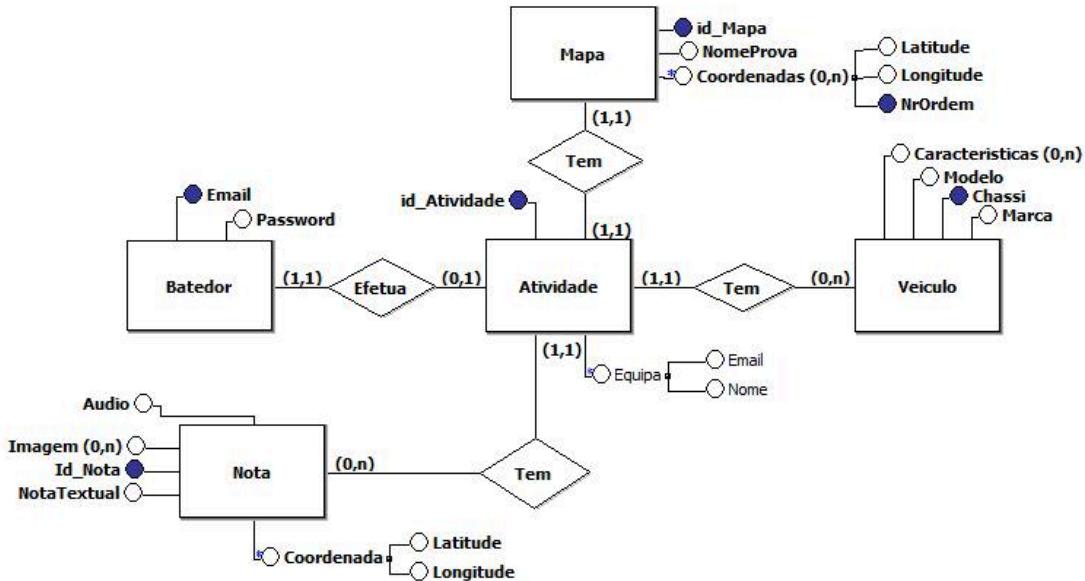


Figura 26 - Modelo Conceptual da Base de Dados *JustTraillt*

4.6. Gramática utilizada pelo batedor

De modo a tornar o sistema mais consistente e credível, foi necessário o desenvolvimento de uma pequena gramática, que resumisse grande parte dos casos que um batedor pode encontrar num percurso de uma prova. Cada palavra ou expressão desta gramática, está associada a uma linguagem simbólica, de forma a otimizar a leitura das notas por parte do copiloto, para este que consiga rapidamente transmitir todas as informações úteis, e não mais do que o necessário, ao piloto.

Num percurso de rally o batedor irá retirar notas essencialmente sobre curvas e outras irregularidades que possam existir no mesmo. As curvas são caracterizadas essencialmente pelo seu ângulo de curva, pela duração e pelo que sucede.

Assim foi desenvolvida a seguinte gramática e associada aos símbolos utilizados em provas oficiais.

- Corner

- Grade

Expressão	Símbolo
Right	R
Left	L
Fast plus	6+
Fast	6
Easy plus	5+
Easy	5
Easy minus	5-
Medium plus	4+
Medium	4
Medium minus	4-
Kay plus	3+
Kay	3
Kay minus	3-
Bad plus	2+
Bad	2
Bad minus	2-
Hairpin plus	1+
Hairpin	1

Tabela 12 - Gramática utilizada para definir o termo Grade

- Duration

Expressão	Símbolo
Less duration than normal	short
Normal duration	
More duration than normal	lg
Very long duration	Vlg
Extra long duration	Xlg
Extreme duration	/xxx

Tabela 13 - Gramática utilizada para definir o termo Duration

- **Further**

Expressão	Símbolo
Late	Late
Closes	>
Tightens	>
Opens	<
Opens and tightens	<>

Tabela 14 - Gramática utilizada para definir o termo Further

- **Road**

Expressão	Símbolo
Caution	!
Double Caution	!!
Triple Caution	!!!
Danger	!!!
maybe	?
Small crest	smCr
Small brow	smCr
Crest	Cr
Brow	Cr
Big crest	BigCr
Action on top of crest	onCr
Small jump	smJmp
Jump	Jmp
Big jump	BigJmp
Short sharp crest	Kick
Bump	Bmp
Big bump	BigBmp
Dip	dip
Over	/
Bridge	
Drop	Drop
Keep	stay
Keep lefthand	stay L
Keep middle	stay middle
Keep righthand	stay R
Through	thru
Cattle guard	guard
No cut	n.c.
Road narrows	nar
Off camber	o.c.
Rough	ruf

Tabela 15 - Gramática utilizada para definir o termo Road

Desta forma torna-se fácil a introdução de informação por parte do batedor. Não existindo redundância na informação obtida.

4.7. Inicialização da Aplicação

A aplicação precisa de ser iniciada tendo conhecimento de algumas informações importantes para o seu funcionamento. Desta forma é importante que de alguma forma esta informação seja agrupada. Para isto, foram estudadas varias possibilidades, contudo escolheu-se criar um ficheiro para representar a informação. Este ficheiro irá conter a informação no formato Json. Este ficheiro conterá a informação do email e a respetiva password da empresa RallyScouts, a gramática e a respetivos símbolos.

Esta solução dá-nos algum conforto para realizar alterações, mas também nos traz uma enorme contrapartida, o ficheiro é visível e não está encriptado. Sendo assim torna-se possível ler qualquer informação nele encontrada. Para isto decidimos assumir o pressuposto de que o Admin do BackOffice é responsável e assim a informação estará em segurança.

```
▼ object {4}
    email : xsolutiondevelopment@gmail.com
    password : rallyscouts
    ► dicionario [64]
    ▼ simbolos {2}
        ▼ corner {3}
            ► grade {18}
            ▼ duration {6}
                Less duration than normal : short
                Normal duration : value
                More duration than normal : lg
                Very long duration : Vlg
                Extra long duration : Xlg
                Extreme duration : /xxx
        ▼ further {5}
            Late : late
            Closes : >
            Tightens : >
            Opens : <
            Opens and tightens : <>
    ► Road {31}
```

Figura 27 - Excerto do ficheiro de configuração

5. Diagrama de Gantt (2^a Fase)

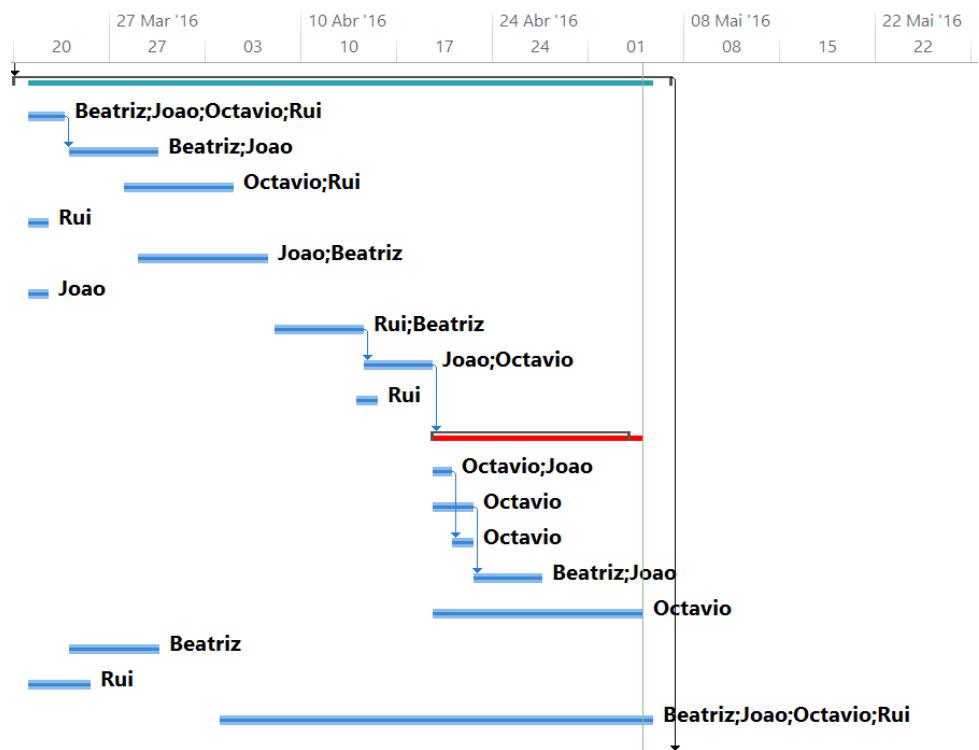


Figura 28 – Diagrama de Gantt

	24,33 dias	Dom 20/03/16	Sex 06/05/16
• 2 Especificação			
2.1 Discussão, de uma forma detalhada sobre o software a desenvolver	2 dias	Seg 21/03/16	Qua 23/03/16
2.2 Construção do Modelo de Domínio	3 dias	Qui 24/03/16	Qua 30/03/16
2.3 Construção de Use Cases	4 dias	Seg 28/03/16	Ter 05/04/16
2.4 Especificação dos Use Case	1 dia	Seg 21/03/16	Ter 22/03/16
2.5 Construção dos Diagramas de Sequência	5 dias	Ter 29/03/16	Qui 07/04/16
2.6 Construção dos Diagramas de Atividade	1 dia	Seg 21/03/16	Ter 22/03/16
2.7 Construção do Diagrama de Classes	3 dias	Sex 08/04/16	Qui 14/04/16
2.8 Especificação da linguagem para o reconhecimento da voz	2 dias	Qui 14/04/16	Ter 19/04/16
2.9 Especificação do ficheiro de configuração	1 dia	Qui 14/04/16	Sex 15/04/16
• 2.10 Base de Dados	7 dias	Ter 19/04/16	Qua 04/05/16
2.10.1 Criação do modelo Conceptual	1 dia	Ter 19/04/16	Qui 21/04/16
2.10.2 Documentação do modelo Conceptual	2 dias	Ter 19/04/16	Sex 22/04/16
2.10.3 Criação do modelo Lógico	1 dia	Qui 21/04/16	Sex 22/04/16
2.10.4 Documentação do modelo Lógico	2 dias	Sex 22/04/16	Qua 27/04/16
2.10.5 Geração da Documentação	7,58 dias	Ter 19/04/16	Qua 04/05/16
2.11 Realização de mockups para BackOffice	3 dias	Qui 24/03/16	Qua 30/03/16
2.12 Realização de mockups para a app mobile	3 dias	Seg 21/03/16	Sex 25/03/16
2.13 Construção do Relatório	16 dias	Seg 04/04/16	Qui 05/05/16

Figura 29 - Lista de tarefas da 2^a Fase

6. Conclusões e Trabalho Futuro

Na primeira parte do trabalho, caracterizamos e organizamos a construção das aplicações. Nesta segunda parte, pensamos ter conseguido analisar e especificar de forma completa todos os requisitos operacionais e funcionais do nosso sistema de software.

Pudemos verificar, analisando o diagrama de Gantt , que todas as tarefas agendadas foram cumpridas com sucesso, ainda que com algum atraso em algumas delas, devendo-se esse atraso à falta de tempo para a realização das mesmas.

Esperamos na terceira parte do trabalho, conseguir implementar as aplicações de forma a que estas cumpram a especificação desenvolvida.

Referências

- Belo, O., Apresentação do Trabalho Prático da Unidade Curricular Laboratórios de Informática IV, Universidade do Minho, 2015-2016
- Biafore, B., The Missing Manual, O'REILLY, 2013
- Connolly, T. and Begg, C. (2004). Database systems A Practical Approach to Design, Implementation, and Management. Harlow, England: Addison-Wesley
- Ian Sommerville, Software Engineering 10th Edition, University of St. Andrews, Scotland, 2016

Lista de Siglas e Acrónimos

BD Base de Dados

GPS Global Positioning System

UML Unified Modeling Language

Anexos

- [Conjunto de imagens exportadas dos Diagramas UML](#)