



Universidade do Minho
Escola de Engenharia
Mestrado Integrado em Engenharia Informática

Unidade Curricular de Laboratórios de Informática IV

Ano Letivo de 2015/2016

Explorador Espacial – Assistente de Campo

Alexandre Silva A72502

Daniel Malhadas A72293

Jéssica Pereira A71164

Joana Arantes A57810

Junho, 2016

LI4

Data de Receção	
Responsável	
Avaliação	
Observações	

Explorador Espacial – Assistente de Campo

Alexandre Silva A72502

Daniel Malhadas A72293

Jéssica Pereira A71164

Joana Arantes A57810

Junho, 2016

Resumo

Este trabalho resume-se ao desenvolvimento de uma aplicação, cujo tema geral é "Agente de Campo". Envolve, por isso, a construção de um programa de grande complexidade, cujo tema específico escolhido é "Exploração Espacial". A nossa aplicação é destinada a assistentes espaciais, em missões da NASA, nosso cliente. Especificamente pedido pelo nosso cliente, implementaremos a aplicação voltada apenas para explorações na Lua, mas que poderá ser, no futuro, voltada para outros astros.

Primeiramente, é feita uma análise completa de todos os requisitos facultados pelo nosso cliente. Apresentamos também um protótipo/modelo da aplicação que vamos desenvolver, com objetivo de esclarecer todos os pontos nos requisitos, de modo a evitar erros e clarificar/corrigir alguns pormenores.

Em segundo lugar, tal como pedido na segunda fase, são analisados novamente todos os requisitos, diferenciando-os como requisitos de utilizador e de sistema, e, em cada um, como requisitos funcionais e não-funcionais.

Apresentaremos a documentação UML, onde incluiremos vários tipos de diagramas, dos quais o Diagrama de Domínio, o Modelo Use Case, o Diagrama de Classes e Diagramas de Sequência.

Será apresentada uma especificação geral de toda a aplicação utilizada como preparação para o desenvolvimento da mesma, da Base de Dados e da interface com o utilizador, com o desenho de *mockups*.

Será apresentada uma análise da nossa Base de Dados, através da implementação dos modelos conceptual, lógico e físico, com a povoação da mesma, incluindo algumas *queries* necessárias para a aplicação.

Será explicado, ainda, o modelo utilizado pela nossa equipa, modelo RUP, que tentamos seguir durante todas as fases do trabalho.

De forma geral, concluímos o presente projeto com a terceira fase, que diz respeito à implementação de toda a especificação detalhada, ou seja, codificação da aplicação e implementação da própria Base de Dados para a aplicação, com toda a informação importante necessária para a atualização dos dados e consistência dos mesmos. Apresentaremos ainda uma breve explicação da plataforma escolhida para a aplicação.

Área de Aplicação: Engenharia Informática, Engenharia de Software, Exploração Espacial, Aplicações, Mobile

Palavras-Chave: Agente de Campo, Exploração Espacial, Engenharia de Software, Assistente, Planeamento, Especificação, Requisitos, Funcionalidade, Bases de Dados

Índice

1. Introdução	1
1.1. Contextualização	1
1.2. Apresentação do Caso de Estudo	2
1.3. Motivação e Objetivos	2
1.4. Estrutura do Relatório	4
2. Planificação	5
2.1. Equipa de Projeto	5
2.2. Plano de Atividades	5
3. Fundamentação da Aplicação	7
3.1. Tipo de Aplicação e Funcionalidades	7
3.2. Análise de Requisitos	7
3.2.1 Requisitos de Utilizador	8
3.2.2 Requisitos de Sistema	9
4. <i>Mockups</i>	13
5. Modelos para Base de Dados	20
5.1. Modelo Conceptual	20
5.1.1 Identificação das Entidades e dos Atributos	20
5.1.2 Identificação dos Relacionamentos	23
5.1.3 Identificação das Chaves Primárias	26
5.2. Modelo Lógico	27
5.2.1 Validação segundo as Regras de Normalização	27
5.2.2 Validação segundo as Transações do Utilizador	28
5.3. Elaboração e Validação do Esquema Lógico da Base de Dados	28
5.4. Tamanho Inicial da BD e Análise do seu Crescimento Futuro	29
5.5. Revisão do Modelo com os Futuros Utilizadores do Sistema da BD	29
5.6. Modelo Físico	29
5.6.1 Tradução do Modelo para um SGBD e sua Implementação	30
6. Especificação UML	33
6.1. Modelo de Domínio	33
6.2. Modelo de Use Case	33
6.3. Diagrama de Classes	35
6.4. Diagrama de Sequência	35
7. Motivos para a Escolha da Plataforma	36
8. Organização e Implementação das Classes - Aplicação	37
9. Resultado Final	39
10. Ferramentas Utilizadas	40

11. Conclusões e Trabalho Futuro	41
I. Anexo 1	45
II. Anexo 2	46
III. Anexo 3	47
IV. Anexo 4	48
V. Anexo 5	49
VI. Anexo 6	50

Índice de Figuras

Figura 1 – Datas/Fases do projeto	5
Figura 2 – Ecrã inicial	13
Figura 3 – Ecrã de Administrador	14
Figura 4 – Ecrã de Astronauta	14
Figura 5 – Ecrã da lista de missões	15
Figura 6 – Ecrã da missão	15
Figura 7 – Ecrã da missão – Detalhes textuais	16
Figura 8 – Ecrã da missão – Detalhes orais	16
Figura 9 – Ecrã da missão – Detalhes fotográficos	17
Figura 10 – Ecrã do percurso/detalhes de um registo	17
Figura 11 – Ecrã dos registos do caderno de notas	18
Figura 12 – Ecrã dos registos fotográficos	18
Figura 13 – Ecrã dos registos textuais	19
Figura 14 – Ecrã dos registos orais	19
Figura 15 – Identificação das chaves primárias	27
Figura 16 – Criação da Base de Dados “Explorador Espacial”	30
Figura 17 – Criação da tabela “Astronauta”	30
Figura 18 – Criação da Tabela “Catalogo”	30
Figura 19 – Criação da tabela “Astronauta acede Catalogo”	31
Figura 20 – Criação da tabela “Missao”	31
Figura 21 – Povoamento da tabela “Astronauta”	32
Figura 22 – Modelo de Use Case	33
Figura 23 – Especificação Ver Missões	34
Figura 24 – Diagrama de sequência	35
Figura 25 – Camadas da aplicação	37
Figura 26 – Camada Bussiness	37
Figura 27 – Classe Astronauta com os seus parâmetros e construtores	38
Figura 28 - Classe Missão com os seus parâmetros	38
Figura 29 – Ecrã inicial	39
Figura 30 – Ecrã das listas das missões disponíveis	39

1. Introdução

Este trabalho diz respeito à unidade curricular de Laboratórios de Informática IV, curso de Engenharia Informática, 3º ano, 2º semestre, cujo tema é Assistente de Campo. Nesta terceira fase serão apresentados todos os conteúdos das fases anteriores e será acrescentada toda a explicação do desenvolvimento do código, diagramas UML melhorados e também o melhoramento dos modelos conceptual e lógico para a Base de Dados, assim como a própria Base de Dados desenvolvida.

1.1. Contextualização

Tradicionalmente, os investigadores de campo, das diferentes áreas de interesse, no decorrer das suas atividades, orientadas e planeadas pela procura de novo ou observação de um determinado conhecimento, muitas vezes necessitam de recolher determinadas amostras que sustentam as suas teses. Depois é feita a análise desses mesmos dados, sendo necessário muitas vezes, de forma a garantir uma base teórica consistente, recorrer a alguns dados recolhidos anteriormente por ele ou por outro investigador. Esse mesmo processo de análise e interpretação tem como objetivo compreender e explicar o problema pesquisado.

Ora, nos últimos 20 anos, tem-se assistido a uma crescente e continuação de adoção, por parte das famílias e das empresas, de ferramentas informáticas (*software* e *hardware*), cada vez mais sofisticadas, complexas e energeticamente eficientes (*hardware*), ao mesmo tempo em que assistimos a uma redução brutal dos seus custos de aquisição e manutenção. Esse desenvolvimento tecnológico permitiu também tornar o trabalho dos investigadores de campo muito mais cómodo e prático, pois foram desenvolvidas peças de *software* que permitem a recolha de dados. Por exemplo, uma ou mais fotos de um determinado fenómeno observado poderá ser imediatamente carregado nas aplicações, passando a estar, se aplicável, disponível, numa plataforma *online*, onde outros possíveis investigadores possam também contribuir no estudo daquele fenómeno. Toda essa comodidade na recolha dos dados deverá também ser pela aplicação, no processo de análise e interpretação dos dados, onde deverá ser possível, de forma intuitiva e rápida, aceder aos nossos registos anteriores previamente catalogados no sistema.

Todo este desenvolvimento tecnológico fornecido fez com que os investigadores de campo "largassem" as suas canetas, cadernos, arquivadores, bússolas e máquinas fotográficas e se dirigissem para o terreno, munidos de um pequeno dispositivo dotado da aplicação, permitindo retirar notas de trabalho de forma oral, tirar fotografias relevantes e aceder em tempo real a toda a uma documentação relevante. A forma como também é possível gerar, de forma automática, relatórios detalhados, tendo por parte as atividades e observações realizadas, é também uma grande valia oferecidos por esses sistemas de *software*.

1.2. Apresentação do Caso de Estudo

O tema geral é “Assistente de Campo”. O nosso caso de estudo está voltado para a Exploração Espacial, envolvendo assim uma aplicação a ser desenvolvida para um assistente de exploração espacial, mais concretamente, da Lua. Foi-nos pedido pela NASA o desenvolvimento desta aplicação, com o objetivo de facilitar e melhorar a atividade dos exploradores espaciais, e permitir assim que estes tenham alguém ou um programa que lhes ajude e assista no seu trabalho em campo. No momento que necessitarem de rever algum dado específico, esta aplicação terá com exatidão e prontidão a oportunidade de deslumbrar o explorador espacial ou o(s) seu(s) explorador(es) de campo, em questão de segundos. Esta aplicação estará disponível também *offline*, visto que se trata de um caso de estudo em que é possível não haver qualquer conexão com a Terra, na maior parte do tempo, e assim permite que, mesmo havendo falhas e não havendo qualquer tipo de comunicação, ela possa intervir e ajudar o assistente de campo em todos os sentidos, desde na procura de algum dado relativo que poderá ser encontrado na *Base de Dados* (BD) local, até à salvaguarda de algum relatório de voz ou imagem criado e guardado na base de dados local.

1.3. Motivação e Objetivos

Para um projeto desta escala (agente de campo) e para algo tão complexo como a exploração espacial, torna-se essencial uma gestão do projeto que tenha em vista, desde cedo, grande parte do horizonte do que será a futura implementação do projeto. Não podemos pensar nas coisas enquanto as fazemos, por essa razão é necessário identificar desde o início todos os pontos importantes do projeto que nos propomos a cumprir. Uma fácil e rápida gestão de todos os recursos dos astronautas é algo essencial nesta área e, como tal, é algo de relevância para o projeto. É necessário, de alguma forma, sermos capazes de guardar toda a informação, para posteriormente gerir os dados do agente de campo. Isto pode ser alcançado com uma base de dados de complexidade média, de forma a evitar possíveis "desaparecimentos" de dados e para evitar falcaturas ou corrupção de informação tão crucial nas missões para que será solicitada.

Outras tecnologias serão também necessárias para pôr em prática a recolha de dados, como por exemplo um leitor de voz, que guarda tanto a voz como uma conversão da mesma para texto, e um sistema para guardar coordenadas geográficas e as associar a outros dados da mesma exploração. Como podemos ver, a exploração espacial é complexa e depende de imensas variáveis, sendo necessário um grande ênfase na precisão da informação, precisão essa que poderá salvar vidas e/ou recursos únicos e caros. Por essa razão, este projeto é algo essencial na área a que se destina e falhas mínimas não serão aceites ou toleradas, sendo mesmo necessário que se implemente, então, uma aplicação que consiga organizar todas essas variáveis, de forma melhor que qualquer humano, motivando assim o nosso projeto.

Perante este caso de estudo, e tendo em conta o que foi requerido, tem-se como objetivo principal desenvolver um agente de campo de exploração espacial capaz de catalogar e fornecer informações, em tempo real, ao utilizador em causa. Sendo esta uma tarefa com bastante complexidade, teremos de encontrar o método mais adequado para a realização desta tarefa com sucesso. Assim, será necessário:

- Desenvolver um assistente com dinamismo suficiente para o utilizador conseguir tirar o máximo proveito do agente de campo;
- Desenvolver um assistente em que o público-alvo seja investigadores de exploração espacial;
- Desenvolver um assistente com uma interface simples e científica que proporcione ao utilizador uma fácil utilização.

Para além destes objetivos, e sendo que este projeto simula uma interação com o mundo real, um outro objetivo a cumprir será pôr em prática as especificações requeridas.

Para alcançarmos estes objetivos, teremos de recolher informações pertinentes, seguindo-se de uma análise do problema, conceptualização de modelos seguidores dos requisitos e sua validação, implementação dos modelos finais e documentação de todo o processo.

1.4. Estrutura do Relatório

Na primeira fase, foi fundamentado o desenvolvimento da aplicação “Explore Assistant”, uma aplicação que ajudará bastante os exploradores espaciais, sendo um assistente de campo específico. Anteriormente, foram explicados os motivos da escolha do caso de estudo e o seu contexto no mundo em geral e aplicativo. Foi também explicado e justificado o modelo de negócio escolhido para a nossa aplicação e a sua caracterização, a fundamentação do seu desenvolvimento e a análise de todos os requisitos. Estes requisitos foram criados a partir do conhecimento que tivemos por parte do nosso cliente e melhorados, consequentemente, com aprovação deste (a NASA).

Na segunda fase, foi apresentada toda a especificação UML e documentação necessária para o desenvolvimento da nossa aplicação. Foram apresentadas melhorias na análise de requisitos, sendo diferenciados agora por requisitos de utilizador e de sistema, e, em cada um, requisitos funcionais e não-funcionais. Foi apresentado um protótipo/modelo da nossa aplicação, e ainda apresentados os modelos conceptual e lógico para a Base de Dados. No que diz respeito à especificação UML, foram apresentados os Diagramas de Domínio, de Classes, de Sequência e Modelos Use Case para os nossos utilizadores. Toda a implementação foi seguida pela técnica utilizada do método RUP pela nossa equipa de trabalho.

Na terceira fase deste projeto foram aplicadas todas as especificações anteriormente detalhadas, seguindo-se da implementação e codificação da aplicação, assim como a própria Base de Dados. Serão apresentados os diagramas UML melhorados, uma breve explicação da organização do código e os motivos da escolha da plataforma para a aplicação desenvolvida.

2. Planificação

2.1. Equipa de Projeto

Como já foi anteriormente especificado, este projeto deverá simular um ambiente adaptável a um mundo real. É da responsabilidade do nosso grupo o desenvolvimento do mesmo, desde o seu planeamento até à sua implementação.

O nosso cliente, como já foi referido, é a NASA, tendo este determinadas exigências em relação ao produto final.

A equipa é constituída pelos alunos Alexandre Silva, Daniel Malhadas, Jéssica Pereira e Joana Arantes, sendo que escolhemos como gestora do projeto a aluna Jéssica Pereira.

2.2. Plano de Atividades

Para manter a organização e cumprir os prazos a que o projeto está sujeito, a nossa gestora do projeto realizou um plano de atividades através de um Diagrama de Gantt que está a ser mantido e cumprido por todos os elementos do grupo, apresentado no anexo 1.

No plano de atividades, encontram-se agendados controlos de execução, isto é, momentos em que, ao se percorrer o plano, se verificam se as atividades agendadas até ao momento foram ou não cumpridas. Caso as atividades planeadas não sejam adequadas ou verificarem atrasos, deverão analisar-se as suas causas, definir meios para os solucionar e, por fim, atualizar o plano.

18-mar	06-mai	19-jun
Fundamentação	Especificação	Construção

Figura 1 – Datas/Fases do projeto

A nossa equipa seguiu o modelo RUP para o desenvolvimento da aplicação, ou seja, voltamos o nosso desenvolvimento numa vertente orientada a objetos, criando assim diagramas UML para uma melhor organização e compreensão do projeto e definimos os responsáveis por cada tarefa, cumprindo os prazos estabelecidos. Temos o planeamento e levantamento de requisitos, a sua análise e, por último, temos a implementação e teste do *software*. Seguimos pela divisão de fases mais gerais, ou seja, fase de conceção, de elaboração, de construção e de transição.

A fase de concepção diz respeito à primeira fase da entrega do projeto (denominada fase de fundamentação), a fase de elaboração diz respeito à segunda fase (denominada fase de especificação) e as duas últimas fases dizem respeito à última fase do projeto (denominada fase de construção). Na fase de concepção houve uma comunicação com o cliente (a NASA) para levantamento e análise de requisitos, análise dos riscos e determinação de prazos. Na fase de elaboração é feita uma análise do domínio do problema e a arquitetura do projeto. Na fase de construção é feito o desenvolvimento do *software*. Na última fase, fase de transição, são feitos os testes antes da entrega do *software* à NASA.

3. Fundamentação da Aplicação

Nesta secção, iremos idealizar um modelo do assistente de campo a criar, pois este é o objetivo desta fase de fundamentação. Assim, foram definidas as seguintes especificações:

- Caracterização geral da aplicação a desenvolver;
- Definição das funcionalidades a implementar, indo de encontro aos requisitos do cliente;
- Definição do modelo de funcionamento da aplicação.

Depois de definidas as funcionalidades que se pretendem implementar, pretende-se detalhar a forma como o utilizador deverá interagir com a aplicação.

3.1. Tipo de Aplicação e Funcionalidades

Mediante o nosso caso de estudo, a nossa aplicação será um programa de fácil intervenção pelo utilizador, de contexto científico, envolvendo um plano de atividades e de acontecimentos. Como funcionalidades terá o reconhecimento de voz e escrita através da leitura da voz *speak to text*, partilha de conteúdo, ser capaz de guardar numa base de dados local e global imagens, e ser capaz de fazer *upload* de vídeos fornecendo apenas os *links* destes. Terá também acesso ao GPS para poder registar a localização atual, e recentes, registar percursos, guiar-se e guiar o utilizador, marcando locais importantes. Deverá permitir que o utilizador fale para a aplicação e esta o entenda e associe os dados de utilização, guardando os dados numa base de dados e voltando a recuperar os dados uma vez autenticado novamente (*login/logout*). Deverá, obviamente, permitir que vários utilizadores se registem na aplicação, para poderem tirar partido desta.

3.2. Análise de Requisitos

Junto com o nosso cliente (representante da NASA), obtivemos os requisitos sobre a aplicação que nos foi pedida. Após a análise dos requisitos primeiramente obtidos, que acreditamos serem os requisitos de utilizador, especificamos melhor alguns deles criando assim requisitos de sistema, explicando o que o sistema deverá fornecer em concreto e como deverá responder. Em cada um destes requisitos distinguimos ainda os funcionais e não-funcionais. Esta análise e especificação será apresentada de seguida.

3.2.1 Requisitos de Utilizador

A. Funcionais

- a) A aplicação deverá disponibilizar ao utilizador mensagens de ajuda.
- b) A aplicação deverá perguntar ao utilizador se pretende guardar a sua sessão e suas credenciais após sido feito o registo e autenticação.
- c) A aplicação deverá permitir ao utilizador tirar fotos, eliminá-las e guardá-las.
- d) O utilizador deverá poder efetuar escolha de registo (se textual, imagem ou voz).
- e) O utilizador poderá efetuar um registo gravando a sua voz, podendo este terminar o registo/gravação quando bem pretender.
- f) O utilizador deverá ter acesso ao plano de acontecimentos e a todos os registos envolvidos.
- g) A aplicação deverá dar permissão ao utilizador de ouvir as gravações em registos efetuados.
- h) A aplicação deverá apenas permitir o registo e autenticação de utilizadores da NASA.
- i) O utilizador deverá ter acesso a todas as pessoas envolvidas num projeto numa determinada busca/exploração espacial/projeto.
- j) A aplicação deverá ter uma lista de tarefas atualizadas (tarefas a realizar e tarefas concluídas).
- k) A aplicação deverá ter conhecimento dos chefes e pessoas a quem vai auxiliar, após feito autenticação/login, para poder fornecer estas informações ao utilizador.
- l) O utilizador deverá poder tomar conhecimento do mapa explorado e por explorar para poder guiar-se na criação e conclusão das suas missões.
- m) Um astronauta poderá criar uma nova missão e adicionar membros à sua missão.
- n) Um astronauta poderá voltar atrás e eliminar missões criadas, cancelando tudo o que fez até ao momento atual, no contexto da criação de missão.
- o) Um astronauta deverá poder eliminar astronautas apenas das suas missões.
- p) Um astronauta não poderá efetuar alterações administrativas em missões que não lhe pertencem.

B. Não-funcionais

I. Requisitos de Produto

- a) A aplicação deverá estar disponível para todos os astronautas 24h, todos os dias.
- b) A aplicação deverá permitir aos utilizadores uma fácil aprendizagem para utilização, diminuindo o tempo de inicialização/preparação para a sua utilização.
- c) A aplicação deverá ter uma interface limpa e a mais clara e minimalista possível.
- d) A aplicação deverá ser de rápida resposta ao utilizador.

II. Requisitos Organizacionais

- a) Os astronautas, utilizadores do programa, deverão autenticar-se com o seu ID fornecido pela instituição da NASA, numerado no seu cartão de identificação.
- b) O chefe terá de permanecer na base para poder coordenar a missão à distância.

III. Requisitos Externos

- a) O sistema deverá implementar provisões de privacidade a todos os astronautas sobre todos os registos submetidos, tal como explicito no regulamento da NASA.

3.2.2 Requisitos de Sistema

A. Funcionais

- a) Ter acesso ao microfone, no momento do registo. Após instalação e registo do utilizador no dispositivo, deverá informar ao utilizador que esta aplicação pretende o acesso ao microfone para, quando pretender, efetuar registos orais.

- b)** Ter acesso à câmara, no momento de registo. Após instalação e registo do utilizador no dispositivo, deverá informar ao utilizador que esta aplicação pretende o acesso à câmara para, quando pretender, efetuar registos e *upload* de fotos e, possivelmente, vídeos.
- c)** Reconhecer voz, através do microfone do sistema operacional usado pelo utilizador, do qual foi permitido o acesso por este no momento de registo. Caso não tenha microfone não será possível efetuar registos de voz e, assim sendo, o sistema não reconhecerá a voz, pois não existe nenhum dispositivo que assim o permita. Neste caso, esta funcionalidade não estará presente, enviando uma mensagem de erro/explicação ao utilizador.
- d)** Reconhecer texto, para introdução de novas mensagens no registo de atividades em possíveis missões.
- e)** Reconhecer imagem, através do acesso à camara, também obtido pela confirmação do utilizador no momento do registo. Se não existir câmara no dispositivo no qual o utilizador efetua a sua autenticação, a funcionalidade de reconhecer imagem não estará disponível mas será apresentada uma mensagem ao utilizador explicando que não consegue aceder à camara.
- f)** Ter acesso ao mapa dos astros existentes na base de dados e ao GPS, para posterior conhecimento da localização do utilizador num dado momento e posteriores cálculos de áreas de percursos efetuadas em possíveis missões.
- g)** Conseguir calcular a localização atual do utilizador para gravação das coordenadas num determinado registo efetuado.
- h)** Cada vez que é adicionado um novo registo ou missão completada, deve atualizar no histórico fazendo este registo parte de um acontecimento passado, para poder ser consultado no futuro.
- i)** O histórico e caderno de notas deverão manter-se actualizados de cada vez que é adicionado um novo registo, de forma correta, preenchendo todas as variáveis necessárias, desde a localização ao registo e possível descrição do evento.
- j)** Ter acesso à base de dados do sistema, ou seja, saber o número total de astronautas/exploradores existentes na base de dados num dado momento, conhecer as informações destes, saber qual o tipo/função que desempenham e o seu ID para poder adicionar missões, sendo os exploradores médicos, cartógrafos, chefes de missão, etc.; conhecer a listagem de missões, histórico, localizações, entre outros.

- k) Fazer *backups* ao fim de cada etapa de uma missão. Não vale a pena fazer periodicamente para não se correr o risco de fazer *backups* sem ter acontecido nada de novo.
- l) Manter a consistência dos dados do *backoffice* com os dados na base de dados (poderá ter-se uma *stack* no *backoffice* que aumenta sempre que uma atividade é completada, sendo que, quando possível, a *stack* é carregada na base de dados e eliminada do *backoffice*).
- m) Ter acesso ao relógio do sistema, para saber a data e momentos específicos, tanto no registo de novas atividades como na própria aplicação, para que o utilizador possa guiar-se por um registo temporal/cronómetro quando pretender. Também útil para o cálculo de datas, data de início e data de fim, e, posteriormente, duração de uma dada atividade.
- n) Guardar fotos de itens/artefactos espaciais numa galeria própria chamada de catálogo, na base de dados, logo após que o utilizador tire fotografia e não a apague, querendo isto dizer que a pretende manter na base de dados.
- o) As alterações administrativas deverão estar presentes no contexto da análise de uma missão, ou seja, alterar o tipo de missão existente, os membros e os percursos a efetuar.

B) Não-funcionais

- a) Poderá haver um número de astronautas infinito, não havendo limitações de números, pois a NASA pode ter novos membros à medida que as suas missões vão sendo conseguidas.
- b) O sistema necessita de ter credenciais de administradores da NASA já codificados antes de ser posto em prática, pois apenas estes poderão registar ou remover astronautas.
- c) Embora apenas seja exigida a possibilidade de explorar a Lua, o sistema deverá ser escalável ao ponto de permitir uma fácil extensão a outros astros.
- d) Embora apenas seja exigida a possibilidade de suportar somente um tipo de veículo, o Rover, o sistema deverá ser escalável ao ponto de permitir uma fácil extensão a outros tipos de veículos.

- e) Embora apenas seja exigida a possibilidade de suportar os seguintes tipos de astronautas: geólogo, médico, cartógrafo, gestor de inventário, o sistema deverá ser escalável ao ponto de permitir uma fácil extensão a outros tipos de astronautas.
- f) Embora apenas seja exigida a possibilidade de suportar os seguintes tipos de missões: exploração com o intuito de cartografar/mapear os astros ou investigação de rochas/minerais, o sistema deverá ser escalável ao ponto de permitir uma fácil extensão a outros tipos de missões.
- g) Uma missão necessita, no mínimo, de astronautas que a executem (pelo menos um), percurso onde irá ocorrer (pelo menos um ponto geográfico), um astronauta que a crie e atribua funções e atividades a realizar. Poderá também ser-lhe atribuído um ou mais veículos, mas tal não é obrigatório. A missão não deverá ser criada se não houverem astronautas livres para a executar ou se o material (veículos) não estiver disponível.
- h) O sistema nunca deverá permitir, em circunstância alguma, fazer uma missão de investigação a rochas ou minerais com um percurso que passe por pontos geográficos que não estejam compreendidos numa área já previamente explorada com missões de exploração para cartografar/mapear. Por essa razão, deve-se sempre conseguir saber os pontos geográficos já explorados no astro em que se pretende realizar a missão de investigação para rochas/minerais.
- i) Uma missão tem 3 fases: preparação, onde se valida o material, intermédia, onde a missão é concretizada, e final, onde os astronautas voltam à base e registam os acontecimentos. Nenhuma missão poderá ser registada no sistema sem que, pelo menos, uma atividade seja referida para cada uma destas fases.
- j) Uma missão, quando iniciada, entrará na fase de preparação e apenas poderá avançar depois do astronauta responsável pelo inventário (gestor de inventário) validar o material necessário.
- k) O astronauta que inicia a missão tornar-se-á o seu chefe/coordenador, sendo exigido que divida os cargos (os cargos essenciais sem os quais a missão não começa são Gestor de Inventário, Coordenador de Campo e Especialista na Missão em questão).

4. Mockups

Todo o processo de explicação de um projeto em Engenharia de Sistemas é algo que, para um leigo na matéria, possa soar como um tanto ou quanto abstrato. Não existe, então, melhor maneira para dissolver e clarificar essa névoa do que aquilo a que damos o nome de *mockups*.

Os *mockups* são como que “desenhos” que demonstram o aspeto final, ou pelo menos o nosso entendimento de o que, nesta fase se espera, ser o aspeto final. Estes *mockups* baseiam-se não na nossa experiência, mas sim na leitura e estudo atento dos requisitos. Com isto queremos afirmar que não queremos que a nossa experiência, mesmo que reduzida, restrinja a nossa visão do projeto. Todo o projeto é um projeto diferente e é necessário que saibamos abstrair a nossa mente, olhar para cada um como se fosse “o projeto” e não simplesmente “um projeto”. Olhamos para este como sendo realmente o projeto e tratamo-lo como tal, nunca deixando os “vícios” que programamos indubitavelmente em nós, em simultâneo com cada linha que escrevemos para programar um projeto e que influencie aquilo que achamos ser possível ou que se pretende realmente fazer.

Seguem-se então os *mockups*. Achamos apenas relevante realçar a importância dos mesmos, sabemos que negligenciá-los apenas destruiria o projeto a longo prazo. Temos a noção de quão bem pensado tem de realmente estar o projeto antes de o começarmos a programar “de qualquer forma” e isso demonstramos agora (assim como em todos os capítulos deste relatório).

Ao iniciar o sistema, a mesma janela será apresentada, seja quem for o utilizador. Nesta janela pedem-se as credenciais sobre as quais o mesmo foi inscrito no sistema.

The image shows a graphical user interface (GUI) mockup for a login window. The window has a title bar with the text 'ficheiro' and standard window control buttons (minimize, maximize, close). Below the title bar is a menu bar with two items: 'sobre' and 'sair'. The main content area of the window contains the text 'Por favor, Indique as suas credenciais' followed by a horizontal line. Below this line are two input fields: one labeled 'ID' and another labeled 'Password'. The 'Password' field contains a series of asterisks. A diagonal line runs from the bottom left towards the bottom right of the main content area. In the bottom right corner, there is a button labeled 'Fechar'.

Figura 2 – Ecrã inicial

Caso as credenciais demonstrem-se ser de Administrador da NASA, então o seguinte ecrã é apresentado. De notar que estes dois botões são a sua única hipótese de interação com o sistema, sendo que pode remover ou inserir astronautas fornecendo os seus dados.

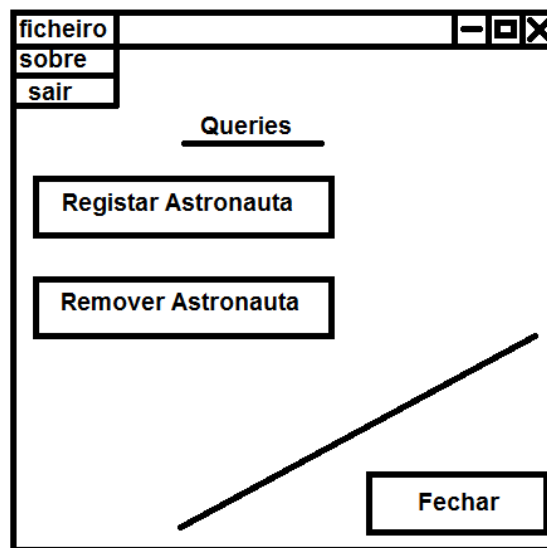


Figura 3 – Ecrã de Administrador

Caso as credenciais mostrem ser um utilizador do tipo Astronauta, então tem acesso às seguintes opções apresentadas na imagem seguinte. Pode criar uma missão ou iniciar uma missão, sendo que assim se torna o Chefe da mesma. Pode consultar missões (antigas, presentes ou futuras) e pode consultar o caderno de notas de todas as missões.



Figura 4 – Ecrã de Astronauta

Caso peça para ver a lista de missões, o seguinte ecrã da seguinte imagem é apresentado. As missões são representadas por nome, sendo que podem ser seleccionadas para ver dados/detalhes sobre as mesmas.

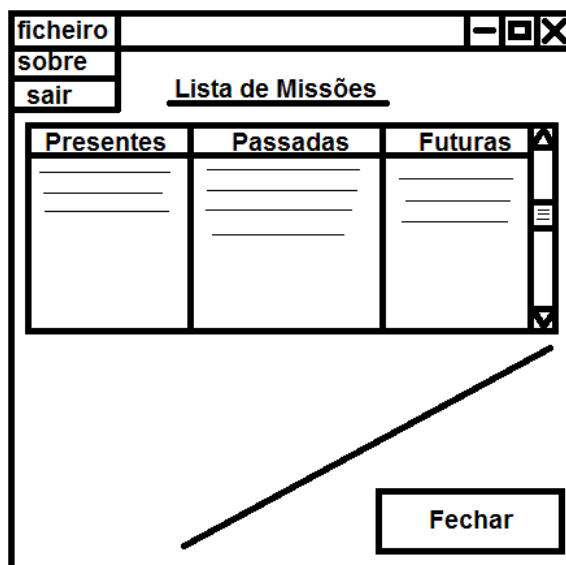


Figura 5 – Ecrã da lista de missões

Ao seleccionar uma missão para ver os seus detalhes, temos estas hipóteses apresentadas na seguinte imagem. Podemos ver registos textuais, orais ou fotográficos, o percurso da missão em questão ou as pessoas envolvidas na dada missão.

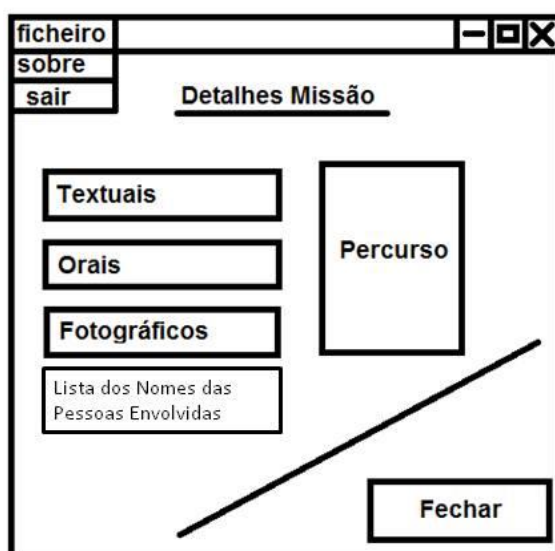


Figura 6 – Ecrã da missão

Ao pedir os detalhes textuais, são apresentados os mesmos representados pelos seus títulos. Podem ser seleccionados para se ver o conteúdo e a sua posição geográfica num mapa interactivo do astro.

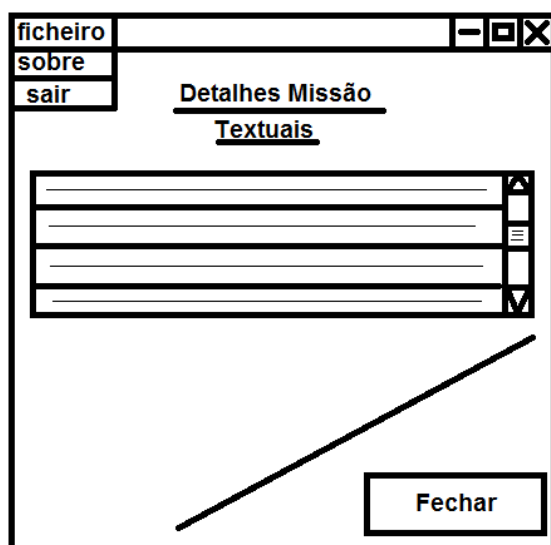


Figura 7 – Ecrã da missão – Detalhes textuais

Ao pedir os detalhes orais, são apresentados os mesmos representados pelos seus títulos. Podem ser seleccionados para se ver o conteúdo e a sua posição geográfica num mapa interactivo do astro.

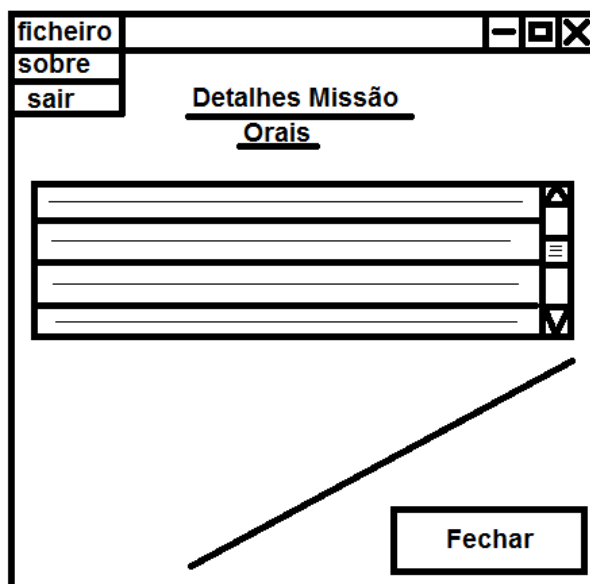


Figura 8 – Ecrã da missão – Detalhes orais

Ao pedir os detalhes fotográficos, são apresentados os mesmos representados pelos seus títulos. Podem ser seleccionados para se ver o conteúdo e a sua posição geográfica num mapa interactivo do astro.

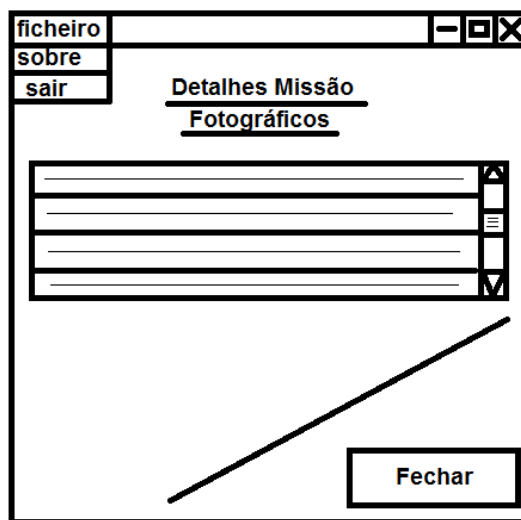


Figura 9 – Ecrã da missão – Detalhes fotográficos

Ao seleccionar a opção de ver o percurso ou pedir os detalhes de um registo, é apresentado então o mapa interactivo da seguinte forma.

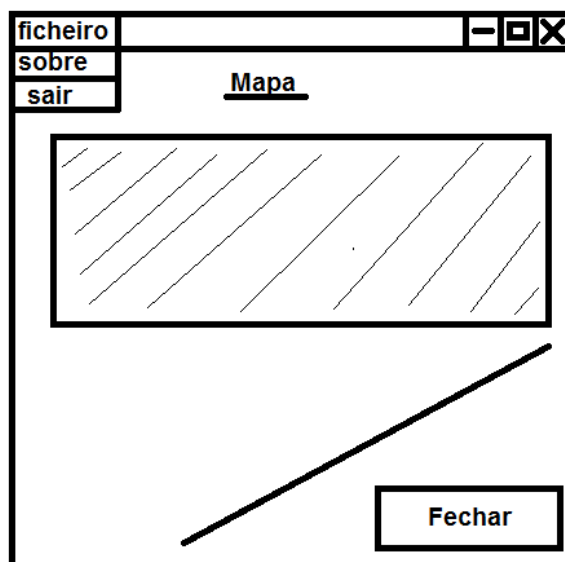


Figura 10 – Ecrã do percurso/detalhes de um registo

Voltando ao menu principal do utilizador Astronauta, caso se peça para consultar o caderno de notas, é-nos dada a possibilidade de consultar estes registos sendo eles globais (relativos a todas as missões).

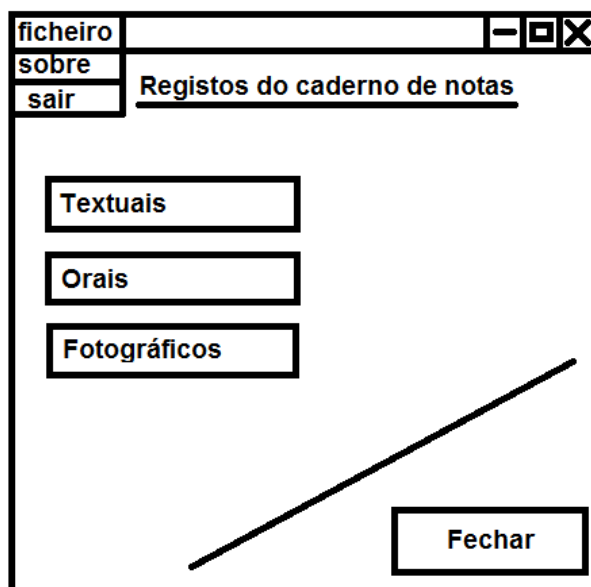


Figura 11 – Ecrã dos registos do caderno de notas

Ao pedir os registos fotográficos globais, são aqui eles apresentados pelo seu título, podendo ser seleccionados para ver os seus detalhes num mapa interativo (igual ao apresentado anteriormente).

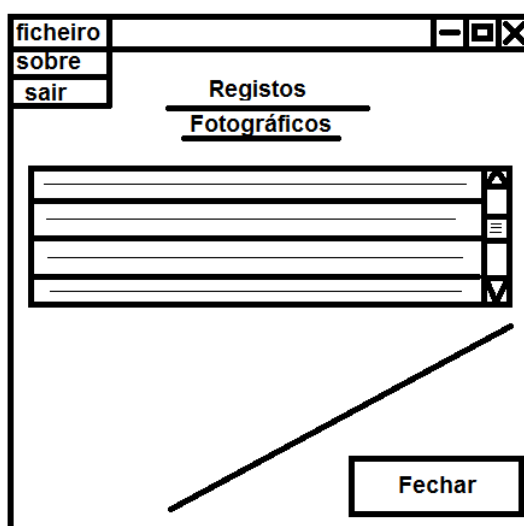


Figura 12 – Ecrã dos registos fotográficos

Ao pedir os registos textuais globais, são aqui eles apresentados pelo seu título, podendo ser seleccionados para ver os seus detalhes num mapa interativo (igual ao apresentado anteriormente).

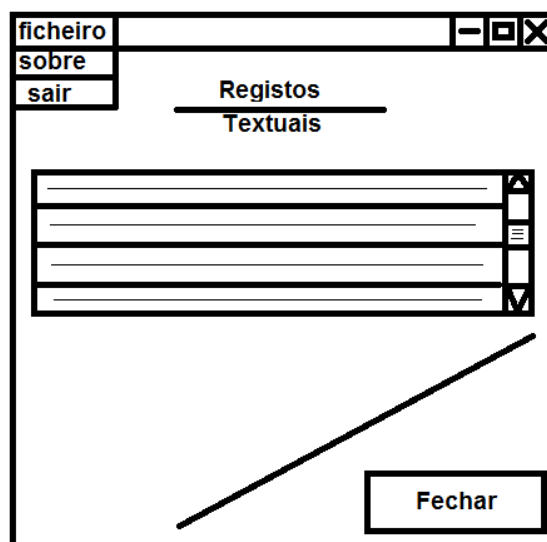


Figura 13 – Ecrã dos registos textuais

Ao pedir os registos orais globais, são aqui eles apresentados pelo seu título, podendo ser seleccionados para ver os seus detalhes num mapa interativo (igual ao apresentado anteriormente).

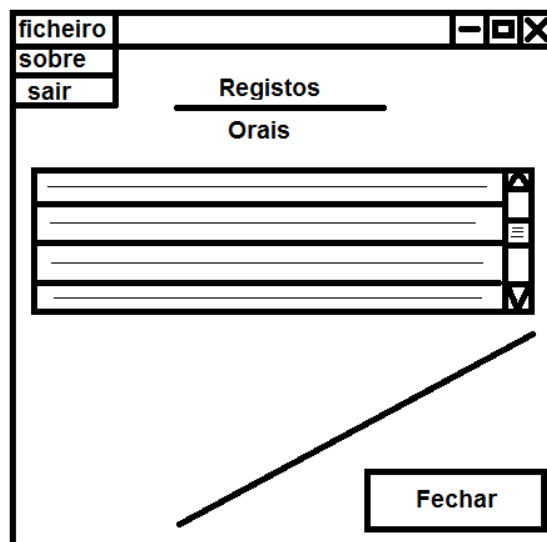


Figura 14 – Ecrã dos registos orais

5. Modelos para Base de Dados

Nesta secção do relatório serão apresentadas todas as explicações no decorrer do desenvolvimento dos modelos conceptual, lógico e físico para a nossa Base de Dados.

5.1. Modelo Conceptual

Todos os requisitos do sistema já foram previamente apresentados, juntamente com os de utilizador, através de entrevistas às chefias da NASA, entrevistas a diferentes tipos de astronautas, observações do local e das respetivas missões, e também pesquisas na Internet. A partir da análise dos requisitos, e havendo sido criado o diagrama de domínio anteriormente a esta fase, começamos a desenvolver o estudo e a conceção do modelo para a base de dados. Após algumas modificações no modelo conceptual, tivemos de adicioná-las, também, no modelo de domínio.

Com base nos requisitos foram, então, criadas as entidades que deverão estar presentes na base de dados. De seguida, foram identificadas as suas respetivas chaves primárias que identificam unicamente a respetiva entidade. Foram também feitas algumas especificações dos atributos de cada entidade. Por último, foram atribuídas conexões entre as entidades, ou seja, os relacionamentos entre estas.

Em anexo a este relatório, encontra-se o nosso modelo conceptual do nosso projeto.

5.1.1 Identificação das Entidades e dos Atributos

Para podermos demonstrar as entidades do projeto em questão, temos primeiro de nos focar no seu núcleo, a entidade Missão, sendo esta a nossa principal entidade, uma vez que todas as outras entidades estão ligadas a esta, seja de forma direta ou indireta.

Atendendo aos requisitos levantados, temos então as seguintes entidades:

A. Missão

Esta aplicação destina-se ao auxílio do trabalho de campo de um explorador espacial e o trabalho deste consiste na realização de várias missões, no contexto espacial. Assim sendo, torna-se claro que a nossa entidade principal é a Missão e vários relacionamentos serão criados à volta deste.

A Missão terá um id, terá um nome e duas datas (data de início e data de fim da realização da missão). A missão pode ser de dois tipos diferentes por isso essa distinção existe na base de dados no atributo denominado TipoMissao.

B. Astronauta

Dado o facto desta base de dados ser parte da implementação de uma aplicação destinada a astronautas/exploradores espaciais, é natural haver a entidade Astronauta. O Astronauta terá como atributos o nome, que na base de dados está descrito como NomeAstronauta, a sua Password, o seu Cargo, e o seu id.

C. Catálogo

No tipo de missão exploração e procura de minerais, pretendemos guardar num catálogo um género de inventário de todos os minerais encontrados, de forma permanente e constantemente atualizada. Daí a criação da entidade Catálogo tornar-se essencial. Este Catálogo tem como atributos o seu id, o seu nome e descrição do catálogo.

D. Mineral

Para que seja possível catalogar os minerais explorados e encontrados nas missões, torna-se essencial a existência desta entidade. Assim, ela tem como atributos o seu id, nome e foto do mineral em questão.

E. Histórico

O Histórico corresponde ao plano de acontecimentos, ou seja, missões que já decorreram ou eventos passados. Existe então o id do histórico e neste, várias missões identificadas como já ocorridas ou concluídas. Cada Histórico tem uma área que é calculada através de outra entidade explicada de seguida.

F. Área Triângulo

A entidade Área Triângulo é uma entidade que fornece a área total para ser guardada no histórico. Uma vez que uma missão concluída é adicionada ao histórico, esta tem uma área que é calculada da mesma forma que é calculada a área de um triângulo, tendo esta entidade como atributos três pontos, ponto a, b, c, e cada um deles um atributo latitude e longitude.

Assim sendo, os atributos ponto a, ponto b e ponto c são atributos compostos da entidade Área Triângulo.

G. Veículo

A cada missão pode estar presente a utilização de um Veículo. Cada Veículo tem o seu id, um tipo e nível de bateria.

H. Astro

No contexto atual, o Astro é uma entidade que permite saber que esta aplicação e esta base de dados apenas se refere ao astro da Lua. Mas no futuro pode expandir-se para outros astros, havendo assim esta característica da nossa aplicação explícita na nossa base de dados, ou seja, que é permitida a extensão da aplicação e base de dados para exploração em outros astros. Para já restringimo-nos à Lua.

Assim sendo, a entidade Astro tem como atributos o id e o nome do astro.

I. Atividade

Uma missão pode ter várias atividades, pois como sabemos pelos requisitos, uma missão tem várias fases. Assim sendo, foi criada a entidade Atividade que tem como atributos o seu id e uma descrição da atividade realizada ou por realizar.

J. Caderno de Notas

Para manter uma lista de todos os planos de atividade existentes e registos, existe a entidade Caderno de Notas que basicamente faz parte de tudo o que o Astronauta irá realizar ou já realizou e que tem como atributos o id do Caderno de Notas. No Caderno de Notas existirão vários registos.

K. Registo

No Caderno de notas pretende-se guardar várias instâncias de Registos. A entidade Registo diz respeito a tudo o que o Astronauta publicar ou realizar, seja um registo textual, fotográfico ou oral, através da gravação de voz. Assim sendo, a entidade Registo é bastante importante e tem como atributos o id, índice de registo, para saber qual registo efetuado primeiro, e respetiva ordem, nome do registo, tipo do registo (textual, oral ou fotografia) e o seu respetivo conteúdo.

L. Ponto de Passagem

Cada Percurso tem vários pontos de passagem. Cada Ponto de Passagem será contextualizado com um índice, ou seja, se foi o primeiro ponto na Missão por onde o Astronauta passou num dado percurso, ou segundo ponto, e assim sucessivamente. A entidade Ponto de Passagem tem assim, como atributos, o índice de passagem, a latitude e longitude como coordenadas e o id.

M. Percurso

Cada missão tem um Percurso e pretendemos guardá-lo na nossa base de dados para poder saber os pontos em que o Astronauta passou e qual o Percurso que realizou numa dada Missão. Como atributos apenas precisamos do seu id.

5.1.2 Identificação dos Relacionamentos

Um relacionamento pode ser definido pelo seu grau e pela sua cardinalidade.

Após termos identificado todas as entidades do nosso projeto, os relacionamentos que as associam e o seu mapa de relacionamentos são os seguintes:

A. Astronauta acede Catálogo

Um Astronauta acede a um catálogo de uma missão já ocorrida.

Cardinalidade: Astronauta(N); Catálogo(N) - Um astronauta pode aceder a várias catálogos e um catálogo pode ser acedido por vários astronautas.

Atributos: Este relacionamento não tem atributos.

B. Catálogo contém Mineral

Um Catálogo contém um ou mais minerais explorados e catalogados.

Cardinalidade: Catálogo(N); Mineral(N) - Um catálogo pode conter vários minerais e um mineral pode estar catalogado em vários catálogos.

Atributos: Este relacionamento não tem atributos.

C. Astronauta chefia Missão

Um Astronauta é chefe de uma missão.

Cardinalidade: Astronauta(1); Missão(N) - Um astronauta pode ser chefe de várias missões, mas uma missão apenas tem um chefe associado.

Atributos: Este relacionamento não tem atributos.

D. Astronauta é membro Missão

Astronauta é membro de uma missão, uma missão tem astronautas envolvidos. Identifica os astronautas envolvidos numa missão e todas as missões em que um determinado astronauta está presente.

Cardinalidade: Astronauta(N); Missão(N) - Um astronauta pode estar envolvido em várias missões e uma missão pode ser realizada por vários astronautas.

Atributos: Este relacionamento não tem atributos.

E. Missão tem associado Histórico

Uma missão após concluída é adicionada ao histórico.

Cardinalidade: Missão(N); Histórico(1) - Um histórico é composto por várias missões, sendo que uma missão pertence a um único existente histórico.

Atributos: Este relacionamento não tem atributos.

F. Histórico tem Área triângulo

Em cada missão pertencente a um histórico existe a área triângulo respetiva.

Cardinalidade: Histórico(1); Área triângulo(N); Um histórico tem várias áreas para cada missão, mas uma área apenas corresponde a uma missão no histórico.

Atributos: Este relacionamento não tem atributos.

G. Missão pode ter Veículo

Uma missão pode ou não ter um veículo associado.

Cardinalidade: Missão(1); Veículo(N) - Uma missão pode ou não ter um veículo associado, mas um veículo apenas estará associado a uma instância de missão.

Atributos: Este relacionamento não tem atributos.

H. Missão relativo a Astro

Uma missão é relativa a um astro, neste caso a Lua.

Cardinalidade: Missão(N); Astro(1) - Uma missão é realizada num astro podendo num astro haver várias missões.

Atributos: Este relacionamento não tem atributos.

I. Missão dispõe actividade

Uma missão tem várias actividades.

Cardinalidade: Missão(1); Actividade(N) - Uma missão tem várias actividades, mas uma actividade está associada a apenas uma missão.

Atributos: Este relacionamento não tem atributos.

J. Missão possui Caderno de notas

Uma missão tem sempre um caderno de notas.

Cardinalidade: Missão(1); Caderno de notas(1) - Uma missão tem um caderno de notas associado, e um caderno de notas corresponde a uma missão.

Atributos: Este relacionamento não tem atributos.

K. Caderno de notas inclui Registo

Caderno de notas é composto ou não por registos.

Cardinalidade: Caderno de Notas(1); Registos(N) - Num caderno de notas podem existir registos de actividades caso o astronauta as publique, mas um registo existente diz respeito a um caderno de notas.

Atributos: Este relacionamento não tem atributos.

L. Registo efetuado em Ponto de passagem

Um registo tem uma localização que é denominada por Ponto de passagem.

Cardinalidade: Registo(N); Ponto de passagem(N) - Um ponto de passagem existe, ou não, num ou em vários registos, e num registo pode existir ou não um determinado ou vários pontos de passagem.

Atributos: Este relacionamento não tem atributos.

M. Caderno de notas é composto Ponto de passagem

Caderno de notas é composto por vários pontos de passagem.

Cardinalidade: Caderno de Notas(N); Ponto de Passagem(N) - Num caderno de notas existem vários pontos de passagem e um ponto de passagem pode estar associado a vários cadernos de notas, de diferentes missões.

Atributos: Este relacionamento não tem atributos.

N. Missão tem Percurso

Uma missão tem um percurso associado.

Cardinalidade: Missão(1); Percurso(1) - Uma missão tem apenas um percurso associado, e este tem de existir.

Atributos: Este relacionamento não tem atributos.

O. Percurso contém Ponto de passagem

Um percurso tem vários pontos de passagem.

Cardinalidade: Percuso(1); Ponto de passagem(N) - Uma missão tem um ou mais pontos de passagem e um ponto de passagem é relativo a um percurso.

Atributos: Este relacionamento não tem atributos.

5.1.3 Identificação das Chaves Primárias

Uma chave primária identifica, de forma única e para cada entidade, um dos seus atributos como um registo da tabela.

Em cada entidade, há atributos que podem ser elementos identificadores únicos de um registo na tabela, chamados de chaves candidatas, aos quais, entre elas, escolhemos as que melhor representam os registos nas tabelas, ou seja, as chaves primárias.

Como as chaves primárias têm de ser únicas, não nulas e, posteriormente, não podem ser alteradas, escolhemos as seguintes chaves primárias:

Entidade	Chave primária
Missão	id_Missao
Astronauta	id_Astronauta
Catálogo	id_Catalogo
Missão	Id_Missao
Histórico	id_Historico
Área triângulo	id_AreaTriangulo
Veículo	id_Veiculo
Astro	id_Astro
Atividade	id_Atividade
Caderno de notas	id_Caderno
Registo	id_Registo
Ponto de passagem	id_Passagem
Percurso	id_Percurso

Figura 15 – Identificação das chaves primárias

5.2. Modelo Lógico

Neste capítulo é explicada a derivação do modelo conceptual para a criação e desenvolvimento do modelo lógico.

O modelo lógico é baseado num modelo de dados específicos, independente de qualquer SGBD, que nos permite verificar as ligações entre as tabelas da nossa BD, assim como as chaves primárias de cada tabela e os seus componentes, ou seja, é a tradução do nosso modelo conceptual num modelo válido para a BD do projeto em questão. O nosso modelo lógico é representado em anexo a este relatório.

5.2.1 Validação segundo as Regras de Normalização

A normalização de dados é uma técnica que nos permite a organização dos dados na BD relacional, de forma a garantir a redundância, a flexibilidade e a integridade dos dados.

Uma BD normalizada deverá ter valores atômicos em todos os atributos, ausência de redundância, utilização mínima de valores nulos e perdas de informação mínimas.

Existem ao todo 8 formas normais, mas as normalizações mais importantes são a Primeira Forma Normal (1FN), a Segunda Forma Normal (2FN) e a Terceira Forma Normal (3FN), pelo que revisamos o nosso modelo lógico até à 3FN.

Ora, visto que o nosso modelo lógico cumpre as três formas normais, como se pode verificar pela imagem do modelo lógico, em anexo a este relatório, podemos afirmar que a nossa BD se encontra normalizada.

5.2.2 Validação segundo as Transações do Utilizador

Uma transação é um conjunto de operações que produz sempre um resultado positivo, contribuindo assim para a garantia da consistência da BD, sendo que as transações têm propriedades ACID (atomicidade, consistência, isolamento e durabilidade).

Como faz todo o sentido que existem utilizadores que tenham apenas acesso à BD para consulta e utilizadores com acesso total, criamos dois tipos de utilizadores (Astronauta e Funcionário, sendo que o Astronauta é também identificado pelo seu cargo – Geólogo, Médico, Cartógrafo e Gestor de Inventário) que terão diferentes questões a fazer à BD. Sendo assim, teremos então diferentes níveis de privilégios que irão condicionar as possíveis transações dos diferentes utilizadores.

5.3. Elaboração e Validação do Esquema Lógico da Base de Dados

Na passagem do modelo conceptual para o seu lógico equivalente, há certas regras a serem respeitadas e que serão descritas de seguida.

No modelo lógico, não é possível existirem relacionamentos de cardinalidade M:N, por isso é necessário decompô-los numa entidade intermédia, com dois relacionamentos 1:N.

Os relacionamentos complexos também devem ser decompostos numa entidade intermédia, com relacionamentos de grau 2.

Os relacionamentos recursivos também devem ser decompostos numa entidade intermédia, de modo a que esteja relacionada com a entidade recursiva em questão.

Os relacionamentos com atributos não devem fazer parte do modelo lógico, pelo que é necessário decompô-los numa entidade fraca e com dois relacionamentos 1:N.

Os relacionamentos multivalor também não devem fazer parte do modelo lógico, pelo que devem ser também decompostos.

Os relacionamentos com cardinalidade 1:1, no modelo lógico, devem ser reexaminados e juntadas.

Em algumas situações, também é necessário remover relacionamentos redundantes, que possam ser obtidos através de outros relacionamentos.

Ora, como se pode verificar pelo nosso modelo lógico em anexo a este relatório, o nosso modelo lógico é válido.

5.4. Definição do Tamanho Inicial da Base de Dados e Análise do seu Crescimento Futuro

No desenvolvimento de um BD, é importante que esta seja extensível e que tenha capacidade de evoluir para suportar novos requisitos com efeito mínimo sobre os utilizadores existentes.

No início, povoamos a nossa BD com poucas entradas de dados, para testar e verificar o funcionamento da mesma, mas a previsão é que possa suportar, no futuro, uma quantidade grande de dados.

5.5. Revisão do Modelo Lógico Final com os Futuros Utilizadores do Sistema de Bases de Dados

Antes de implementarmos fisicamente a BD, é necessário rever o modelo lógico com os utilizadores para garantir que estes considerem que o modelo representa exatamente os requisitos pretendidos.

Sendo assim, verificamos que o nosso modelo cumpria com os requisitos e objetivos propostos, pelo que procedemos à próxima e última etapa da elaboração da BD em causa.

5.6. Modelo Físico

Neste capítulo é explicada a derivação do modelo lógico para a criação e desenvolvimento do modelo físico.

O modelo físico permite-nos descrever como os dados estão organizados e guardados na BD, contendo o formato e ordem dos registos e os caminhos para acesso aos dados. Permite-nos mapear o modelo de dados representacional no modelo de dados físico. O nosso modelo lógico é representado em anexo a este relatório.

Foi necessário adicionar uma nova tabela chamada Funcionários, pois a aplicação precisa de diferenciar os utilizadores denominados Astronautas que vão participar em missões, dos utilizadores que apenas podem inserir e remover Astronautas, não tendo estes conexão com todas as outras funcionalidades da aplicação.

5.6.1 Tradução do Modelo Lógico para um SGBD e Consequente Implementação

O SGBD utilizado para o nosso projeto foi o SQL Server.

De seguida é apresentada algum código para o desenvolvimento da nossa Base de Dados.

```
-- -----  
-- Esquema ExploradorEspacial  
-- -----  
  
CREATE DATABASE ExploradorEspacial  
GO  
  
USE ExploradorEspacial  
GO
```

Figura 16 – Criação da Base de Dados “Explorador Espacial”

```
-- -----  
-- Tabela `ExploradorEspacial`.`Astronauta`  
-- -----  
  
CREATE TABLE Astronauta (  
    id_Astronauta INT PRIMARY KEY NOT NULL AUTO_INCREMENT,  
    NomeAstronauta VARCHAR(45) NOT NULL,  
    PasswordAstronauta VARCHAR(100) NOT NULL,  
    CargoAstronauta VARCHAR(45) NOT NULL)  
GO
```

Figura 17 – Criação da tabela “Astronauta”

```
-- -----  
-- Tabela `ExploradorEspacial`.`Catalogo`  
-- -----  
  
CREATE TABLE Catalogo (  
    id_Catalogo INT PRIMARY KEY NOT NULL AUTO_INCREMENT,  
    NomeCatalogo VARCHAR(45) NOT NULL,  
    DescricaoCatalogo VARCHAR(100) NOT NULL)  
GO
```

Figura 18 – Criação da Tabela “Catalogo”

```

-----
-- Tabela `ExploradorEspacial`.`Astronauta_acede_Catalogo`
-----
CREATE TABLE Astronauta_acede_Catalogo (
  Astronauta_id_Astronauta INT NOT NULL,
  Catalogo_id_Catalogo INT NOT NULL,
  CONSTRAINT fk_Astronauta_acede_Catalogo_Astronauta1
    FOREIGN KEY (Astronauta_id_Astronauta)
    REFERENCES Astronauta (id_Astronauta),
  CONSTRAINT fk_Astronauta_acede_Catalogo_Catalogo1
    FOREIGN KEY (Catalogo_id_Catalogo)
    REFERENCES Catalogo (id_Catalogo))
GO

```

Figura 19 – Criação da tabela “Astronauta acede Catalogo”

```

-----
-- Tabela `ExploradorEspacial`.`Missao`
-----
CREATE TABLE Missao (
  id_Missao INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
  DataInicio DATE NOT NULL,
  DataFim DATE NOT NULL,
  TipoMissao VARCHAR(45) NOT NULL,
  NomeMissao VARCHAR(45) NOT NULL,
  Astronauta_id_Astronauta INT NOT NULL,
  Caderno_de_Notas_id_Caderno INT NOT NULL,
  Historico_id_Historico INT NOT NULL,
  Percurso_id_Percurso INT NOT NULL,
  Astro_id_Astro INT NOT NULL,
  CONSTRAINT fk_Missao_Astronauta1
    FOREIGN KEY (Astronauta_id_Astronauta)
    REFERENCES Astronauta (id_Astronauta),
  CONSTRAINT fk_Missao_Caderno1
    FOREIGN KEY (Caderno_de_Notas_id_Caderno)
    REFERENCES Caderno_de_Notas (id_Caderno),
  CONSTRAINT fk_Missao_Historico1
    FOREIGN KEY (Historico_id_Historico)
    REFERENCES Historico (id_Historico),
  CONSTRAINT fk_Missao_Percurso1
    FOREIGN KEY (Percurso_id_Percurso)
    REFERENCES Percurso (id_Percurso),
  CONSTRAINT fk_Missao_Astro1
    FOREIGN KEY (Astro_id_Astro)
    REFERENCES Astro (id_Astro))
GO

```

Figura 20 – Criação da tabela “Missao”

```

-----
-- Povoamento da tabela `ExploradorEspacial`.`Astronauta`
-----

begin
insert into Astronauta values
  (1,'Maria Helena','mariahelena52','Gestor de Missao')
insert into Astronauta values
  (2,'Pedro Cunha','pcunha92','Especialista de Rochas')
insert into Astronauta values
  (3,'Pedro Alves','pedroaalves','Medico')
insert into Astronauta values
  (4,'Joana Arantes','joarantes91','Cartografo')
insert into Astronauta values
  (5,'Daniel Malhadas','dmalhadas5','Geologo')
insert into Astronauta values
  (6,'Jessica Pereira','jessyper','Gestor de Missao')
insert into Astronauta values
  (7,'Rita Rua','ruar','Geologo')
insert into Astronauta values
  (8,'Susana Mendes','sukinha1904','Cartografo')
insert into Astronauta values
  (9,'Maria Pereira','mary_14','Especialista de Rochas')
insert into Astronauta values
  (10,'Marcos Pereira','marcos1234','Medico')
insert into Astronauta values
  (11,'Alexandre Silva','alex1234','Gestor de Missao')
end

```

Figura 21 – Povoamento da tabela “Astronauta”

6. Especificação UML

UML é uma linguagem de modelagem que permite representar um sistema de forma padronizada que auxilia a visualizar a comunicação entre os objetos.

Para isso, neste capítulo, iremos apresentar a nossa especificação UML para a nossa aplicação em causa.

6.1. Modelo de Domínio

O modelo de domínio é a representação visual das classes conceptuais num domínio do problema.

O nosso modelo de domínio encontra-se em anexo a este relatório.

6.2. Modelo de Use Case

Os modelos de Use Case descrevem as funcionalidades propostas para o sistema.

O nosso modelo de Use Case está representado de seguida.

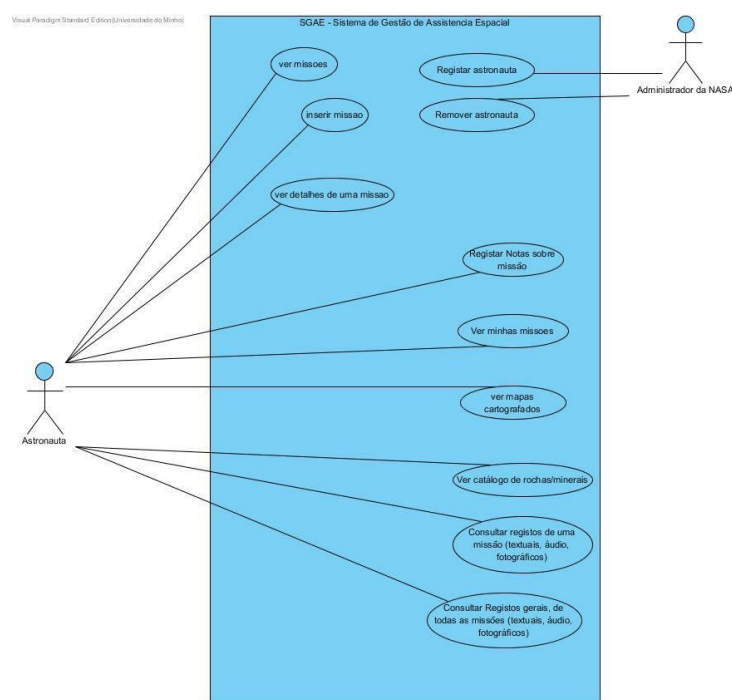


Figura 22 – Modelo de Use Case

ver missoes

Info

Use Case Notes

Flow of Events

Details

Requirements

Diagrams

Test Plan

References

Description

Agency FB

8

Full

Super Use Case

Primary Actor

Secondary Actor(s)

Brief Description

Preconditions

Flow of Events

Post-conditions

Alternative flows and exceptions

Astronauta

Use case que devolve a lista das missoes que se encontram no sistema.

O utilizador encontra-se credenciado no sistema

	Actor Input	System Response
1	Utilizador solicita a lista das missoes que se encontram armazenadas no sistema	
2		O sistema calcula internamente a lista de missoes
3		O sistema devolve ao utilizador a lista de missoes que se encontram no sistema

O utilizador teve acesso a lista de missoes

2-1 O sistema verifica que a lista é vazia e informa o utilizador

Lista Vazia(passo 2)

Figura 23 – Especificação Ver Missões

Como já foi referido, existem vários tipo de missões, das quais:

- Cartografar/Mapear a Lua: Identificar características do solo.
- Montar peças: Telescópios.
- Verificar a lista de instrumentos utilizados para permitir ou não outra pessoa de o usar: Se alguém precisar de um determinado instrumento mas ele estiver a ser usado por outrem, permitir saber quando esse instrumento está disponível para esta nova pessoa poder efetuar a sua tarefa/missão.
- Fazer perfurações para preparar local de trabalho para encontrar pedras.
- Trazer pedras: "As 2 200 pedras trazidas pelas seis missões Apollo que pousaram na superfície da Lua cerca de 400 kg de rochas, 75 novas variedades de minerais, a maioria silicatos".
- Encher depósito de Rover.
- Atribuir missões (dar permissões a dados utilizadores).

34

6.3. Diagrama de Classes

Os diagramas de classes definem todas as classes que o sistema necessita possuir. O nosso diagrama de classes encontra-se em anexo a este relatório, melhorado.

6.4. Diagrama de Sequência

Os diagramas de sequência representam a sequência de processos. O nosso diagrama de sequência apresenta-se de seguida.

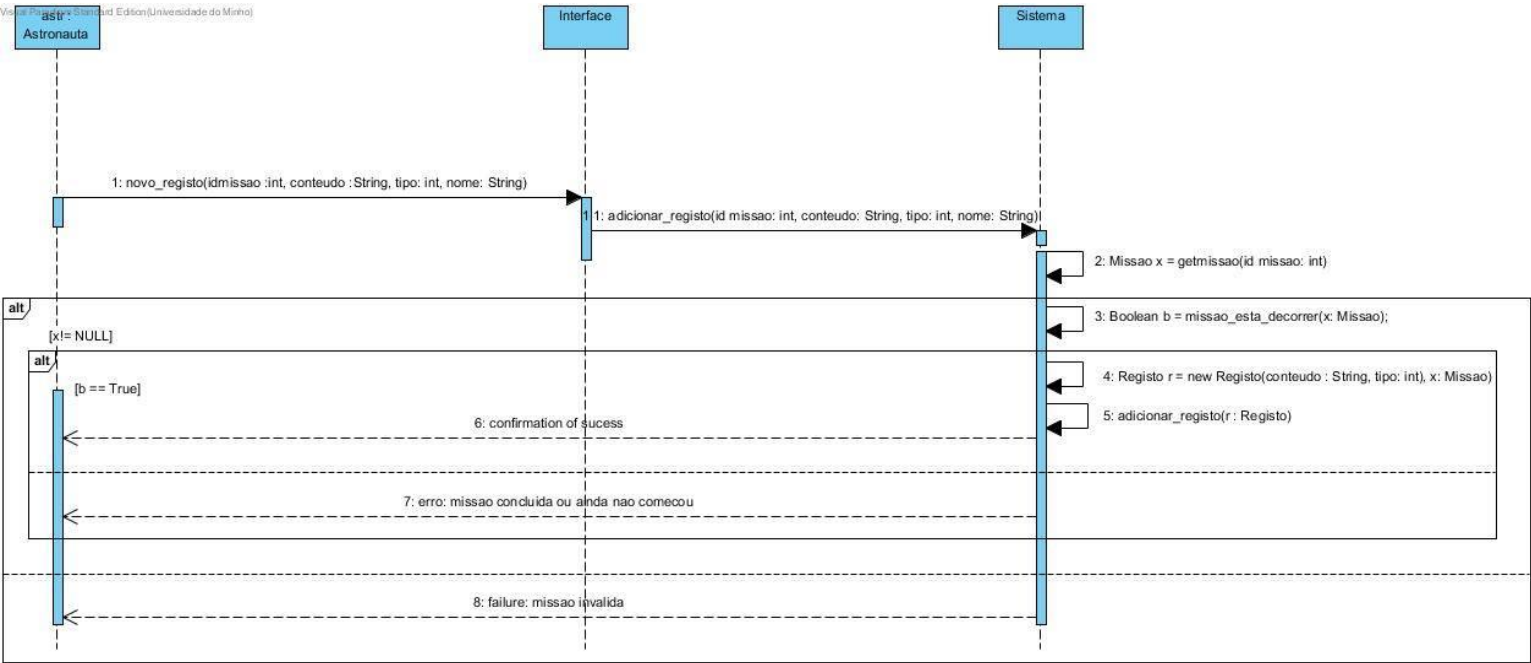


Figura 24 – Diagrama de sequência

7. Motivos para a Escolha da Plataforma

Uma vez que nos foi solicitado desenvolver uma aplicação recorrendo a *software* de desenvolvimento da Microsoft, achamos que seria ideal que a aplicação fosse desenvolvida também para a plataforma Windows mais recente: Windows 10.

A razão desta escolha deve-se ao facto de que usar as *APIs* unificadas da Microsoft entre as ferramentas de desenvolvimento e o sistema operativo onde a aplicação irá funcionar torna o desenvolvimento do produto final mais fácil. Se implementássemos a aplicação, por exemplo, para a plataforma Android, seria mais trabalhoso e complicado conseguir "cruzar" as *APIs* da Microsoft, nomeadamente Microsoft SQL Server, com as *APIs* de desenvolvimento Android. Seria necessário, para tal, arranjar *drivers* que fizessem a ligação. De facto, esses *drivers* existem, mas necessitaríamos de um tempo suplementar para uma correta aprendizagem e manuseamento.

Não devemos esquecer que a plataforma Windows 10 é uma plataforma unificada, ou seja, trata-se rigorosamente do mesmo sistema operativo nos dispositivos em que opera, nomeadamente: portáteis, *tablets* e *smartphones*. Por isso, mesmo que os testes à nossa aplicação sejam feitos num portátil, sabemos que esse mesmo executável Windows (.exe), a ser executado num *tablet* ou num *smartphone*, funcionaria de igual forma. Também sabemos que temos recurso a emuladores no Microsoft Visual Studio que nos permitem ter uma visualização de como será apresentada a nossa aplicação num *smartphone* ou outro tipo de sistema mobile.

Esta escolha faz que a nossa aplicação "Agente de Campo" possa ser usada de forma móvel, por exemplo, quando o astronauta se encontra em missão e pretende registar um determinado acontecimento observado, ou também ser usada, por exemplo, no final de um dia, num portátil, quando um determinado astronauta pretende criar uma missão para um determinado dia e apontar os membros que pretende que ingressem a missão.

O uso de uma plataforma única em dispositivos de diferente natureza, faz também que, dado o tempo limitado de desenvolvimento que dispusemos para desenvolver a aplicação, o tempo para *debugging* seja consideravelmente menor.

8. Organização e Implementação das Classes - Aplicação

Seguindo a especificação UML, e mais especificamente o Diagrama de Classes, tentamos criar as diferentes camadas para a criação da aplicação: a camada data que diz respeito à implementação da Base de Dados em SQL Server, a camada Business que diz respeito ao desenvolvimento, e disponibilização de métodos importantes para apresentar à ultima camada, camada de interface UI.

Assim sendo, vimos que as classes mais importantes a serem implementadas logo de inicio, na camada Business, seriam as classes Astronauta, contendo todos os parâmetros necessários a este inerentes, tais como ID, Password e Nome para login, Missão e Caderno de Notas para serem guardados os Registos dos Astronautas.

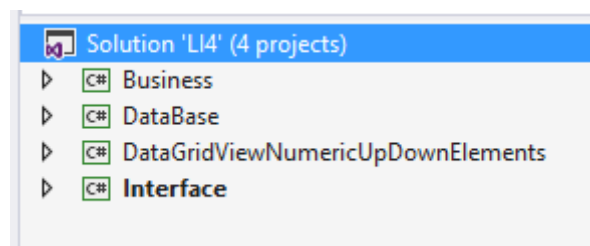


Figura 25 – Camadas da aplicação

De seguida, são apresentadas as camadas acima explicitadas.

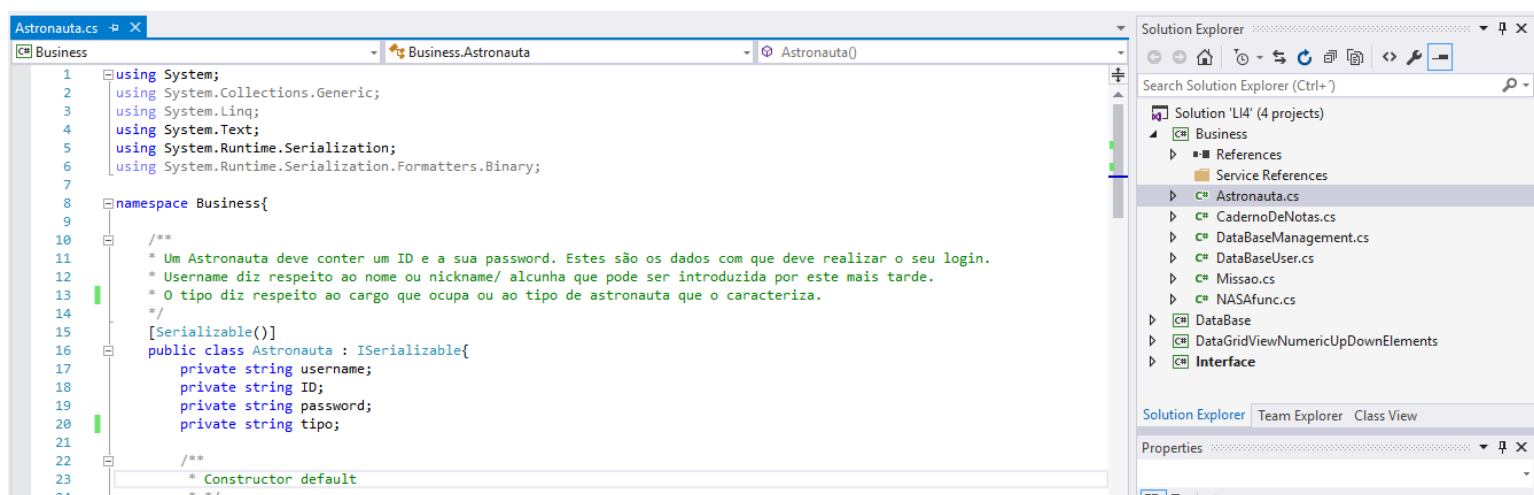


Figura 26 – Camada Bussiness

```

    * Constructor with Astronauta
    * */
public Astronauta(Astronauta u){
    username = u.Username;
    ID       = u.Id;
    tipo     = u.Tipo;
    password = u.Password;
}

//TODO falta completar com os novos parametros tambem
/**
 * Deserialization Constructor
 * */
public Astronauta(SerializationInfo info, StreamingContext ctxt){
    username = (string)info.GetValue("Username", typeof(string));
    ID       = (string)info.GetValue("ID",      typeof(string));
    tipo     = (string)info.GetValue("tipo",    typeof(string));
    password = (string)info.GetValue("Password", typeof(string));
}

/**
 * Methods get and set
 * */
public string Username
{
    get {
        return username;
    }
    set {
        username = value;
    }
}

public string Id
{

```

Figura 27 – Classe Astronauta com os seus parâmetros e construtores

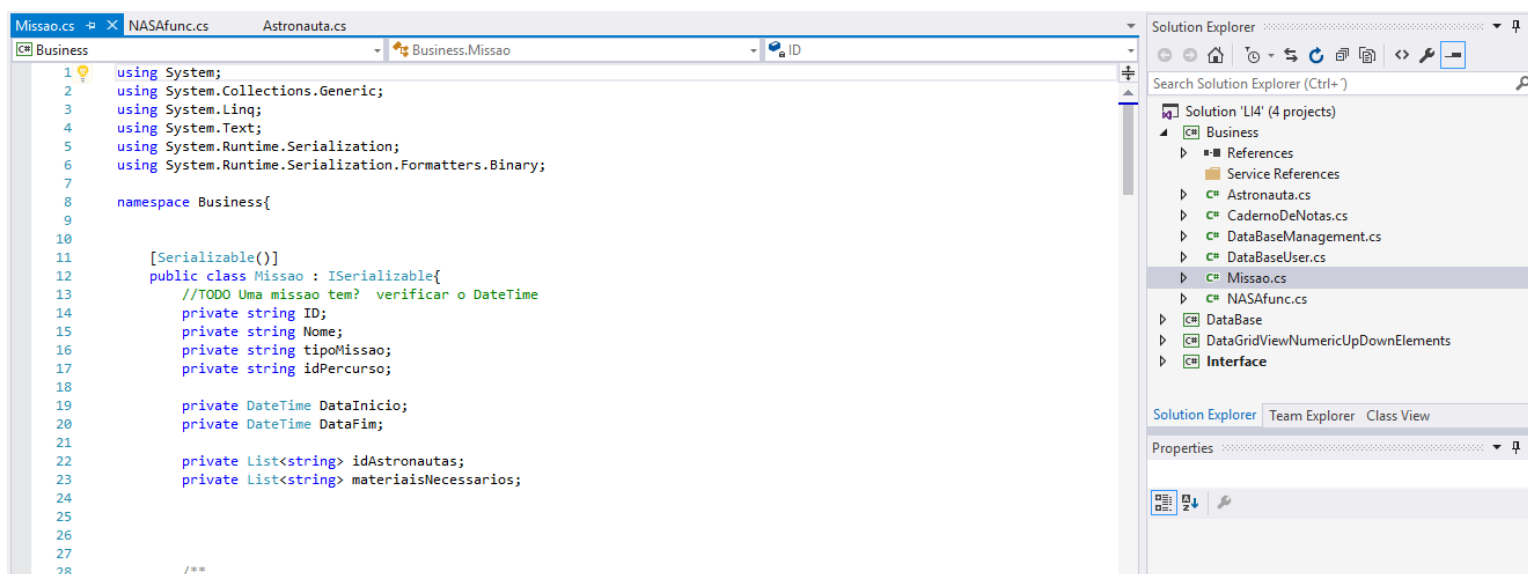


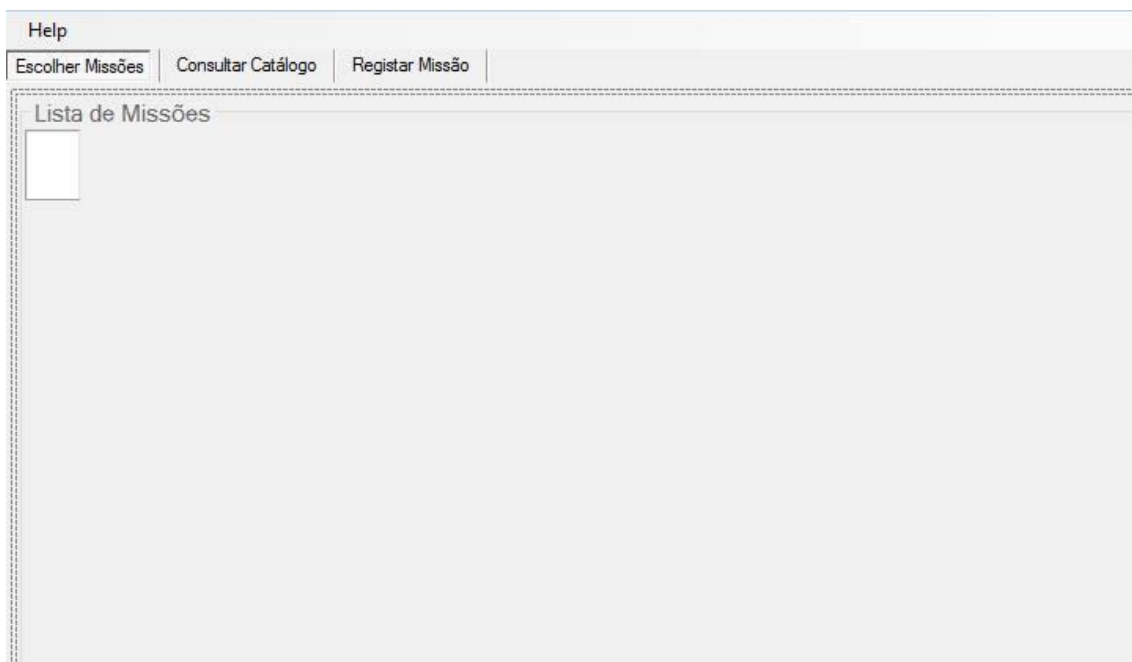
Figura 28 - Classe Missão com os seus parâmetros

9. Resultado Final



The screenshot shows a window titled "Agente de Campo - Explorador Espacial". Inside, there is a section titled "Registrar Astronauta" with the instruction "Por favor, registre o Astronauta." Below this, there are four input fields: "Nome", "ID", "Password", and "Confirmação de Password". To the right of the "Confirmação de Password" field are two buttons: "Registrar" and "Cancelar".

Figura 29 – Ecrã inicial



The screenshot shows the same window, but now displaying the "Lista de Missões" screen. At the top, there is a "Help" button and three tabs: "Escolher Missões", "Consultar Catálogo", and "Registrar Missão". The "Escolher Missões" tab is selected. Below the tabs, there is a large empty area labeled "Lista de Missões".

Figura 30 – Ecrã das listas das missões disponíveis

10. Ferramentas Utilizadas

Para a realização deste projeto, foi necessário a utilização das seguintes ferramentas:

- Microsoft Office Word para a escrita do relatório;
- Microsoft Office PowerPoint para a apresentação do projeto;
- Visual Paradigm para a especificação UML;
- SQL Server e Visual Studio para o desenvolvimento da aplicação.

11. Conclusões e Trabalho Futuro

Na primeira fase, foi elaborada a fundamentação da aplicação. Esta parte foi bem conseguida, devido ao tempo que nos foi dado para a sua realização e pelo contacto imediato e bem conseguido com o nosso cliente (a NASA). Os requisitos foram tidos em conta como a parte mais importante do trabalho, pois a partir daí temos uma ideia do que temos de elaborar para realizar estas tarefas e a futura aplicação.

Na segunda fase especificamos o desenvolvimento da aplicação com documentação UML detalhada, usando um método organizado para a realização desta tarefa. Cumprimos com as tarefas de realização de diagramas *Use Case*, diagramas de classe, entre outros. Geramos a documentação de uma forma rápida para podermos partir rapidamente para a construção da aplicação.

Na terceira fase, apresentamos a arquitetura do sistema, descrevendo os módulos desta, a organização e desenvolvimento da aplicação e os motivos de escolha da plataforma Windows 10, que permitiu a criação de uma aplicação que é universal a todos os dispositivos, desde que tenham o sistema operativo do Windows 10.

Como trabalho futuro, temos em conta a importância destas tarefas e das suas realizações no futuro. Prevemos que a aplicação poderá ser aperfeiçoada, como por exemplo, com a implementação de um perfil de administrador que permita a gestão dos utilizadores e a manutenção da aplicação.

Referências

1. Sommerville, Ian, 2011, *Software Engineering*, Ninth Edition, Addison-Wesley.
2. National Aeronautics and Space Administration, 2004, *The Vision for Space Exploration*. [pdf] National Aeronautics and Space Administration. Available at: https://www.nasa.gov/pdf/55583main_vision_space_exploration2.pdf [Accessed March 2016].

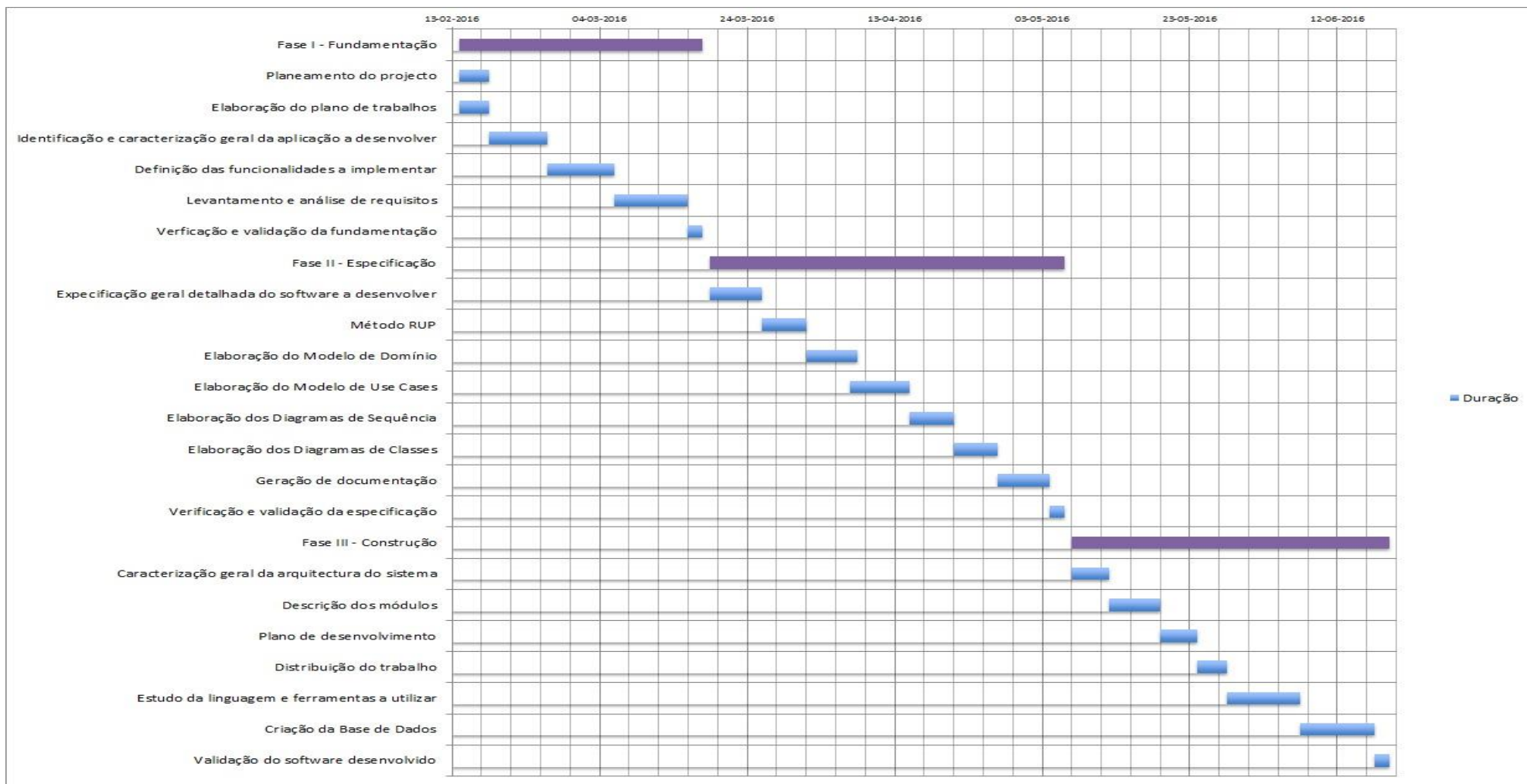
Lista de Siglas e Acrónimos

NASA	National Aeronautics and Space Administration
BD	Base de Dados
GPS	Global Positioning System
SGBD	Sistema de Gestão de Base de Dados
UML	Unified Modeling Language
ID	Identity
SQL	Structured Query Language
API	Application Programming Interface

Anexos

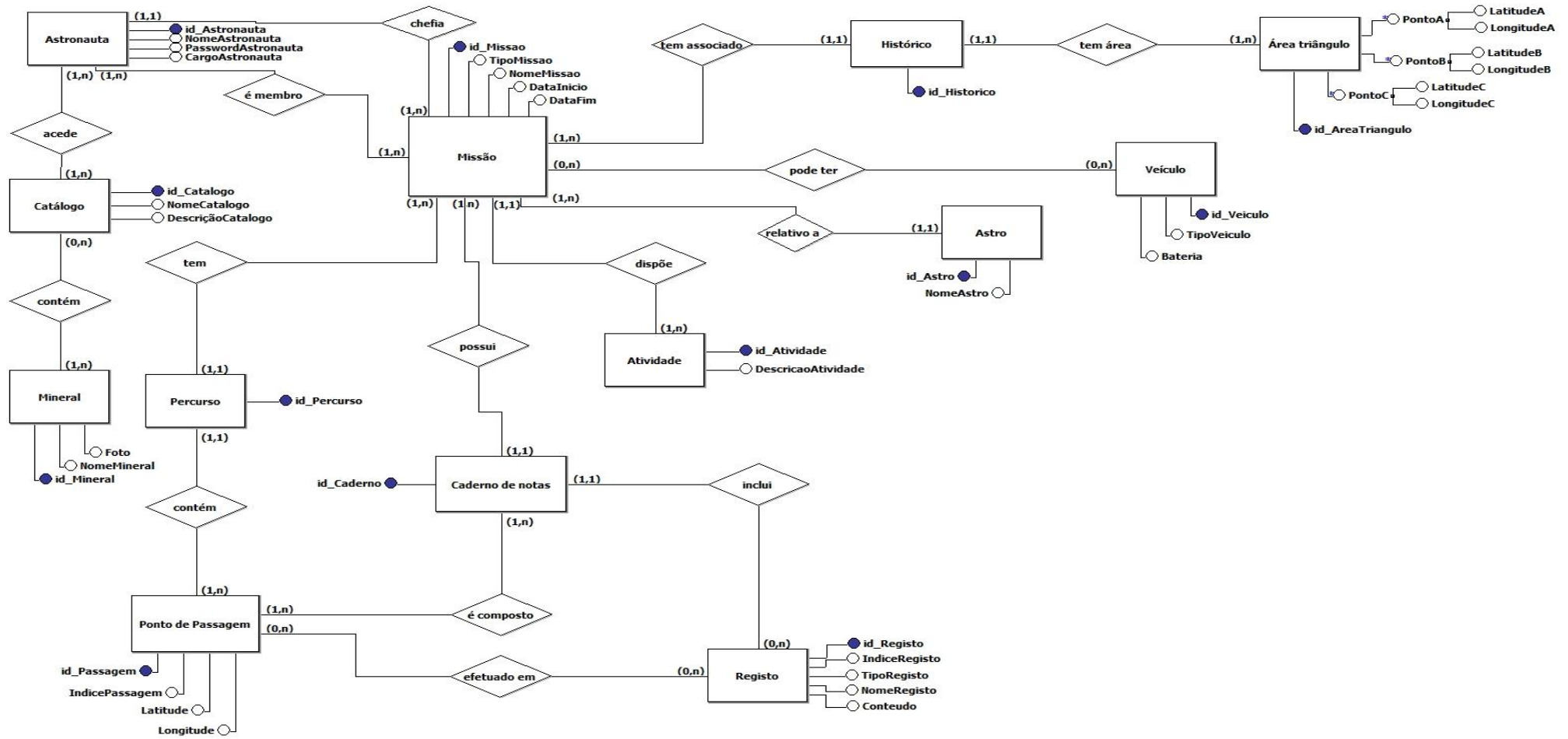
I. Anexo 1

Plano de atividades – Diagrama de Gantt



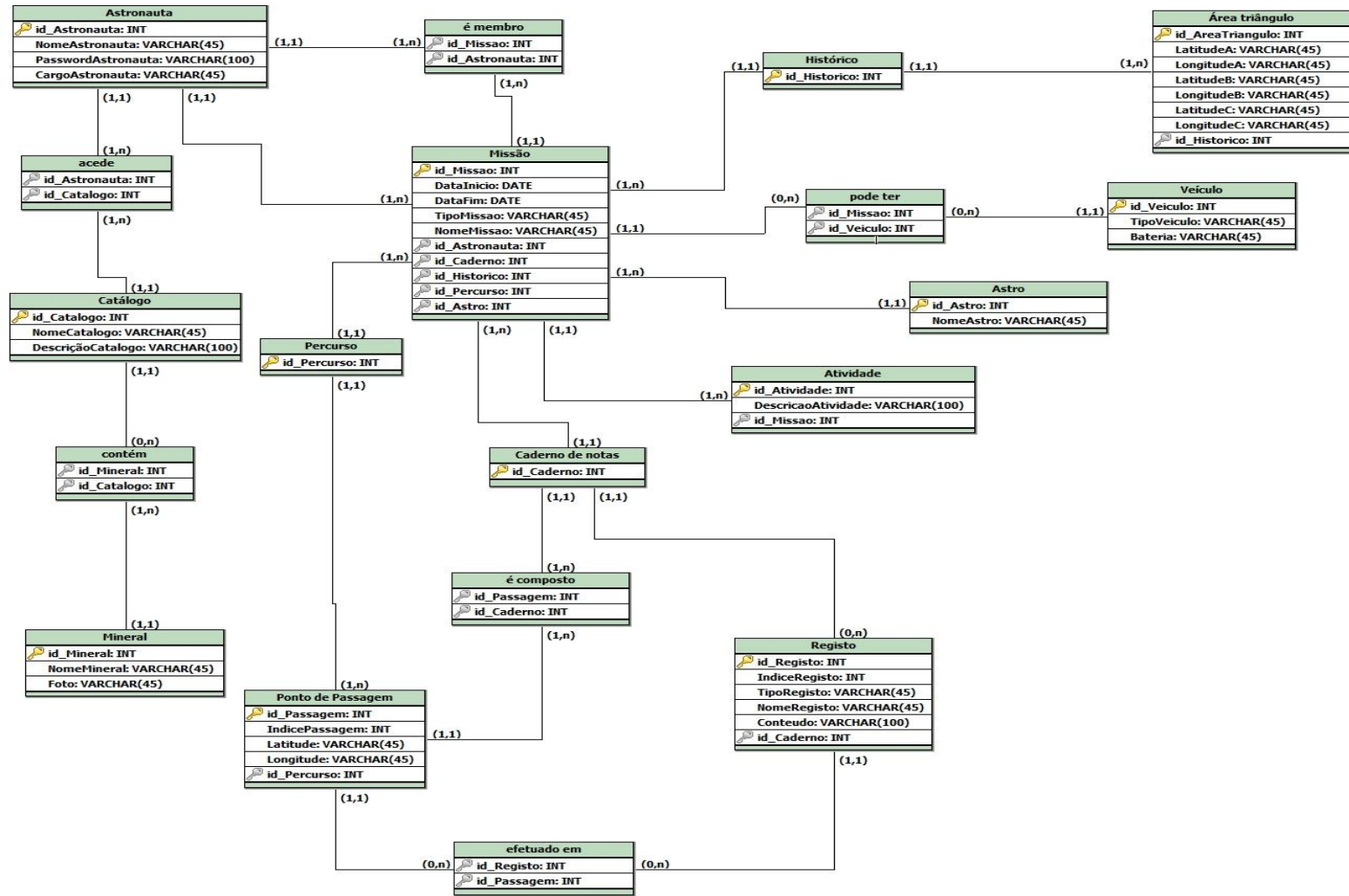
II. Anexo 2

Modelo conceitual



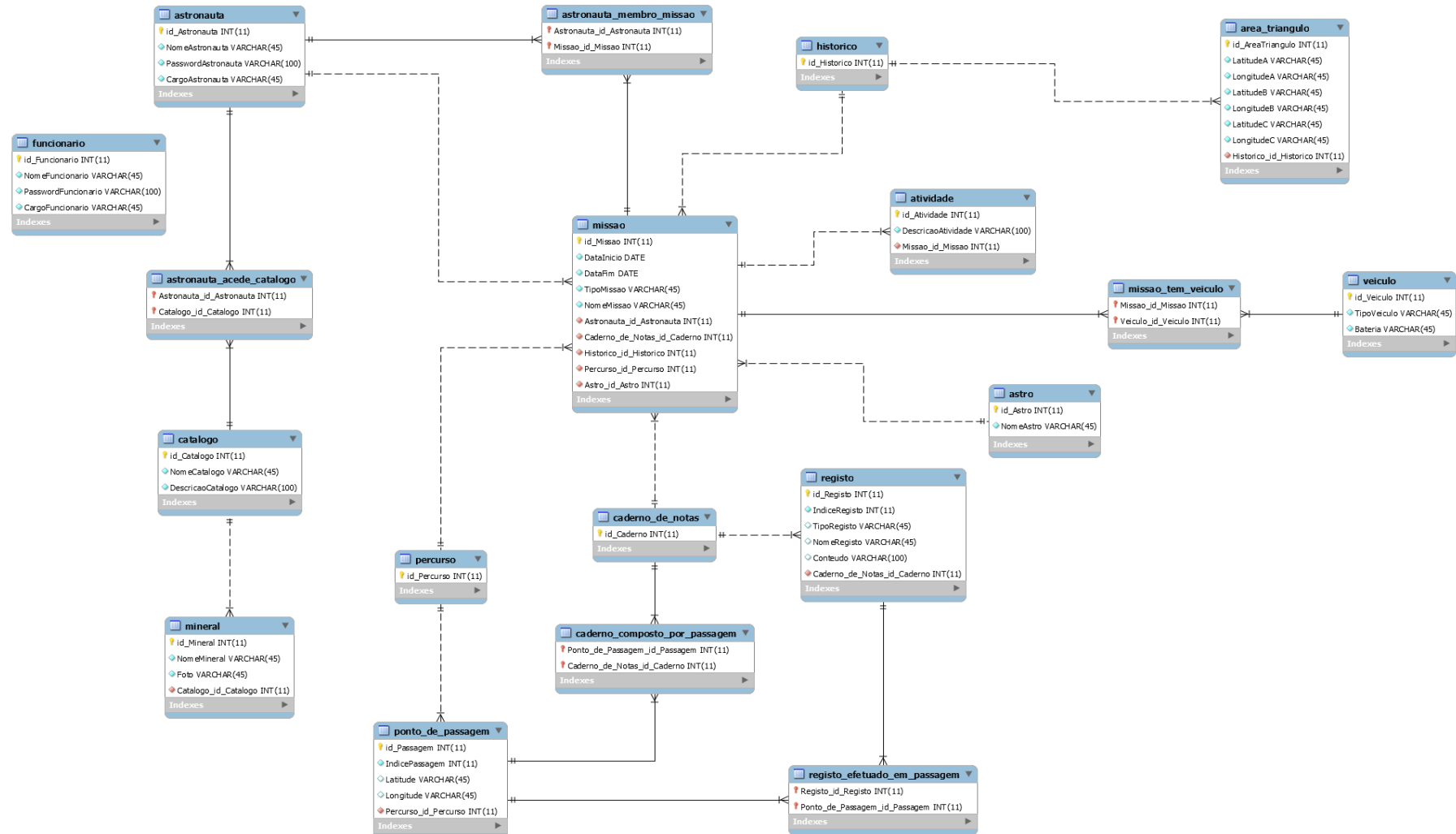
III. Anexo 3

Modelo lógico



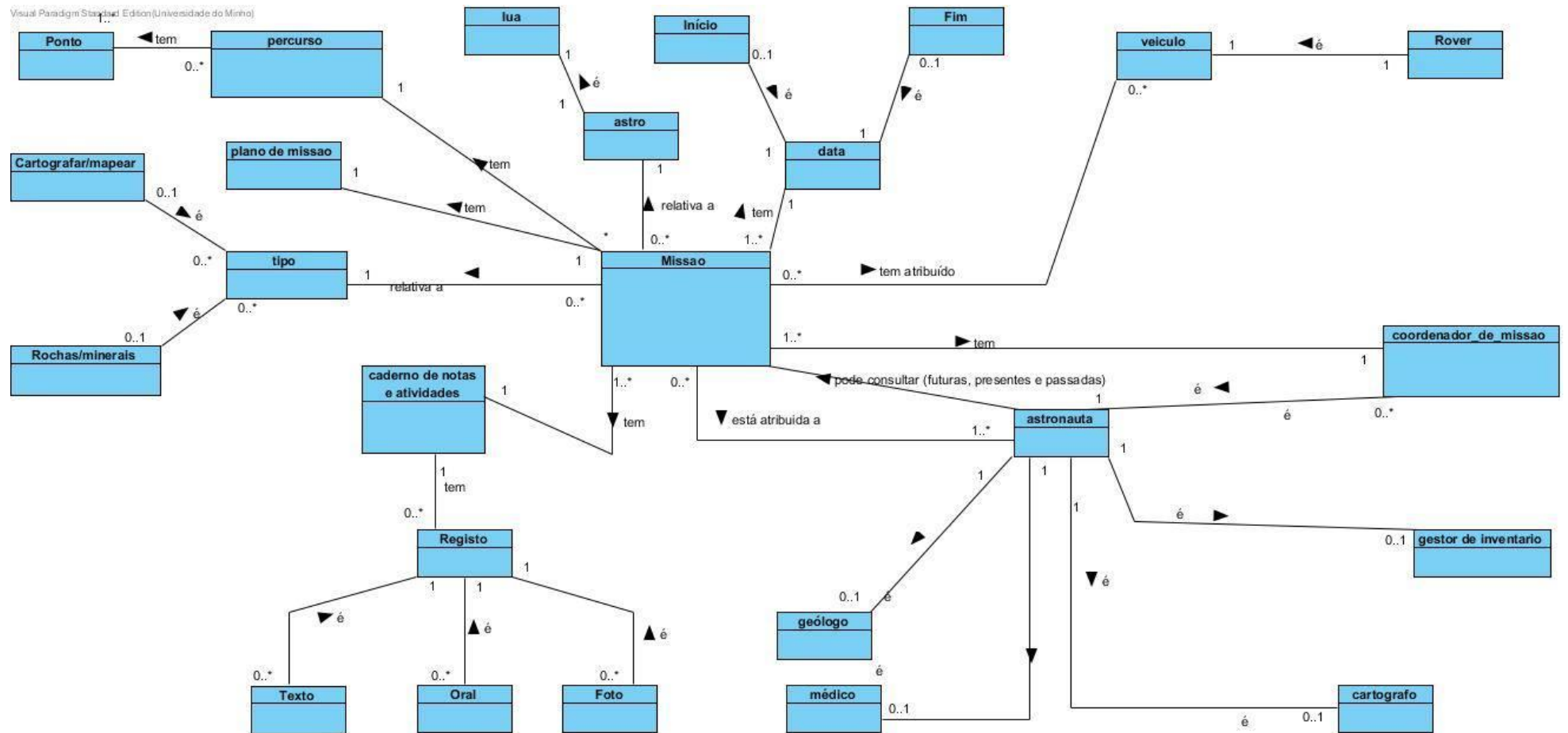
IV. Anexo 4

Modelo físico



V.Anexo 5

Modelo de Domínio



VI. Anexo 6

Diagrama de Classes

