



Universidade do Minho

Escola de Engenharia

Mestrado Integrado em Engenharia Informática

Unidade Curricular de Laboratórios de Informática IV

Ano Letivo de 2020/2021

TDelivery

Daniela Fernandes (A73768)

José Gomes (A82418)

Rui Fernandes (A89138)

Tiago Rodrigues (A84276)

7 de junho de 2021

LI4

Data de Receção	
Responsável	
Avaliação	
Observações	

TDelivery

Daniela Fernandes (A73768)

José Gomes (A82418)

Rui Fernandes (A89138)

Tiago Rodrigues (A84276)

7 de junho de 2021

Resumo

Este documento diz respeito ao trabalho prático da Unidade Curricular de Laboratórios de Informática IV, que tem como objetivo o desenvolvimento de um sistema de *software* orientado à monitorização de eventos. Depois de uma profunda reflexão, optamos por desenvolver um sistema de *tracking* em tempo real de entregas de encomendas de uma empresa com vários entregadores.

Neste relatório é apresentada e detalhada a primeira e segunda etapa do desenvolvimento desta solução de *software*. A primeira etapa compreendeu a apresentação do domínio da aplicação, a apresentação do sistema que nos propusemos a desenvolver, a descrição do modelo do sistema a apresentar, a apresentação do plano de desenvolvimento, uma elaboração sobre a viabilidade da solução proposta assim como a identificação das métricas para a avaliação do grau de satisfação dos objetivos.

A segunda etapa do projeto marcou o início do processo de materialização das ideias e expectativas apresentadas. Este início foi marcado pela recolha e análise dos requisitos funcionais e não funcionais do sistema. De seguida, em conformidade com o *Rational Unified Process* (RUP), estes requisitos deram lugar a *use cases* concretos que, quando apresentados e especificados, juntamente com um modelo de domínio, estabelecem as fundações do sistema a desenvolver.

Posteriormente, num processo de modelação iterativo, foi concebido um diagrama de classes, resultado da análise e discussão dos aspetos estruturais do sistema a desenvolver, que representa o desenho arquitetural da proposta de *software*.

Por outro lado, para que os dados, dos diferentes componentes guardadas, se mantivessem, quando sujeitos a alteração por vários utilizadores, íntegros e desta forma disponíveis para consulta durante todo o tempo de vida do produto final, foram desenhados o modelo conceptual e lógico da Base de Dados, que servirão de fio condutor na implementação desta.

Posto isto, na terceira e última fase foram implementadas todo o sistema.

Deste modo, foram percorridas todas as etapas que consideramos necessárias para que o trabalho seja o mais conciso, metódico e bem fundamentado possível. No final deste documento, as diversas fases do projeto estão devidamente planificadas, de forma a facilitar a sua realização. Por fim, é de salientar que, tratando-se de um processo iterativo e incremental onde a aplicação gerada foi sendo moldada para melhor servir os seus utilizadores, esta possa ser projetada e assim, conquistar o seu merecido espaço no mercado.

Área de Aplicação: Engenharia de *Software*, .

Palavras-Chave: Conceção e Desenvolvimento *Software*, Diagrama de *Gantt*, Modelo *Waterfall*, Arquitetura Multicamada, Gestão de Projeto.

Índice

1	Introdução	1
1.1	Contextualização	2
1.2	Motivação e Objetivos	2
1.3	Estrutura do Relatório	3
2	Fundamentação, Justificação e Utilidade do Sistema	4
2.1	Definição da Identidade do Sistema a Desenvolver	4
2.2	Funcionamento do sistema	5
3	Identificação dos Recursos Necessários	6
4	Modelo do Sistema a Implementar	7
5	Definição de medidas de sucesso	9
6	Plano de desenvolvimento	10
7	Levantamento e Análise de Requisitos	12
7.1	Requisitos Funcionais	13
7.1.1	Registo	13
7.1.2	Autenticação	13
7.1.3	Término de Sessão	14
7.1.4	Apresentar Funcionalidades da Aplicação	14
7.1.5	Apresentar o Contacto dos diferentes Entregadores	14
7.1.6	Consulta de Dados Pessoais	15
7.1.7	Edição de Dados Pessoais	15
7.1.8	Remoção do Perfil	16
7.1.9	Consulta de uma Encomenda	16
7.1.10	Categorizar Encomendas	16
7.1.11	Listagem de Encomendas	17
7.1.12	Consulta de Favoritos	17
7.1.13	Editar Lista de Favoritos	18
7.1.14	Gestão de Utilizadores	18
7.1.15	Exportação de Dados	18
7.2	Requisitos Não Funcionais	19
7.2.1	Requisitos de Produto	19
7.2.2	Requisitos Organizacionais	19

7.2.3	Requisitos Externos	20
8	Modelação	21
8.1	Modelo de Domínio	21
8.1.1	Atores	22
8.2	Diagramas de <i>Use Case</i>	23
8.3	Especificação de <i>Use Cases</i>	24
8.3.1	Efetuar Registo	24
8.3.2	Iniciar sessão	25
8.3.3	Terminar sessão	26
8.3.4	Consultar funcionalidades da aplicação	27
8.3.5	Consultar Estatísticas	27
8.3.6	Consultar Encomenda	28
8.3.7	Listagem de Encomendas	29
8.3.8	Consultar favoritos	29
8.3.9	Adicionar encomenda aos favoritos	30
8.3.10	Remover encomenda dos favoritos	31
8.3.11	Categorizar Encomenda	31
8.3.12	Filtrar Encomendas	32
8.3.13	Consultar Dados do Perfil	33
8.3.14	Editar perfil	34
8.3.15	Remover Perfil	34
8.3.16	Consultar Contacto dos Entregadores	35
8.3.17	Remover Utilizador	36
9	Diagramas de Sequência	37
9.1	Efetuar Registo	38
9.2	Iniciar Sessão	39
9.3	Terminar Sessão	39
9.4	Consultar Funcionalidades da Aplicação	40
9.5	Consultar Estatísticas	40
9.6	Consultar Encomenda	41
9.7	Listagem de Encomendas	42
9.8	Consultar Favoritos	42
9.9	Adicionar Encomenda aos Favoritos	43
9.10	Remover Encomenda dos Favoritos	43
9.11	Categorizar Encomenda	44
9.12	Filtrar Encomendas	45
9.13	Consultar Dados do Perfil	45
9.14	Editar Perfil	46
9.15	Remover Perfil	47
9.16	Consultar Contacto dos Entregadores	48
9.17	Remover Utilizador	49

10 Diagramas de Atividade	50
10.1 Editar Perfil de Utilizador	50
10.2 Editar Lista de Encomendas Favoritas	51
10.2.1 Consultar Encomenda	52
11 Máquina de Estados	53
12 Arquitetura da Camada de Negócios	55
12.1 Dicionário das principais Classes	55
12.2 Diagrama de Classes	56
12.3 Diagrama de Classes ORM	57
13 Modelação da Camada de Dados	58
13.1 Modelo Conceptual	58
13.1.1 Identificação e Caracterização das Entidades	58
13.1.2 Identificação e Caracterização dos Relacionamentos	59
13.1.3 Identificação e caracterização dos atributos com as entidades e relacionamentos	60
13.1.4 Chaves Primárias	62
13.1.5 Diagrama ER	63
13.2 Modelo Lógico	63
13.2.1 Validação do Modelo Lógico	64
13.3 Estimativa do espaço em disco	64
14 Prototipagem	66
15 Implementação do Sistema	72
15.1 Povoamento da Base de Dados	73
15.2 Ferramentas Utilizadas	75
15.2.1 <i>Framework</i> Blazor	75
15.2.2 Visual Studio	75
15.2.3 SQL Server	75
15.2.4 Outras ferramentas	75
15.3 Interface	76
15.3.1 Página Inicial	76
15.3.2 Registo e Autenticação	76
15.3.3 Consultar & Editar Perfil	78
15.3.4 Contactos	79
15.3.5 Consulta de encomenda	80
15.3.6 Sobre Nós	81
15.3.7 <i>Dashboard</i> do Administrador	82
15.3.8 Equipa	83
15.3.9 Termos e Condições	84
15.3.10 Término de Sessão	84
15.4 Tempo & Custo	85

15.5 Desenvolvimento Futuro	85
16 Conclusões e Trabalho Futuro	86
Lista de Siglas e Acrónimos	88

Lista de Figuras

2.1	Logótipo do sistema.	5
4.1	Maqueta do sistema de <i>software</i> a desenvolver	7
6.1	Planeamento e distribuição de tarefas	11
6.2	Diagrama de Gantt.	11
8.1	Modelo de Domínio	22
8.2	Diagrama de <i>Use Case</i>	23
9.1	Diagrama de Sequência correspondente ao <i>use case</i> Efetuar Registo	38
9.2	Diagrama de Sequência correspondente ao <i>use case</i> Iniciar Sessão	39
9.3	Diagrama de Sequência correspondente ao <i>use case</i> Terminar Sessão	39
9.4	Diagrama de Sequência correspondente ao <i>use case</i> Consultar Funcionalidades da Aplicação	40
9.5	Diagrama de Sequência correspondente ao <i>use case</i> Consultar Estatísticas	40
9.6	Diagrama de Sequência correspondente ao <i>use case</i> Consultar Encomenda	41
9.7	Diagrama de Sequência correspondente ao <i>use case</i> Listagem de Encomendas	42
9.8	Diagrama de Sequência correspondente ao <i>use case</i> Consultar Favoritos	42
9.9	Diagrama de Sequência correspondente ao <i>use case</i> Adicionar Encomenda aos Favoritos	43
9.10	Diagrama de Sequência correspondente ao <i>use case</i> Remover Encomenda dos Favoritos	43
9.11	Diagrama de Sequência correspondente ao <i>use case</i> Categorizar Encomenda	44
9.12	Diagrama de Sequência correspondente ao <i>use case</i> Filtrar Encomendas	45
9.13	Diagrama de Sequência correspondente ao <i>use case</i> Consultar Dados do Perfil	45
9.14	Diagrama de Sequência correspondente ao <i>use case</i> Editar Perfil	46
9.15	Diagrama de Sequência correspondente ao <i>use case</i> Remover Perfil	47
9.16	Diagrama de Sequência correspondente ao <i>use case</i> Consultar Contacto dos Entregadores	48
9.17	Diagrama de Sequência correspondente ao <i>use case</i> Remover Utilizador	49
10.1	Diagrama de Atividade relativo à edição do perfil de utilizador	50
10.2	Diagrama de Atividade relativo à edição da lista de encomendas favoritas	51
10.3	Diagrama de Atividade relativo à consulta de uma encomenda	52
11.1	Máquina de Estados	53
11.2	Máquina de Estados relativa ao registo na aplicação	54

12.1 Diagrama de Classes	56
12.2 Diagrama de Classe ORM	57
13.1 Diagrama ER	63
13.2 Modelo Lógico da Base de Dados	64
14.1 Proposta de interface para a <i>landing page</i>	67
14.2 Proposta de interface para a página inicial	67
14.3 Proposta de interface para a página de registo	68
14.4 Proposta de interface para a página de autenticação	68
14.5 Proposta de interface para a página relativa ao estado de uma encomenda	69
14.6 Proposta de interface para o histórico de encomendas	69
14.7 Proposta de interface para a lista de favoritos	70
14.8 Proposta de interface para o perfil do utilizador	70
14.9 Proposta de interface para a <i>dashboard</i> do administrador	71
15.1 Arquitetura de <i>software</i>	72
15.2 Página inicial do sistema	76
15.3 Página de registo de um utilizador	77
15.4 Página de autenticação de um utilizador	78
15.5 Página de consultar e editar perfil	78
15.6 Página com contactos das empresas que efetuam entregas	79
15.7 Página com as informações de uma encomenda	80
15.8 Página sobre nós	81
15.9 Página da <i>dashboard</i> do administrador	82
15.10Página da equipa	83
15.11Página termos e condições	84

Lista de Tabelas

2.1	Identidade do sistema.	4
13.1	Dicionário de Dados das Entidades	59
13.2	Relacionamentos entre as entidades	59
13.3	Atributos da entidade Utilizador	60
13.4	Atributos da entidade Encomenda	61
13.5	Atributos da entidade Entregador	62
13.6	Atributos do relacionamento Acompanha	62
13.7	Espaço ocupado pelos diferentes tipos de dados	65
13.8	Espaço ocupado por cada entrada nas diferentes tabelas	65
13.9	Estimativa do espaço em disco ocupado pela BD após um ano	65

1 Introdução

Nesta introdução visa-se apresentar o projeto, desde a motivação, o nome, a ideia e os objetivos por detrás do mesmo. Neste sentido, vai ser feita uma contextualização do tema selecionado pelo grupo para a realização do projeto de forma a facilitar a interpretação de todos os pontos abordados neste documento. Além disso, outro tópico abordado é o porquê da escolha do tema e quais os objetivos que pretendemos atingir no final da realização do projeto.

Note-se que todo o desenvolvimento do projeto será guiado pelo modelo *Waterfall* [1]. Desta forma, para se proceder ao correto e funcional desenvolvimento de um sistema de *software*, é essencial ser percorrido um conjunto de passos fundamentais:

1. Definição dos requisitos;
2. Modelação do sistema de *software*;
3. Implementação;
4. Integração e testes;
5. Operacionalidade e manutenção do sistema.

Tendo em conta este modelo, iremos seguir uma metodologia sequencial para a realização de todas as fases da aplicação. Para além disso, tivemos em consideração o esforço de cada um dos elementos do grupo, para numa fase final ser possível fazer uma estimativa dos recursos necessários à realização do projeto.

1.1 Contextualização

Todos os dias recebemos e enviamos uma carta ou uma encomenda, e para isso utilizamos empresas destinadas a esse fim. Por vezes queremos rastrear onde um pacote se encontra, se já chegou ao seu destinatário, se falta muito tempo para chegar ao seu destino, ou se ficou preso em algum posto. São diversas as empresas que fazem entregas de vários tipos de encomendas, comida, mobiliário, roupa, cartas, e, ao contrário do que é expectável, nem todas nos garantem uma informação atualizada em tempo real sobre localização da encomenda, ou de qual a duração para a entrega estar concluída.

De acordo com inúmeras pesquisas, o hábito de fazer compras *online* cresceu significativamente em Portugal desde o início da década. Trata-se de uma prática impulsionada pela massificação da tecnologia, aliada a um crescimento da literacia digital, assim como um aumento da oferta por parte das empresas. De facto, segundo o Instituto Nacional de Estatística, em 2020, no contexto da pandemia da Covid-19, a percentagem de utilizadores de internet aumentou 3% em relação ao período homólogo de 2019, sendo que a percentagem de utilizadores de comércio eletrónico registou em 2020, o maior aumento desde 2002 - Em 2020, 44,5% das pessoas dos 16 aos 74 anos fizeram encomendas pela internet. Aumentou, também, a importância despendida por encomenda, em mais de 15,7% [2]. Por outro lado, quase 1/5 do volume de negócios das empresas com 10 ou mais trabalhadores corresponde a comércio eletrónico, sendo que as vendas de bens e serviços através do mesmo representaram 19% do total do volume de negócios em 2019 [3].

Com uma maior preocupação para que as encomendas não fiquem perdidas, ou demorem mais tempo do que é publicitado pelas empresas, surgiu a necessidade de implementar mecanismos de *tracking* em tempo real. É com este propósito que o nosso projeto vai ser desenvolvido: *tracking* em tempo real de entregas de encomendas de uma empresa com vários entregadores.

1.2 Motivação e Objetivos

Como a entrega de encomendas já é um serviço utilizado há bastante tempo, são diversas as empresas neste setor, sendo que algumas mantiveram o seu modo de funcionamento desde que foram criadas, sem se adaptarem à evolução das tecnologias. Foi esse o principal motivo pelo o qual o nosso grupo decidiu prosseguir com este tema.

Com esta aplicação pretende-se informatizar e tornar todo o processo de entregas mais simples, adaptado a cada utilizador. Pretendemos também melhorar a comunicação entre a empresa e os utilizadores, evitar perda de informação e erros de custos, entre outras consequências negativas para o negócio.

De forma sucinta, no sentido de automatizar todo este processo, a aplicação desenvolvida permitirá que o utilizador, após efetuar o registo, no qual terá de fornecer alguns dados relevantes à sua identificação, como o nome, *e-mail*, e uma *password*, estará perante uma

interface simples e intuitiva, que lhe permite consultar quais encomendas estão em curso e o histórico das suas encomendas. Além disso, também poderá adicionar uma determinada encomenda aos seus favoritos, que lhe permitirá receber uma notificação, via SMS, quando a entrega estiver concluída. Todo este processo será posteriormente detalhado através da justificação da viabilidade e utilidade do sistema.

Por fim, importa salientar que apesar de sabermos que este não é um conceito inovador, esta aplicação resolve um problema generalizado que afeta uma grande parte dos clientes de comércio *online*, pelo que é certo que tornará mais cómodo o problema de monitorização de uma encomenda.

1.3 Estrutura do Relatório

Apresentada toda esta idealização da aplicação TDelivery, a motivação e os objetivos que levaram a todo este enredo, as secções seguintes do presente relatório descrevem detalhadamente a forma como o grupo pretende garantir o sucesso da aplicação no mercado *web*.

De seguida, é apresentada uma cuidada análise sobre a viabilidade e crescimento futuro do sistema que se pretende desenvolver. Terminada esta parte, são identificados os recursos que o grupo considerou necessários para o desenvolvimento do produto final. É, então, apresentado o plano de trabalho que pretendemos seguir. Assim, de forma sucinta mas explícita, iremos abordar e sistematizar os tópicos fundamentais para uma posterior especificação de todo o projeto, abordando, então, as seguintes temáticas:

- Definição da identidade do sistema a desenvolver, abordando o conceito da aplicação e como todo o sistema irá funcionar e se irá relacionar com o utilizador;
- Justificação da viabilidade e utilidade do sistema. Nesta parte surge uma ampla explicação sobre os princípios fulcrais da aplicação e também levantamento de requisitos;
- Identificação dos recursos necessários para o desenvolvimento do sistema;
- Definição das medidas de sucesso necessárias para todo o projeto;
- Plano de desenvolvimento e elaboração de um diagrama de *Gantt* ilustrando o progresso das diferentes etapas do projeto com base nas datas de entrega estabelecidas para cada etapa deste projeto.

Além disso, procedeu-se a uma cuidada análise de requisitos, essencial para descrever o funcionamento do sistema a implementar e, posteriormente, apresenta-se um conjunto de diagramas UML que, juntamente com os requisitos, permitem especificar e documentar o sistema.

Por fim, apresenta-se a modelação da camada de dados, recorrendo, para isso, ao modelo conceptual e ao modelo lógico, assim como um dicionário de dados.

2 Fundamentação, Justificação e Utilidade do Sistema

Considerando o que foi explicado nas secções 1.1 e 1.2, a TDelivery tem como objetivo implementar um sistema que permite melhorar o processo de encomendas, recorrendo a ferramentas *online*.

Num mundo cada vez mais evoluído em termos tecnológicos, o ser humano procura tornar o dia a dia menos cansativo e o mais confortável possível. Nesta linha de pensamento surge, então, a aplicação TDelivery, um sistema de monitorização de encomendas, com intuito automatizar e tornar mais conveniente o problema de *tracking* de uma encomenda.

Desta forma, pretende-se que o sistema de *software* a desenvolver forneça um leque de serviços de forma a solucionar os problemas acima referidos, disponíveis através de um simples clique. Em suma, realça-se a escassez deste tipo de aplicações à disposição do público como uma das principais motivações para avançar com o projeto em questão.

2.1 Definição da Identidade do Sistema a Desenvolver

Nome	TDelivery
Categoria	Sistema de monitorização em tempo real de encomendas
<i>Slogan</i>	<i>The Leading Tracking Solution</i>
Idioma	Inglês
Público-alvo	Clientes de comércio <i>online</i>
Equipa de Desenvolvimento	Daniela Fernandes, José Gomes, Rui Fernandes, Tiago Rodrigues

Tabela 2.1: Identidade do sistema.



Figura 2.1: Logótipo do sistema.

2.2 Funcionamento do sistema

Para uma entrega confiável e dentro dos prazos prometidos, a TDelivery tem como objetivo tornar mais cómodo o acompanhamento de todas as entregas dos seus clientes. Posto isto, a pretende-se criar um *website*, em que se mostre todos os serviços e que permita ao cliente registar-se na plataforma.

Neste sentido, espera-se que os utilizadores sejam capazes de criar e eliminar a sua conta, alterar as credenciais e outros dados relativos à sua conta. Depois de registado e devidamente autenticado, o utilizador deve ser capaz acompanhar o estado de uma encomenda, visualizando o percurso da mesma no mapa. O cliente deve também, se assim entender, ser notificado, via SMS, de atualizações relativas ao estado da encomenda, caso a tenha adicionada à sua lista de favoritos. Futuramente, poderia também ser implementado um sistema de avaliação de cada uma das empresas responsáveis pelas entregas, o que realça o potencial de expansão e crescimento futuro associado a este sistema de *software*. Desta forma, finalizada a entrega, o cliente deverá avaliar o serviço prestado pela empresa transportadora responsável pela entrega em questão.

Por outro lado, no que diz respeito ao administrador da empresa, espera-se que este seja capaz de aceder a um *dashboard*, no qual lhe serão apresentadas um conjunto de métricas e estatísticas, como por exemplo o número de entregas realizadas num dado intervalo de tempo e o respetivo tempo de entrega, o que lhe permite ter uma visão mais geral do estado atual do negócio, permitindo-lhe ter uma melhor noção de quais as opções mais adequadas a tomar num determinado período de tempo.

Da mesma forma, considera-se imperativa uma interface intuitiva e *user-friendly*, tanto para os clientes como para o administrador da empresa.

3 Identificação dos Recursos Necessários

Para o desenvolvimento de uma aplicação, vários recursos precisaram de ser alocados até se obter uma peça final de *software*. Antes de mais, é essencial ter à disposição uma equipa de desenvolvedores, que serão os responsáveis por concretizar todas as etapas, desde a especificação até à implementação e *deployment* do sistema de *software*. Consequentemente, é também imprescindível a presença de um gestor de projeto, garantindo que todo o trabalho será devidamente concretizado, com qualidade, e cumprindo todos os prazos estabelecidos. Por fim, para garantir o sucesso da aplicação junto do público-alvo, foi também fundamental contactar indivíduos comuns, os futuros utilizadores do *software*, para perceber qual seria a melhor forma de estabelecer a interação entre o utilizador e a plataforma, para que desta forma os desenvolvedores possam proporcionar aos utilizadores finais uma experiência o mais personalizada possível.

Neste sentido, estes recursos serão utilizados para:

- Fazer uma pesquisa;
- Definir e fundamentar o projeto a desenvolver;
- Definir objetivos e plano de trabalho;
- Especificar o sistema através de diversos diagramas;
- Modelar e implementar a base de dados;
- Desenvolver o código do *software*.
- Documentar o sistema desenvolvido;
- Testar o *software* desenvolvido;

4 Modelo do Sistema a Implementar

Com o objetivo de especificar a arquitetura do *software* que será utilizada ao longo do desenvolvimento do projeto, foi elaborada uma maquete que demonstra o modo de funcionamento da aplicação.

Tendo por base os princípios da Arquitetura Multicamada, a aplicação será dividida em três camadas, nomeadamente, a camada de apresentação, a camada de negócio e, por fim, a camada de dados, sendo que cada uma das camadas é independente das outras. Note-se, também, que cada camada oferece um serviço à camada superior e solicita um serviço à camada inferior.

Desta forma, a camada de apresentação é a componente visual da aplicação responsável por mostrar os dados e interagir com o utilizador. A camada de negócio, por sua vez, implementa as regras de negócio e requisitos do próprio sistema. Por fim, a camada de dados é responsável pela persistência dos dados, permitindo uma correta manipulação de dados, tendo em consideração aspectos de controlo de concorrência e integridade dos dados.

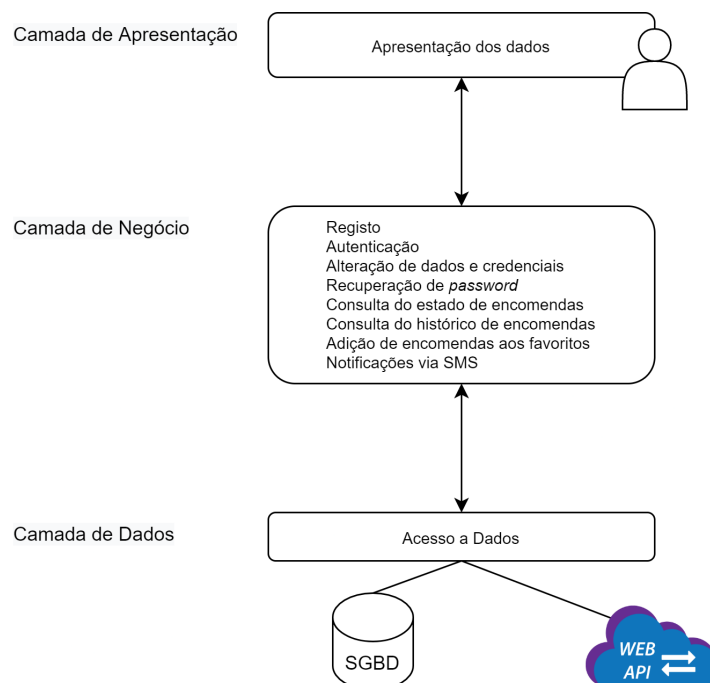


Figura 4.1: Maquete do sistema de *software* a desenvolver

Esta separação em camadas torna todo o sistema mais flexível, permitindo que as camadas possam ser alteradas de forma independente. As funcionalidades da camada de negócio podem, por sua vez, ser divididas em classes, que podem ser agrupadas em *packages*, reduzindo as dependências entre eles, o que permite a reutilização por diferentes partes da aplicação e, eventualmente, até por diferentes aplicações.

5 Definição de medidas de sucesso

Consideraríamos que o projeto seria um caso de sucesso, se, na fase final, o sistema de *software* cumprisse todos os requisitos iniciais e fosse aceite pelo cliente final. Outro parâmetro que consideramos importante, é a concretização do projeto dentro dos prazos estipulados e de acordo com o plano de trabalho desenvolvido. Se isto acontecesse, significaria que houve uma boa dinâmica entre todos os elementos do grupo.

Seria também fundamental, a utilização exclusiva dos recursos previstos, assim como poucas alterações ao longo do desenvolvimento do sistema.

Com estas medidas, todo o trabalho elaborado no futuro será com a intenção de as cumprir, tendo como objetivo um projeto bem sucedido.

6 Plano de desenvolvimento

Para se proceder ao correto e funcional desenvolvimento de um sistema que consiga dar resposta a todos os requisitos estabelecidos, é essencial ser percorrido um conjunto de passos fundamentais. Este desenvolvimento é um processo sequencial e planificado, que necessita que desde o principio seja definido um conjunto de diretivas de acordo com aquilo que se pretende desenvolver na totalidade. Além disso, para que todo o projeto se desenvolva de forma organizada, coerente e acima de tudo, procedimental, esta primeira fase consistiu em fundamentar todo o desenvolvimento do sistema de *software* idealizado. Fases posteriores irão consistir em analisar e especificar de forma completa todos os requisitos funcionais e não-funcionais do sistema de software a implementar. Depois, iremos desenvolver, validar, documentar e instalar o sistema de *software* idealizado.

Desta forma, o desenvolvimento assenta sobre os seguintes princípios:

- Delineamento de uma estratégia logística e flexível de desenvolvimento;
- Desenvolvimento iterativo e incremental ao longo do projeto;
- Desenvolvimento ágil, progressivo e melhorado do produto.

Por fim, de forma a tornar todo este processo de desenvolvimento organizado e estruturado, fez-se um planeamento do trabalho que será efetuado ao longo da realização do projeto, elaborando um diagrama de *Gantt* com o auxílio da ferramenta *Microsoft Project*, tendo em conta os prazos de entrega estabelecidos e a duração de cada uma das etapas. Através do diagrama é possível ver o tempo que se pretende dedicar a cada uma das várias tarefas para que o trabalho seja concluído no prazo estipulado.

Este diagrama está dividido em duas partes, **Especificação** e **Implementação**, omitindo-se a fase de fundamentação porque quando construído essa fase já tinha sido terminada.

Para a fase de especificação foi dada especial importância ao levantamento e análise de requisitos, assim como à criação do diagrama de classes e modelação da Base de Dados. Tomamos esta decisão uma vez que consideramos que estes modelos são essenciais para o sucesso do projeto.

Por fim, como se pode ver, decidimos também deixar alguns dias de margem antes do prazo limite para que se possa compensar algum imprevisto que ocorra durante o desenvolvimento do *software*.

4 Especificação	36 days	Mon 22/03/21	Mon 10/05/21	
Levantamento e Análise de Requisitos	3 days	Mon 22/03/21	Wed 24/03/21	Todos
Modelo de Domínio	5 days	Wed 24/03/21	Tue 30/03/21	José
Especificação e Diagramas de Use Case	5 days	Tue 30/03/21	Mon 05/04/21	Rui
Diagramas de Sequência	7 days	Mon 05/04/21	Tue 13/04/21	Daniela, José
Diagrama de Classes	4 days	Wed 07/04/21	Mon 12/04/21	Rui, Tiago
Validação dos Diagramas	2 days	Sat 10/04/21	Mon 12/04/21	Todos
4 Modelação da Base de Dados	12 days	Mon 12/04/21	Tue 27/04/21	
Modelo Conceptual	5 days	Mon 12/04/21	Fri 16/04/21	Daniela, Tiago
Modelo Lógico	3 days	Fri 16/04/21	Tue 20/04/21	José, Rui
Validação da Modelação da BD	2 days	Wed 21/04/21	Thu 22/04/21	Todos
Elaboração de um protótipo	10 days	Fri 23/04/21	Thu 06/05/21	Todos
Validação Final	2 days	Thu 06/05/21	Fri 07/05/21	Todos
Relatório	36 days	Mon 22/03/21	Mon 10/05/21	Todos
Apresentação	1 day	Sat 08/05/21	Sat 08/05/21	Todos
4 Implementação	19 days	Wed 12/05/21	Mon 07/06/21	
Implementação física da BD	3 days	Wed 12/05/21	Fri 14/05/21	José, Rui
Povoamento da BD	2 days	Fri 14/05/21	Mon 17/05/21	Daniela, Tiago
Implementação da camada de negócio	11 days	Tue 18/05/21	Tue 01/06/21	José, Rui
Implementação da camada de apresentação	11 days	Tue 18/05/21	Tue 01/06/21	Daniela, Tiago
Testes e correção de erros	3 days	Wed 02/06/21	Fri 04/06/21	Todos
Instalação do Sistema	2 days	Fri 04/06/21	Mon 07/06/21	Daniela
Relatório	19 days	Wed 12/05/21	Mon 07/06/21	Todos
Apresentação	1 day	Mon 07/06/21	Mon 07/06/21	Todos

Figura 6.1: Planeamento e distribuição de tarefas

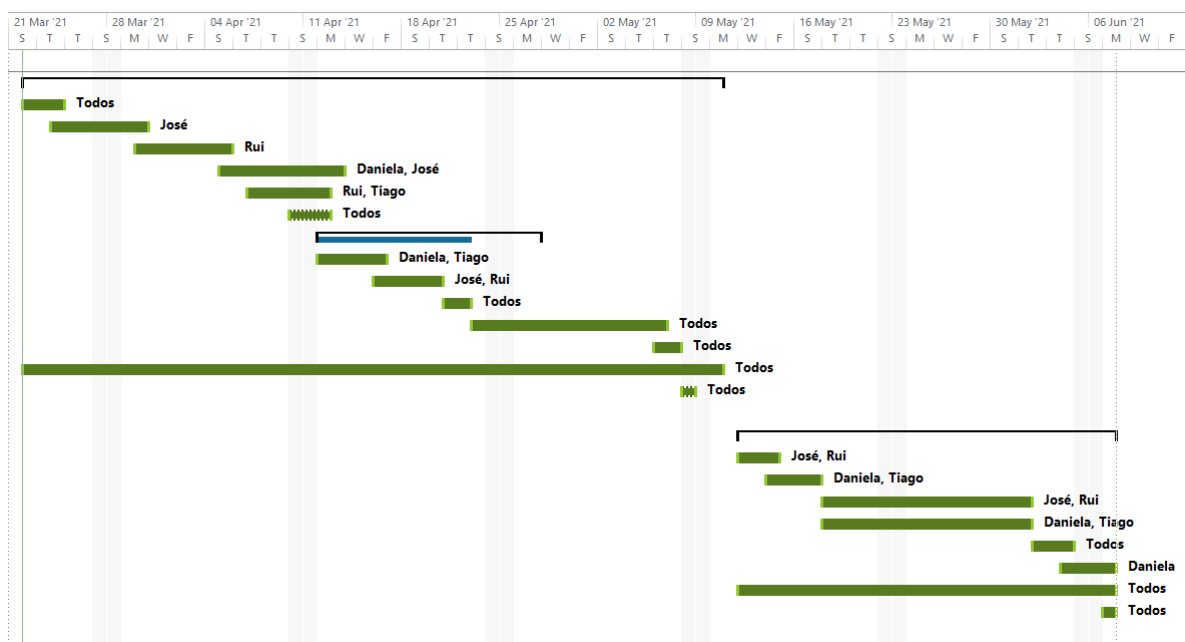


Figura 6.2: Diagrama de Gantt.

7 Levantamento e Análise de Requisitos

No sentido a identificar os requisitos necessários para a elaboração do projeto, decidimos reunir com o responsável da empresa TDelivery. Desta forma, pretendíamos perceber quais os objetivos deste relativamente ao *software* que iria ser implementado.

Assim, nesta secção, tendo em conta as funcionalidades pretendidas, serão definidos os requisitos do utilizador, e, para cada um destes, serão especificados os requisitos funcionais do sistema a si associados. Para além disto, serão também identificados os requisitos não funcionais do sistema.

Por se considerar um padrão intuitivo, optou-se pela priorização de requisitos de acordo a técnica *MoSCoW prioritization* [4], segundo a qual os requisitos são classificados da seguinte forma:

- **Must Have:** Requisitos que têm de ser considerados;
- **Should Have:** Requisitos que devem de ser considerados;
- **Could Have:** Requisitos desejáveis, mas que não são estritamente necessários;
- **Won't Have This Time:** Requisitos que se considera interessante ter, que poderão, ou não, ser considerados no futuro.

Assim, seguindo esta abordagem, chegamos aos requisitos apresentados nas secções 7.1 e 7.2.

7.1 Requisitos Funcionais

De seguida, para cada requisito de utilizador, apresentam-se os de sistema que lhe correspondem.

7.1.1 Registo [*Must Have*]

Definição de Requisitos do utilizador:

1. O cliente deve registar-se na plataforma de modo a posteriormente poder autenticar-se na mesma.

Especificação de Requisitos do sistema:

1. O sistema deve apresentar um campo de registo para o cliente, onde este possa introduzir nome, *username*, *email*, palavra-passe, e contacto;
2. O sistema deve ser capaz de registar um novo cliente na base de dados, com os dados por ele especificados;
3. Ao registar um cliente o sistema deve verificar se os dados introduzidos são válidos;
4. Se um cliente introduzir informação inválida no registo, o sistema deve notificá-lo do erro, e ignorar a informação inserida;
5. O sistema deve ser capaz de verificar se o cliente preencheu todos os campos;
6. O sistema não deve permitir que existam clientes com o mesmo *username* ou *email*;
7. O sistema deve assegurar que a palavra-passe tem pelo menos 8 caracteres.

7.1.2 Autenticação [*Must Have*]

Definição de Requisitos do utilizador:

1. O utilizador deve autenticar-se na plataforma para poder fazer uso da mesma.

Especificação de Requisitos do sistema:

1. O sistema deve apresentar um campo de autenticação para o utilizador onde permita introduzir *e-mail* e palavra-passe;

2. O sistema deve conseguir autenticar um utilizador, verificando se o *e-mail* e a palavra-passe introduzidos estão de acordo com a informação presente na base de dados;
3. Se um utilizador introduzir informação inválida (*e-mail* ou palavra-passe incorretos) na autenticação, o sistema deve notifica-lo do erro, não lhe permitindo prosseguir;
4. O sistema só deve permitir ao utilizador aceder às restantes funcionalidades do sistema, caso este esteja devidamente autenticado.

7.1.3 Término de Sessão [*Must Have*]

Definição de Requisitos do utilizador:

1. O utilizador deve ser capaz terminar sessão na plataforma.

Especificação de Requisitos do sistema:

1. O sistema deve permitir ao utilizador terminar a sua sessão na plataforma;
2. O sistema deve conseguir regressar à página de registo/autenticação.

7.1.4 Apresentar Funcionalidades da Aplicação [*Must Have*]

Definição de Requisitos do utilizador:

1. O utilizador deve ser capaz de visualizar um menu com as principais funcionalidades da aplicação.

Especificação de Requisitos do sistema:

1. O sistema deve exibir uma janela com as principais funcionalidades que possui, assim como uma breve descrição de cada uma.

7.1.5 Apresentar o Contacto dos diferentes Entregadores [*Should Have*]

Definição de Requisitos do utilizador:

1. O utilizador deve poder consultar a lista de empresas de entregas existentes no sistema.

Especificação de Requisitos do sistema:

1. O sistema deverá mostrar a lista das empresas de entregas registados no sistema;
2. A lista de empresas de entregas poderá ser ordenada por diferentes critérios, por exemplo, alfabeticamente, ou considerando o número de entregas realizadas;
3. Cada empresas deverá ter um contacto associado.

7.1.6 Consulta de Dados Pessoais [*Must Have*]

Definição de Requisitos do utilizador:

1. O cliente pode consultar os seus dados pessoais.

Especificação de Requisitos do sistema:

1. O sistema deve ser capaz de apresentar ao cliente os seus dados pessoais.

7.1.7 Edição de Dados Pessoais [*Must Have*]

Definição de Requisitos do utilizador:

1. O cliente pode editar os seus dados pessoais.

Especificação de Requisitos do sistema:

1. O sistema deve permitir ao cliente editar os seus dados pessoais;
2. Se o cliente não introduzir dados válidos na edição (contacto ou palavra-passe inválidos), o sistema deve notificá-lo do erro, e a atualização não é efetuada;
3. O sistema deve registar as alterações atualizando a informação presente na base de dados.

7.1.8 Remoção do Perfil [*Should Have*]

Definição de Requisitos do utilizador:

1. Um utilizador deve ser capaz de remover o seu perfil da aplicação.

Especificação de Requisitos do sistema:

1. O sistema deve permitir que um utilizador, se assim desejar, possa remover o seu perfil;
2. O sistema deverá apresentar um ecrã ao utilizador onde este deve confirmar a intenção de remover o seu perfil;
3. O sistema deverá remover os dados do utilizador previamente armazenados;
4. O sistema deve, aquando da remoção de um utilizador, remover também a informação relativa às suas encomendas.

7.1.9 Consulta de uma Encomenda [*Must Have*]

Definição de Requisitos do utilizador:

1. O utilizador deve ser capaz de verificar o estado de uma encomenda.

Especificação de Requisitos do sistema:

1. O sistema deve permitir que um utilizador consulte o estado de uma encomenda;
2. O sistema deve notificar um utilizador, via SMS, de eventuais alterações no estado de uma encomenda, caso esta tenha sido adicionada aos favoritos;
3. O sistema deve apresentar ao utilizador, num mapa, a localização atual da encomenda;
4. O sistema de apresentar ao utilizador informações sobre a encomenda, nomeadamente a data em que a encomenda foi feita, qual a empresa de entregas responsável pela entrega, assim como o respetivo contacto.

7.1.10 Categorizar Encomendas [*Should Have*]

Definição de Requisitos do utilizador:

1. O utilizador deve ser capaz de atribuir uma categoria a uma determinada encomenda.

Especificação de Requisitos do sistema:

1. O sistema deve permitir que um utilizador atribua uma categoria a uma determinada encomenda;
2. O sistema deve permitir que o utilizador altere a categoria em que uma determinada encomenda se insere;
3. O sistema deve ser capaz de filtrar e listar todas as encomendas que se inserem numa determinada categoria.

7.1.11 Listagem de Encomendas [*Must Have*]

Definição de Requisitos do utilizador:

1. O utilizador deve poder consultar quais as encomendas que acompanha, assim como aquelas que acompanhou no passado.

Especificação de Requisitos do sistema:

1. O sistema deve apresentar ao utilizador as encomendas (se existentes) que este acompanha, assim como aquelas que acompanhou no passado;
2. O sistema deve restringir o acesso ao histórico, sendo que apenas o próprio utilizador pode ter acesso ao mesmo.
3. O sistema deve exibir a listagem de produtos de uma encomenda pertencente ao histórico, caso o utilizador assim deseje.

7.1.12 Consulta de Favoritos [*Should Have*]

Definição de Requisitos do utilizador:

1. O utilizador deve ter a possibilidade de consultar o estado das suas encomendas favoritas.

Especificação de Requisitos do sistema:

1. O sistema deve permitir que um utilizador marque uma encomenda como favorito;
2. O sistema deve restringir o acesso à listagem de favoritos, sendo que apenas o utilizador pode ter acesso à sua lista.

7.1.13 Editar Lista de Favoritos [*Should Have*]

Definição de Requisitos do utilizador:

1. O utilizador deve ser capaz de adicionar ou remover uma encomenda dos favoritos, permitindo um acesso fácil e rápido ao estado da encomenda em questão.

Especificação de Requisitos do sistema:

1. O sistema deve permitir que um utilizador adicione ou remova uma encomenda dos favoritos;
2. O sistema deve notificar, via SMS, o utilizador futuras atualizações relativas ao estado da encomenda em questão;
3. Após a remoção de uma encomenda dos favoritos, o sistema não deve notificar o utilizador sobre futuras relativas ao estado da encomenda em questão.

7.1.14 Gestão de Utilizadores [*Should Have*]

Definição de Requisitos do utilizador:

1. O administrador do sistema deverá poder ver e gerir a lista de utilizadores do sistema.

Especificação de Requisitos do sistema:

1. O sistema deverá disponibilizar a lista de utilizadores, juntamente com detalhes dos mesmos, não sendo, no entanto, capaz de ver informação sobre as suas encomendas;
2. O sistema deve permitir a remoção de um determinado utilizador;

7.1.15 Exportação de dados [*Could Have*]

Definição de Requisitos do utilizador:

1. O utilizador deve ser capaz de exportar informação relativa ao seu histórico de encomendas e à sua lista de favoritos.

Especificação de Requisitos do sistema:

1. O sistema deve permitir a exportação da informação relativa a um cliente e respetivas encomendas, por exemplo, em formato JSON.

7.2 Requisitos Não Funcionais

Em adição aos requisitos definidos acima, foram também identificados os seguintes requisitos não funcionais do sistema.

7.2.1 Requisitos de Produto

- O sistema deve estar disponível 24 horas por dia;
- O sistema deve ser produzido de forma a funcionar corretamente em todos os *browsers*;
- O sistema deve suportar o registo de 1500 utilizadores no espaço de 1 ano;
- O sistema deve ser de fácil uso, com uma interface o mais simples e intuitiva possível;
- O sistema deve ser de fácil utilização por utilizadores com pouca literacia digital;
- O tempo de resposta do sistema deve ser o mais curto possível de forma a não influenciar de forma negativa a experiência do utilizador;
- Em caso de *downtime* por parte do *website*, este não poderá ser superior a 30 minutos por dia, em intervalos de *downtime* nunca maiores que 3 minutos, espaçados por 10 minutos;
- O tempo de resposta a eventos do sistema deve ser inferior a 20 segundos;
- Toda a interação com o sistema deverá requerer a autenticação prévia do utilizador.

7.2.2 Requisitos Organizacionais

- A aplicação será desenvolvida para *web*;
- A linguagem de programação utilizada será C# juntamente com a *framework* Blazor;
- O Sistema de Gestão de Bases de Dados a usar será o SQL Server.

7.2.3 Requisitos Externos

- O sistema deve recorrer a uma API externa de mapas para representar a localização das encomendas do utilizador;
- O sistema deve recorrer a uma API externa de forma a ser capaz de notificar o utilizador, via SMS, de atualizações no estado da sua encomenda.
- O sistema deve manter a privacidade dos utilizadores e dos seus dados, sendo estes visíveis apenas para cada utilizador. Desta forma, cada utilizador da plataforma só conseguirá aceder aos registos da BD associados à sua conta na plataforma. O próprio administrador da plataforma apenas poderá aceder à gestão de contas de plataforma e não à informação relativa a cada um dos utilizadores;
- O administrador encontra-se previamente registados na plataforma.

8 Modelação

Após a recolha e descrição detalhada dos requisitos, procedeu-se à modelação do sistema a ser implementado. A modelação pode ser vista como uma abstração do sistema que será construído – na medida em que abstrai muito do detalhe de implementação – e não como uma representação alternativa do mesmo.

Nesta secção será então abordada a modelação realizada do sistema. Para tal foi utilizada a linguagem UML (*Unified Modeling Language*), uma linguagem-padrão para a elaboração da estrutura de projetos de *software*. Para uma especificação fundamentada do sistema de *software*, optou-se por elaborar, como é possível constatar ao longo desta secção, o Modelo de Domínio, Diagrama de *Use Cases* e respetivas especificações, assim como Diagramas de Sequência e de Atividade. Desta forma, os modelos de sistema elaborados têm como intuito descrever e demonstrar as relações entre os utilizadores e o sistema (plataforma *web TDelivery*).

Importa salientar que estes modelos criados em UML ajudarão futuramente na tarefa de implementação e mais tarde ainda, quando for necessário documentar a estrutura do sistema e o modo como este opera.

8.1 Modelo de Domínio

O modelo de domínio permite representar o problema de uma forma mais prática, destacando as entidades principais e os relacionamentos entre elas. Efetivamente, o modelo de domínio ajuda a começar a idealização correta do sistema a desenvolver, tratando-se do ponto de partida ideal para a modelação. Posto isto, em seguida tem-se a explicação do modelo.

A evolução deste modelo foi algo que esteve sempre em aberto e o resultado final não é um diagrama que se possa dizer que esteja bem ou mal, mas após a especificação e adição de algumas entidades, e a remodelação de algumas relações entre entidades, chegamos a um resultado final que acreditamos que identifique na íntegra o domínio do sistema.

Tendo por base os requisitos previamente recolhidos e analisados, foi desenvolvido o seguinte modelo de domínio que visa recolher e representar de maneira geral os conceitos, e algumas relações entre entidades que estarão incluídas no sistema a desenvolver.

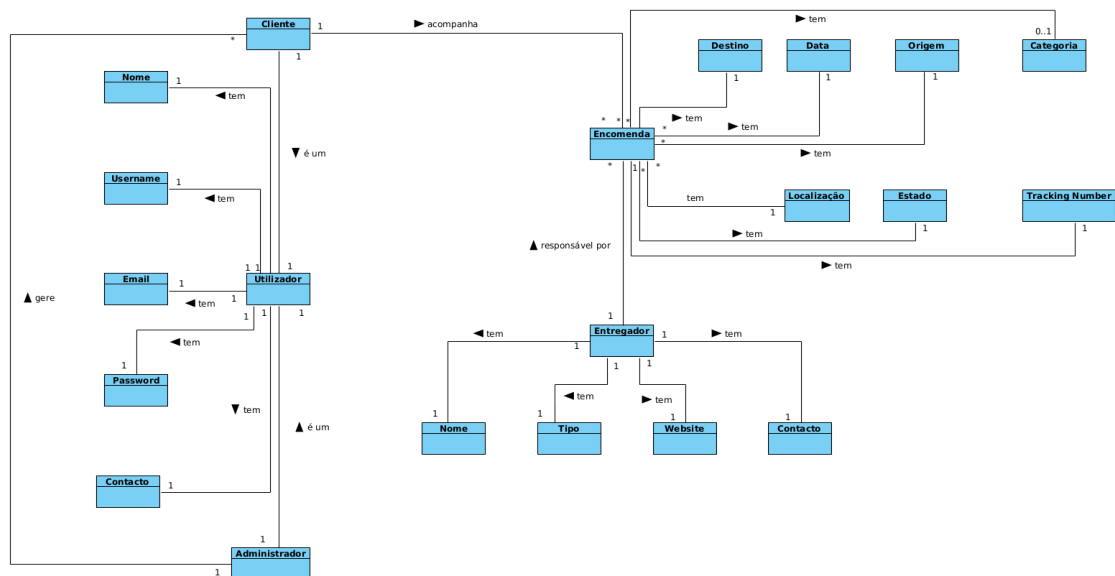


Figura 8.1: Modelo de Domínio

8.1.1 Atores

Tendo em conta o diagrama da Figura 8.1, os atores identificados foram os seguintes:

- **Administrador** – Utilizador que, quando autenticado, trata da gestão da aplicação e de utilizadores.
- **Utilizador não registado** – Utilizador que, não estando registado na aplicação, apenas tem acesso à página de registo/autenticação;
- **Utilizador registado** – Utilizador que, registado e autenticado na aplicação, solicita pedidos ao sistema, podendo consultar informações sobre as suas encomendas, assim como o seu perfil.

8.2 Diagramas de *Use Case*

Este diagrama representa as interações possíveis entre o sistema e os seus utilizadores. Logicamente, grande parte dos requisitos funcionais são transpostos neste diagrama, pelo que assume extrema importância na modelação do nosso sistema.

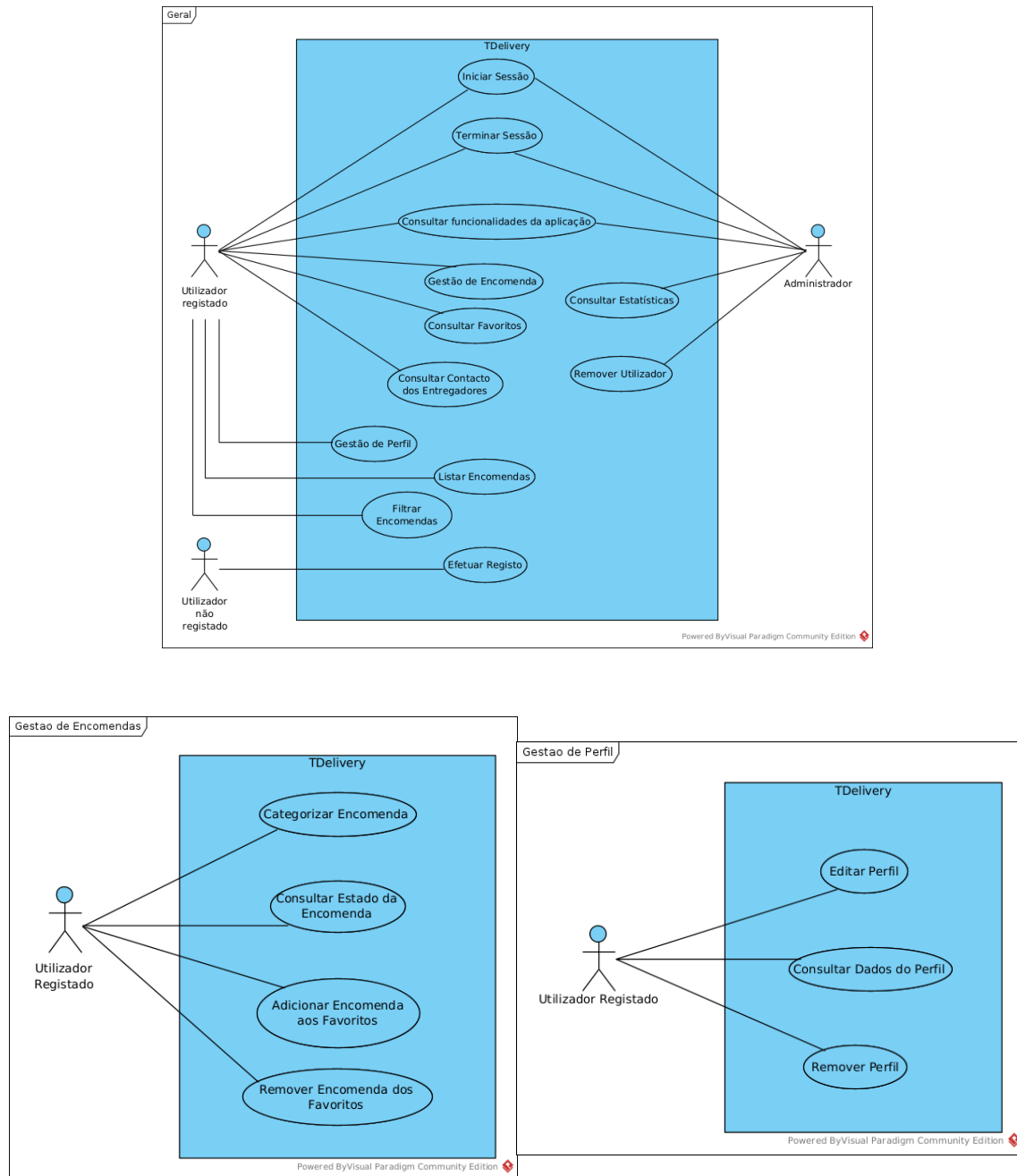


Figura 8.2: Diagrama de *Use Case*

8.3 Especificação de *Use Cases*

A secção anterior mostrou a aglomeração de todos os *use cases* que cada ator pode requerer ao sistema. No entanto, é também importante saber detalhadamente como cada uma das ações progride, e perante esta necessidade, surgiu a especificação textual de cada um dos *use cases*.

As especificações de *use cases* ajudam a perceber qual o processo por detrás de um *use case* entre o sistema e o ator. Neste ponto de vista, torna-se, então, importante descrever o processo, visto poder esclarecer o processo de desenvolvimento de *software*.

De seguida apresenta-se a especificação dos *use cases* resultantes do levantamento e análise de requisitos efetuado anteriormente.

8.3.1 Efetuar Registo

Use Case	Efetuar Registo
<i>Descrição:</i>	De modo a registar um utilizador, é necessário fornecer o <i>username</i> desejado, o seu <i>email</i> e a <i>password</i> . Caso o <i>username</i> inserido exista na base de dados, o utilizador é obrigado a escolher outro. Caso contrário prossegue-se com o registo e é pedido ao utilizador para fornecer determinadas informações como por exemplo o seu nome e número de telemóvel. Tendo todos os campos preenchidos, o registo fica então concluído e o utilizador é informado do mesmo.
<i>Ator:</i>	Utilizador não registado
<i>Pré-condição:</i>	Utilizador não registado
<i>Pós-condição:</i>	Utilizador registado

Fluxo Normal:

1. O utilizador indica que se pretende registar no sistema;
2. O sistema solicita o *username*, *email* e *password*;
3. O utilizador introduz o *username*, *email* e *password*;
4. O utilizador submete o registo;
5. O sistema verifica os dados introduzidos;
6. O sistema confirma se os dados introduzidos são válidos;
7. O sistema regista o utilizador na base de dados;
8. O sistema notifica do utilizador do sucesso ao efetuar o registo.

Fluxo de Exceção 1 [Utilizador cancela registo] (Passo 3)

- 3.1 O utilizador cancela o registo.

Fluxo de Exceção 2 [Registo inválido] (Passo 6)

- 6.1 O sistema notifica o utilizador de que o registo é inválido.
-

8.3.2 Iniciar sessão

Use Case	Iniciar sessão
<i>Descrição:</i>	Para iniciar sessão é necessário que o utilizador/administrador forneça as suas credenciais, as quais serão verificadas logo de seguida e, caso estejam incorretas, este é informado do mesmo, sendo obrigado a reintroduzi-las. Se nada estiver errado, então o utilizador/administrador é autenticado.
<i>Ator:</i>	Utilizador registado Administrador
<i>Pré-condição:</i>	Utilizador não autenticado
<i>Pós-condição:</i>	Utilizador autenticado

Fluxo Normal:

1. O utilizador/administrador indica que pretende iniciar sessão;
2. O sistema solicita os dados de acesso à aplicação (*username* e *password*);
3. O utilizador/administrador introduz o *username* e *password*;
4. O sistema verifica os dados;
5. O sistema autentica o utilizador/administrador.

Fluxo de Exceção 1 [Credenciais inválidas] (Passo 4)

- 4.1 O sistema informa o utilizador/administrador de que as credenciais introduzidas não são válidas.
-

8.3.3 Terminar sessão

Use Case	Terminar sessão
<i>Descrição:</i>	Para fazer <i>logout</i> , o utilizador/administrador apenas tem de indicar que pretende terminar a sessão e o sistema é responsável por processar o seu pedido e, posteriormente, desautenticá-lo.
<i>Ator:</i>	Utilizador registado Administrador
<i>Pré-condição:</i>	Utilizador autenticado
<i>Pós-condição:</i>	Utilizador não autenticado

Fluxo Normal:

1. O utilizador/administrador indica que pretende terminar a sessão na aplicação;
 2. O sistema processa o pedido;
 3. O utilizador/administrador é desautenticado.
-

8.3.4 Consultar funcionalidades da aplicação

Use Case	Consultar funcionalidades da aplicação
<i>Descrição:</i>	O sistema possui um separador destinado a ajudar os utilizadores a utilizar a aplicação e a elucidá-los acerca das várias funcionalidades da mesma.
<i>Ator:</i>	Utilizador autenticado Administrador
<i>Pré-condição:</i>	Utilizador autenticado
<i>Pós-condição:</i>	É apresentado um menu explicativo sobre as funcionalidades da aplicação
<i>Fluxo Normal:</i>	
1. O sistema apresenta ao utilizador/administrador um menu explicativo sobre cada funcionalidade da aplicação.	

8.3.5 Consultar Estatísticas

Use Case	Consultar Estatísticas
<i>Descrição:</i>	O sistema possui um separador, apenas acessível ao administrador, destinado a apresentar um conjunto de métricas e estatísticas, que lhe permite ter uma visão mais geral do estado atual do negócio.
<i>Ator:</i>	Administrador
<i>Pré-condição:</i>	Administrador autenticado
<i>Pós-condição:</i>	É apresentada uma <i>dashboard</i> com alguns dados estatísticos
<i>Fluxo Normal:</i>	
1. O sistema apresenta ao administrador uma <i>dashboard</i> com estatísticas relativas aos utilizadores da aplicação e respetivas encomendas.	

Fluxo de Exceção 1 [Dados indisponíveis] (Passo 1)

- 1.1 O sistema informa que alguns dos dados não se encontram disponíveis.
-

8.3.6 Consultar Encomenda

Use Case	Consultar Encomenda
<i>Descrição:</i>	O utilizador deve ser capaz de acompanhar o estado de uma encomenda, obtendo informação sobre a localização atual assim como a empresa responsável pela entrega.
<i>Ator:</i>	Utilizador registado
<i>Pré-condição:</i>	Utilizador autenticado
<i>Pós-condição:</i>	Utilizador consulta o estado da encomenda
<i>Fluxo Normal:</i>	
<ol style="list-style-type: none">1. O utilizador indica que pretende consultar o estado de uma encomenda;2. O sistema apresenta a lista de encomendas do utilizador;3. O utilizador seleciona a encomenda cujo estado pretende consultar;4. O sistema apresenta uma página com informação relativa ao estado da encomenda.	
<i>Fluxo Alternativo 1 [Utilizador sem histórico de encomendas] (Passo 2)</i>	
<ol style="list-style-type: none">2.1 O sistema solicita o <i>tracking number</i> da encomenda;2.2 O sistema apresenta uma página com informação relativa ao estado da encomenda.	
<i>Fluxo de Exceção 1 [Tracking number inválido] (Passo 2.1)</i>	
<ol style="list-style-type: none">2.1.1 O sistema informa o utilizador de que o <i>tracking number</i> introduzido não é válido	
<i>Fluxo de Exceção 2 [Dados indisponíveis] (Passos 4 e 2.2):</i>	
<ol style="list-style-type: none">4.1 O sistema informa que alguns dos dados não se encontram disponíveis.	

8.3.7 Listagem de Encomendas

Use Case	Consultar histórico de Encomendas
<i>Descrição:</i>	Para que o utilizador possa aceder a todas as encomendas associadas ao seu perfil, este só tem de indicar que o pretende efetuar e o sistema apresenta-lhe depois um conjunto de informações sobre as mesmas.
<i>Ator:</i>	Utilizador registado
<i>Pré-condição</i>	Utilizador autenticado
<i>Pós-condição</i>	O histórico de encomendas é apresentado
<i>Fluxo Normal:</i>	
<ol style="list-style-type: none">1. O utilizador indica que pretende consultar o histórico de encomendas;2. O sistema processa o pedido;3. O sistema apresenta as encomendas relativas ao utilizador, assim como informações sobre as mesmas.	

8.3.8 Consultar favoritos

Use Case	Consultar favoritos
<i>Descrição:</i>	Para consultar a lista de encomendas favoritas, o utilizador apenas tem de indicar que pretende consultá-las e o sistema fornece a respetiva lista caso esta exista. Se o utilizador não possuir encomendas na lista de favoritos, a consulta fica sem efeito.
<i>Ator:</i>	Utilizador registado
<i>Pré-condição:</i>	Utilizador autenticado
<i>Pós-condição:</i>	A lista de favoritos é apresentada

Fluxo Normal:

1. O utilizador indica que pretende consultar a lista de favoritos;
2. O sistema processa o pedido;
3. O sistema apresenta a lista de favoritos.

Fluxo de Exceção 1 [O utilizador não tem favoritos] (Passo 3)

- 3.1 O sistema notifica o utilizador de que este não tem encomendas lista de favoritos.
-

8.3.9 Adicionar encomenda aos favoritos

Use Case	Adicionar encomenda aos favoritos
<i>Descrição:</i>	Caso o utilizador pretenda adicionar uma encomenda à lista de favoritos, este apenas necessita de indicar qual a encomenda em questão e, posteriormente, o sistema adiciona-a à lista, caso esta não faça parte da mesma. Caso contrário, o sistema notifica o usuário de que a encomenda já consta da lista.
<i>Ator:</i>	Utilizador registado
<i>Pré-condição</i>	Utilizador autenticado
<i>Pós-condição</i>	A encomenda é adicionada aos favoritos

Fluxo Normal:

1. O utilizador indica que pretende adicionar uma encomenda aos favoritos;
2. O sistema adiciona a encomenda aos favoritos;
3. O sistema notifica o utilizador de que a encomenda foi adicionada aos favoritos.

Fluxo de exceção 1 [Encomenda já está nos favoritos] (passo 2)

- 2.1 O sistema indica que a encomenda já está na lista de favoritos.
-

8.3.10 Remover encomenda dos favoritos

Use Case	Remover encomenda dos favoritos
<i>Descrição:</i>	Caso o utilizador pretenda remover uma encomenda da sua lista de favoritos, este apenas precisa de o indicar e o sistema fica encarregue de executar o seu pedido.
<i>Ator:</i>	Utilizador registado
<i>Pré-condição</i>	Utilizador autenticado && encomenda faz parte da lista de favoritos
<i>Pós-condição</i>	Encomenda removida da lista de favoritos
<i>Fluxo Normal:</i>	
<ol style="list-style-type: none">1. O utilizador indica que pretende remover a encomenda aos favoritos;2. O sistema remove a encomenda dos favoritos;3. O sistema notifica o utilizador de que a encomenda foi removida dos favoritos.	

8.3.11 Categorizar Encomenda

Use Case	Categorizar Encomenda
<i>Descrição:</i>	No sentido de catalogar as suas encomendas, o utilizador deve ser capaz de atribuir uma categoria a uma encomenda.
<i>Ator:</i>	Utilizador registado
<i>Pré-condição:</i>	Utilizador autenticado
<i>Pós-condição:</i>	A categoria é atribuída à encomenda

Fluxo Normal:

1. O utilizador indica que pretende categorizar a encomenda;
2. O sistema solicita qual a categoria a atribuir à encomenda;
3. O utilizador seleciona a categoria a atribuir;
4. O sistema atribui a categoria à encomenda.

Fluxo Alternativo 1 [Encomenda já categorizada] (Passo 1)

- 1.1 O sistema indica que a encomenda já tem uma categoria associada;
- 1.2 O sistema solicita o utilizador sobre o facto de este querer substituir a categoria atual;
- 1.3 O sistema substitui a categoria da encomenda pela categoria introduzida pelo utilizador.

Fluxo de Exceção 1 [Utilizador cancela a operação] (Passo 1.2)

- 1.2.1 O utilizador cancela a operação.
-

8.3.12 Filtrar Encomendas

Use Case	Filtrar Encomendas
<i>Descrição:</i>	De modo a ter uma melhor perceção sobre as suas encomendas, o utilizador deve ser capaz de visualizar quais as encomendas que se inserem numa determinada categoria.
<i>Ator:</i>	Utilizador registado
<i>Pré-condição:</i>	Utilizador autenticado
<i>Pós-condição:</i>	É apresentada ao utilizador a lista das encomendas que pertencem à categoria em questão

Fluxo Normal:

1. O utilizador seleciona a categoria pela qual pretende filtrar as encomendas;
2. O sistema apresenta ao utilizador a lista das encomendas que pertencem à categoria em questão.

Fluxo de Exceção 1 [Não existem encomendas na categoria] (Passo 2.2)

1. O sistema informa o utilizador de que nenhuma das suas encomendas pertencem à categoria introduzida.
-

8.3.13 Consultar Dados do Perfil

Use Case	Consultar Dados do Perfil
<i>Descrição:</i>	O utilizador da aplicação deve ser capaz de consultar os dados relativos ao seu perfil na aplicação.
<i>Ator:</i>	Utilizador registado
<i>Pré-condição:</i>	Utilizador autenticado
<i>Pós-condição:</i>	Utilizador consulta os dados do seu perfil

Fluxo Normal:

1. O utilizador indica que pretende consultar os dados relativos ao seu perfil na aplicação;
 2. O sistema processa o pedido;
 3. O sistema apresenta as informações de perfil do utilizador.
-

8.3.14 Editar perfil

Use Case	Editar perfil
<i>Descrição:</i>	De modo a editar o seu perfil, o sistema tem que, primeiramente, processar o pedido do utilizador e depois apresentar as informações do mesmo. Assim, o usuário fica possibilitado de alterar qualquer campo desejado. Após submetidas as novas informações, o sistema processa-as e guarda-as na base de dados. Note-se que caso o usuário queira pode também cancelar a edição do seu perfil, mantendo as informações inalteradas.
<i>Ator:</i>	Utilizador registado
<i>Pré-condição:</i>	Utilizador autenticado
<i>Pós-condição:</i>	Perfil do utilizador editado
<i>Fluxo Normal:</i>	<ol style="list-style-type: none">1. O utilizador indica que pretende editar o seu perfil;2. O sistema processa o pedido;3. O sistema apresenta as informações de perfil do utilizador;4. O utilizador altera os campos pretendidos;5. O utilizador submete as mudanças;6. O sistema processa as mudanças pretendidas.
<i>Fluxo de Exceção 1 [Utilizador cancela a edição do perfil] (Passo 5)</i>	
5.1 O utilizador cancela a edição do perfil.	

8.3.15 Remover Perfil

Use Case:	Remover Perfil
<i>Descrição:</i>	O utilizador deve ser capaz de remover a sua conta da aplicação.
<i>Pré-condição:</i>	Utilizador autenticado

Pós-condição: Utilizador removido

Fluxo Normal:

1. O utilizador indica que pretende remover o seu perfil na aplicação;
2. O sistema solicita as credenciais do utilizador;
3. O utilizador insere as credenciais;
4. O sistema remove a conta do utilizador.

Fluxo de Exceção 1 [Utilizador cancela a operação] (Passo 1)

- 1.1 Utilizador cancela a operação.

Fluxo de Exceção 2 [Credenciais Inválidas] (Passo 3)

- 3.1 O sistema informa o utilizador de que as credenciais introduzidas não são válidas.
-

8.3.16 Consultar Contacto dos Entregadores

Use Case:	Consultar Contacto dos Entregadores
------------------	--

<i>Descrição:</i>	O sistema possui um separador destinado a listar todas as empresas de entregas associadas às encomendas e respetivo contacto
-------------------	--

<i>Ator:</i>	Utilizador registado
--------------	----------------------

<i>Pré-Condição:</i>	Utilizador autenticado
----------------------	------------------------

<i>Pós-condição:</i>	Utilizador consulta os contactos das empresas de entregas
----------------------	---

Fluxo Normal

1. O sistema apresenta ao utilizador uma lista com todas as empresas de entregas associadas às encomendas registadas na aplicação, assim como o respetivo contacto.
-

Fluxo de Exceção 1 [Informação indisponíveis] (Passo 1)

- 1.1 O sistema informa que parte da informação não se encontra disponível.
-

8.3.17 Remover Utilizador

Use Case	Remover utilizador
<i>Descrição:</i>	O administrador deve ser capaz de remover utilizadores da aplicação quando considerar pertinente.
<i>Pré-condição:</i>	Administrador autenticado
<i>Pós-condição:</i>	Utilizador removido
<i>Fluxo Normal:</i>	
<ol style="list-style-type: none">1. O administrador indica que se pretende remover um utilizador da aplicação;2. O sistema solicita o <i>username</i> do utilizador a remover;3. O administrador introduz o <i>username</i> do utilizador a remover;4. O sistema remove o utilizador.	
<i>Fluxo de Exceção 1: [Administrador cancela a operação] (Passo 1)</i>	
<ol style="list-style-type: none">1.1 O administrador cancela a operação.	
<i>Fluxo de exceção 2 [Utilizador não existente] (Passo 3)</i>	
<ol style="list-style-type: none">3.1 O sistema informa que o utilizador não existe	

9 Diagramas de Sequência

Este tipo de diagrama representa a sequência dos acontecimentos que foram descritos na especificação dos *Use Cases*, abordado no capítulo anterior.

Para a elaboração dos diagramas consideramos a existência de dois subsistemas que interagem com a plataforma *web*:

- Subsistema Utilizadores;
- Subsistema Encomendas.

Todos os *use cases* definidos e especificados anteriormente foram também especificados através de diagramas de sequência, que se apresentam de seguida.

9.1 Efetuar Registo

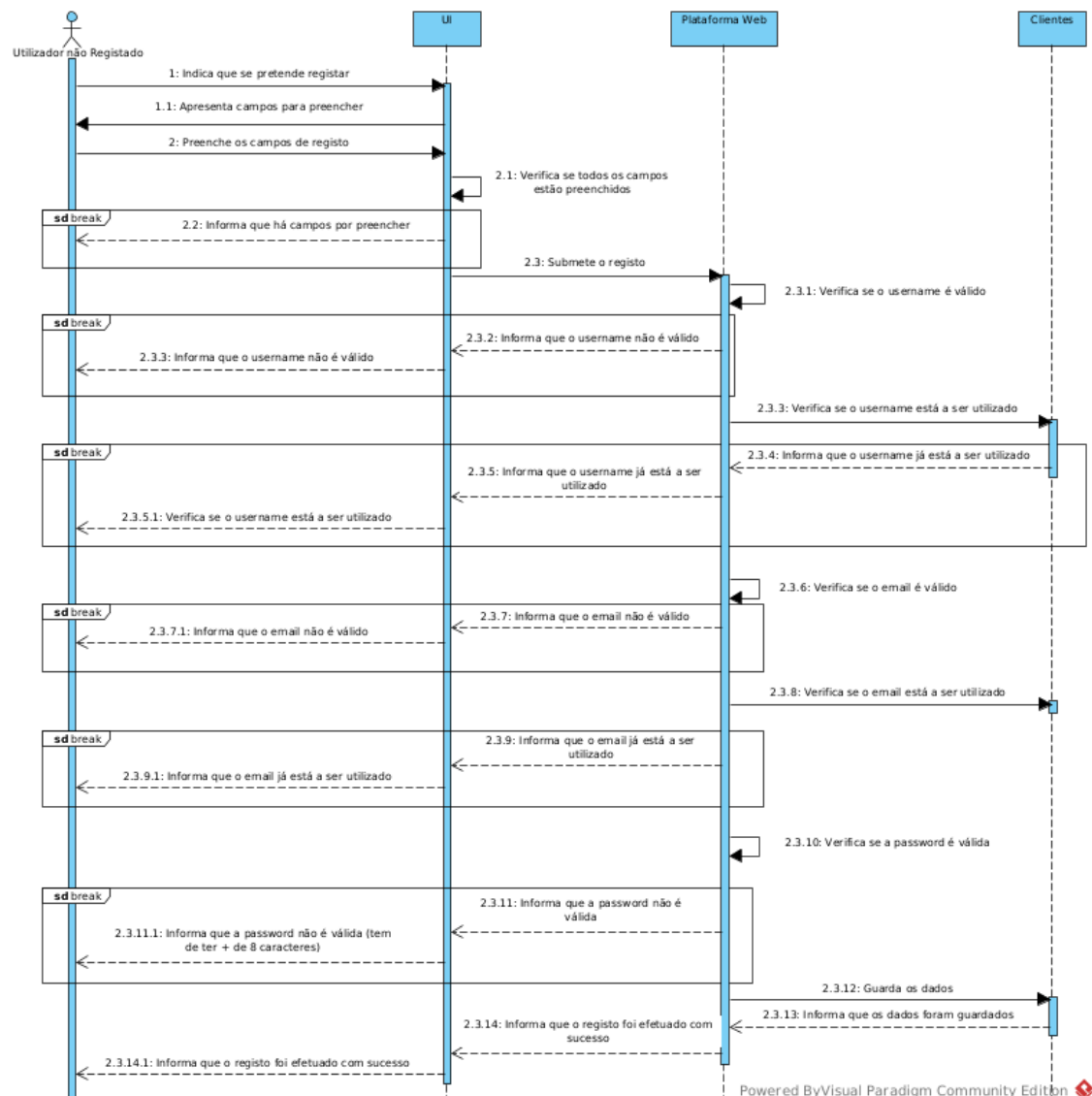


Figura 9.1: Diagrama de Sequência correspondente ao *use case* Efetuar Registo

9.2 Iniciar Sessão

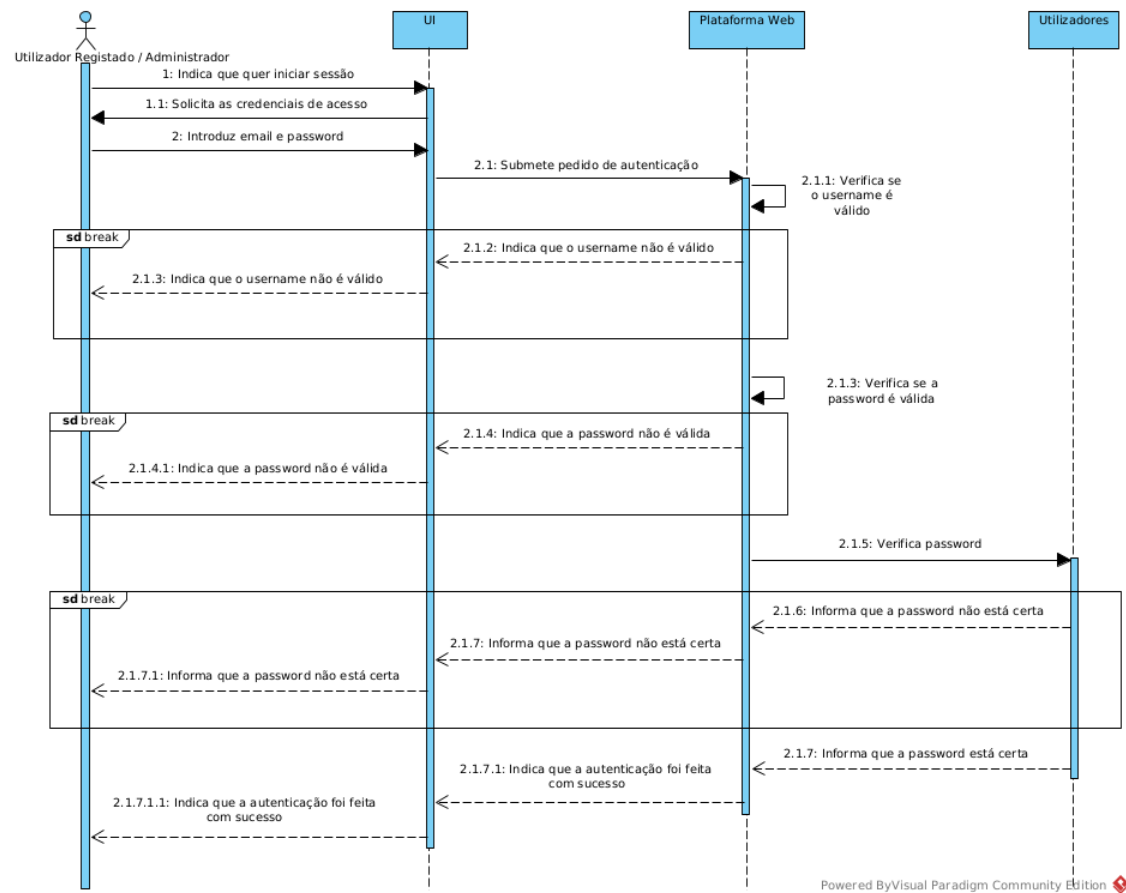


Figura 9.2: Diagrama de Sequência correspondente ao *use case* Iniciar Sessão

9.3 Terminar Sessão

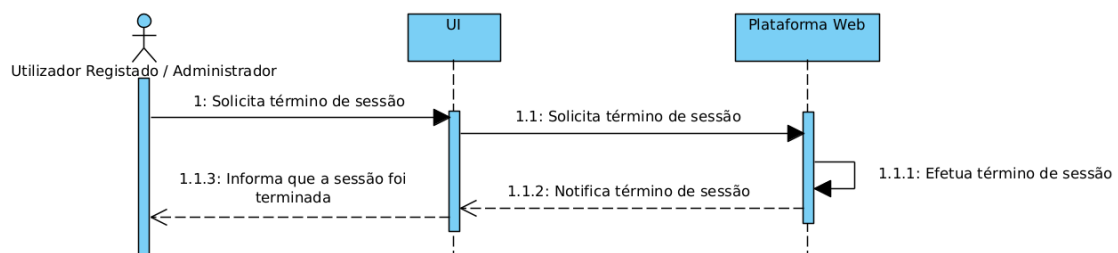


Figura 9.3: Diagrama de Sequência correspondente ao *use case* Terminar Sessão

9.4 Consultar Funcionalidades da Aplicação

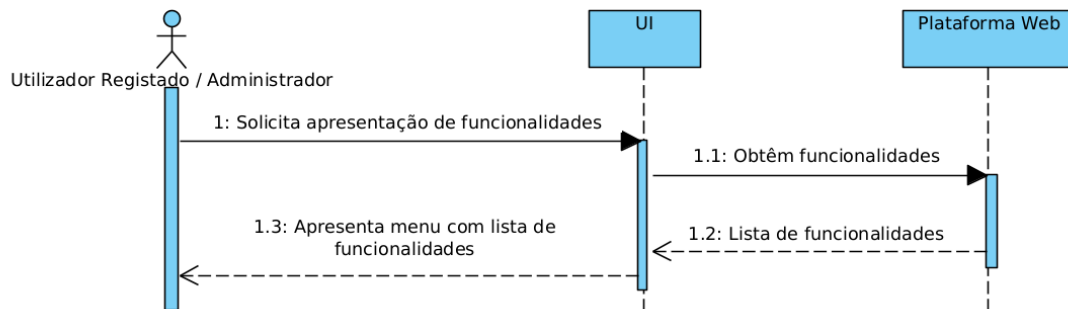


Figura 9.4: Diagrama de Sequência correspondente ao *use case* Consultar Funcionalidades da Aplicação

9.5 Consultar Estatísticas

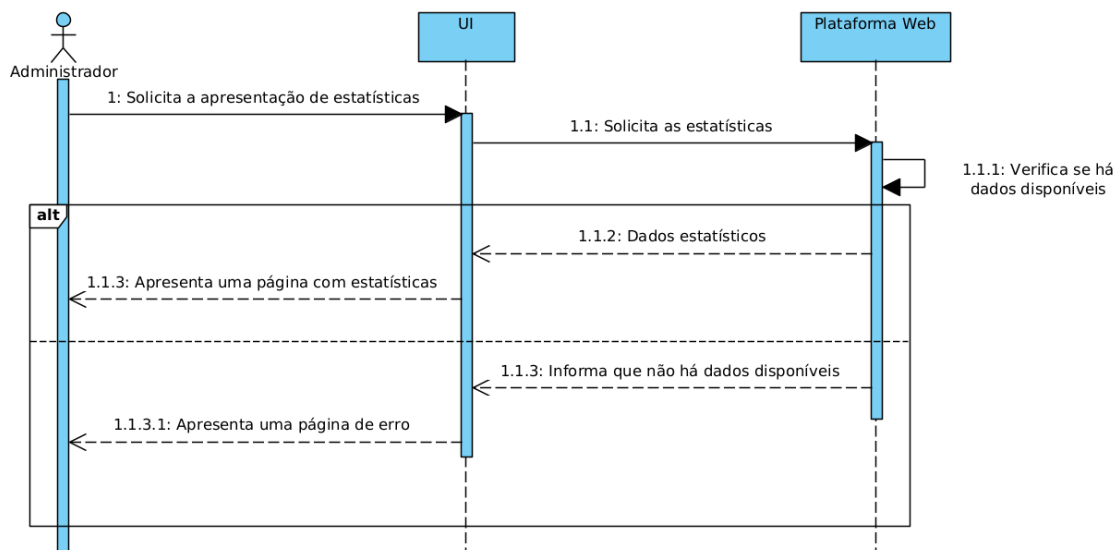


Figura 9.5: Diagrama de Sequência correspondente ao *use case* Consultar Estatísticas

9.6 Consultar Encomenda

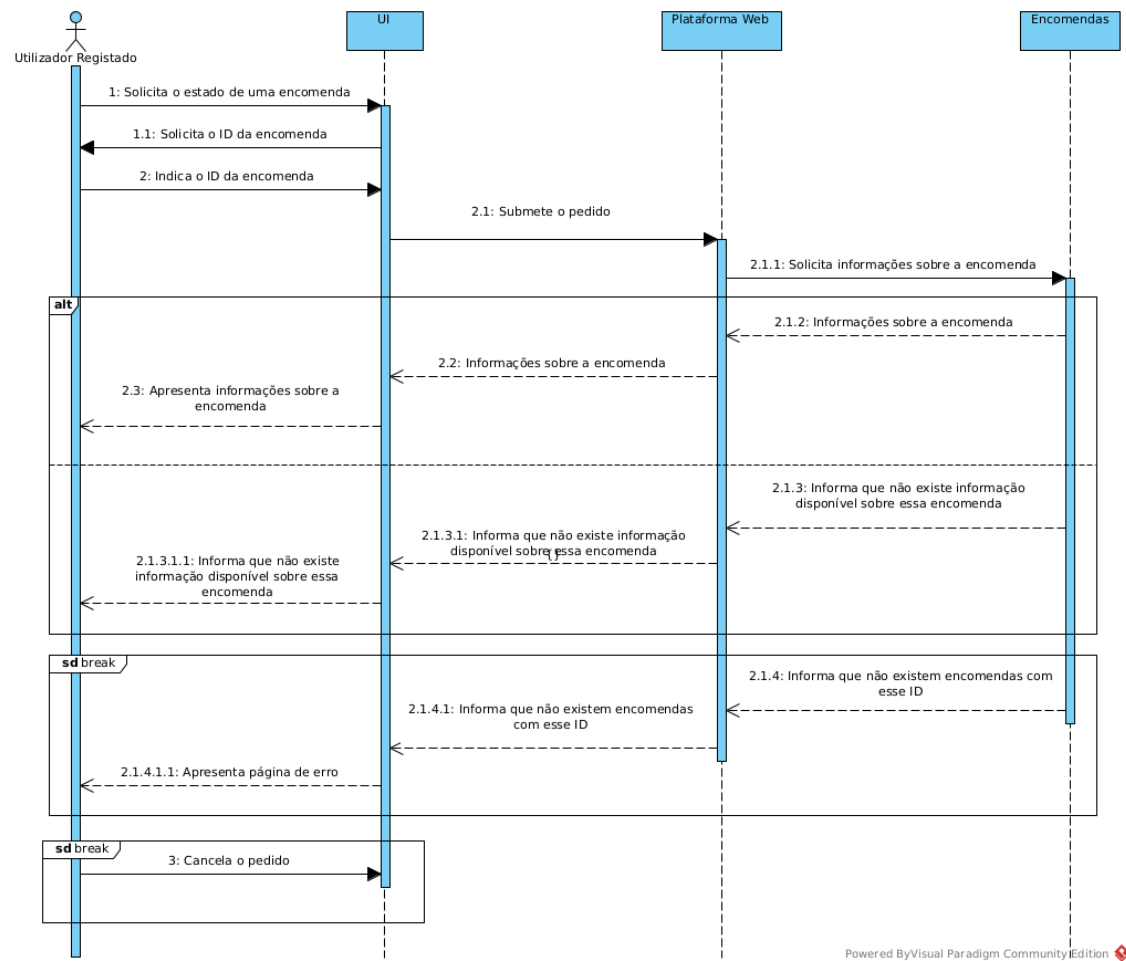


Figura 9.6: Diagrama de Sequência correspondente ao *use case* Consultar Encomenda

9.7 Listagem de Encomendas

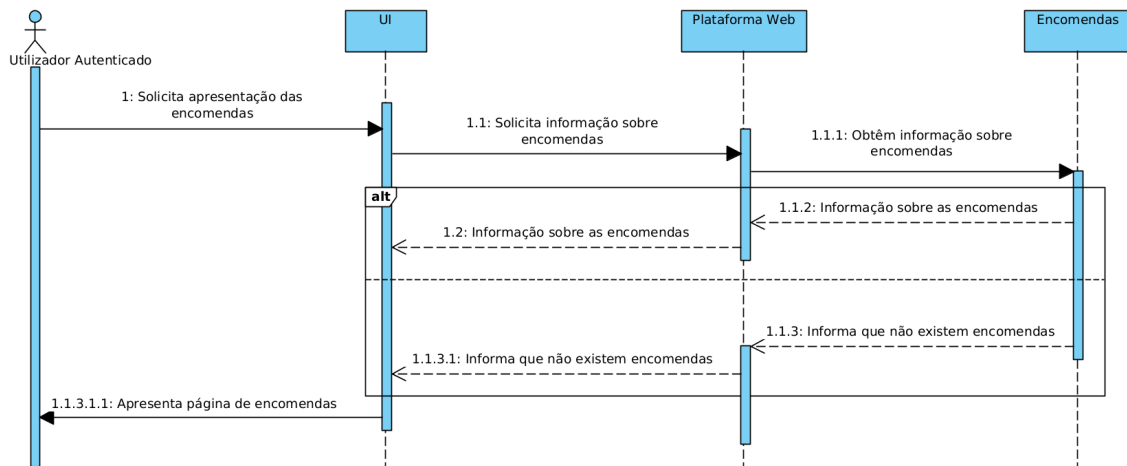


Figura 9.7: Diagrama de Sequência correspondente ao *use case* Listagem de Encomendas

9.8 Consultar Favoritos

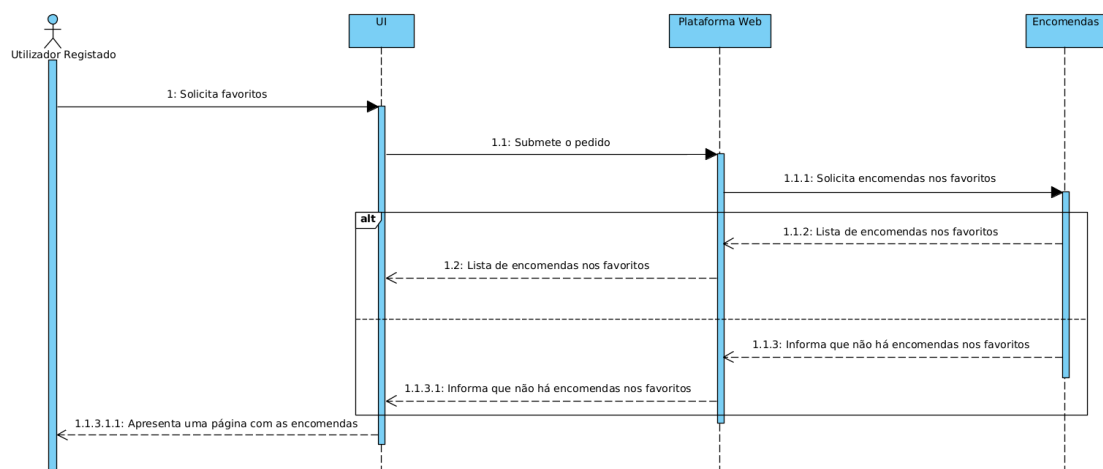


Figura 9.8: Diagrama de Sequência correspondente ao *use case* Consultar Favoritos

9.9 Adicionar Encomenda aos Favoritos

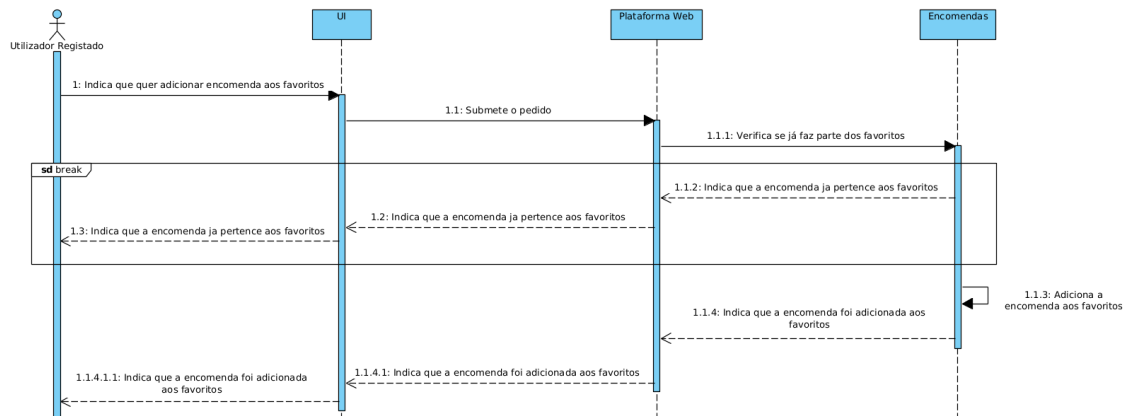


Figura 9.9: Diagrama de Sequência correspondente ao *use case* Adicionar Encomenda aos Favoritos

9.10 Remover Encomenda dos Favoritos

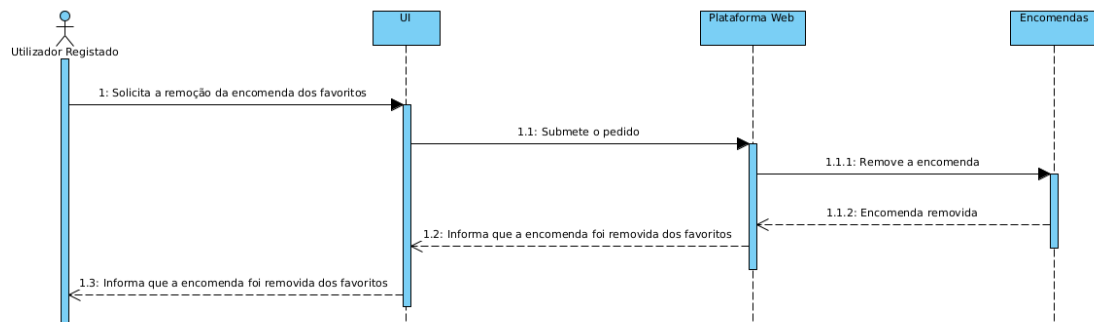


Figura 9.10: Diagrama de Sequência correspondente ao *use case* Remover Encomenda dos Favoritos

9.11 Categorizar Encomenda

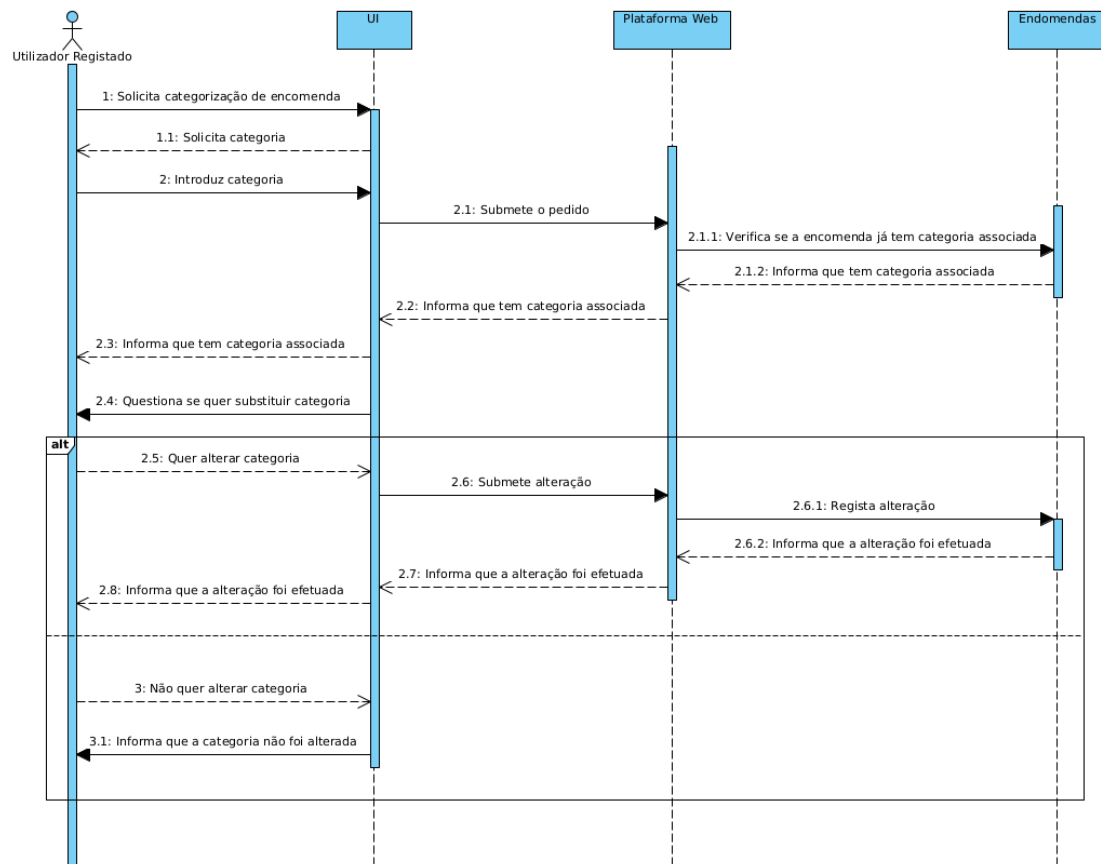


Figura 9.11: Diagrama de Sequência correspondente ao *use case* Categorizar Encomenda

9.12 Filtrar Encomendas

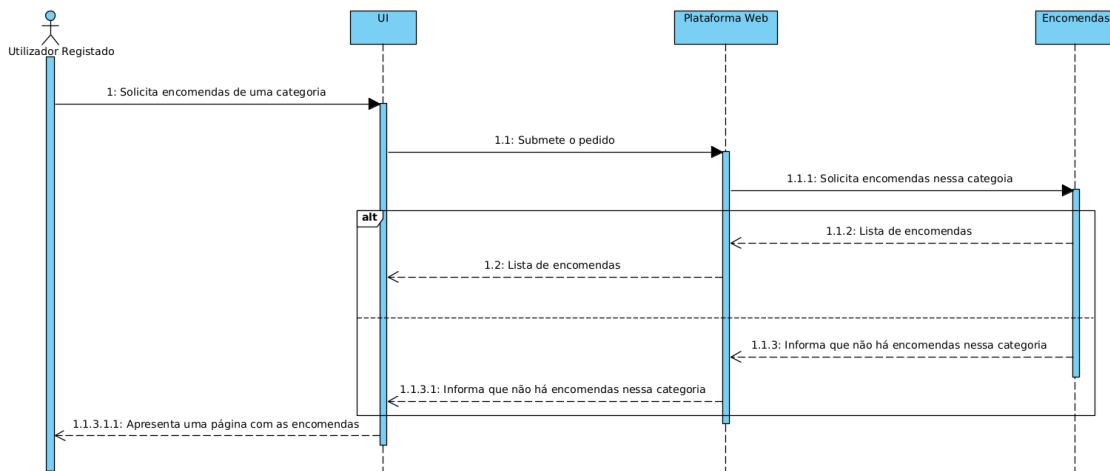


Figura 9.12: Diagrama de Sequência correspondente ao *use case* Filtrar Encomendas

9.13 Consultar Dados do Perfil

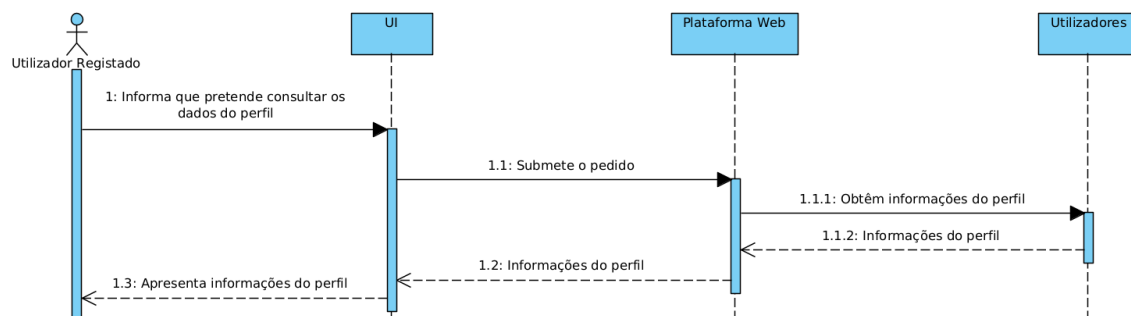


Figura 9.13: Diagrama de Sequência correspondente ao *use case* Consultar Dados do Perfil

9.14 Editar Perfil

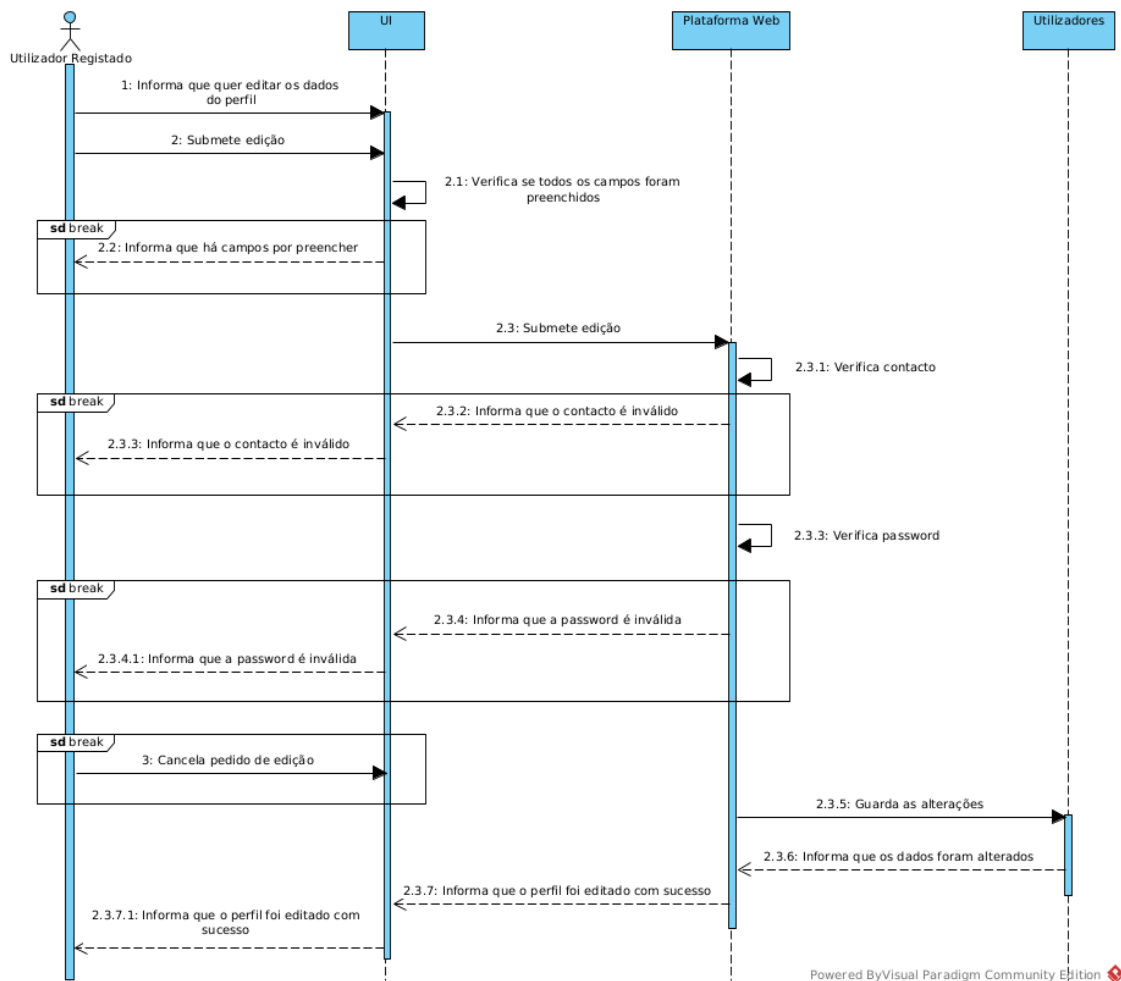


Figura 9.14: Diagrama de Sequência correspondente ao *use case* Editar Perfil

9.15 Remover Perfil

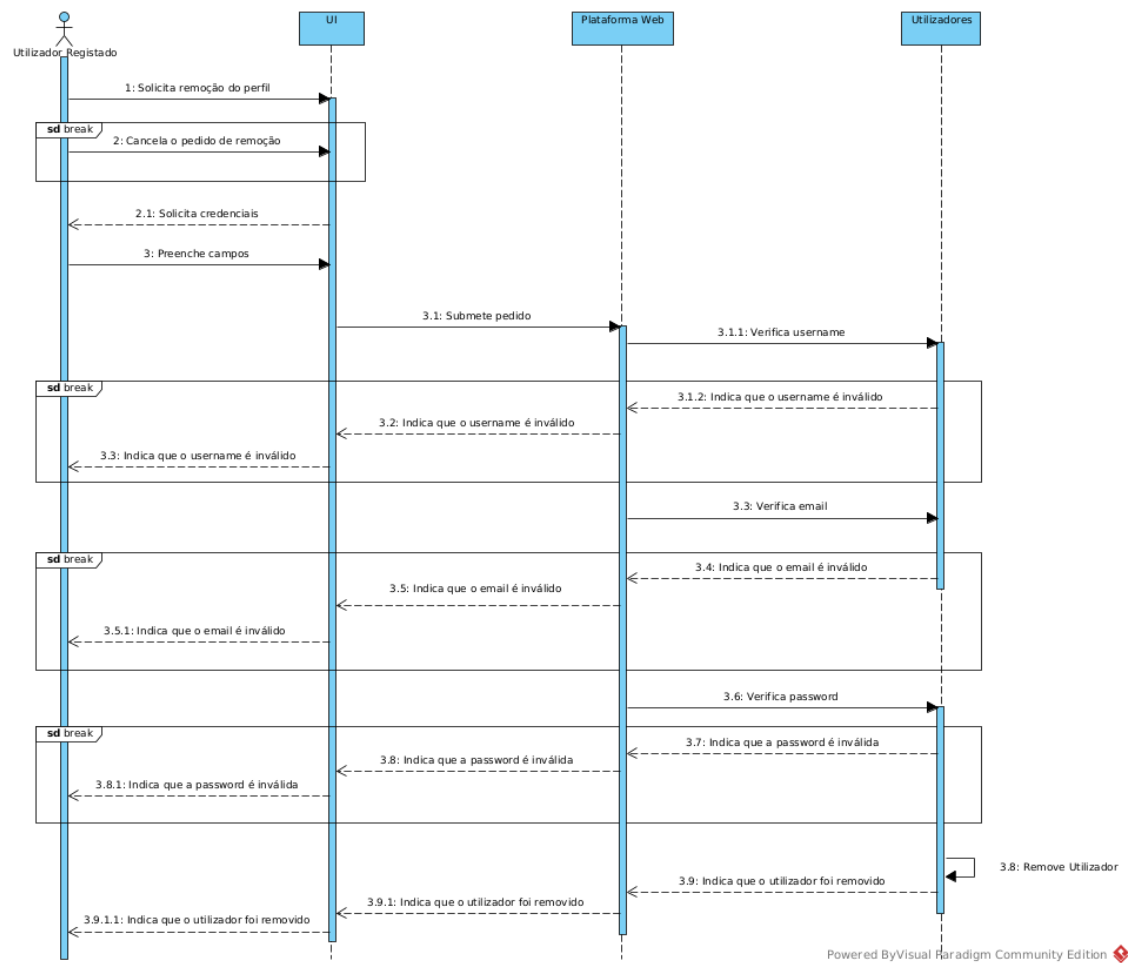


Figura 9.15: Diagrama de Sequência correspondente ao *use case* Remover Perfil

9.16 Consultar Contacto dos Entregadores

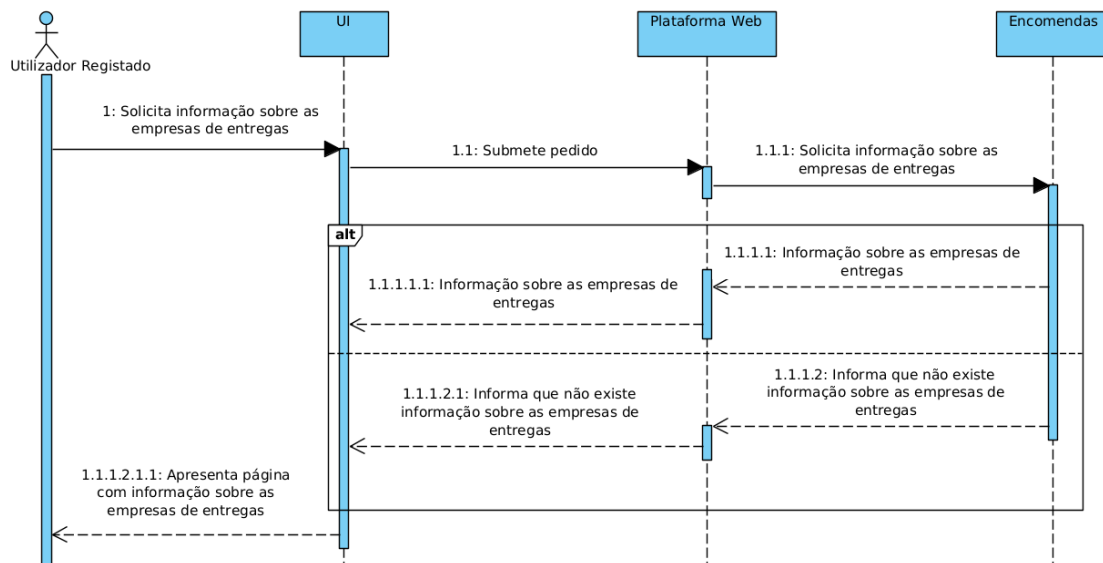


Figura 9.16: Diagrama de Sequência correspondente ao *use case* Consultar Contacto dos Entregadores

9.17 Remover Utilizador

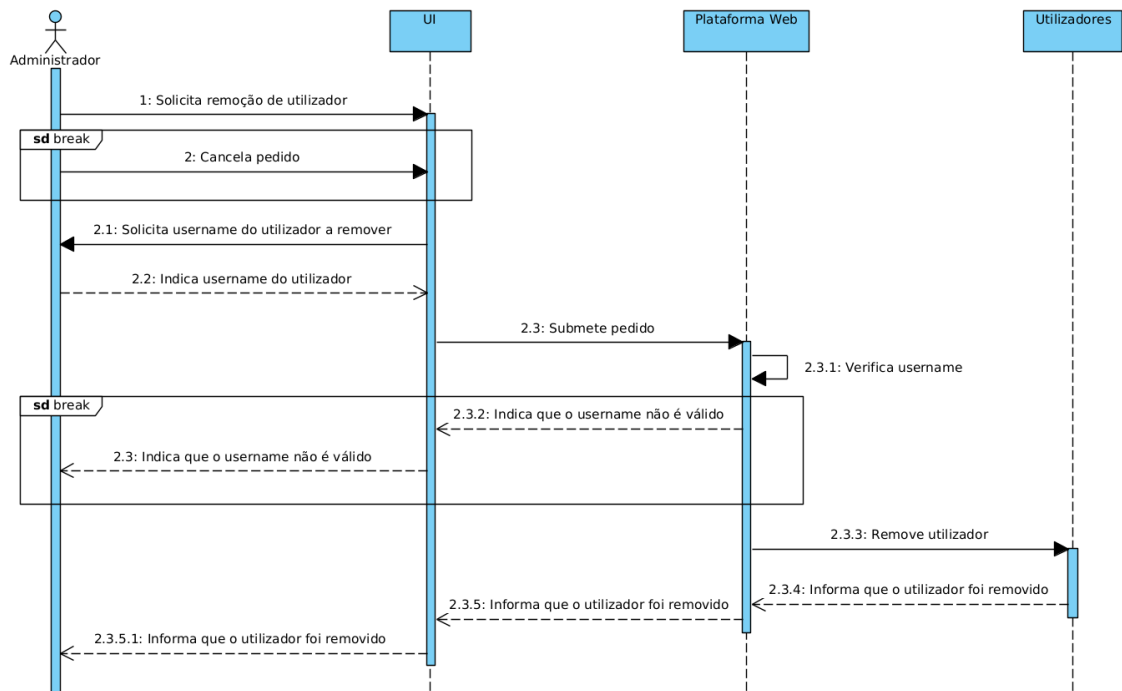


Figura 9.17: Diagrama de Sequência correspondente ao *use case* Remover Utilizador

10 Diagramas de Atividade

Tendo em conta que o projeto a desenvolver se trata de uma aplicação, é fulcral retratar o comportamento dos utilizadores. Assim, utilizando diagramas de atividade, baseados em modelos de fluxo, é possível retratar as interações entre o utilizador e o sistema. Assim, é modelado o fluxo de decisões efetuado ao longo do processo.

10.1 Editar Perfil de Utilizador

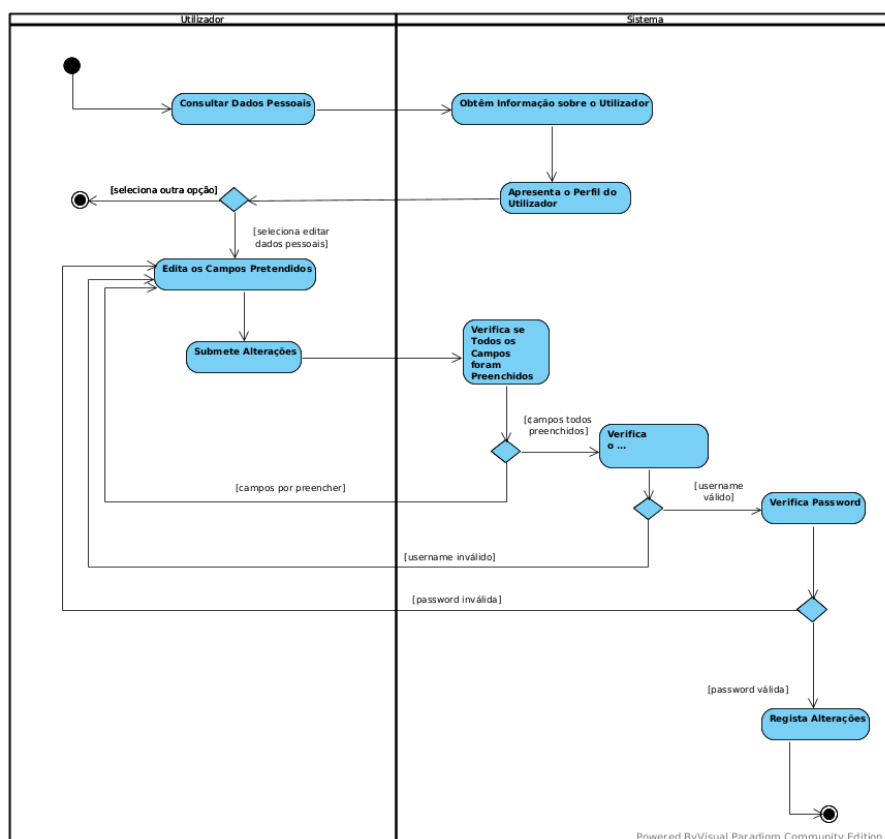


Figura 10.1: Diagrama de Atividade relativo à edição do perfil de utilizador

10.2 Editar Lista de Encomendas Favoritas

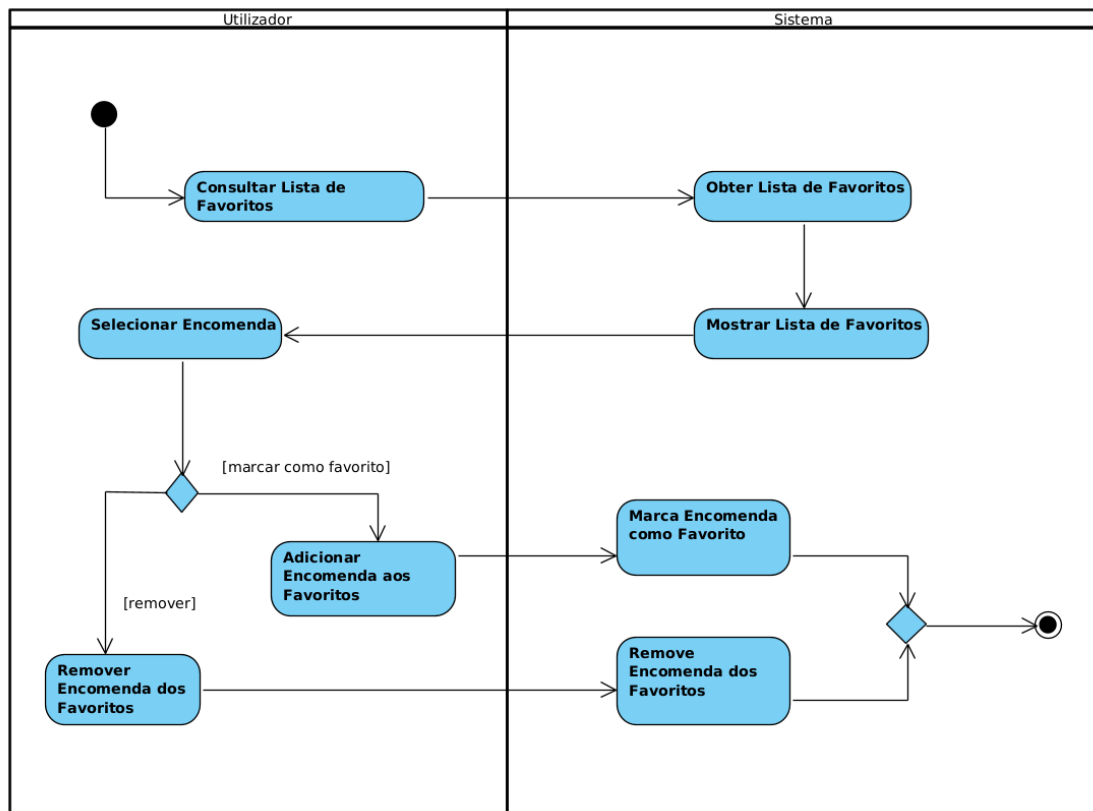


Figura 10.2: Diagrama de Atividade relativo à edição da lista de encomendas favoritas

10.2.1 Consultar Encomenda

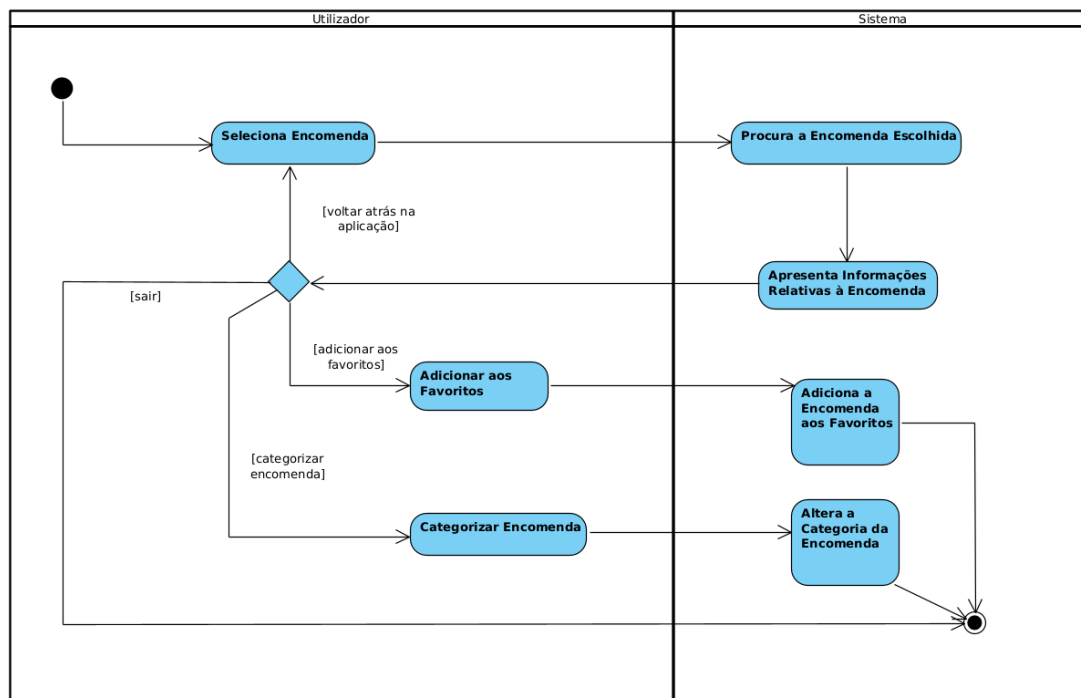


Figura 10.3: Diagrama de Atividade relativo à consulta de uma encomenda

11 Máquina de Estados

Os Diagramas de Estado permitem modelar o comportamento de um dado sistema de forma global. A ênfase é colocada no estado do sistema – modelam-se todos os estados possíveis que o sistema atravessa em resposta aos eventos que podem ocorrer. Desta forma, estes diagramas revelam-se extremamente úteis para modelar o comportamento do sistema de forma transversal aos *use cases*, ou para modelar o sistema como um todo.

De seguida, na Figura 11.1, apresenta-se uma máquina de estados que modela o funcionamento geral do sistema a desenvolver. Por fim, na Figura 11.2, apresenta-se uma máquina de estados que modela o funcionamento geral do registo no sistema.

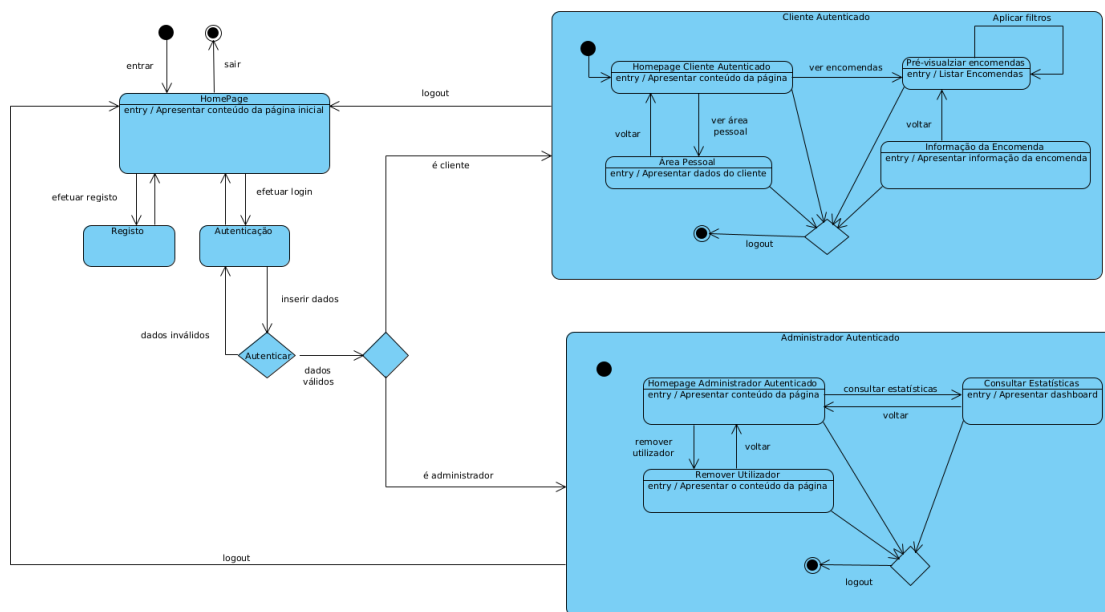


Figura 11.1: Máquina de Estados

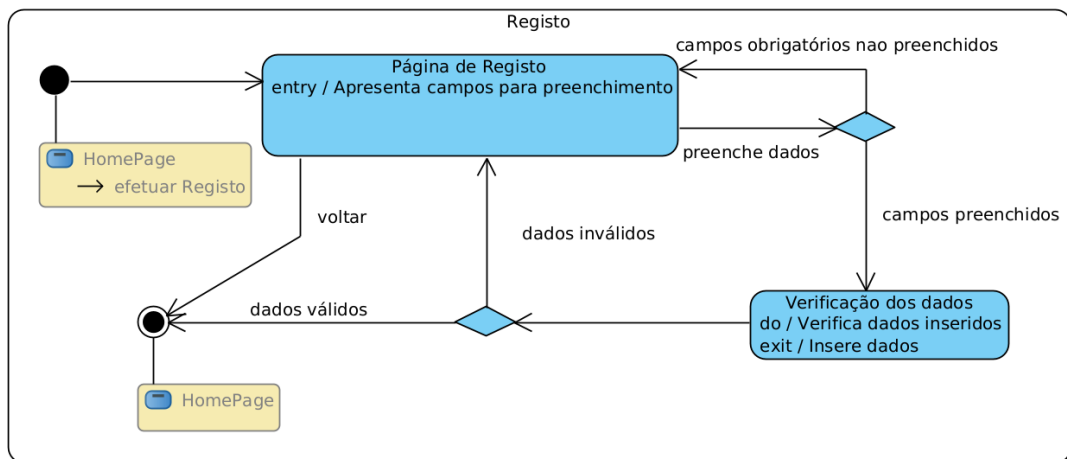


Figura 11.2: Máquina de Estados relativa ao registo na aplicação

12 Arquitetura da Camada de Negócios

Como consequência lógica da análise conjunta dos requisitos e do modelo de domínio apresentado, identificamos e caracterizamos as classes e desenvolvemos uma pequena descrição da arquitetura da solução a desenvolver. Esta descrição foi essencial na construção o diagrama de classes representativo da arquitetura da camada de negócios.

A criação dos seguintes diagramas teve como objetivo mediar a interação com toda a camada de negócios para que a solução apresentada fosse modular e composicional com um sistema congruente com uma arquitetura de três camadas.

12.1 Dicionário das principais Classes

- **Utilizador** – Corresponde à representação no sistema dos utilizadores da aplicação e contém a informação pessoal, informação de acesso e informação sobre as respetivas encomendas;
- **Entregador** – Corresponde à representação no sistema de uma empresa de entregas;
- **Encomenda** – Corresponde à representação no sistema de uma encomenda;
- **InfoEncomenda** – Classe que descreve a relação existente entre um utilizador a uma encomenda, isto é, a categoria em que este a inseriu e se a adicionou aos favoritos ou não.

12.2 Diagrama de Classes

Um diagrama de classes é usado para desenvolver um sistema orientado a objetos que mostre as classes nele presentes e as associações entre elas. É nesta parte da modelação que, no contexto do projeto, foram identificados os objetos essenciais para o seu funcionamento, e foi realizada a representação destes mesmos em classes.

Este diagrama representa, em alto nível, as classes do sistema bem como os seus atributos e métodos. O diagrama foi construído com auxílio dos requisitos previamente apresentados.

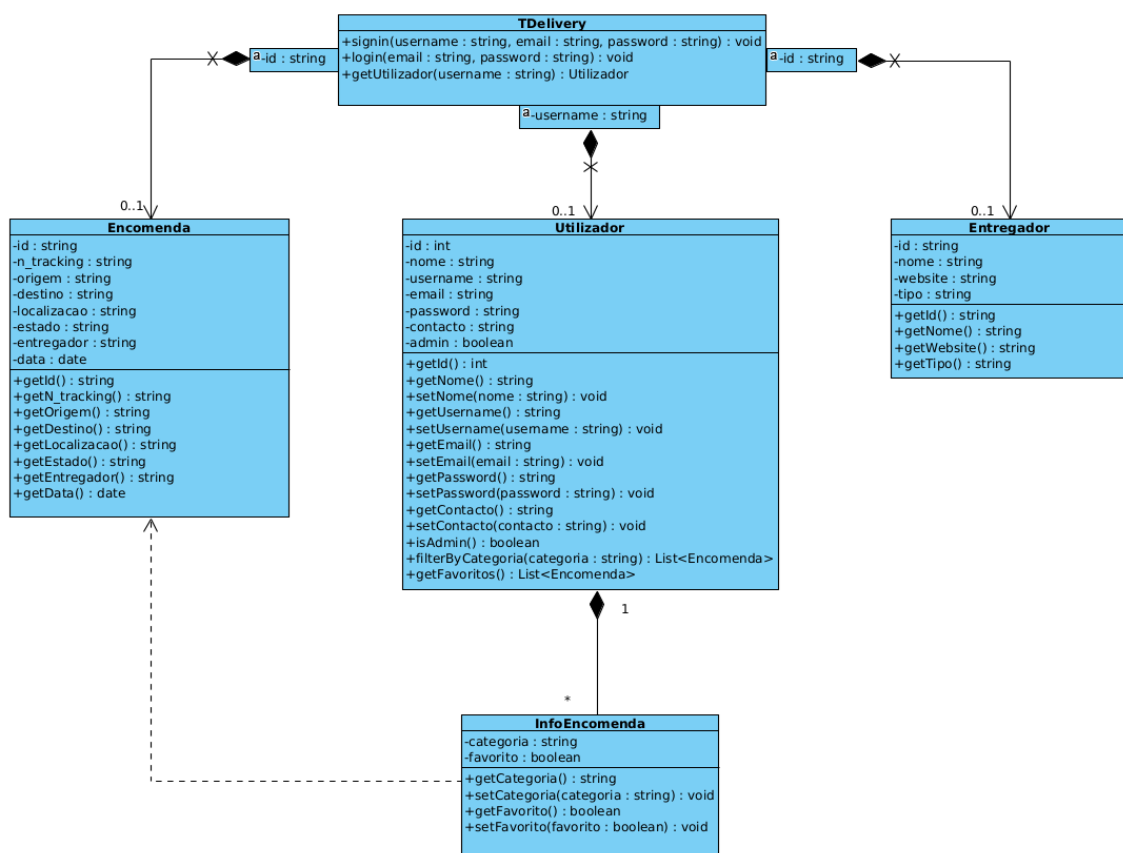


Figura 12.1: Diagrama de Classes

12.3 Diagrama de Classes ORM

Para garantir persistência, foi elaborado um diagrama, ilustrado na Figura 12.2, que insere, partindo do Diagrama de classes previamente apresentado, em alternativa a coleções, unidades de acesso a dados.

A classe `UtilizadorDAO` é necessária para poder ser feito o acesso a todos os utilizadores do sistema, sejam eles administradores ou clientes. De forma análoga, a classe `EntregadorDAO` garante o acesso à informação associada aos entregadores existentes no sistema.

Já a classe `EncomendaDAO` é necessária para poder aceder a todas as encomendas existentes no sistema bem como tudo o que a si está relacionado.

Por fim, a classe `InfoEncomendaDAO` permite aceder informação relativa à relação que um utilizador estabelece com uma encomenda, isto é, a categoria em que este a inseriu e se a adicionou aos favoritos ou não.

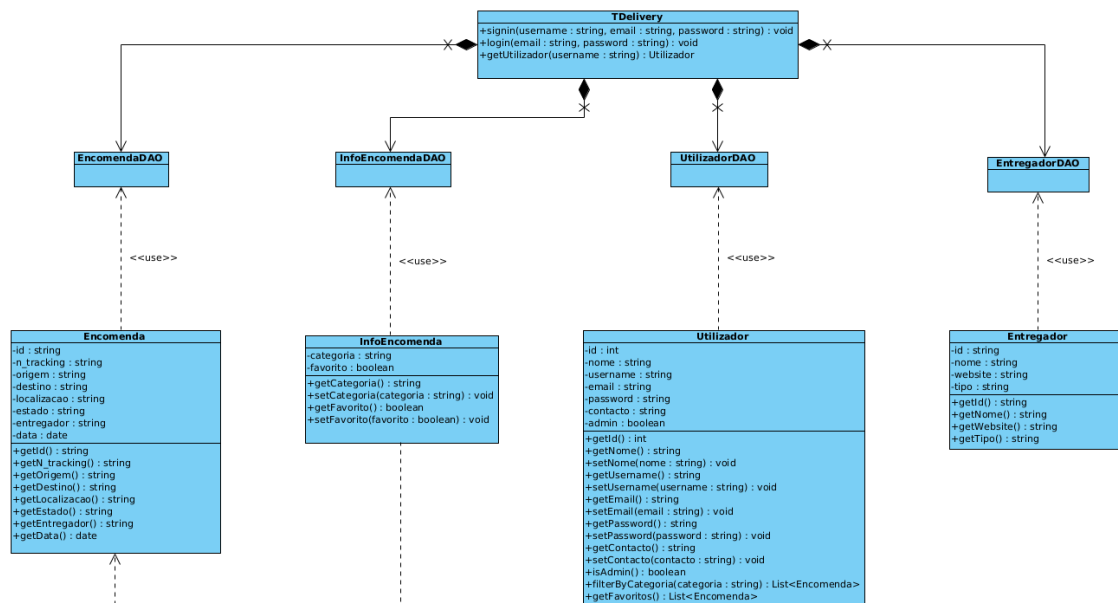


Figura 12.2: Diagrama de Classe ORM

13 Modelação da Camada de Dados

A camada de dados, responsável por suportar a natureza distribuída da solução de *software* proposta, terá que garantir a consistência, integridade, segurança e acessibilidade aos dados.

Para armazenar de modo persistente os dados que consideramos indispensáveis ao bom funcionamento da aplicação, implementamos um Sistema de Gestão de Base de Dados (SGBD) consistente, com o mínimo de redundância, persistente e seguro, capaz de garantir a integridade e independência dos dados. Nessa medida, pretende-se que o SGBD partilhe os dados de uma forma lógica, permitindo aos utilizadores consultá-los.

13.1 Modelo Conceptual

O processo de modelação adotado consiste em, inicialmente, identificar e caracterizar as entidades e relacionamentos entre as várias entidades. De seguida, é necessário definir atributos das entidades e relacionamentos entre as mesmas, determinando o domínio de dados destes. É também fundamental definir as chaves primárias e alternativas mediante as chaves candidatas existentes. Após ser desenvolvido o modelo conceptual será necessário analisar o mesmo de forma a garantir a sua validação.

13.1.1 Identificação e Caracterização das Entidades

Após a leitura e interpretação dos requisitos, identificaram-se vários objetos chave. De entre estes, surgiram as seguintes entidades.

Entidade	Descrição	Ocorrência
Utilizador	Termo geral que descreve as características principais associadas a um utilizador.	Um utilizador pode ser um cliente ou administrador.
Encomenda	Termo geral que refere as características associadas a uma encomenda.	Cada encomenda realizada pelo utilizador.
Entregador	Termo geral que escreve o responsável pela entrega de uma encomenda.	Cada entregador responsável pela entrega de uma encomenda.

Tabela 13.1: Dicionário de Dados das Entidades

13.1.2 Identificação e Caracterização dos Relacionamentos

Uma vez definidas as entidades do sistema, torna-se imperativo estabelecer as relações entre as mesmas. Novamente, por leitura e interpretação dos requisitos, identificaram-se os relacionamentos e respetiva cardinalidade, apresentados na tabela seguinte.

Entidade	Relacionamento	Entidade	Descrição	Cardinalidade
Utilizador	Acompanha	Encomenda	Um utilizador acompanha o estado de várias encomendas.	1:N
Entregador	Entrega	Encomenda	Um entregador é responsável pela entrega de várias encomendas.	1:N

Tabela 13.2: Relacionamentos entre as entidades

13.1.3 Identificação e caracterização dos atributos com as entidades e relacionamentos

Após ter sido feita a identificação das entidades e dos relacionamentos, foi necessário identificar os atributos de cada um destes.

Entidades

Entidade	Atributo	Descrição	Tipo de Dados	Nulo	Tipo de Atributo
Utilizador	ID	Identificador único do utilizador	Número Inteiro	Não	Simples
	Nome	Nome do utilizador	Sequência de caracteres	Não	Simples
	Username	<i>Username</i> do utilizador	Sequência de caracteres	Não	Simples
	Email	<i>Email</i> do utilizador	Sequência de caracteres	Não	Simples
	Password	<i>Password</i> do utilizador	Sequência de caracteres	Não	Simples
	Contacto	Contacto do utilizador	Sequência de caracteres	Não	Simples
	Admin	Booleano que indica se o utilizador é um cliente ou o administrador	Bit	Não	Simples

Tabela 13.3: Atributos da entidade Utilizador

Entidade	Atributo	Descrição	Tipo de Dados	Nulo	Tipo de Atributo
Encomenda	ID	Identificador único da encomenda	Número Inteiro	Não	Simples
	Número de <i>Tracking</i>	Número de <i>tracking</i> da encomenda, fornecido pela empresa de entregas	Sequência de caracteres	Não	Simples
	Data	Data em que a encomenda foi efetuada	Data	Não	Simples
	Origem	Local de origem da encomenda	Sequência de caracteres	Não	Simples
	Destino	Local onde a encomenda deve ser entregue	Sequência de caracteres	Não	Simples
	Estado	Estado atual da encomenda – Entregue ou Em Trânsito	Sequência de caracteres	Não	Simples
	Entregador	Identificador único do entregador responsável pela entrega	Número Inteiro	Não	Simples

Tabela 13.4: Atributos da entidade Encomenda

Entidade	Atributo	Descrição	Tipo de Dados	Nulo	Tipo de Atributo
Entregador	ID	Identificador único do entregador	Número Inteiro	Não	Simples
	Nome	Nome da empresa de entregas	Inteiro	Não	Simples
	Tipo	Tipo de entrega – <i>Express</i> ou Normal	Sequência de caracteres	Sim	Simples
	Website	Website da empresa responsável pela entrega	Sequência de caracteres	Sim	Simples
	Contacto	Contacto da empresa responsável pela entrega	Sequência de caracteres	Sim	Simples

Tabela 13.5: Atributos da entidade Entregador

Relacionamentos

Relacionamento	Atributo	Descrição	Tipo de Dados	Nulo	Tipo de Atributo
Acompanha	Favorito	Booleano que indica se a encomenda pertence aos favoritos do utilizador	Bit	Não	Simples
	Categoria	Categoria em que o utilizador inseriu a encomenda	Sequência de caracteres	Sim	Simples

Tabela 13.6: Atributos do relacionamento Acompanha

13.1.4 Chaves Primárias

A escolha de chaves primárias para cada uma das entidades referidas anteriormente passou por um processo sequencial e idêntico, uma vez que, para a maioria destas, foi decidido a utilização de um atributo ID como seu identificador.

No caso do utilizador, importa salientar que, ainda que *email* e o *username* identifiquem univocamente um utilizador na base de dados, sendo, portanto, chaves candidatas, este pode,

se assim entender, alterar estes dados, pelo que se torna imperativo a criação de um ID.

13.1.5 Diagrama ER

Ao longo da elaboração do diagrama ER, tentou-se representar da forma mais clara possível as entidades mais importantes, assim como os seus relacionamentos, tendo sempre em consideração a legibilidade do diagrama e a clareza de modelação.

Assim, após a realização de todas as etapas definidas, obtivemos como resultado o seguinte modelo conceptual:

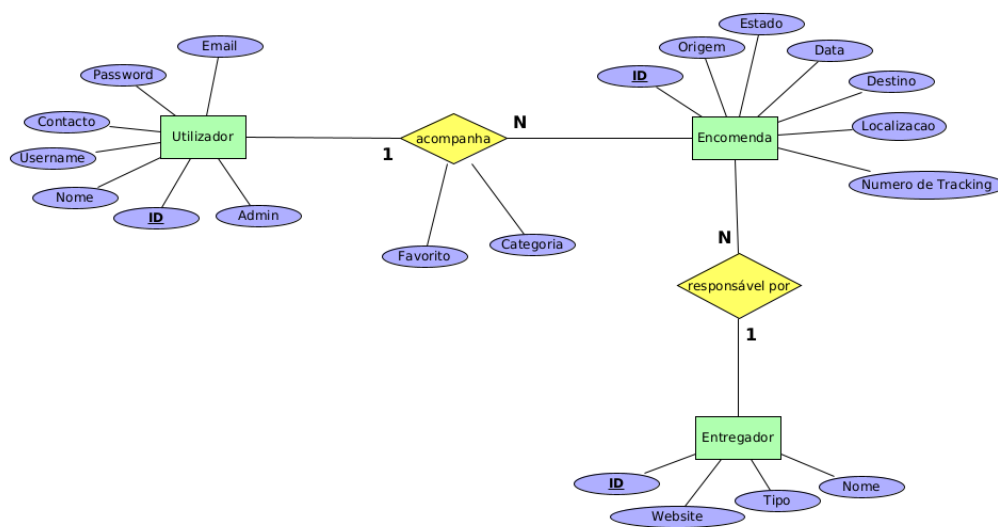


Figura 13.1: Diagrama ER

13.2 Modelo Lógico

Numa segunda fase da modelação da BD, é necessária a criação de um modelo lógico (ML) tendo por base o modelo conceptual apresentado anteriormente. Este processo dá origem a diversas tabelas de registo de dados. Nestas existem chaves primárias ou estrangeiras, que são chaves primárias de outras tabelas. Os atributos, bem como o seu tipo de dados e tamanho estão ainda presentes neste tipo de tabelas.

Assim, apresenta-se o modelo lógico da BD na Figura 13.2.

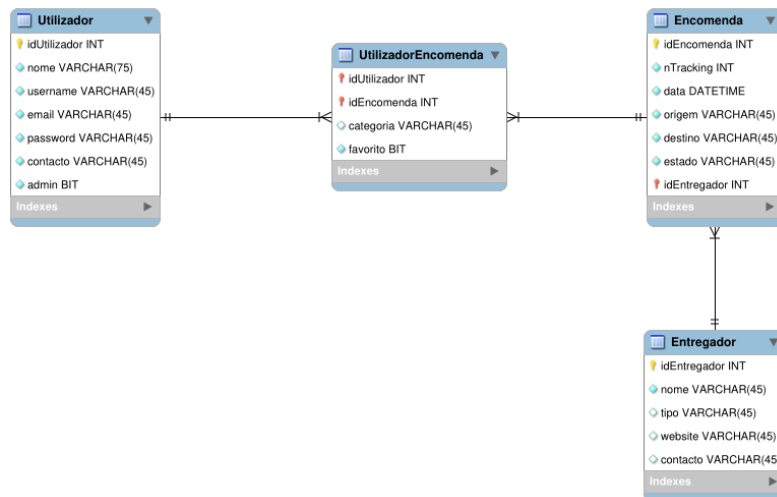


Figura 13.2: Modelo Lógico da Base de Dados

13.2.1 Validação do Modelo Lógico

Efetuada a validação do ML através da normalização, é necessário avaliar o mesmo modelo com os utilizadores finais da BD, de modo a perceber se o modelo criado vai de encontro às suas expectativas de utilização.

Neste sentido, os *stakeholders* foram requeridos para rever o modelo lógico, de modo a confirmar se este representa fidedignamente os dados necessários para o bom funcionamento da ideia em causa.

13.3 Estimativa do espaço em disco

Um passo importante no desenvolvimento de uma BD é perceber o seu desempenho no futuro, nomeadamente o espaço em disco que irá ocupar. Para tal, calculou-se o tamanho em disco de cada uma das tabelas da BD, tendo por base uma estimativa de crescimento previsto do tamanho das mesmas com a introdução de dados. A estimativa é calculada tendo o domínio dos atributos de cada tabela. Para isso, é necessário, inicialmente, investigar os tipos de dados dos atributos dessas mesmas entidades.

Tipo de Dados	Tamanho, em <i>bytes</i>
BIT	0.125
INT	4
DATE	3
VARCHAR(N)	$N + 2$

Tabela 13.7: Espaço ocupado pelos diferentes tipos de dados

Com base na informação da tabela acima e assumindo, o máximo de caracteres possíveis para os atributos textuais assim como não existências de valores nulos, calculou-se os seguintes resultados por cada inserção numa tabela em *bytes*:

Tabela	Tamanho, em <i>bytes</i>
Utilizador	269.125
Encomenda	156
Entregador	192
UtilizadorEncomenda	55.125

Tabela 13.8: Espaço ocupado por cada entrada nas diferentes tabelas

Considerando a análise de mercado levada a cabo anteriormente, é espectável que, após um ano, a BD contenha registos relativos a 50 entregadores, 25000 encomendas e a 5000 utilizadores, sendo que é esperado que cada um acompanhe, em média, 50 encomendas. Desta forma, calculam-se os seguintes valores:

Tabela	Tamanho, em <i>bytes</i>
Utilizador	$269.125 \times 5000 = 1345625$
Encomenda	$156 \times 25000 = 3900000$
Entregador	$192 \times 50 = 9600$
UtilizadorEncomenda	$55.125 \times 5000 \times 50 = 1.378125E7$
Σ	$1.9036475E7$

Tabela 13.9: Estimativa do espaço em disco ocupado pela BD após um ano

Em suma, no fim de um ano, serão necessários cerca de $1.9036475E7$ *bytes*, *i.e.* cerca de 19MB de espaço em disco para suportar a BD. Contudo, tendo em conta que determinados campos podem ser nulos ou que nem todos podem utilizar os caracteres máximos disponíveis, este valor pode ser inferior.

14 Prototipagem

Na maioria dos sistemas de *software* orientados ao público em geral e a uma utilização diária, um dos aspetos essenciais para o sucesso do produto é, sem dúvida, a sua interface com o utilizador. Tendo em conta o objetivo principal desta aplicação, rapidamente se percebe que a interface é uma das partes mais importantes do projeto.

As propostas de interface – *mockups* – assumem um papel importante na fase da especificação do sistema na medida em que:

- Permitem definir alguns detalhes, como o esquema de cores, tipos de letra e esquemas visuais a serem utilizados posteriormente, na fase de implementação;
- Permitem desenvolver o programa mais próximo do ponto de vista do utilizador;
- Permitem definir o comportamento da aplicação prematuramente, detetando problemas de usabilidade antes de iniciar o desenvolvimento do *software*.

Para além disto, os *mockups* complementam-se como parte da modelação do projeto, uma vez que a modelação contribui para decisões a tomar no aspeto visual do *website* e o próprio desenho concetual do projeto desencadeia alterações na modelação UML do projeto, visto ser mais fácil enquadrar-nos na perspetiva de utilizador, e consequentemente, detetar falhas na estruturação do projeto.

Neste sentido, a interface foi desenhada de modo a que este seja um sistema *user-friendly*, capaz de ser utilizado por pessoas com pouca literacia digital, mas também complexo o suficiente para responder a todos os requisitos no que à funcionalidade do sistema diz respeito.

De seguida, apresentam-se os *mockups* desenvolvidos.

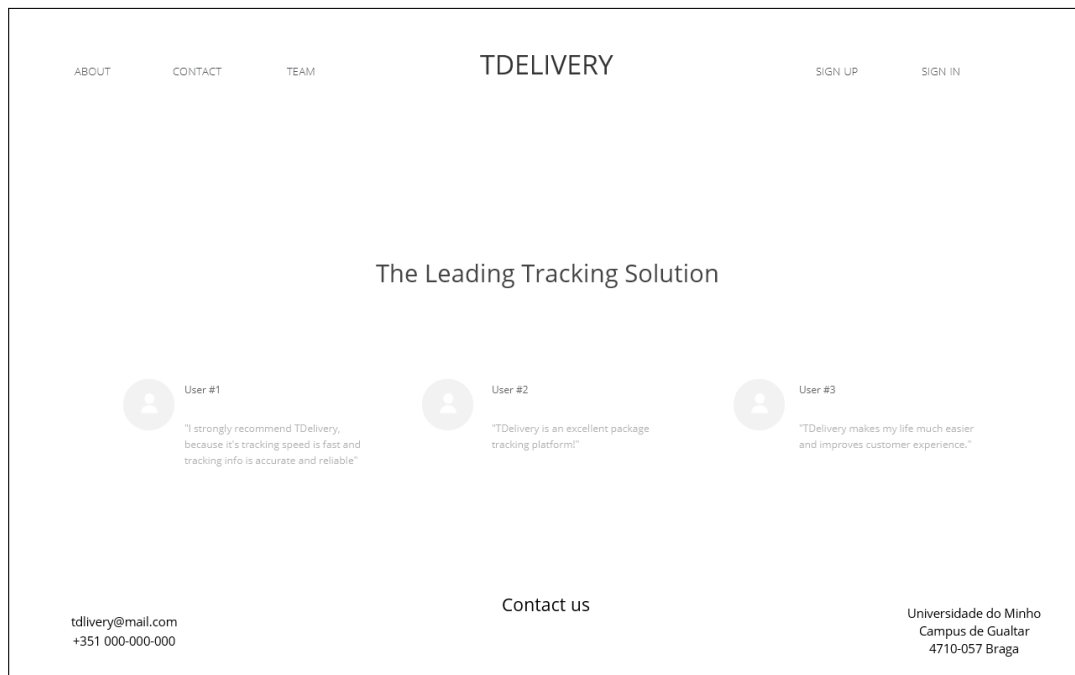


Figura 14.1: Proposta de interface para a *landing page*

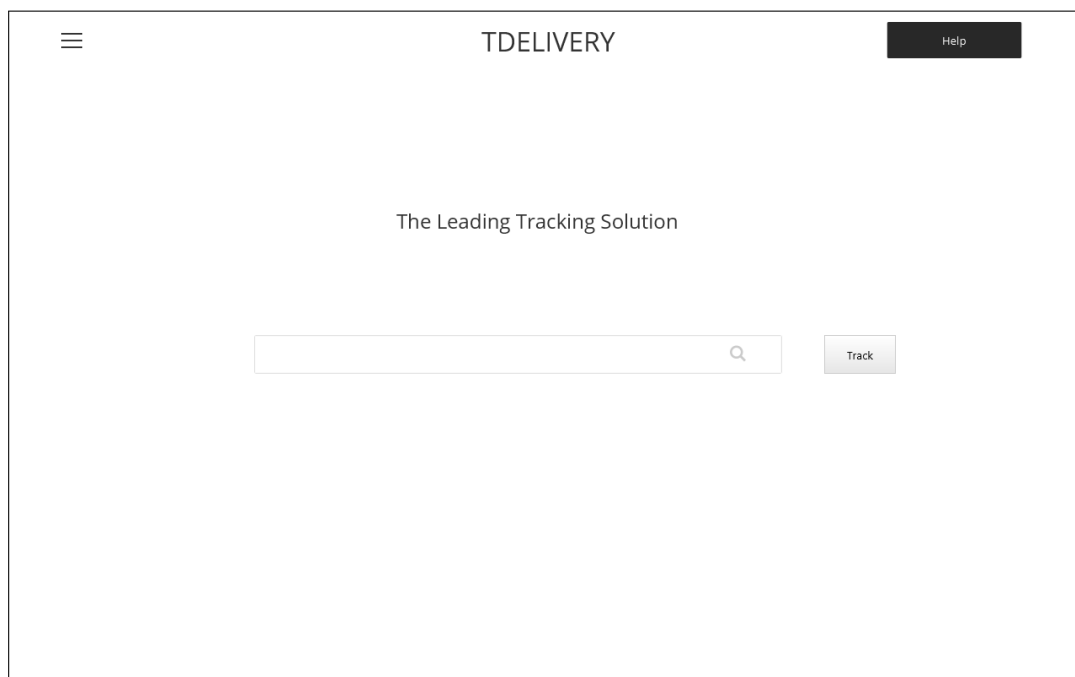


Figura 14.2: Proposta de interface para a página inicial

TDELIVERY

×

Sign Up

Username

Email

Password

Confirm password

☒ I agree with terms of use and privacy

SIGN UP

Figura 14.3: Proposta de interface para a página de registo

TDELIVERY

×

Sign In

Username

Password

SIGN IN

Forgot password?

Not a member yet?
Register

Figura 14.4: Proposta de interface para a página de autenticação

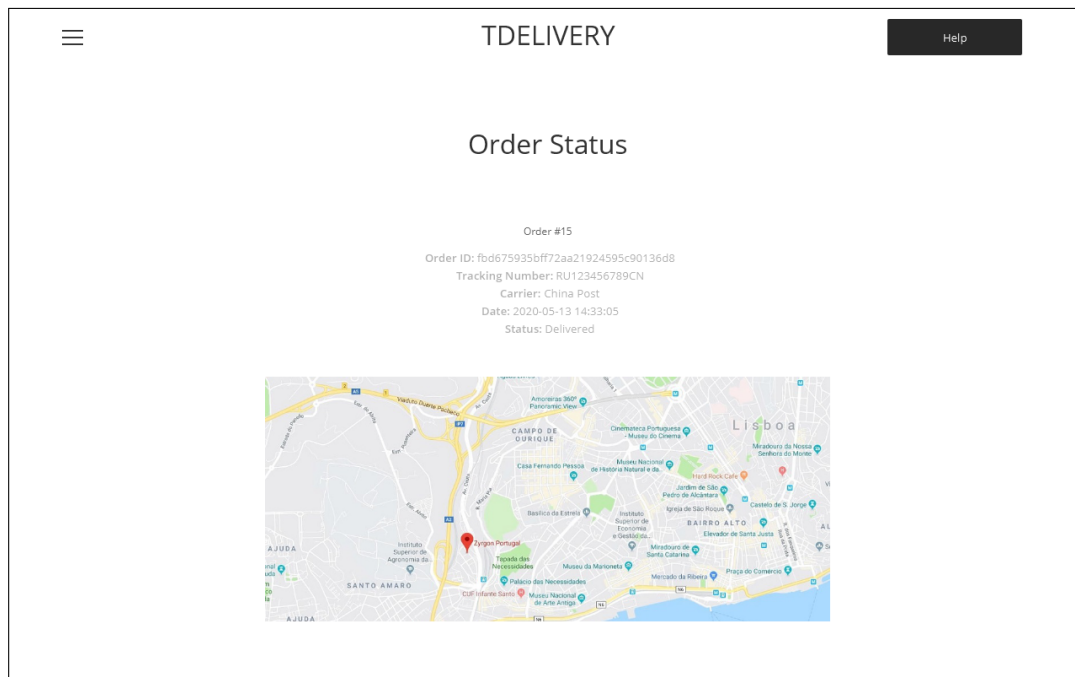


Figura 14.5: Proposta de interface para a página relativa ao estado de uma encomenda

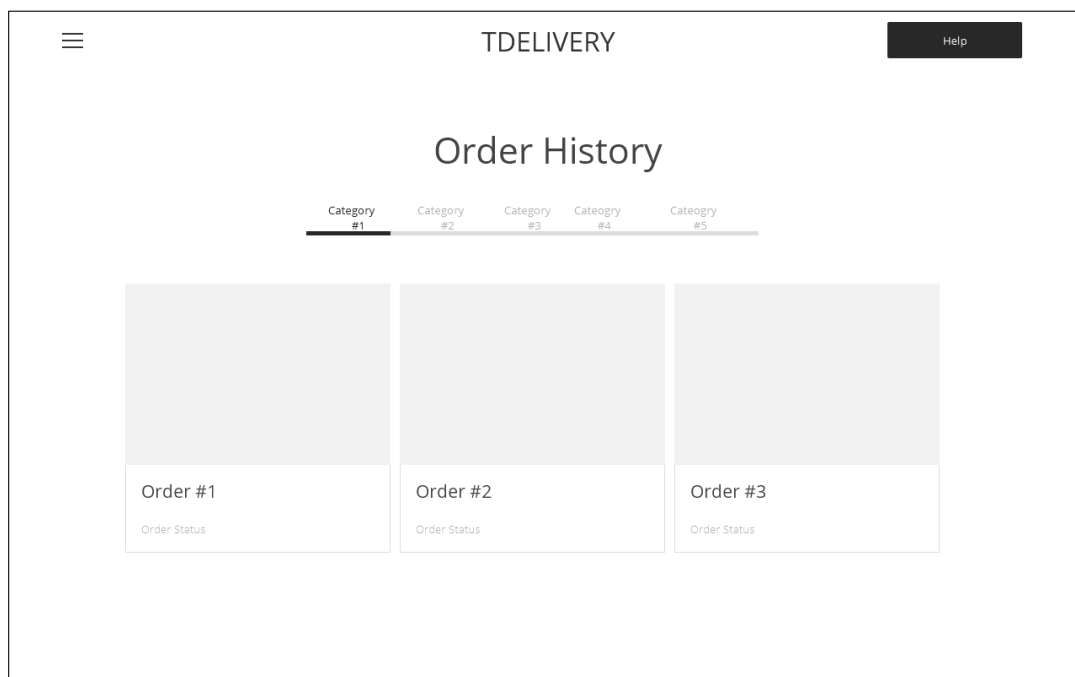


Figura 14.6: Proposta de interface para o histórico de encomendas

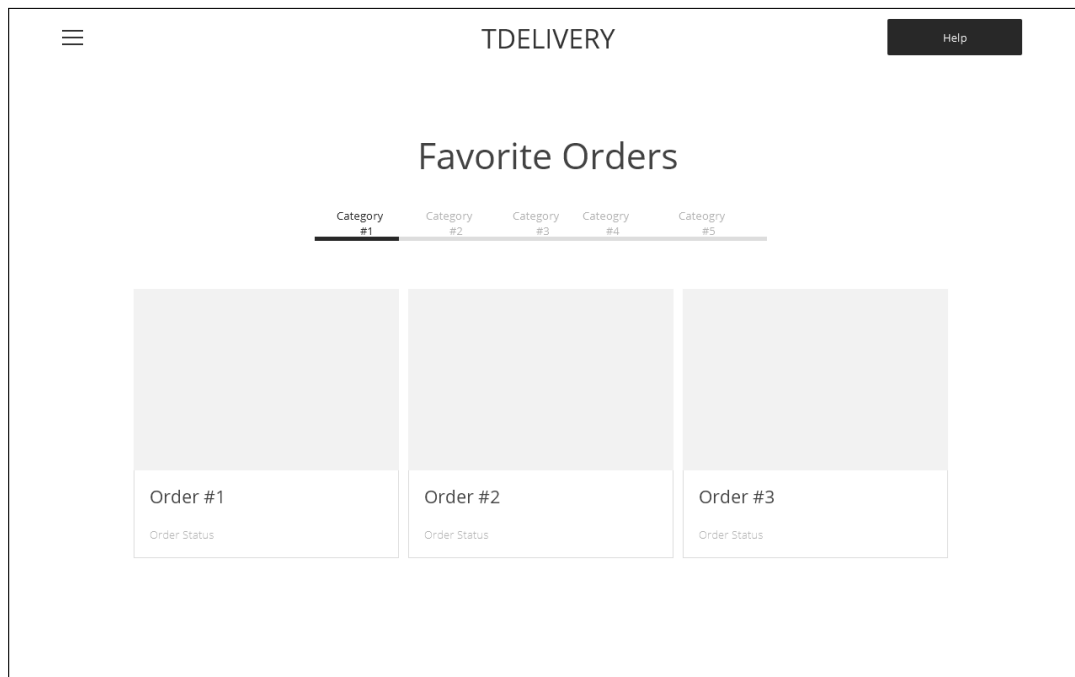


Figura 14.7: Proposta de interface para a lista de favoritos

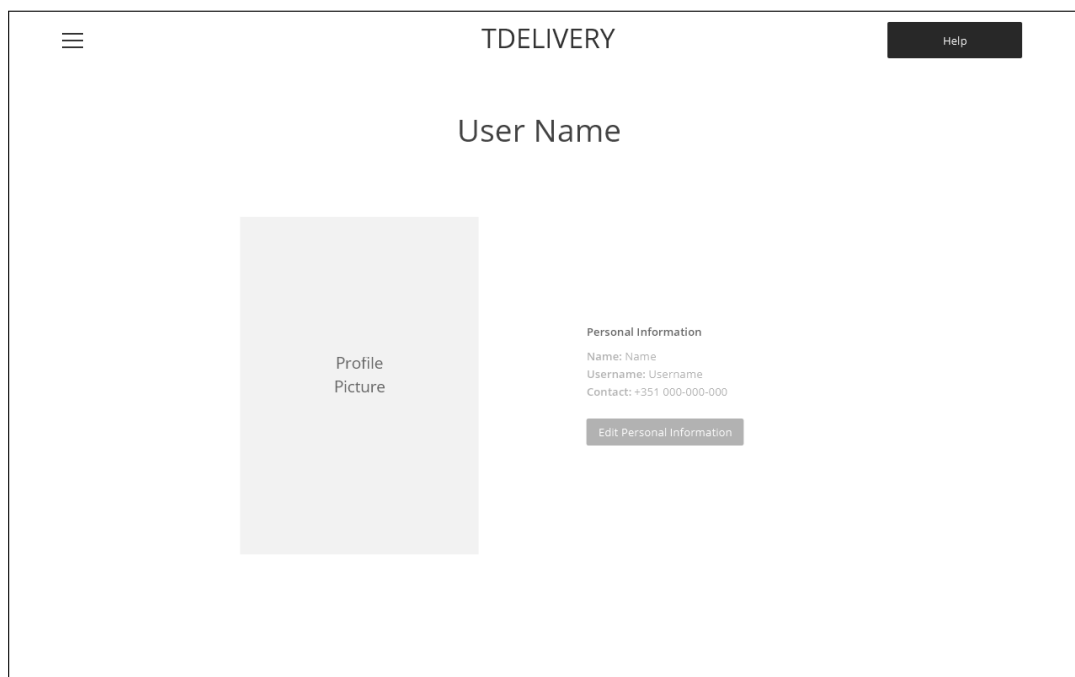


Figura 14.8: Proposta de interface para o perfil do utilizador

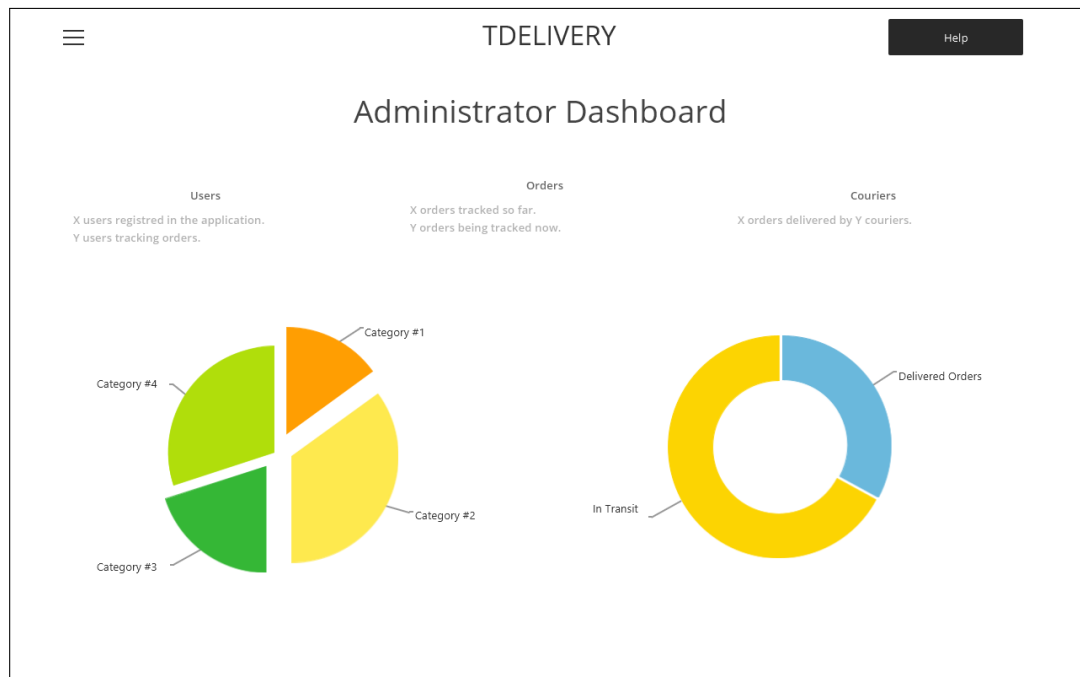


Figura 14.9: Proposta de interface para a *dashboard* do administrador

15 Implementação do Sistema

A terceira e última fase deste trabalho prático consiste na implementação do sistema de *software* modelado anteriormente. Aqui criamos a base de dados e o código com toda a lógica do sistema e a interface que queremos desenvolver.

Nesta fase, tivemos que decidir sobre quais as ferramentas a utilizar, de acordo com as necessidades correntes.

Inicialmente tivemos que criar a base de dados especificada na fase anterior, no entanto implementamos com algumas diferenças. Além da base de dados que implementamos, utilizamos também APIs externas de modo a cumprir os requisitos que implementamos inicialmente.

Tendo em conta as funcionalidades descritas e os requisitos estipulados, serão expostas as abordagens adotadas para cada caso, e serão apresentados os resultados obtidos no que toca à interação com o utilizador.

Tendo isto em conta, o seguinte diagrama que descreve a arquitetura adotada para o *software* a implementar.

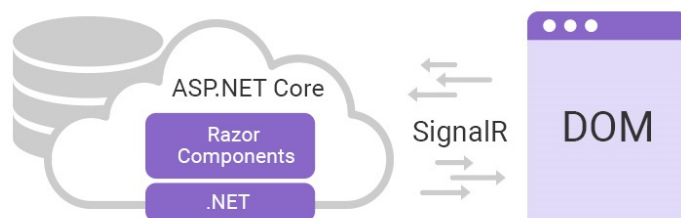


Figura 15.1: Arquitetura de *software*

15.1 Povoamento da Base de Dados

Uma vez que a empresa TDelivery não permite o registo de administradores, estes são pré-registados na BD, sendo este trabalho feito pelo administrador da base de dados. Tendo em conta os trabalhadores atuais da empresa, o administrador gerou um *script* com a informação relativa a estes.

Para além disto, também as informações sobre as empresas de entregas são registados pelo administrador através do seguinte *script* de povoamento.

```
INSERT INTO [dbo].[Entregadores] VALUES
('CTT', 'Express', 'ctt.pt', '+351707262626'),
('CTT', 'Normal', 'ctt.pt', '+351707262626'),
('Australia Post', 'Normal', 'apost.au', '+61388479980'),
('Brazil Correios', 'Normal', 'bcorreios.br', '556130030100'),
('Russia Post', 'Express', 'rpost.ru', '+7(495)956-20-67'),
('Poste Italiane', 'Normal', 'postei.it', '39803160'),
('Yanwen', 'Normal', 'yanwen.ch', '86 400-108-5656'),
('DHL Germany', 'Normal', 'dhl.de', '+4922828609898'),
('Mexico Post', 'Express', 'mexpost.mx', '(55)53403300'),
('DHL Parcel NL', 'Normal', 'dhl.nl', '+31(D)900-2222120'),
('TNT Italy', 'Normal', 'tnt.it', '+39199803868'),
('Ukraine Post', 'Normal', 'upost.ua', '+380443232099'),
('Japan Post', 'Express', 'jpost.jp', '+810570-046111'),
('Czech Post', 'Normal', 'cpost.cz', '420950292102'),
('BuyLogic', 'Express', 'buylogic.bl', '86-755-8145512286'),
('Macao Post', 'Normal', 'macaopost.mo', '+84328574491'),
('Korea Post', 'Express', 'korapost.kr', '82-2-2108-9895'),
('Romania Post', 'Normal', 'romaniapost.ro', '+400219393111');
```

Listing 15.1: Povoamento da BD com informação sobre as empresas de entregas

```
INSERT INTO [dbo].[Encomendas] VALUES
('gFrPA', '2021-04-30', 'Povoa de Lanhoso', 'Braga', 'Entregue',
4),
('fVzjB', '2021-04-30', 'Porto', 'Braga', 'Entregue', 1),
('n0ukx', '2021-05-01', 'Aveiro', 'Faro', 'Entregue', 18),
('v6bbs', '2021-05-02', 'Lisboa', 'Coimbra', 'Em Transito', 3),
('Yb5lg', '2021-05-03', 'Setubal', 'Guarda', 'Entregue', 15),
('O50al', '2021-05-04', 'Guimarães', 'Aveiro', 'Entregue', 7),
('LaZcJ', '2021-05-04', 'Famalicão', 'Lisboa', 'Em Transito', 8),
('PSiko', '2021-05-04', 'Barcelos', 'Guimarães', 'Alfândega', 9),
,
```

```
( '4fUnW', '2021-05-10', 'Braga', 'Faro', 'Entregue', 10),
( '2hsKF', '2021-05-10', 'Braga', 'Barcelos', 'Em Transito', 13),
( '1xmqm', '2021-05-11', 'Ermesinde', 'Porto', 'Em Transito', 10)

( 'P1Dd8', '2021-05-15', 'Povoa de Varzim', 'Braga', 'Em
  Processamento', 3),
( 'ZlewP', '2021-05-15', 'Esposende', 'Braga', 'Alfândega', 2),
( 'd3Yja', '2021-05-22', 'Viana do Castelo', 'Coimbra', 'Em
  Processamento', 11),
( 'TCosV', '2021-05-23', 'Frankfurt', 'Aveiro', 'Em Processamento
  ', 15),
( 'rO00v', '2021-05-28', 'Milao', 'Valongo', 'Em Transito', 5),
( 'Ui7rv', '2021-06-01', 'Nice', 'Alfena', 'Entregue', 6),
( '87Ckx', '2021-06-01', 'Londres', 'Porto', 'Em Transito', 8),
( 'F7o5H', '2021-06-03', 'Stuttgart', 'Beja', 'Em Processamento',
  6),
( 'WGewU', '2021-06-03', 'Coimbra', 'Oeiras', 'Em Processamento',
  9),
( '2wTCH', '2021-06-03', 'Napoles', 'Portalegre', 'Em Transito',
  1),
( 'lc2Yg', '2021-06-03', 'Madrid', 'Braga', 'Alfândega', 17),
( 'tREsW', '2021-06-03', 'Toledo', 'Vila Verde', 'Em
  Processamento', 15),
( 'vREKU', '2021-06-05', 'Maiorca', 'Bragança', 'Em Transito',
  13),
( 'n1LyK', '2021-06-05', 'Funchal', 'Vieira do Minho', 'Em Viagem
  ', 16),
( 'nrZGo', '2021-06-07', 'Praia da Vitoria', 'Guarda', 'Em Viagem
  ', 14),
( 'cwmM5', '2021-06-07', 'Nordeste', 'Braga', 'Em Processamento',
  4);
```

Listing 15.2: Povoamento da BD com informação sobre as encomendas

15.2 Ferramentas Utilizadas

15.2.1 *Framework* Blazor

Utilizando a *framework* Blazor, foi possível criar uma aplicação escrita em C# que corre no *browser*. Desta forma, aplicação desenvolvida é composta, na sua maioria, pelas seguintes linguagens: C#, CSS e HTML-based *razor syntax*.

15.2.2 Visual Studio

O Visual Studio 2019, sendo um poderoso IDE, funciona perfeitamente com a *framework* escolhida uma vez que reúne um conjunto de características e ferramentas que proporcionam um enorme apoio ao desenvolvimento de *software*. Além disso, o facto de incluir extensões que permitem a relação com a ferramenta SQL Server também foi uma enorme ajuda.

15.2.3 SQL Server

O SQL Server é um sistema de bases de dados relacional construído pela Microsoft que permite o armazenamento e a gestão de dados bem como o fornecimento de dados caso estes sejam requisitados por outro *software*. É um SGBD rápido e eficaz e, por isso, faz todo o sentido a sua utilização.

15.2.4 Outras ferramentas

De mencionar, foram também utilizadas as seguintes ferramentas:

- **Bing Maps:** Visualização interativa de mapas e imagens de satélite;
- **TrackingMore:** Informação relativa ao estado de encomendas.

15.3 Interface

15.3.1 Página Inicial

Ao inicializar o sistema, a página inicial é a seguinte.

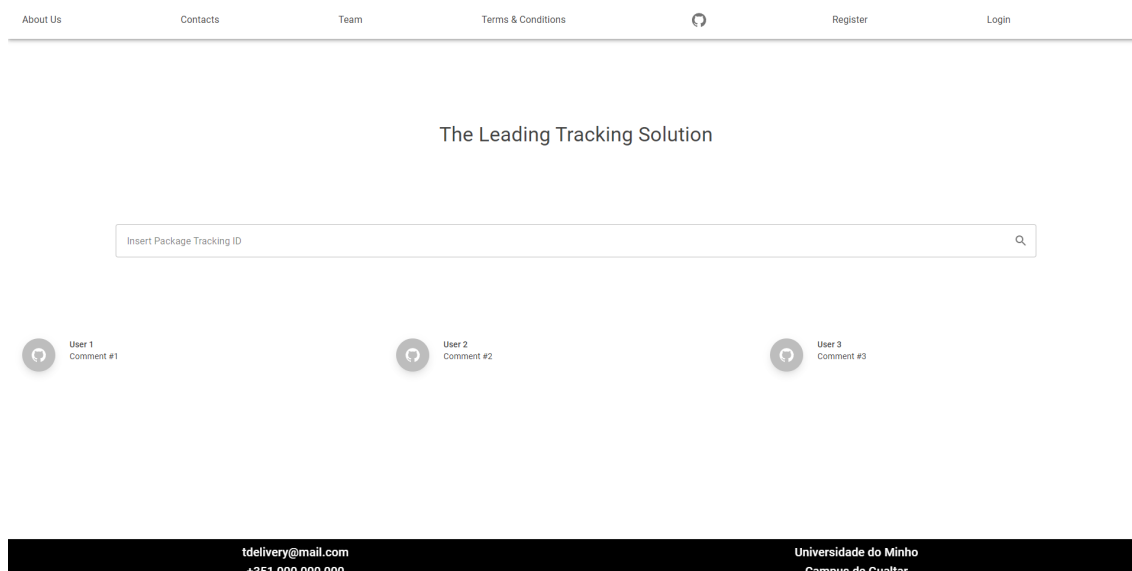


Figura 15.2: Página inicial do sistema

15.3.2 Registo e Autenticação

A página inicial da aplicação é a mesma tanto para um utilizador como para o administrador. Aqui, um novo cliente pode registar-se, ou, caso já se encontrem registados, o administrador ou um cliente podem autenticar-se.

Contrariamente ao estabelecido na fase de modelação, optou-se por permitir que um utilizador não autenticado consulte o estado de uma encomenda dado o seu *tracking number*. Optou-se por esta implementação uma vez que, caso se trate de alguém que raramente faz compras *online* e que, portanto, as perspetivas de voltar a utilizar a aplicação são diminutas, estar-se-ia a "desperdiçar" espaço na base de dados com informações relativas a um utilizador que utilizaria a aplicação uma única vez.

No que diz respeito à autenticação, uma vez que encriptar as *passwords* recorrendo única e exclusivamente a funções de *hash* criptográficas é considerado uma má prática, optou-se por utilizar as funcionalidades *built-in* da *framework* Blazor para este fim. Aliás, tome-se como o exemplo o algoritmo de sintetização de mensagem MD5, que já não se considera adequado

para fins criptográficos[5] uma vez que sofre de extensas vulnerabilidades¹. Caso se tivesse optado guardar as *passwords* dos utilizadores recorrendo a este algoritmo, ou até mesmo a algoritmos de *hashing* considerados adequados, como por exemplo o SHA-1[6], a segurança dos dados dos utilizadores estaria comprometida, uma vez que aplicar uma função de *hash* às *passwords* não é um método suficientemente seguro.

Desta forma, o esquema da Base de Dados acabou por ser um pouco diferente do projetado inicialmente.

TDelivery Register Login

Register

Create a new account.

Email

Password

Confirm password

Register

Use another service to register.

There are no external authentication services configured. See [this article about setting up this ASP.NET application to support logging in via external services.](#)

Figura 15.3: Página de registo de um utilizador

¹Um requisito básico de uma função de *hash* criptográfica é que seja computacionalmente inviável encontrar duas mensagens distintas com o mesmo valor de *hash*. O MD5 falha nesse requisito – *collision attacks* realizados em 2005 mostram que já não é adequado para fins criptográficos. Ainda assim, este permanece adequado para outros fins não criptográficos, como por exemplo a verificação da integridade de dados, no entanto apenas para corrupção não intencional.

TDelivery Register Login

Log in

Use a local account to log in.

Email

Password

☐ Remember me?

[Log in](#)

[Forgot your password?](#)

[Register as a new user](#)

[Resend email confirmation](#)

Use another service to log in.

There are no external authentication services configured. See this article about setting up this ASP.NET application to support logging in via external services.

© 2021 - TDelivery

Figura 15.4: Página de autenticação de um utilizador

15.3.3 Consultar & Editar Perfil

O utilizador da aplicação é ainda capaz de consultar os dados relativos ao seu perfil na plataforma e, se assim desejar, alterá-los.

TDelivery Hello 1234@gmail.com! Logout

Manage your account

Change your account settings

- Profile**
- Email
- Password
- Two-factor authentication
- Personal data

Profile

Username

Phone number

[Save](#)

© 2021 - TDelivery

Figura 15.5: Página de consultar e editar perfil

15.3.4 Contactos

O utilizador é ainda capaz de consultar o contacto de todas as empresas de entregas registadas na aplicação.


About Us	Contacts	Team	Terms & Conditions		Register	Login
Courier Contacts						
Courier	Type	Phone				
DHL Express	express					
UPS	express	+1 800 742 5877				
Fedex	express	1 800 247 4747				
TNT	express	1 800 558 5555				
China EMS	globalpost	0086-11183				
China Post	globalpost	86 20 11185				
Singapore Post	globalpost	0065 - 6841 2000				
Singapore Speedpost	globalpost	(+65) 6222 5777				
Hong Kong Post	globalpost	852 2921 2222				
Swiss Post	globalpost	0848 888 888				
USPS	globalpost	1-800-275-6777				
UK Royal Mail	globalpost	44 1752387112				
Parcel Force	globalpost	0344 800 4466				
PostNL International	globalpost	+31 88 868 6161				
Netherlands Post	globalpost					
tdelivery@mail.com +351 000 000 000			Universidade do Minho Campus de Gualtar			

Figura 15.6: Página com contactos das empresas que efetuam entregas

15.3.5 Consulta de encomenda

Quando um utilizador consulta o estado de uma encomenda, é-lhe apresentado um conjunto de informações relativas à mesma, que inclui:

- *Tracking number* da encomenda;
- Estado da encomenda – *entregue* ou *em trânsito*;
- Data em que a encomenda foi realizada;
- Entregador responsável pela encomenda.

Ainda assim, algumas respostas da API não devolvem qualquer informação relativa a localização da encomenda, apenas devolvem o seu estado – *em trânsito* ou *entregue*. Nestes casos, representou-se no mapa uma hipotética localização da encomenda considerando *default values*.

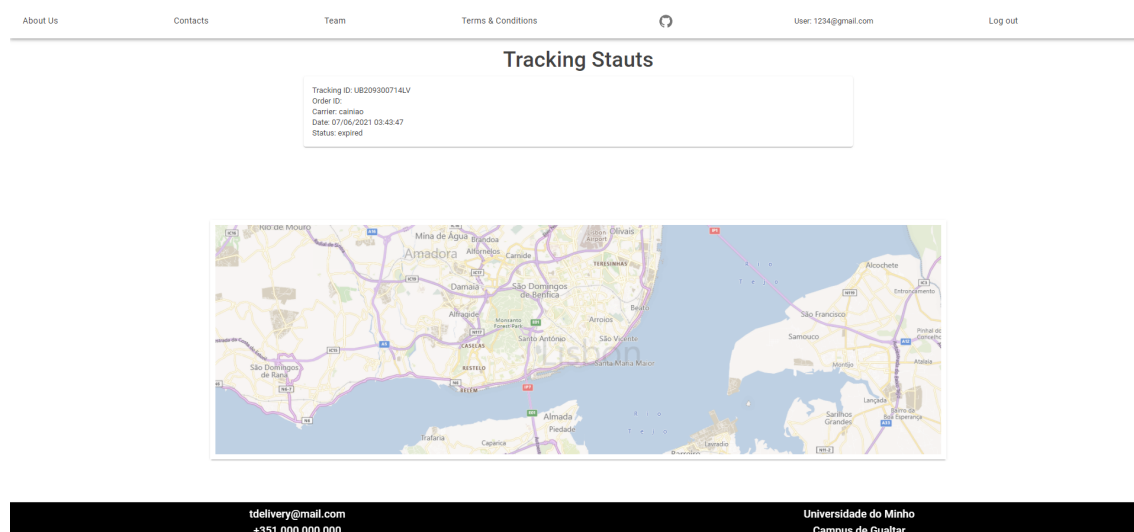


Figura 15.7: Página com as informações de uma encomenda

15.3.6 Sobre Nós

Esta é uma página que explica sucintamente quais os serviços prestados pela empresa.

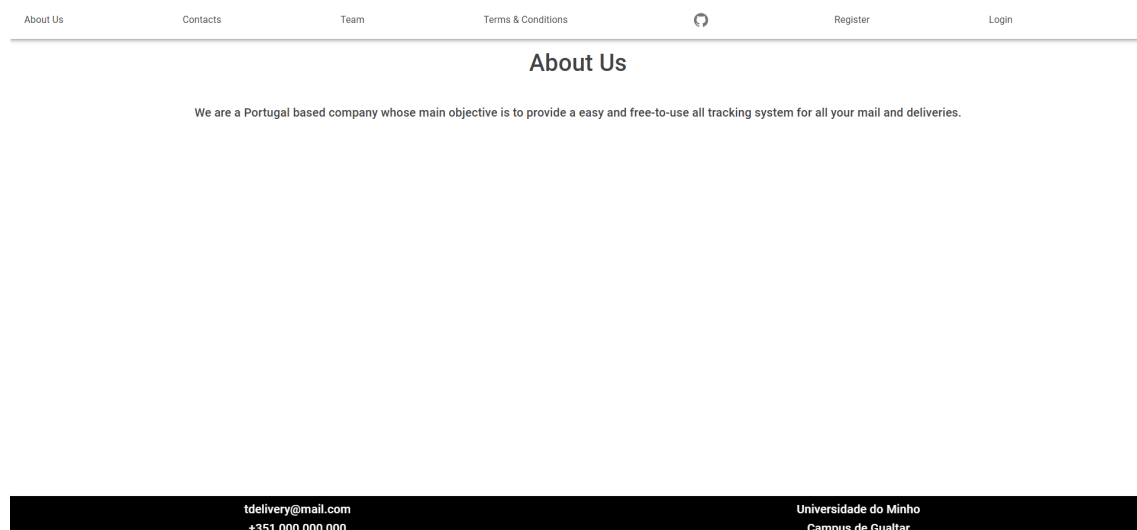


Figura 15.8: Página sobre nós

15.3.7 *Dashboard* do Administrador

O sistema desenvolvido é capaz de apresentar ao administrador uma *dashboard* com estatísticas relativas aos utilizadores da aplicação e respetivas encomendas. Uma vez mais, tendo em consideração que a API escolhida não fornece informação suficiente, alguma da informação apresentada é fictícia.

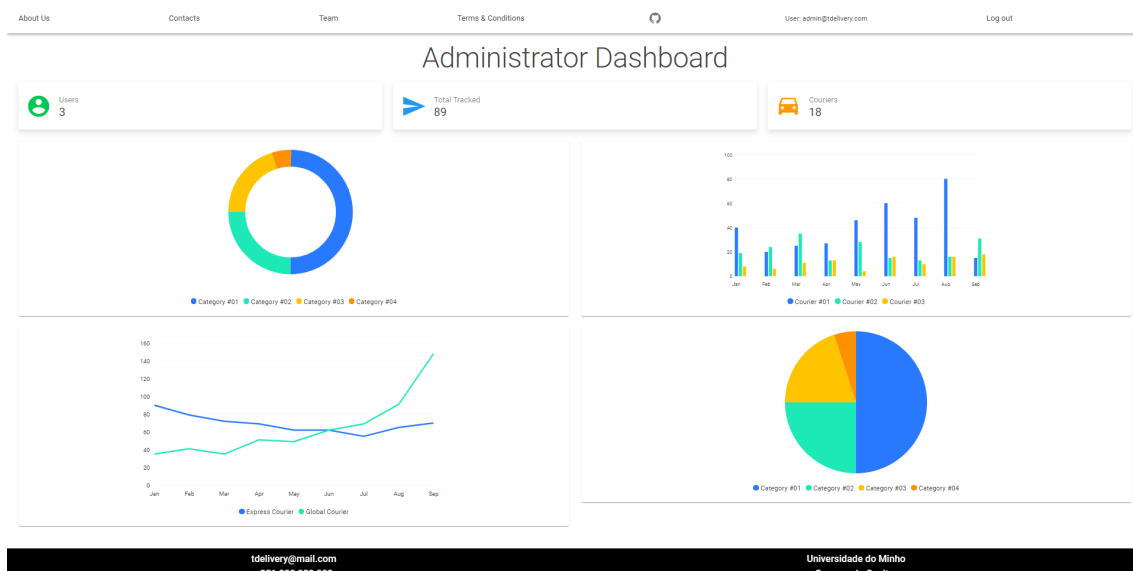


Figura 15.9: Página da *dashboard* do administrador

15.3.8 Equipa

Nesta página, apresenta-se a equipa foi responsável pelo desenvolvimento e modelação de todo o sistema.

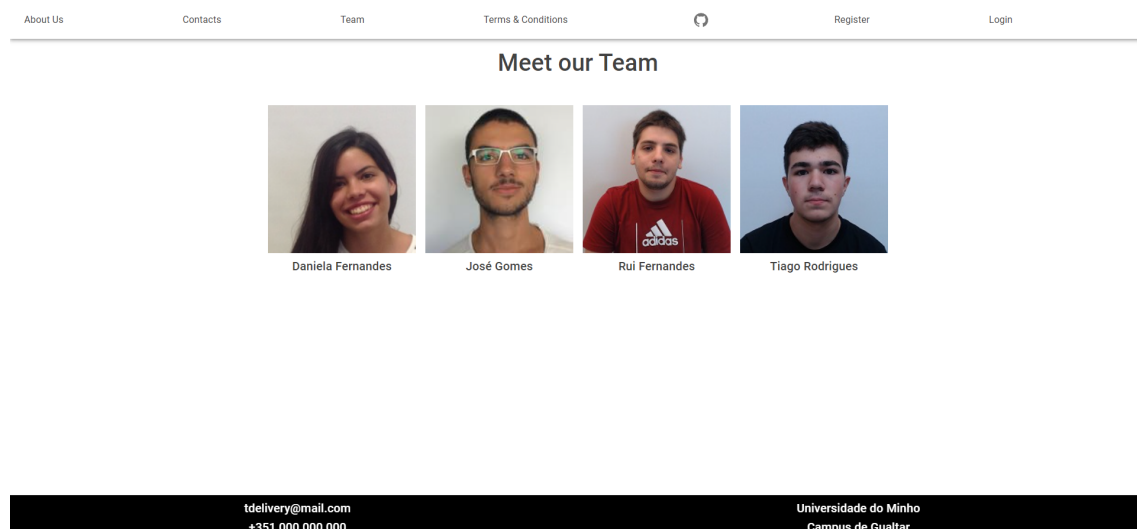


Figura 15.10: Página da equipa

15.3.9 Termos e Condições

Nesta página apresentam-se os termos e condições que estabelecemos para a utilização do nosso sistema.

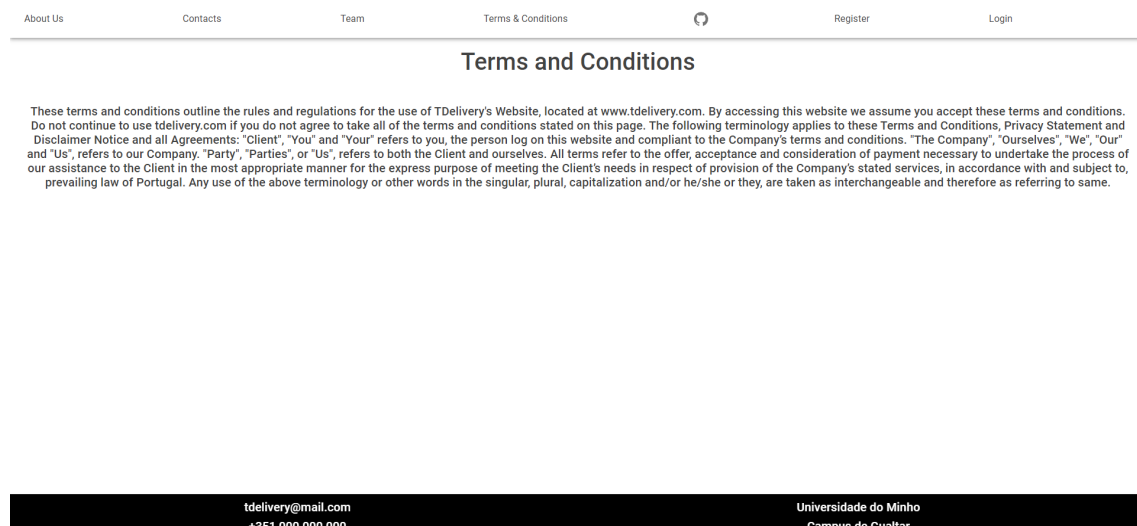


Figura 15.11: Página termos e condições

15.3.10 Término de Sessão

Enquanto está autenticado, o utilizador pode pretender terminar sessão na plataforma. Para isto, basta clicar no botão respetivo da barra de navegação e, imediatamente, será redirecionado para a página inicial da empresa.

15.4 Tempo & Custo

O plano de desenvolvimento elaborado na primeira parte deste trabalho prático foi cumprido integralmente, pelo que não houve atrasos na entrega do *software* encomendado pelo cliente. Relativamente ao custo do projeto, para além do custo fixo total do *software* utilizado no desenvolvimento do projeto, há um custo anual que terá de ser suportado pelo cliente, relativamente a um servidor onde o *website* estará alojado e de um domínio. No entanto este custo é residual para uma empresa como a TDelivery.

15.5 Desenvolvimento Futuro

Tal como todos os sistemas de *software*, sendo que este trabalho não é uma exceção, há sempre novas funcionalidades que podem ser acrescentadas para tornar a aplicação ainda mais completa. Caso a aplicação desenvolvida seja um sucesso nesta primeira fase de desenvolvimento, as atualizações que lhe seriam feitas, são de seguida apresentadas, por ordem de prioridade:

- Implementação de notificações, categorização de encomendas e favoritos;
- Encriptação dos dados entre cliente e servidor, usando um mecanismo de chave pública e chave privada.
- Implementação de suporte a publicidade na aplicação;
- Suporte a várias línguas, quando a aplicação atingir nível internacional.

16 Conclusões e Trabalho Futuro

Com a elaboração deste relatório, e com base no que está planeado ser feito nas próximas fases do projeto, podemos notar a importância desta abordagem inicial e daquilo que definimos ser relevante para o projeto final. Depois de estruturar todo o problema com o auxílio de ferramentas disponibilizadas pela *Microsoft*, pretendemos ainda criar todos os mecanismos que permitam dar resposta ao que a aplicação se propõe. Montou-se, ainda, uma ideia muito geral, daquilo que será o sistema em termos de funcionalidades e respostas ao cliente, usando como suporte uma maquete que aborda a lógica do problema.

Na segunda etapa de especificação, procedeu-se à modelação do sistema assente nos princípios estabelecidos na fase anterior. Elaboraram-se alguns diagramas UML considerados relevantes para a compreensão do sistema a ser desenvolvido, estruturou-se a base de dados sólida e sem redundância de dados e esboçaram-se *mockups*, idealizando a interface e camadas de apresentação do sistema.

Na terceira e última etapa do projecto, debruçamo-nos sobre a produção do *software* em si. Tendo por base todo o trabalho desenvolvido referente à modelação, foi criada uma aplicação *web* que cumpre as diretrizes mais relevantes para os futuros utilizadores. Ainda assim, a maior dificuldade sentida assentou no uso de novas tecnologias com as quais este nunca estivemos em contacto, barreira essa que foi rapidamente ultrapassada aquando da realização de intensivas pesquisas acerca do projeto em mãos. Desta forma, foi realmente enriquecedor estar em contacto com tecnologias como a *framework* Blazor, Microsoft Azure, Docker, API's com as quais nunca antes tínhamos trabalhado – TrackingMore e Bing Maps –, e ainda ganhar confiança com uma nova linguagem, o C#.

Como aspetos a melhorar considera-se de especial relevância a consistência na elaboração do trabalho, direcionada à divisão equitativa das diferentes tarefas presentes em cada fase, com o objetivo final de estabelecer um equilíbrio a nível de quantidade a realizar em cada parte e momento.

Referências

- [1] Roger S. Pressman. *Software Engineering: A Practitioner's Approach*. McGraw-Hill, 5th edition, 2000.
- [2] Instituto Nacional de Estatística. Inquérito à Utilização de Tecnologias da Informação e da Comunicação pelas Famílias, 2020.
- [3] Instituto Nacional de Estatística. Quase 1/5 do volume de negócios das empresas com 10 ou mais trabalhadores corresponde a comércio eletrónico, 2019.
- [4] Barbee Davis. *Agile practices for Waterfall Projects: Shifting Processes for Competitive Advantage*. J. Ross Pub., Plantation, FL, 2013.
- [5] Xiaoyun Wang and Hongbo Yu. How to Break MD5 and Other Hash Functions. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, pages 19–35, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.
- [6] How Not to Store Passwords: SHA-1 Fails Again. <https://securityintelligence.com/how-not-to-store-passwords-sha-1-fails-again/>. (Acedido a 20/05/2021).

Lista de Siglas e Acrónimos

API *Application Programming Interface*

BD Base de Dados

CSS *Cascading Style Sheets*

DAO *Data Access Objects*

ER Entidade-Relacionamento

HTML *Hypertext Markup Language*

ID Identificador

IDE *Integrated Development Environment*

MB *Megabyte*

MD *Message Digest*

ML Modelo Lógico

ORM *Object-relational Mapping*

RUP *Rational Unified Process*

SGBD Sistema de Gestão de Bases de Dados

SHA *Secure Hash Algorithm*

SMS *Short Message Service*

UML *Unified Modeling Language*