

UNIVERSIDADE DO MINHO  
MIEI

ARQUITECTURAS DE SOFTWARE

# ESS Trading Platform

**Grupo:**

Luís Braga (a82088)

Luís Martins (a82298)

Braga, Portugal  
30 de Novembro de 2019

# Resumo

O relatório surge no contexto da progressão dos planos e objectivos da **Easy & Safe Software Limited (ESS Ltd.)** em estabelecer-se como corretora no mercado financeiro global. Após um longo processo de reflexão, discussão e análise a ESS Ltd. decidiu utilizar como meio para as suas novas metas a criação de um produto que permitisse a negociação de *Contracts For Differences*. Este produto será idealizado na forma de uma plataforma que permite a a criação de contratos sobre ativos financeiros. Surgiu, portanto, a *ESS Trading Platform*, uma plataforma de *trading* de CDFs nos mercados financeiros.

Ao longo do relatório, serão expostos os passos que a equipa de trabalho teve de efectuar no que diz respeito ao processo de desenvolvimento, dando ênfase acrescida à arquitectura do sistema. Numa primeira fase, são apresentados no decorrer do presente relatório a análise de requisitos e a sua categorização como funcionais ou qualitativos, a modelação e compreensão do domínio da compra e venda de CDFs nos mercados financeiros, a análise do domínio de forma a perceber e entender as funcionalidades a implementar, e a exploração e especificação dessas mesmas funcionalidades. Tendo todo este processo como finalidade o desenho de uma peça de software foram também realizados os diagramas essenciais que traduzem a estrutura e o comportamento.

**Área de aplicação:** Especificação e desenvolvimento de software

**Palavras-chave:** Trading, plataforma, UML, Requisitos.

# Índice

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Análise de Requisitos</b>	<b>1</b>
2.1	Atributos de Qualidade . . . . .	1
2.2	Requisitos Funcionais . . . . .	3
<b>3</b>	<b>Modelação de Domínio</b>	<b>8</b>
3.1	Análise do diagrama . . . . .	8
3.2	Entidades Relevantes . . . . .	9
<b>4</b>	<b>Use Cases</b>	<b>10</b>
4.1	Diagrama Geral . . . . .	10
4.1.1	Actores . . . . .	11
4.2	Especificação de Use Case . . . . .	11
4.2.1	Registar Utilizador . . . . .	11
4.2.2	Entrar na Conta . . . . .	12
4.2.3	Alterar Dados Pessoais . . . . .	13
4.2.4	Carregar dinheiro para a Conta . . . . .	14
4.2.5	Consultar activos disponíveis . . . . .	14
4.2.6	Realizar um contrato sobre um activo . . . . .	15
4.2.7	Consultar o portfólio de contratos . . . . .	16
4.2.8	Consultar um contrato do seu portfólio . . . . .	16
4.2.9	Alterar Dados do Contrato . . . . .	17
4.2.10	Encerrar Contrato . . . . .	17
4.2.11	Consultar Histórico de Contratos . . . . .	18
4.2.12	Consultar Contrato Encerrado do histórico . . . . .	18
4.2.13	Levantar Dinheiro . . . . .	19
<b>5</b>	<b>Diagrama de Packages</b>	<b>20</b>
<b>6</b>	<b>Diagrama de Classes</b>	<b>23</b>
6.1	Escolha dos Design Patterns . . . . .	25
6.1.1	Pattern - Facade . . . . .	25
6.1.2	Pattern - DAO . . . . .	25

6.1.3	Pattern - Singleton . . . . .	25
6.1.4	Pattern - Factory . . . . .	25
6.1.5	Pattern - Strategy . . . . .	26
6.2	Outros pormenores . . . . .	26
6.3	Novo requisito . . . . .	26
<b>7</b>	<b>Diagramas de Máquina de Estados</b>	<b>28</b>
7.1	Máquina de Estado Login . . . . .	29
7.2	Máquina de Estado do Menu do Utilizador . . . . .	30
<b>8</b>	<b>Base de Dados</b>	<b>32</b>
<b>9</b>	<b>Interface</b>	<b>33</b>
9.1	Entrada na Plataforma . . . . .	33
9.2	Login . . . . .	34
9.3	Registar . . . . .	34
9.4	Menu Principal . . . . .	35
9.5	Alterar Dados Pessoais . . . . .	35
9.6	Histórico . . . . .	36
9.7	Consulta de Contrato Fechado . . . . .	37
9.8	Consulta de Activos Disponíveis . . . . .	38
9.9	Realização de um Contrato . . . . .	39
9.10	Consulta do Portfólio . . . . .	40
9.11	Consulta de um Contrato do Portfólio . . . . .	41
9.12	Depositar Fundos . . . . .	42
9.13	Levantar Fundos . . . . .	42
<b>10</b>	<b>Conclusão e Trabalho Futuro</b>	<b>43</b>
<b>11</b>	<b>Anexo</b>	<b>44</b>
11.0.1	Antigas Interfaces . . . . .	44
11.1	Entrada na Plataforma . . . . .	44
11.2	Login . . . . .	45
11.3	Registar . . . . .	46
11.4	Menu Principal . . . . .	48
11.5	Alterar Perfil . . . . .	49
11.6	Histórico . . . . .	50
11.7	Consulta de Contrato Encerrado . . . . .	51
11.8	Consulta de Activos Disponíveis . . . . .	51
11.9	Realização de um Contrato . . . . .	52
11.10	Consulta do Portfólio . . . . .	55
11.11	Consulta de um Contrato do Portfólio . . . . .	56

# 1. Introdução

A empresa **Easy & Safe Software Ltd. (ESS Ltd.)**, pretende alargar ainda mais a sua área de negócio e, para tal, depois de várias reuniões, decidiu enveredar pela negociação de *Contract For Differences (CFD)*, uma vez que possui alta capacidade para competir no mercado actual.

De forma a estabelecer-se como corretora no mercado financeiro global a ESS Ltd. pretende criar um produto para a negociação de CFDs. Tal produto traduzir-se-ia numa plataforma que permitisse a compra e venda de CFDs.

De forma a alcançar o objectivo principal foi proposto que o sistema deverá ser responsável por manter os valores de activos a serem negociados via CFD, permitir a abertura de contas de utilizador, que possuirão um *plafond* inicial e permitir que utilizadores abram posições de compra/venda sobre os activos disponíveis. Para posições abertas deverá ser possível definir limites de perda e ganho mediante os quais o contrato deverá ser fechado, ou seja, limites de *Take Profit* e *Stop Loss*. O utilizador também deverá poder fechar um contrato manualmente e monitorizar em tempo real o valor de cada activo adquirido e presente no portfólio.

Ao longo deste relatório será apresentado tudo o que foi realizado no sentido de produzir esta peça de software.

## 2. Análise de Requisitos

### 2.1 Atributos de Qualidade

Nesta secção iremos descriminar os atributos de qualidade da plataforma.

#### Persistência

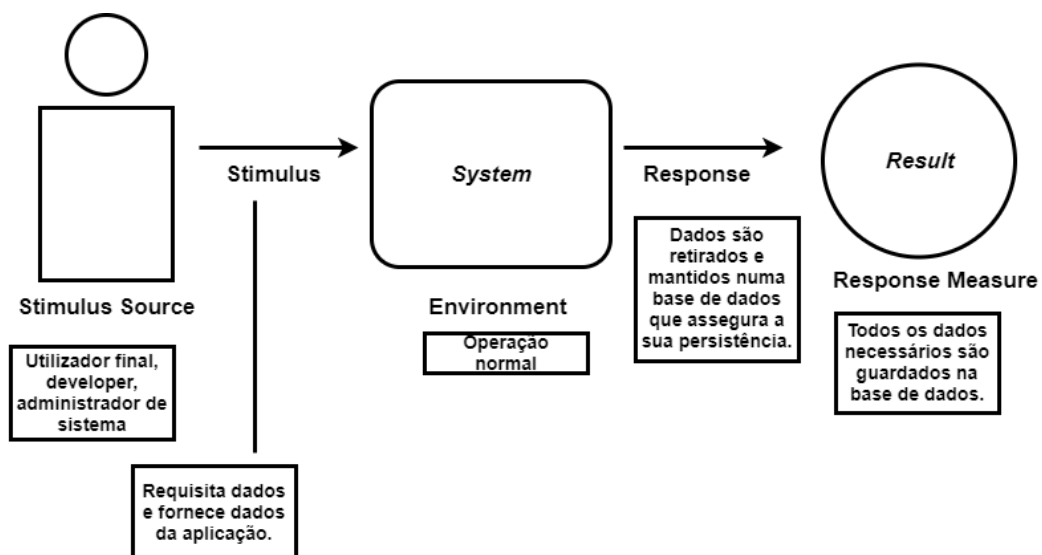


Figura 2.1: Cenário para atributo de qualidade persistência.

#### Reusabilidade

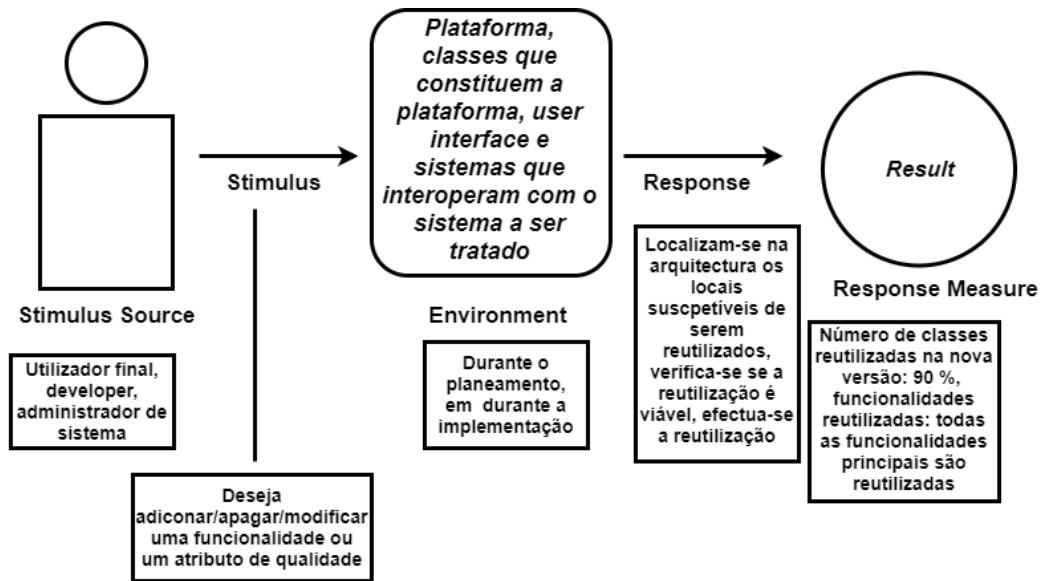


Figura 2.2: Cenário para atributo de qualidade Reusabilidade.

## Modificabilidade

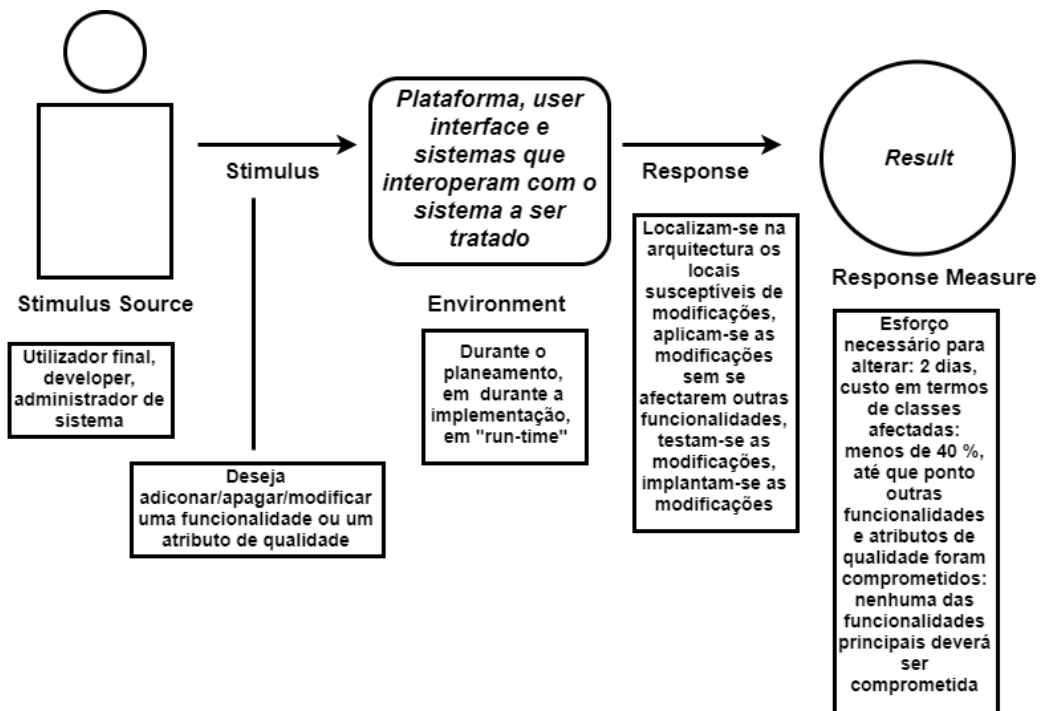


Figura 2.3: Cenário para atributo de qualidade modificabilidade.

A fim de implementar a solução para o problema proposto, o grupo de trabalho decidiu implementar o sistema usando um padrão arquitectural designado por Arquitectura por

Camadas. A Arquitectura por camadas utilizada pode ser representada por camadas horizontais que se sobrepõe umas às outras. Cada camada têm a sua responsabilidade, responsabilidade esta que é dividida internamente. A camada superior usufrui dos serviços da adjacente camada inferior e não tem conhecimento das camadas inferiores.

A modificabilidade é mais elevada neste tipo de arquitecturas, pois a organização por camadas permite alterações cirúrgicas, que aliada aos design patterns torna o sistema cada vez mais modificável.

A reusabilidade também é um dos apanágios da arquitectura por camadas, porque quando é alterada uma camada, especialmente uma camada superior, as inferiores não sofrem alterações.

A persistência assegura-se através da camada da base de dados.

## 2.2 Requisitos Funcionais

De acordo com a interpretação do grupo de trabalho relativamente ao que foi fornecido pela empresa, numa primeira fase há um usufruidor da aplicação: o utilizador.

O utilizador deverá:

1. Conseguir criar uma conta;
2. Ter acesso à sua conta;
3. Alterar os seus dados;
4. Carregar dinheiro para a sua conta;
5. Consultar activos disponíveis;
6. Realizar contratos sobre um activo;
7. Aquando da realização de um contrato:
  - Definir limite de *Take Profit*;
  - Definir limite de *Stop Loss*.
8. Consultar o seu portfólio de contratos;
9. Consultar um contrato do seu portfólio;
  - Quando se acede às informações de um contrato é possível alterar o valor de *Take Profit* e *Stop Loss*.
10. Encerrar contratos;
11. Consultar histórico de contratos;
12. Consultar um contrato do seu histórico;
13. Levantar dinheiro;



De forma a melhor perceber as acções do sistema e a interacção entre a díade Utilizador-Sistema a equipa de trabalho desenvolveu as seguintes especificações das funcionalidades acima elencadas.

## **1. Registar Utilizador**

### **Especificação dos Requisitos de Utilizador**

1. O utilizador tem de se registar na plataforma para a poder utilizar

### **Especificação dos Requisitos de Sistema**

1. O sistema solicita os dados do cliente: username, password, email e NIF.
2. O sistema não pode permitir o registo de utilizadores com o mesmo email ou username.
3. O sistema tem de armazenar os dados do cliente na BD.
4. O sistema tem de atribuir um *plafond* inicial ao utilizador.

## **2. Realizar "login"**

### **Especificação dos Requisitos de Utilizador**

1. O utilizador tem de se autenticar para poder utilizar a aplicação.

### **Especificação dos Requisitos de Sistema**

1. O sistema tem de solicitar o username e a password, de modo a proceder com a autenticação.
2. O sistema tem de verifica a validade de tanto o username e a password, não sendo possível um utilizador autenticar-se com dados errados.

## **3. Alterar Dados Pessoais**

### **Especificação dos Requisitos de Utilizador**

1. O utilizador tem de poder editar informação relativa ao seu perfil.

### **Especificação dos Requisitos de Sistema**

1. O sistema pode permitir a alteração dos campos todos campos.
2. O sistema armazena estas alterações na BD.

## **4. Introduzir Dinheiro**

### **Especificação dos Requisitos de Utilizador**

1. Um utilizador deve ser capaz de introduzir dinheiro na sua conta.

### **Especificação dos Requisitos de Sistema**

1. O sistema realiza a transferência.
2. O sistema actualiza o saldo do utilizador.

## **5. Consultar activos disponíveis**

### **Especificação dos Requisitos de Utilizador**

1. O utilizador tem de conseguir consultar os activos disponíveis no mercado.

### **Especificação dos Requisitos de Sistema**

1. O sistema consulta através da API os valores dos activos disponíveis.
2. O sistema apresenta ao utilizador estas informações.

## **6. Realizar Contrato**

### **Especificação dos Requisitos de Utilizador**

1. O utilizador tem de conseguir realizar um *Contract For Differences*.

### **Especificação dos Requisitos de Sistema**

1. O sistema permite que sejam realizados contratos de compra/venda.
2. O sistema permite que o utilizador escolha leverage se assim o entender.
3. O sistema não permite que o utilizador realize contratos que custem mais do que o capital que possui.
4. O sistema não permite que o utilizador realize contratos sobre activos que o sistema não comercializa.
5. O sistema armazena o contrato numa BD.
6. O sistema adiciona o contrato à "pool" de contratos a ser comercializada.

## **7. Consultar Portfólio**

### **Especificação dos Requisitos de Utilizador**

1. O utilizador tem de poder consultar uma elencagem de contratos activos.

### **Especificação dos Requisitos de Sistema**

1. O sistema deverá restringir o acesso desta lista apenas ao utilizador em causa.

2. O sistema tem de poder mostrar informação relativa ao conjunto de contratos activos, referindo o activo a ser negociado, o tipo de contrato e a data de negociação.

## **8. Consultar Contrato Activo**

### **Especificação dos Requisitos de Utilizador**

1. O utilizador tem de poder consultar um conjunto de dados relativos a um contrato activo.

### **Especificação dos Requisitos de Sistema**

1. O sistema deverá restringir o acesso desta informação a apenas ao cliente em causa.
2. O sistema tem de poder mostrar informação em cada contrato referente ao activo, ao tipo de contrato (venda/compra), quantidade de activo negociada, valor do activo no início do contrato, valor actual do activo, valor gasto e balanço actual e data de começo do pacto, o leverage e os valores de *Take Profit* e *Stop Loss* utilizados.

## **9. Alterar Condições do Contrato**

### **Especificação dos Requisitos de Utilizador**

1. O utilizador tem de poder alterar os valores de *Take Profit* e *Stop Loss* dos seus contratos abertos.

### **Especificação dos Requisitos de Sistema**

1. O sistema verifica se o valor de *Stop Loss* é válido de forma a não permitir que em caso de perda o utilizador fique com saldo negativo.

## **10. Encerrar Contrato**

### **Especificação dos Requisitos de Utilizador**

1. O utilizador deve poder encerrar uma posição no mercado quando assim o desejar.

### **Especificação dos Requisitos de Sistema**

1. O sistema deve encerrar uma posição de determinado utilizador.
2. O sistema actualiza o saldo do utilizador findo o contrato.

## **11. Consultar Histórico**

### **Especificação dos Requisitos de Utilizador**

1. O utilizador tem de poder consultar uma elencagem de contratos encerrados.

## **Especificação dos Requisitos de Sistema**

1. O sistema deverá restringir o acesso desta lista apenas ao utilizador em causa.
2. O sistema tem de poder mostrar informação relativa ao conjunto de contratos activos, referindo o activo negociado, o tipo de contrato e a data de término.

## **12. Consultar Contrato Encerrado**

### **Especificação dos Requisitos de Utilizador**

1. O utilizador tem de poder consultar um conjunto de dados relativos aos contratos encerrados.

### **Especificação dos Requisitos de Sistema**

1. O sistema tem de poder apresentar ao utilizador os contratos já encerrados.
2. O sistema deverá restringir o acesso desta lista a apenas ao cliente em causa.
3. O sistema tem de poder mostrar informação em cada um desses contratos referente ao activo, ao tipo de contrato (venda/compra), quantidade de activo negociada, valor do activo no início do contrato, valor do activo no fim do contrato, valor gasto, balanço final do negócio, data do início do contrato e data de finalização, o leverage usado e os valores de Take Profit e Stop Loss utilizados.

## **13. Levantar Dinheiro**

### **Especificação dos Requisitos de Utilizador**

1. Um utilizador deve ser capaz de levantar dinheiro da sua conta.

### **Especificação dos Requisitos de Sistema**

1. O sistema soma o valor absoluto de lucro/prejuízo do utilizador.
2. O sistema verifica se o utilizador pretende levantar mais do que o valor mínimo permitido.

### 3. Modelação de Domínio

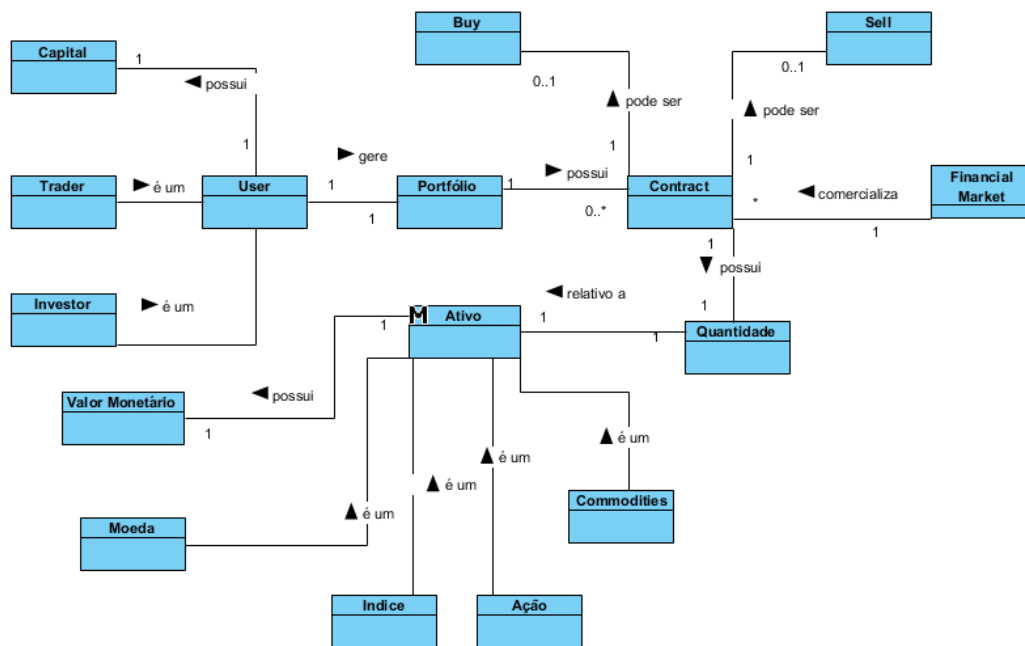


Figura 3.1: Modelo de Domínio

#### 3.1 Análise do diagrama

No sistema de existe apenas uma entidade que usufruirá do sistema: o utilizador. Um utilizador registado no sistema é classificado de duas maneiras consoante o seu comportamento, isto é, um *trader*, se se envolver em transacções mais frequentes e é apelidado de *investor* se realizar negociações a longo prazo, chegando a sustentar posições durante anos. Todos os utilizadores possuirão um saldo ou capital disponível para efectuar negociações. Os utilizadores também possuirão um conjunto de contratos activos, intitulado *portfólio*, podendo este estar vazio.

O utilizador é a entidade responsável pela realização de contratos. Consoante a sua deliberação sobre o mercado poderá realizar contratos de venda e compra, sendo que após

estipular as condições do seu contrato, este estará pronto a ser comercializado no mercado financeiro.

Um contrato, independentemente do seu tipo, negocia uma quantidade de activo financeiro. Esse activo financeiro possui um valor monetário dinâmico e variável temporalmente. No comércio financeiro existe um leque variado de activos, desde a sua área até à flutuação do valor. Exemplos desses activos são *Criptomoedas*, *Exchange-Traded Fund (ETF)*, *Acções*, *Índices*, *Matérias-primas* e *Moedas*.

## 3.2 Entidades Relevantes

O modelo de domínio inclui todas as entidades que são consideradas basilares para o funcionamento da aplicação. Contudo, de todas elas podem-se destacar algumas, tais como:

**Utilizador:** Todo o investimento deste projecto tem como objectivo obter o número máximo possível de utilizadores, pois isso traduzirá o sucesso da aplicação. Estes irão gerar o lucro da aplicação e sem eles a aplicação não funcionará.

**Contrato:** É em torno desta entidade que a aplicação gira; O principal objectivo da aplicação é negociar contratos.

**Activo:** Todos os contratos referenciam activos e são estes que iram ditar a o ganho ou perda de capital por parte do utilizador, logo é uma entidade crucial para o sistema.

## 4. Use Cases

### 4.1 Diagrama Geral

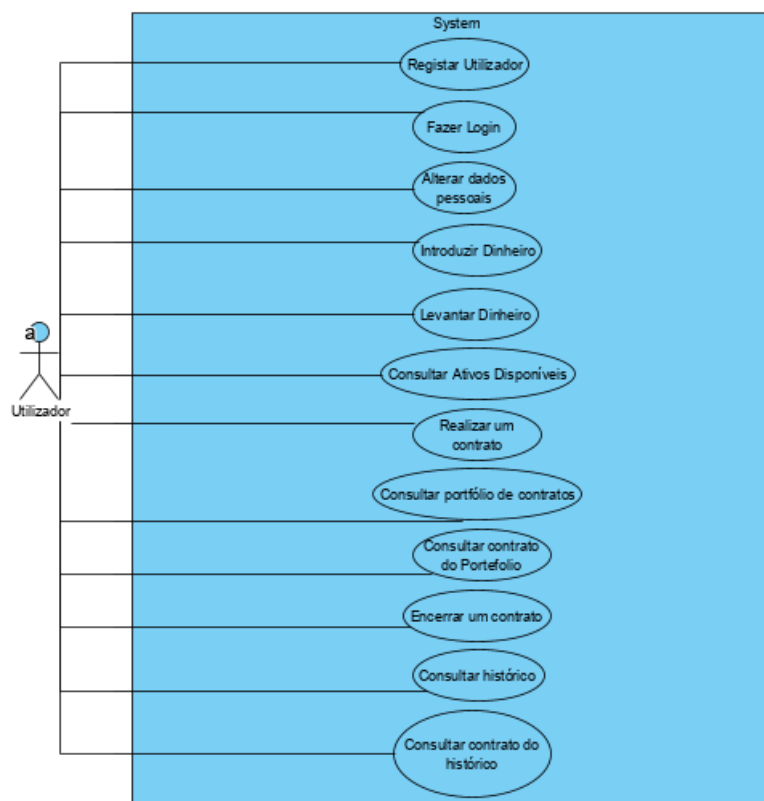


Figura 4.1: Diagrama de use cases.

No diagrama acima (Figura 4.1), encontram-se os use cases identificados, isto é, as principais funcionalidades que utilizador terá quando interagir com o sistema.

O utilizador poderá registar-se no sistema fornecendo os dados necessários, bem como depois de estar registado, fazer a autenticação no mesmo. Além disso, poderá ainda alterar os seus dados se, por ventura, assim o desejar.

No que diz respeito a um utilizador autenticado, ele poderá introduzir dinheiro que

ficará associado ao seu capital/saldo, bem como levantar desde que tenha dinheiro mínimo suficiente. Conseguirá também consultar os activos disponibilizados pela plataforma e, evidentemente, poderá realizar contratos de diferença (compra/venda) sobre estes mesmos activos. Por outro lado, será igualmente possível ao usuário consultar o portfólio (contratos realizados), bem como consultar um desses contratos e, seguidamente, encerrá-los. Assim que o contrato é encerrado passa a pertencer ao histórico e o utilizador poderá solicitá-lo, bem como outros contratos que já terá fechado a posição.

#### 4.1.1 Actores

**Utilizador:** Representa o actor que irá usufruir todas as funcionalidades fornecidas pelo sistema.

## 4.2 Especificação de Use Case

### 4.2.1 Registar Utilizador

Use Case:	Registar utilizador	
Actor:	Utilizador	
Pré condição:		
Pós condição:	O novo cliente é adicionado ao sistema	
Cenário Normal	Actor input	System response
		1. Apresenta os campos a preencher pelo utilizador.
	2. Preenche esses campos.	
		3. Verifica a validade dos dados.
		4. Verifica que o utilizador não existe.
Exception 1 [Dados inválidos] (passo 3)		5. Cria o cliente de acordo com os dados preenchidos.
		3.1. Avisa sobre a invalidade dos dados.
Exception 2 [Utilizador já existe]		4.1 Avisa acerca do cliente já existir.

Figura 4.2: Especificação de use case registar utilizador.

Tal como foi dito anteriormente para um utilizador poder usufruir e utilizar as funcionalidades da plataforma necessita de efectuar um registo, no qual terá de fornecer as seguintes informações:

- Nome;
- Username;



- Password;
- Email.

#### 4.2.2 Entrar na Conta

Use Case:	Fazer Login	
Actor:	Utilizador	
Pré condição:	Utilizador Registado	
Pós condição:	Utilizador Autenticado	
Cenário Normal	Actor input	System response
		1. Pede as credenciais do utilizador
	2. Coloca o seu username e password	3.Verifica que as credenciais estão corretas
Alternativo 1 [Credenciais estão erradas] (passo 3)		3.1 Avisa que as credenciais estão erradas
		3.2.Volta ao passo 1

Figura 4.3: Especificação de use case fazer login.

Um utilizador, estando registado, pode livremente entrar na sua conta desde que introduza as suas credenciais correctamente (username/email e password).

### 4.2.3 Alterar Dados Pessoais

Use Case:	Alterar dados pessoais	
Actor:	Utilizador	
Pré condição:	Utilizador autenticado	
Pós condição:	O dados do cliente são alterados	
Cenário Normal	Actor input	System response
		1. Apresenta os campos a preencher
	2.Preenche os campos	
		3. Valida os campos
Exception 1 [Campo inválido]		4. Altera os dados pessoais do cliente conforme os campos alterados
		3.1 Avisa acerca da invalidez do campo

Figura 4.4: Especificação de use case alterar dados pessoais.

Um utilizador também pode editar o seu perfil, isto é, pode alterar algumas das informações que fornece durante o processo de registo. Assim pode alterar os seguintes campos:

- NIF;
- Password;
- Email.

#### 4.2.4 Carregar dinheiro para a Conta

Use Case:	Introduzir dinheiro	
Actor:	Utilizador	
Pré condição:	Utilizador autenticado	
Pós condição:	Capital de investimento atualizado	
Cenário Normal	Actor input	System response
	1. Indica ao sistema que pretende introduzir dinheiro.	
		2. Pede ao utilizador dados necessários.
	3. Avança com o carregamento.	
	4. Confirma a ação	
		5. Verifica validade dos dados.
		6. Atualiza saldo.
Exceção 1 (Passo 2 e 4) [Utilizador pretende cancelar operação]	{2,4}.1. Cancela operação.	
Cenário Alternativo 1 (Passo 5) [Dados inválidos]		5.1. Informa cliente sobre a invalidade dos dados inseridos.
		5.2. Volta ao passo 2.

Figura 4.5: Especificação de use case introduzir dinheiro.

Um utilizador pode carregar dinheiro para sua conta de forma a conseguir a investir no mercado.

#### 4.2.5 Consultar activos disponíveis

Use Case:	Consultar activos disponíveis	
Actor:	Utilizador	
Pré condição:	Utilizador autenticado	
Pós condição:	Lista de ativos e respetivos preços apresentada	
Cenário Normal	Actor input	System response
	1. Pede para consultar os ativos disponíveis	
		2. Apresenta a lista de ativos com respetivos preços

Figura 4.6: Especificação de use case consultar activos disponíveis.

Inclusive, um utilizador poderá consultar os activos disponíveis e os seus valores mais actualizados de maneira a definir qual o tipo de negócio e sobre que activos irá pactuar.

#### 4.2.6 Realizar um contrato sobre um activo

Use Case:	Realizar um contrato	
Actor:	Utilizador	
Pré condição:	Utilizador autenticado && lista de ativos consultadas	
Pós condição:	Contrato realizado && adicionado ao portefólio	
Cenário Normal	Actor input	System response
	1. Seleciona o ativo da lista	
		2. Apresenta os campos a preencher
	3. Diz que deseja comprar	
	4. Escolhe uma opção de leverage disponibilizada	
	5. Indica o montante	
		6. Atualiza a quantidade correspondente
	7. Indica o Take Profit	
	8. Indica o Stop Loss	
		9. Verifica que os campos foram preenchidos corretamente
		10. Calcula e verifica que o saldo do utilizador é suficiente
	11. Confirma o contrato	
		12. Atualiza o saldo e envia para o portefólio do utilizador
Alternativo 1 [Utilizador deseja vender] Passo 3	3.1. Diz que deseja vender	
		3.2. Avança para o passo 4
Alternativo 2 [Saldo Insuficiente] Passo 10		8.1.1. Avisa que não tem saldo suficiente para tal montante
		8.1.2. Avisa qual o montante máximo a colocar
		8.1.3. Volta ao passo 4
Alternativo 3 [Stop Loss demasiado alto] Passo 10		8.2.1. Avisa que não tem saldo suficiente para tal Stop Loss
		8.2.2. Volta ao passo 7
Exceção 1 [Utilizador não deseja confirmar] Passo 11	10. Cancela a ação	

Figura 4.7: Especificação de use case realizar um contrato.

De referir que do passo 3 até ao 8, não será sequencial, apenas para se saber quais os campos que o utilizador terá que preencher.

Um utilizador pode fazer contratos sobre activos que estão disponíveis de momento. Aquando da realização de um contrato, o utilizador deverá indicar se pretende tirar partido de leverage, ou seja, qual o factor multiplicativo (x1 significa que não deseja leverage), qual o valor de monetário, a quantidade de activo, o valor de Take Profit e Stop Loss. Tudo isto deverá ser válido conforme o seu saldo e tendo em conta os Stop Loss's dos contratos abertos que o utilizador, eventualmente, já possua no seu portefólio.

#### 4.2.7 Consultar o portfólio de contratos

Use Case:	Consultar contrato do Portefólio	
Actor:	Utilizador	
Pré condição:	Utilizador autenticado	
Pós condição:	Apresenta o CFD aberto	
Cenário Normal	Actor input	System response
	1. Especifica contrato que pretende consultar.	
		2. Apresenta o contrato aberto.

Figura 4.8: Especificação de use case consultar portfólio.

De maneira a melhor poder controlar os seus investimentos o utilizador pode verificar e informar-se sobre quais os seus investimentos activos. Pode consultar as seguintes informações: activo a ser negociado, o tipo de contrato e o balanço atual do contrato.

#### 4.2.8 Consultar um contrato do seu portfólio

Use Case:	Consultar contrato do Portefólio	
Actor:	Utilizador	
Pré condição:	Utilizador autenticado	
Pós condição:	Apresenta o CFD aberto	
Cenário Normal	Actor input	System response
	1. Especifica contrato que pretende consultar.	
		2. Apresenta o contrato aberto.

Figura 4.9: Especificação de use case consultar um contrato do portfólio.

No seguimento do último use case especificado também será útil averiguar informações mais específicas sobre um contrato activo. Pode consultar as seguintes informações: activo, ao tipo de contrato (venda/compra), quantidade de activo negociada, valor do activo no início do contrato, valor actual do activo, valor gasto e balanço actual e data de começo do pacto.

### 4.2.9 Alterar Dados do Contrato

Use Case:	Alterar Dados do Contrato	
Actor:	Cliente	
Pré condição:	Cliente autenticado & contrato do Portfólio consultado	
Pós condição:	O dados do contrato são alterados	
Cenário Normal	Actor input	System response
	1. Diz que pretende alterar Take Profit ou Stop Loss.	
		2. Apresenta campo a preencher.
	3. Preenche os campos	
		4. Valida os campos 5. Altera os dados do contrato.
Exception 1 [Valor de Stop Loss inválido] (passo 4)		4.1 Avisa acerca da invalidez do campo

Figura 4.10: Especificação de use case alterar dados de um contrato.

Um utilizador deve poder modificar os dados dos contratos, nomeadamente, os valores de *Take Profit* e *Stop Loss*. O sistema deixa a alteração do *Take Profit* à consideração do cliente, contudo, o valor de *Stop Loss* deve ser validado de forma a não ser menor do que o valor que o utilizador tem em saldo.

### 4.2.10 Encerrar Contrato

Use Case:	Encerrar contrato	
Actor:	Utilizador	
Pré condição:	Utilizador autenticado && contrato do portfólio consultado	
Pós condição:	Contrato arquivado & contas normalizadas	
Cenário Normal	Actor input	System response
	1. Pede ao sistema que lhe encerre o contrato selecionado.	
		2. Atribuir ao contrato o valor de fecho.
		3. Calcular lucro total.
		4. Atualizar saldo da corretora.
		5. Atualizar saldo do utilizador.
		6. Arquivar contrato.

Figura 4.11: Especificação de use case encerrar contrato.

Depois de consultado um contrato do portfólio, o utilizador poderá encerrá-lo. Para tal, terá que pedir ao sistema e o sistema mostrará todas as informações relevantes, bem como actualizará as contas do utilizador e da corretora. Por último, o contrato irá para o histórico.

#### 4.2.11 Consultar Histórico de Contratos

Use Case:	Consultar histórico	
Actor:	Utilizador	
Pré condição:	Utilizador autenticado	
Pós condição:	Apresenta uma elencação dos CFD's fechados	
Cenário Normal	Actor input	System response
		1. Consulta contratos fechados do cliente.
		2. Apresenta o histórico de contratos encerrados.

Figura 4.12: Especificação de use case consultar histórico de contratos.

Um utilizador pode consultar o seu histórico de contratos, onde estão presentes os contratos encerrados. Destes pode ver informações gerais como: activo negociado, o tipo de contrato e a data de término.

#### 4.2.12 Consultar Contrato Encerrado do histórico

Use Case:	Consultar contrato do histórico	
Actor:	Utilizador	
Pré condiçã	Utilizador autenticado	
Pós condiçã	Apresenta o CFD fechado	
Cenário Normal	Actor input	System response
	1. Especifica contrato que pretende consultar.	
		2. Apresenta o contrato fechado.

Figura 4.13: Especificação de use case consultar um contrato do histórico.

Um utilizador pode consultar um único contrato da sua lista de contratos encerrados. Pode consultar as seguintes informações: activo, ao tipo de contrato (venda/compra),

quantidade de activo negociada, valor do activo no início do contrato, valor do activo no fim do contrato, valor gasto, balanço final do negócio, data do início do contrato e data de finalização.

#### 4.2.13 Levantar Dinheiro

Use Case:	Levantar Dinheiro	
Actor:	Utilizador	
Pré condição:	Utilizador autenticado	
Pós condição:	Dinheiro transferido para a conta do utilizador	
Cenário Normal	Actor input	System response
	1. Solicita o levantamento (transferência) de dinheiro	
		2. Verifica que já realizou, pelo menos, contratos com prejuízo/lucro (valor absoluto) de valor superior à plafond inicial
		3. Pede ao utilizador os dados necessários
	4. Preenche os dados	5. Verifica que os dados são válidos
		6. Pergunta ao utilizador se pretende avançar
	7. Confirma que pretende avançar	8. Transfere o dinheiro para a conta do utilizador
		9. Atualiza o saldo
Exception 1 [Não cumpriu os contratos necessários] Passo 3		2.1. Verifica que não realizou contratos necessários 2.2. Avisa o utilizador
Exception 2 [Não preenche os dados] Passo 4	4.1. Diz que não pretende preencher os dados	
Alternativo 1 [Dados Inválidos] Passo 5	5.2. Volta ao passo 3	5.1. Avisa o utilizador sobre a invalidade dos dados
Exception 2 [Não pretende avançar] Passo 7	7.1. Diz que não pretende avançar	

Figura 4.14: Especificação de use case levantar dinheiro.

O utilizador terá que solicitar o levantamento (transferência) de dinheiro e será possível desde que já tenha contratos no histórico, cuja a soma do valor absoluto do balanço desses contrato tem que ser superior ao plafond inicial atribuído. Depois os dados terão que ser válidos, ou seja, os dados da conta terão que estar correctos e também terá que ser uma transferência de, no mínimo, 10 euros.



## 5. Diagrama de Packages

A fim de implementar a solução para o problema proposto, o grupo de trabalho decidiu implementar o sistema usando um padrão arquitetural designado por Arquitetura por Camadas. A Arquitetura por camadas utilizada pode ser representada por camadas horizontais que se sobrepõe umas às outras. Cada camada têm a sua responsabilidade, responsabilidade esta que é dividida internamente. A camada superior usufrui dos serviços da adjacente camada inferior e não tem conhecimento das camadas inferiores.

Dado que seria complicado conseguir apresentar neste capítulo uma figura que retratasse as camadas com os respectivos packages e classes pertencentes a esses packages, o grupo decidiu dividir.

Na seguinte imagem, serão apresentadas as diferentes camadas do sistema, bem como respectivos packages.

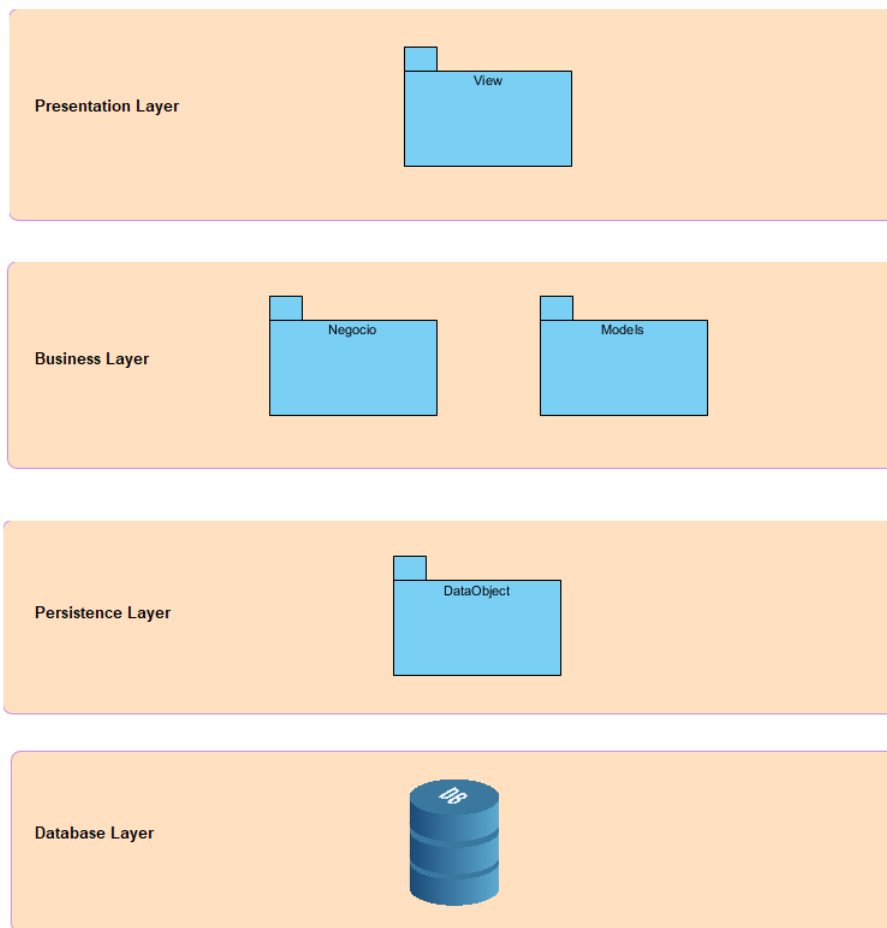


Figura 5.1: Camadas do projecto e respectivas

De forma a especificar com detalhe que classes constituem os packages elaborou-se o seguinte diagrama.

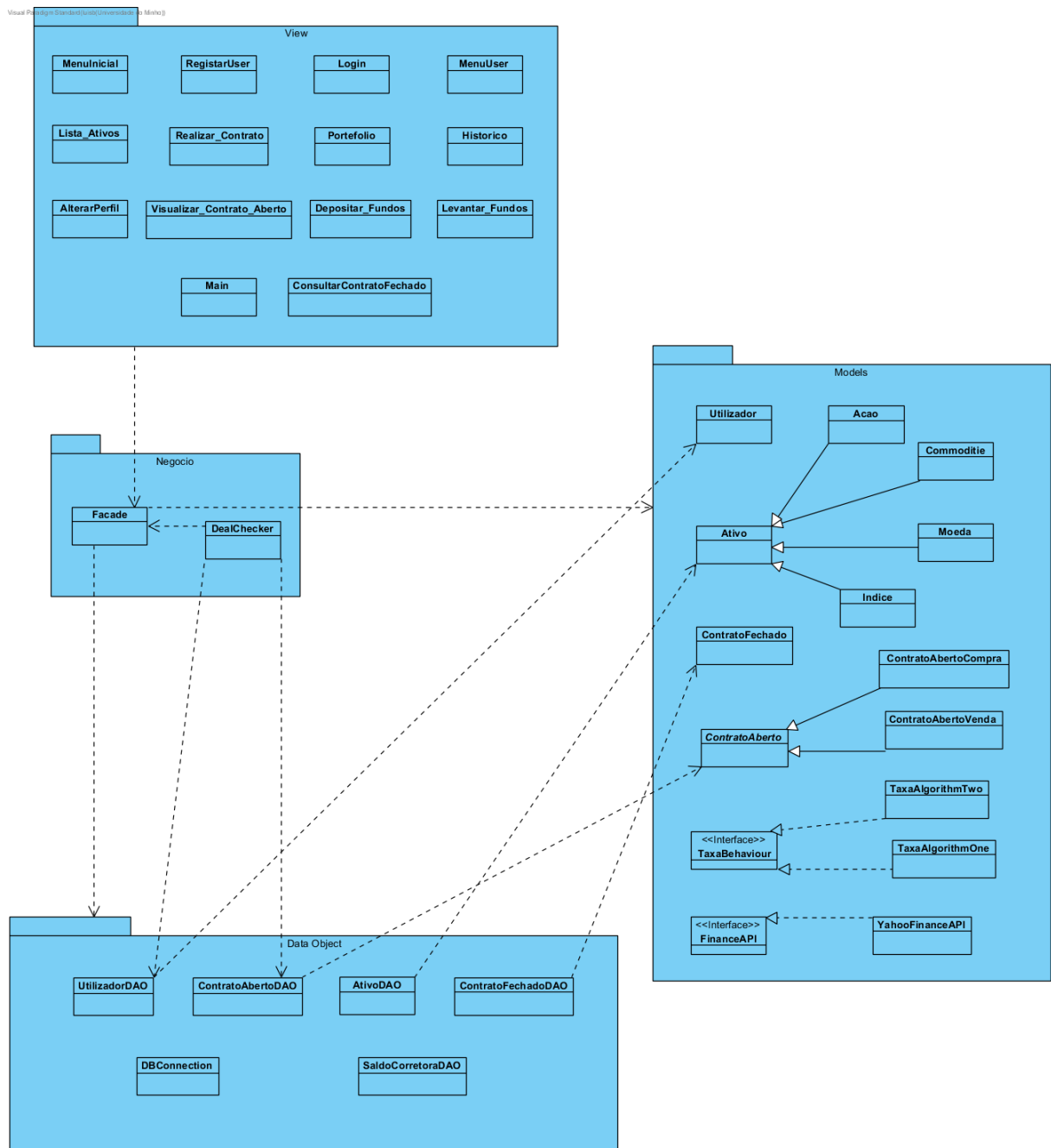


Figura 5.2: Packages e respectivas classes.

## 6. Diagrama de Classes

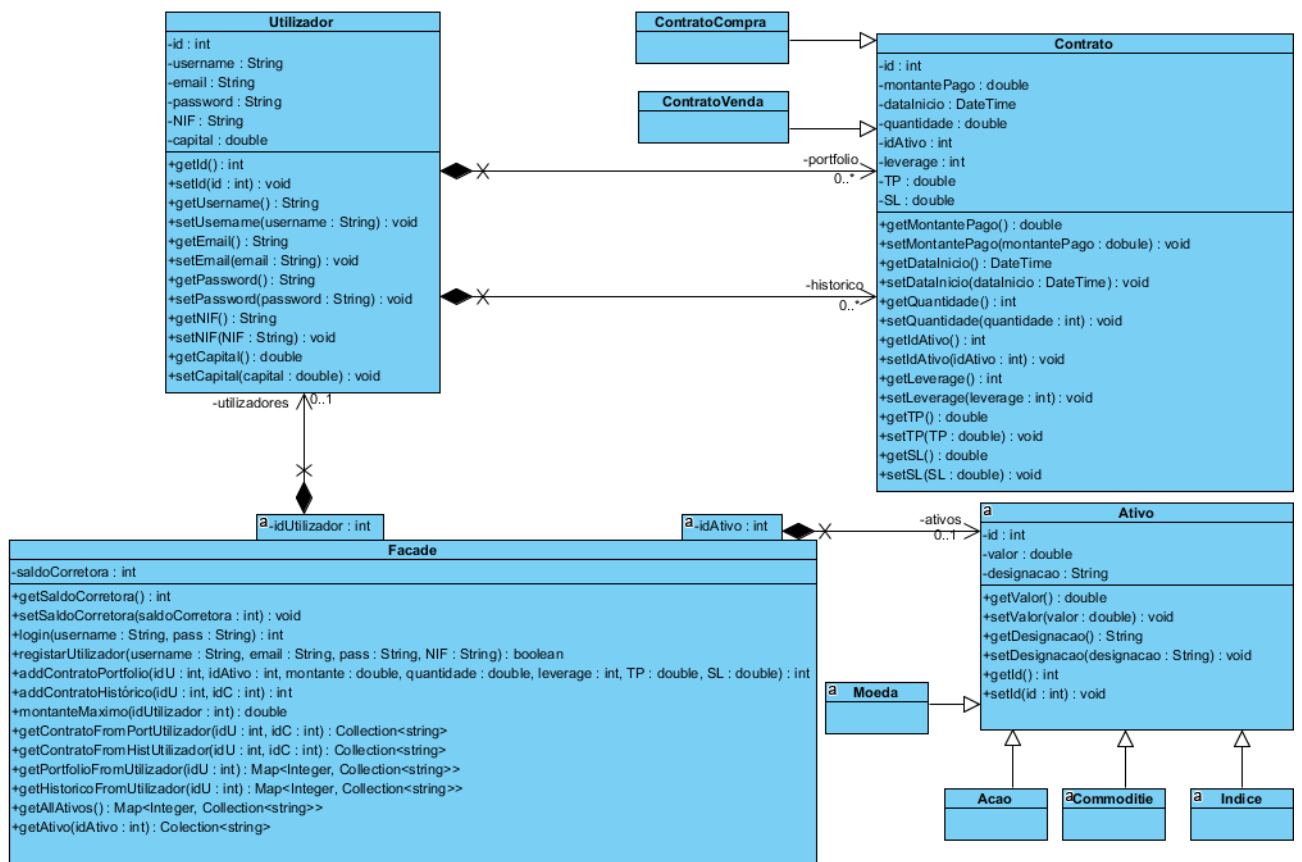


Figura 6.1: Primeiro Diagrama de Classes.

Numa fase inicial do trabalho, depois da análise de todos os diagramas elaborou-se o diagrama de classes apresentado acima.

Após deliberar sobre o uso de patterns e a sua adequação aos problemas e "stumbling blocks" encontrados, o grupo de trabalho concebeu o seguinte diagrama de classes.

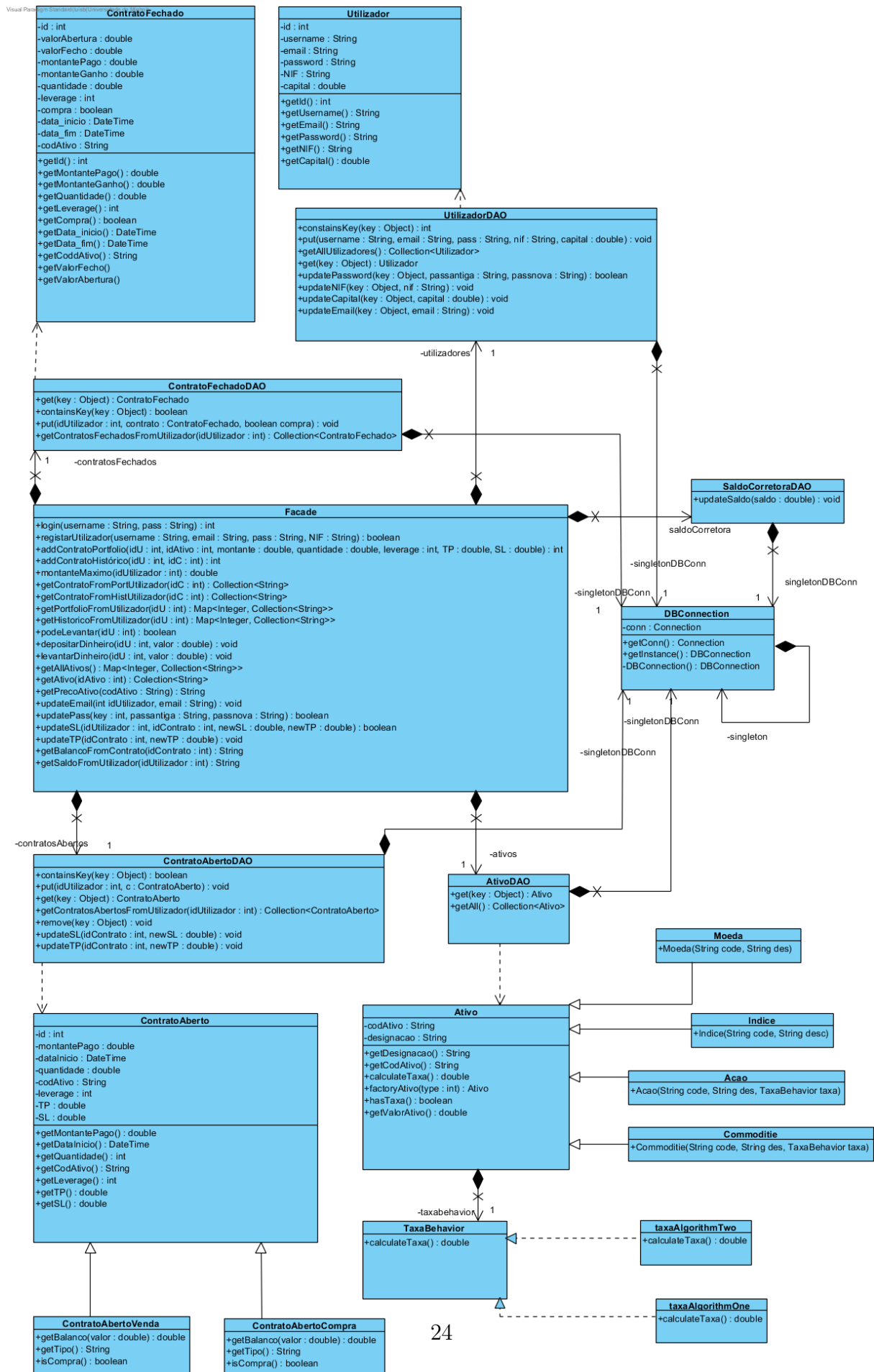


Figura 6.2: Diagrama de Classes final.

## 6.1 Escolha dos Design Patterns

### 6.1.1 Pattern - Facade

A escolha deste Pattern já tinha sido tomada no primeiro Diagrama de Classes e, assim, se manteve, na medida em que, é um bom mediador entre a Base de Dados e a interface do utilizador. Por outro lado, o Facade é capaz de apresentar ao utilizador o sistema de uma forma simplificada.

### 6.1.2 Pattern - DAO

De modo a garantir a persistência dos dados, usou-se este pattern, conjuntamente com uma Base de Dados relacional, cujo o modelo lógico será apresentado no capítulo 8. Surge também no sentido de garantir que caso o sistema possua um elevado número de dados, esses objetos não estejam guardados na memória.

Em primeiro lugar, é necessário identificar as classes que irão armazenar dados, pois essas são as que devem ser persistidas. O foco é nas associações, onde tem que se dar importância a tudo que sejam listas e map's armazenadas. Portanto, relativamente ao primeiro diagrama e tendo em conta o requisito do histórico, conclui-se que as seguintes classes iriam ter DAO's associados:

- Ativo - AtivoDAO;
- Utilizador - UtilizadorDAO;
- ContratoAberto - ContratoAbertoDAO;
- ContratoFechado - ContratoFechadoDAO.

### 6.1.3 Pattern - Singleton

Este pattern foi utilizado no sentido das classes DAO's têm uma instância (igual) em comum que é a conexão à Base de Dados (classe `DBConnection`). Tendo em conta o que foi referido, assim pode apenas ser criada uma vez, sendo que seria desnecessário criar o objeto várias vezes, pelo que haverá um custo reduzido.

### 6.1.4 Pattern - Factory

Durante o processo de planeamento o grupo deparou-se com a criação de vários tipos de activos, nomeadamente Acao, Commodity, Indice e Moeda. Para tal, foi decidido que se deveria usar o factory pattern. Para o efeito, criou-se um método que trata da criação dos diferentes tipos de ativo e devolve um ativo do tipo pretendido.

Com a utilização deste pattern o cliente apenas precisa de comunicar qual o tipo de objecto que necessita e a responsabilidade da criação do objecto é passada para a classe Ativo e, conseqüentemente, a criação do Ativo fica totalmente desacoplada de quem o cria. Este pattern é implementado através do método `factoryAtivo(type : int) : Ativo` na classe `Ativo`.

### 6.1.5 Pattern - Strategy

Este padrão de arquitectura é usado em duas situações no sistema desenvolvido.

No primeiro caso, é usado na classe `Ativo` no qual a taxa cobrada pela correctora varia conforme o activo, sendo até inexistente em determinados activos. Para tal, criou-se a interface `TaxaBehavior` que possui o método `+calculateTaxa() : double` e as classes `TaxaAlgorithmOne` e `TaxaAlgorithmTwo`. A classe `Ativo` possui uma variável de instância `taxaBehavior` e na criação do activo conforme o seu tipo cria ou não o seu `taxaBehavior`.

No segundo caso, o *Strategy Pattern* é empregado para facilitar a troca de API de dados financeiros e reduzir a dependência de uma única API. Para o efeito criou-se uma interface `FinanceAPI` e uma classe concreta que encapsula o uso da API `YahooFinance` e implementa o método `+getPriceAPI(codAtivo : String) : double`. O Facade detém uma variável de instância `financeAPI` e através dela requisita os valores de cada ativo.

Este padrão permite que uma família de algoritmos possa ser definida como uma hierarquia de classes e pode ser usada para alterar o comportamento da aplicação sem alterar a sua arquitectura, permite adicionar novos algoritmos facilmente, é possível mudar de estratégia em "run-time" e as estruturas de dados usadas para implementar o algoritmo estão encapsuladas na "Strategy class", portanto a classe que lhe dá contexto não é afectada.

## 6.2 Outros pormenores

Nesta secção, é abordada outras questões como a herança e a delegação.

### Classe Contrato

Esta classe foi criada no contexto do que o `ContratoFechado` e `ContratoAberto` têm em comum. Portanto, foi realizada uma delegação.

### Classe ContratoAberto

Classe abstrata, que será herdada pelas classes `ContratoAbertoCompra` e `ContratoAbertoVenda`, na medida em que, são dois tipos de contrato e existe diferença no cálculo de balanço, sendo que esse valor será simétrico entre os dois.

## 6.3 Novo requisito

O novo requisito fornecido pelo professor no final do dia 28 de Novembro, tinha como objetivo fazer com que os utilizadores possam seguir um determinado número de ativos, sendo notificados quando um destes sofrer uma alteração significativa no seu valor.

As alterações necessárias foram a

- Criaram-se duas novas tabelas na base de dados:

- A tabela **Gostos**, que faz a ligação entre um utilizador e os activos que pretende seguir e ser notificado quando estes sofrem uma variação significativa.
- A tabela **notificacoes**, que guarda as notificacoes recebidas aquando de uma variação significativa de um activo.
- Implementou-se o padrão observador em que o Facade é o "*observer*" e o "*subject*" é o ativo. Para tal concebeu-se a interface Observer e a Subject que o Ativo e o Facade implementam, respectivamente. Com a utilização deste pattern o subject apenas sabe que o observador implementa a interface observer. Também não é necessário realizar qualquer tipo de alteração à classe que implementa o subject para conseguir adicionar e remover observadores. Para além disto podemos reutilizar o subject e o observer independentemente um do outro.
- Criou-se uma nova "*user interface*" de forma a permitir ao utilizador final seleccionar ativos de forma a ser notificado.
- Na classe DealChecker alterou-se de forma a notificar também utilizadores que seguem um determinado ativo.
- Conceberam-se os métodos necessários para ligar e fazer uso destas componentes



## 7. Diagramas de Máquina de Estados

De seguida, e tendo já em mente o prototipo da interface a implementar, foram desenvolvidas as máquinas de estado, que representam a interacção que um utilizador da aplicação terá com o sistema desenvolvido pelo grupo e que tipos de consequências acontecem quando um utilizador interage com o sistema. O intuito relacionado com o desenvolvimento destas máquinas de estado é de melhor perceber como um utilizador poderá usufruir da nossa plataforma e de que maneira, de modo a que, no desenvolvimento da interface e das funcionalidades, estes estejam de acordo com este plano inicial. Como tal, foram desenvolvidas duas máquinas de estado, para as particularidades cruciais ao funcionamento do sistema, todas relacionadas com o utilizador e a sua experiência na plataforma.

## 7.1 Máquina de Estado Login

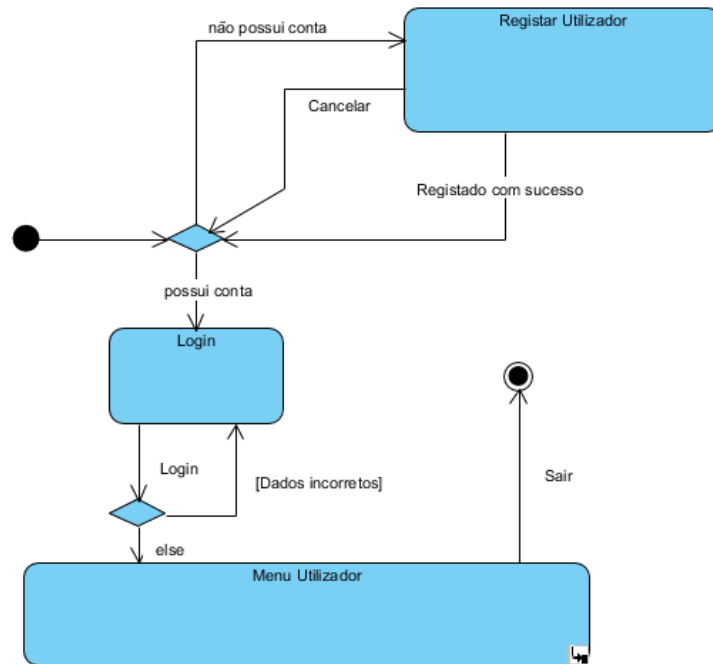


Figura 7.1: Máquina de Estado Login.

Nesta máquina de estados que modela o processo de login, um utilizador, caso não possua conta, terá de transitar para o estado em que se regista, existindo um conjunto de acções que acontecem nesse mesmo estado. Caso o utilizador possua conta apenas necessita de se autenticar e digitar as credenciais corretas de acesso à sua conta. Se assim o desejar, poderá também fechar a aplicação.

## 7.2 Máquina de Estado do Menu do Utilizador

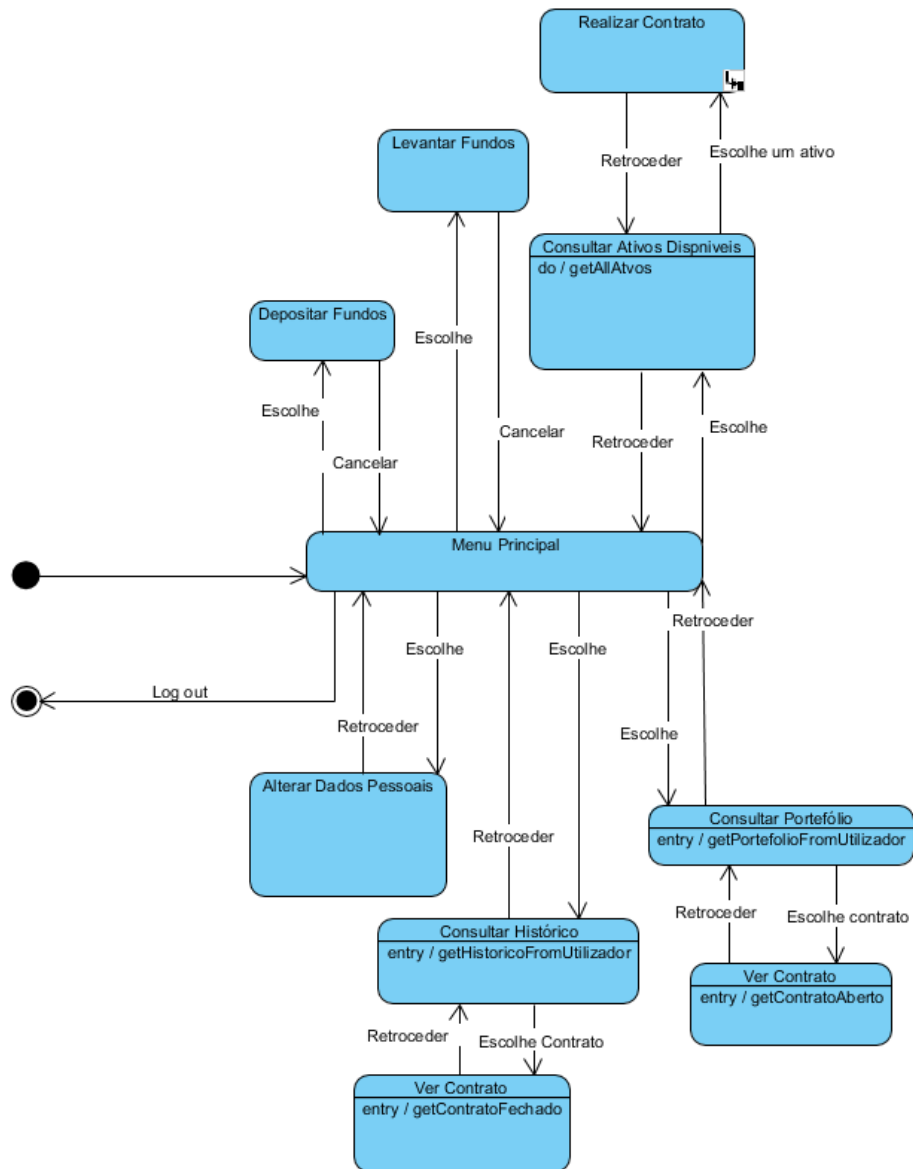


Figura 7.2: Máquina de Estado Menu Utilizador

Após o utilizador se ter autenticado, este irá transitar para o menu principal onde, de acordo com as suas escolhas, irá circular para estados diferentes.

No menu "Consultar Activos Disponíveis", poderá realizar um contrato sobre um desses activos, em que na próxima secção será apresentada e descrita a máquina de estado deste mesmo, uma vez que se trata de um processo de maior complexidade.

A partir do menu "Consultar Portfólio", poderá aceder a um contrato específico dos que lá se encontram, onde serão apresentados detalhes sobre o mesmo. Depois de ter o contrato escolhido, se assim o desejar, poderá encerrar o contrato, ou seja, fechar a

posição e, consequentemente, o contrato será reencaminhado para o histórico e o saldo actualizado. Outra opção será alterar o Take Profit que atribuiu anteriormente e, da mesma maneira, poderá ser alterado o Stop Loss.

Se optar pelo menu "Consultar Histórico", ser-lhe-á apresentado todos os contratos, cujo o utilizador fechou a posição, sendo que poderá optar por ver um destes em específico.

No menu principal, terá ainda direito a poder introduzir dinheiro, de modo a aumentar o seu saldo disponível na plataforma. Por outro lado, poderá fazer o caminho inverso, isto é, levantar dinheiro do sistema e ver o seu capital ser reduzido consoante o montante desse levantamento.

De referir que os menus estão todos interligados, pelo que o utilizador poderá retroceder sempre para o anterior, assim como terminar sessão quando se encontra no menu principal.

Posteriormente ao utilizador ter visualizado os activos disponíveis na plataforma e ter escolhido um destes, terá a opção de realizar um contrato sobre este.

Aquando da realização de um contrato deverá escolher se pretende um contrato de compra ou um contrato de venda, sendo que poderá também cancelar e regressar à apresentação dos activos.

Em seguida, o sistema perguntará ao utilizador se deseja ter leverage no contrato, apresentando as respectivas opções e, dentro dessas opções, poderá escolher uma diferente da pré-definida (x1). Em que seguida deverá escolher o montante e, conforme o seu saldo seja suficiente ou não, irá avançar para a decisão do Stop Loss ou para repetir este passo, respectivamente. O mesmo se aplica no Stop Loss, se tiver capital necessário seguirá para a escolha do Take Profit, caso contrário terá que repetir a sua escolha. Consequentemente, decidirá o Take Profit, sendo que neste caso não existe qualquer restrição. Por último, pode confirmar ou cancelar o contrato que estava a realizar.

Uma vez mais, se o utilizador desejar poderá voltar atrás e alterar as definições que teria colocado anteriormente para o contrato que está a realizar.

## 8. Base de Dados

Como consequência do Diagrama de Classes realizado e de maneira a obter maior persistência dos dados (com os cuidados que se deve ter no mapeamento de classes em tabelas), criou-se uma Base de Dados relacional com o seguinte modelo lógico:

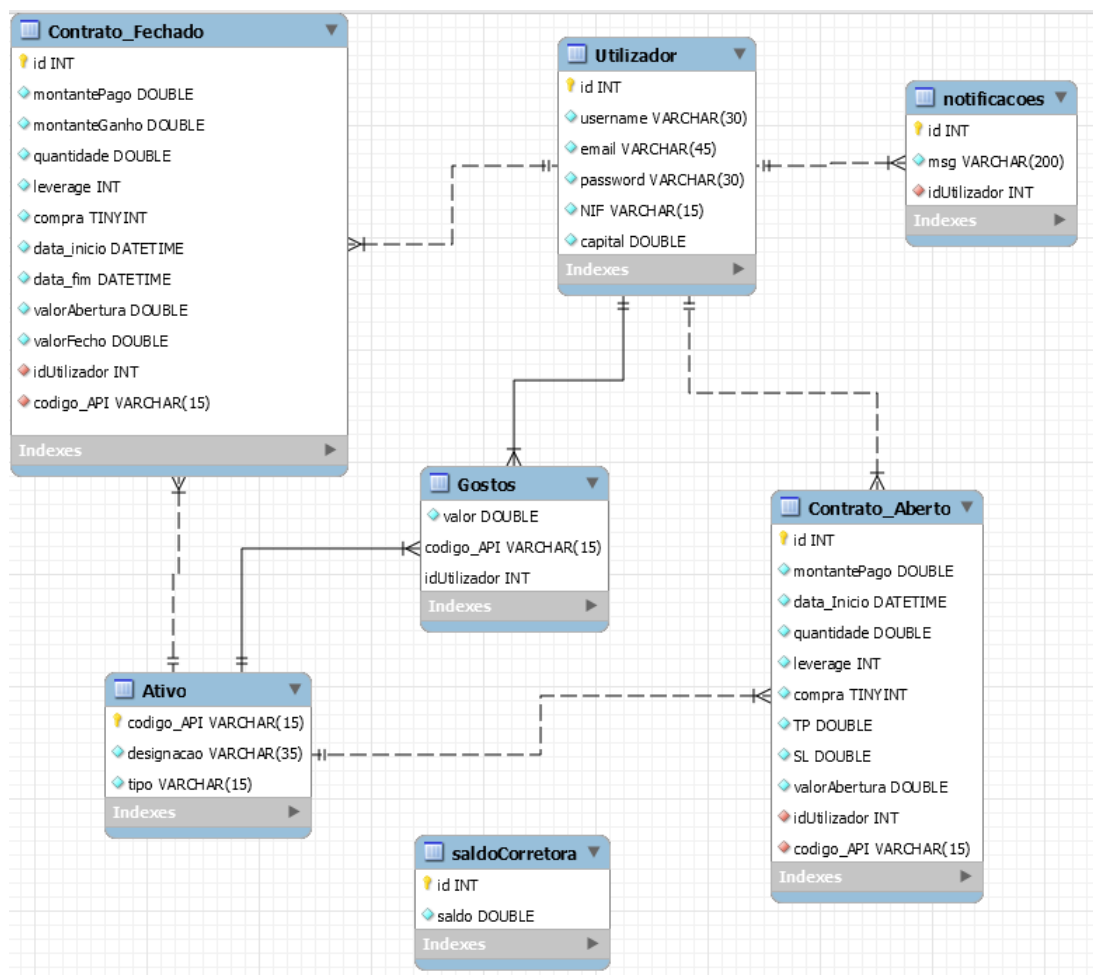


Figura 8.1: Base de Dados relacional.

## 9. Interface

Nesta secção são explicitados os esboços da interface, feitos tendo sempre em atenção os diagramas de máquina de estados elaborados.

### 9.1 Entrada na Plataforma

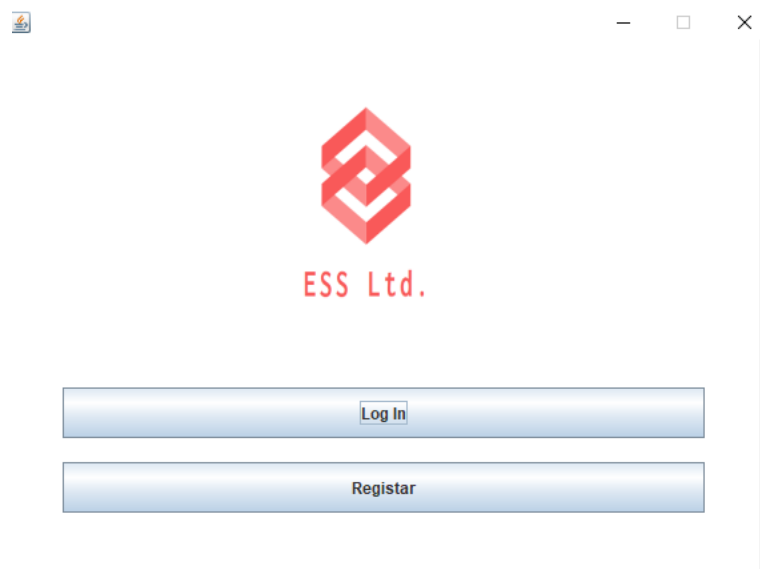
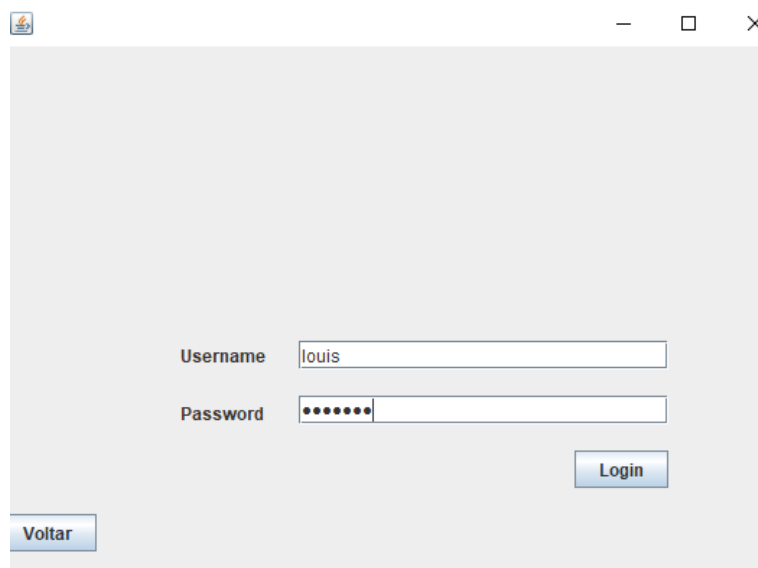


Figura 9.1: Interface inicial da plataforma.

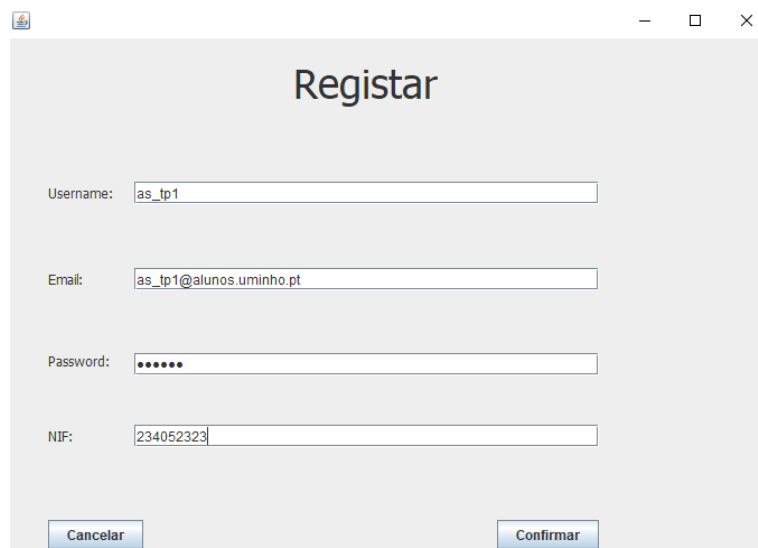
## 9.2 Login



A screenshot of a login window. The window has a title bar with a small icon on the left and standard minimize, maximize, and close buttons on the right. The main area is light gray. In the center, there are two input fields. The first is labeled 'Username' and contains the text 'louis'. The second is labeled 'Password' and contains seven dots. Below the password field is a blue button labeled 'Login'. In the bottom-left corner, there is a blue button labeled 'Voltar'.

Figura 9.2: Interface de login.

## 9.3 Registrar



A screenshot of a registration window titled 'Registrar'. The window has a title bar with a small icon on the left and standard minimize, maximize, and close buttons on the right. The main area is light gray. It contains four input fields: 'Username' with 'as\_tp1', 'Email' with 'as\_tp1@alunos.uminho.pt', 'Password' with seven dots, and 'NIF' with '234052323'. At the bottom, there are two blue buttons: 'Cancelar' on the left and 'Confirmar' on the right.

Figura 9.3: Interface registrar.

A um utilizador que se pretenda registar surgirão as seguintes interfaces e ser-lhe-ão requisitados os seguintes dados por ordem, podendo, ou não, aparecer as mensagens de erro associadas.

## 9.4 Menu Principal



Figura 9.4: Interface menu principal.

## 9.5 Alterar Dados Pessoais

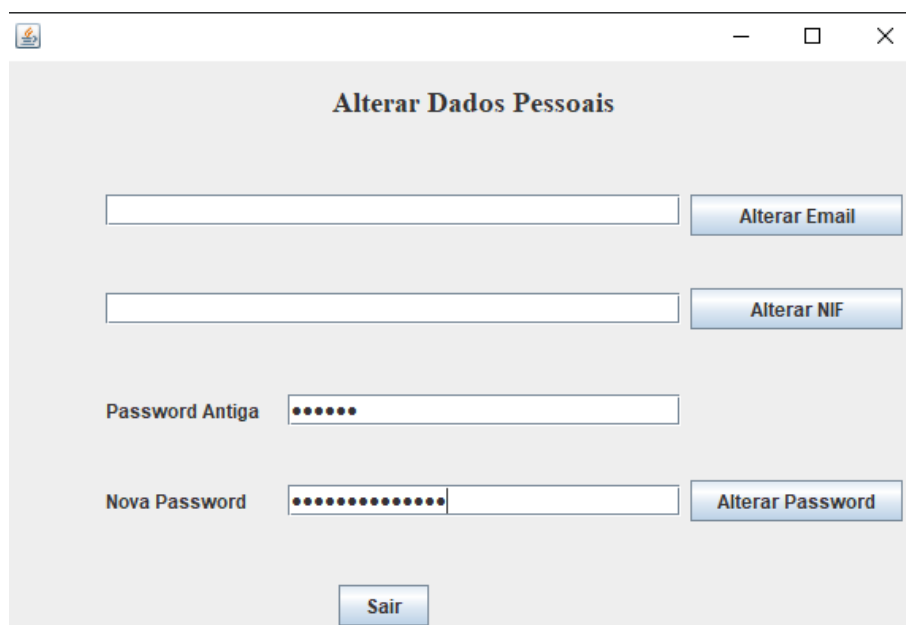
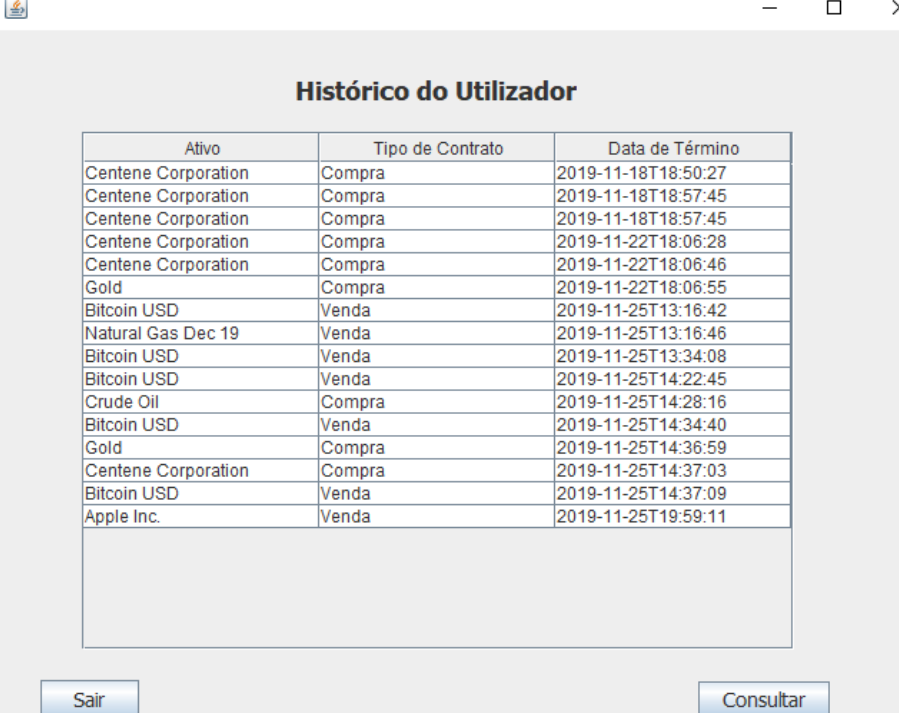


Figura 9.5: Interface email inválido.



## 9.6 Histórico



**Histórico do Utilizador**

Ativo	Tipo de Contrato	Data de Término
Centene Corporation	Compra	2019-11-18T18:50:27
Centene Corporation	Compra	2019-11-18T18:57:45
Centene Corporation	Compra	2019-11-18T18:57:45
Centene Corporation	Compra	2019-11-22T18:06:28
Centene Corporation	Compra	2019-11-22T18:06:46
Gold	Compra	2019-11-22T18:06:55
Bitcoin USD	Venda	2019-11-25T13:16:42
Natural Gas Dec 19	Venda	2019-11-25T13:16:46
Bitcoin USD	Venda	2019-11-25T13:34:08
Bitcoin USD	Venda	2019-11-25T14:22:45
Crude Oil	Compra	2019-11-25T14:28:16
Bitcoin USD	Venda	2019-11-25T14:34:40
Gold	Compra	2019-11-25T14:36:59
Centene Corporation	Compra	2019-11-25T14:37:03
Bitcoin USD	Venda	2019-11-25T14:37:09
Apple Inc.	Venda	2019-11-25T19:59:11

Sair Consultar

Figura 9.6: Interface histórico.

## 9.7 Consulta de Contrato Fechado

**Contrato Fechado**

**Activo:** Apple Inc.

**Tipo de Contrato:** Venda

**Quantidade de Activo:** 37,75      **Quantia investida:** 10000


**Valor de abertura do activo:** 264,9      **Valor de fecho do activo:** 265,21

**Data de abertura da posição:** 2019-11-25T19:52:52      **Data de fecho da posição:** 2019-11-25T19:59:11

**Leverage:** 1      **Balanço Final:** -11,68

Figura 9.7: Interface contrato encerrado.

## 9.8 Consulta de Activos Disponíveis



The screenshot shows a web application window with the title "Ativos Disponíveis". Inside the window, there is a table with three columns: "Ativo", "Tipo de Ativo", and "Valor por unidade". The table lists various assets such as Aurora Cannabis Inc., NVIDIA Corporation, Apple Inc., Cisco Systems, Inc., Gold, Ethereum USD, Ford Motor Company, Natural Gas Dec 19, Nokia Corporation, Vix, Crude Oil, Dow 30, Microsoft Corporation, Bitcoin USD, FTSE 100, S&P 500, Advanced Micro Devices, Russell 2000, Centene Corporation, Silver, EUR/USD, and RBOB Gasoline Dec 19. Below the table, there are two buttons: "Sair" and "Escolher Ativo".

Ativo	Tipo de Ativo	Valor por unidade
Aurora Cannabis Inc.	Acao	2,51
NVIDIA Corporation	Acao	221,21
Apple Inc.	Acao	266,37
Cisco Systems, Inc.	Acao	45,45
Gold	Ativo	1454,7
Ethereum USD	Moeda	146,43
Ford Motor Company	Acao	9
Natural Gas Dec 19	Commoditie	2,52
Nokia Corporation	Acao	3,48
Vix	Indice	11,9
Crude Oil	Commoditie	58,02
Dow 30	Indice	28066,47
Microsoft Corporation	Acao	151,23
Bitcoin USD	Moeda	7164,63
FTSE 100	Indice	7396,29
S&P 500	Indice	3133,64
Advanced Micro Devices, ...	Acao	39,79
Russell 2000	Indice	1622,29
Centene Corporation	Acao	61,18
Silver	Commoditie	16,88
EUR/USD	Moeda	1,1
RBOB Gasoline Dec 19	Commoditie	1,68

Sair Escolher Ativo

Figura 9.8: Interface activos disponíveis.

## 9.9 Realização de um Contrato



**Contrato Aberto**

Saldo da conta: 75539,11

Ativo: Bitcoin USD      Valor do Ativo: 7167,51

Tipo de Contrato: Compra ▼

Montante desejado: 8000,00      Quantidade: 1,12

Leverage: 1 ▼

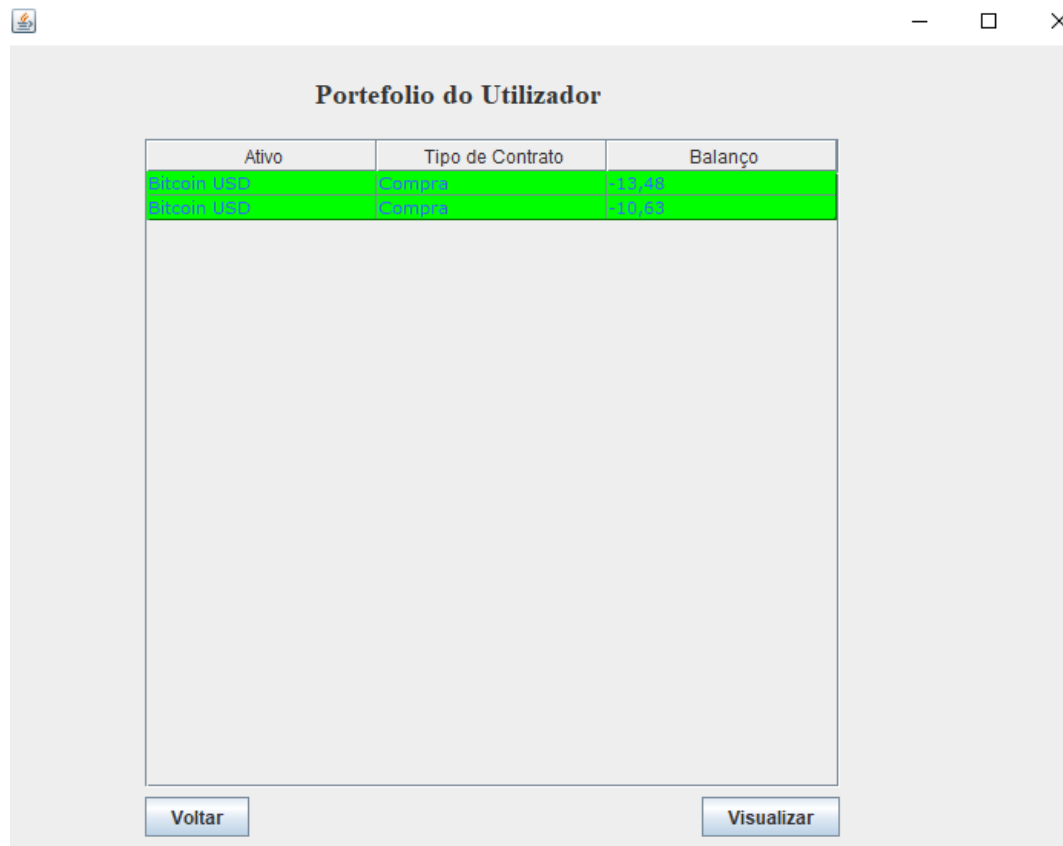
Take Profit: 300,00

Stop Loss: -300,00

Cancelar      Abrir Posição

Figura 9.9: Interface realização de contrato, definição do tipo de contrato.

## 9.10 Consulta do Portfólio



**Portefolio do Utilizador**

Ativo	Tipo de Contrato	Balanço
Bitcoin USD	Compra	-13,48
Bitcoin USD	Compra	-10,63

Voltar Visualizar

Figura 9.10: Interface consulta do portfólio.

## 9.11 Consulta de um Contrato do Portfólio

**Contrato Aberto**

Ativo: Bitcoin USD      Balanço: -10,63

Tipo de Contrato: Compra      Data Início: 2019-11-25T14:29:12

Quantidade: 1      Montante Investido: 7181,8      Leverage: 1

Take Profit:

Stop Loss:

Figura 9.11: Interface consulta contrato do portfólio.

## 9.12 Depositar Fondos



**Depositar Fondos**

Conta

Valor (euros)

Figura 9.12: Interface Depositar fondos.

## 9.13 Levantar Fondos



**Levantar Fondos**

Conta

Valor (euros)

Figura 9.13: Interface Levantar fondos.

## 10. Conclusão e Trabalho Futuro

Neste projecto, foi elaborada a especificação e implementação do sistema. Na especificação foram construídos os diagramas UML, onde se mostrou vital aplicar os conhecimentos de arquitectura e planeamento de forma a delinear e facilitar o processo de construção até se obter o produto final. Pelo que, esta fase tem uma influência crucial, sendo um passo crítico no processo de desenvolvimento. Na implementação bastou cumprir o que foi delineado anteriormente.

Num primeiro passo, produziu-se o modelo de domínio, o que foi um certo desafio, devido à complexidade da área do projecto. Após a concepção do modelo de domínio estabilizar o restante processo foi facilitado.

O próximo grande "stumbling block" viria a ser o diagrama de classes. A aplicação dos "*design patterns*" revelou-se como o grande problema, pois exigiu o estudo de todos os padrões, as vantagens e desvantagens que trariam e, por fim, e a adequação da sua utilização perante o problema apresentado.

A "user interface" foi numa primeira fase idealizada em terminal. Contudo, o grupo de trabalho optou por realizar uma interface mais avançada concebida em Java Swing, de modo a obter um produto com mais qualidade e mais fácil de utilizar.

Relativamente ao requisito dado propositadamente perto do limite de entrega podemos afirmar que teria sido muito mais difícil de adicionar e implementar se não fosse empregue qualquer tipo de "*design pattern*".



# 11. Anexo

## 11.0.1 Antigas Interfaces

Estas foram as interfaces que o grupo de trabalho idealizou numa primeira fase e que são apresentadas em anexo para se melhor ver a evolução e benefícios que as novas interfaces trouxeram para o uso do sistema.

Foi tomada a decisão de quando surgem menus de escolhas múltiplas, a escolha é feita através das teclas numéricas. Portanto, nas imagens que se seguem irá surgir várias vezes o caso "[número] - [opção]".

## 11.1 Entrada na Plataforma



Figura 11.1: Interface inicial da plataforma.

## 11.2 Login

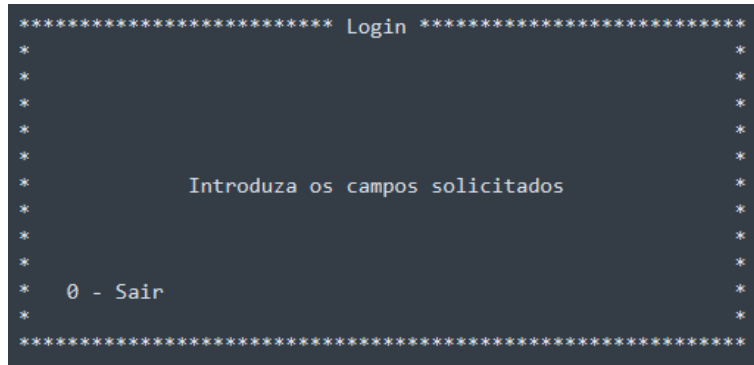


Figura 11.2: Interface de login.

Os campos solicitados serão:

```
Username: [User Input]
```

Figura 11.3: Interface pedido de username.

```
Password: [User Name]
```

Figura 11.4: Interface pedido de password.

Contudo, se os dados estiverem errados surgirá a seguinte mensagem:

```
!!!! Dados Inválidos !!!!!
```

Figura 11.5: Interface dados inválidos.

### 11.3 Registar

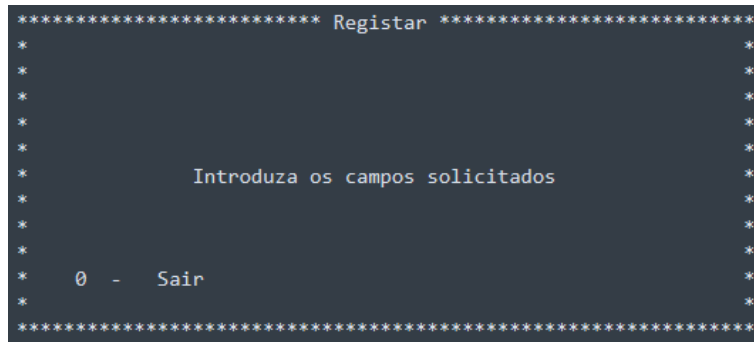


Figura 11.6: Interface registrar.

A um utilizador que se pretenda registar surgirão as seguintes interfaces e ser-lhe-ão requisitados os seguintes dados por ordem, podendo, ou não, aparecer as mensagens de erro associadas.

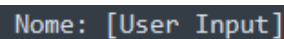


Figura 11.7: Interface de inserção do nome.

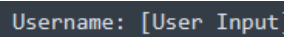


Figura 11.8: Interface de inserção do username.

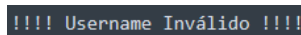


Figura 11.9: Interface username inválido.

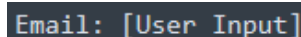


Figura 11.10: Interface de inserção do email.

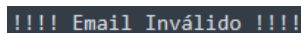
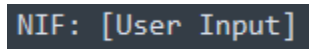
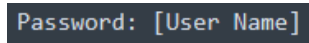


Figura 11.11: Interface email inválido.

A terminal window with a dark background. The text 'NIF: [User Input]' is displayed in a light blue monospace font. The text is positioned on the left side of the terminal, with a small gap between 'NIF:' and '[User Input]'.

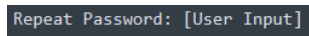
```
NIF: [User Input]
```

Figura 11.12: Interface de inserção do NIF.

A terminal window with a dark background. The text 'Password: [User Name]' is displayed in a light blue monospace font. The text is positioned on the left side of the terminal, with a small gap between 'Password:' and '[User Name]'.

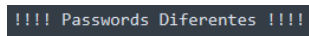
```
Password: [User Name]
```

Figura 11.13: Interface de inserção da password.

A terminal window with a dark background. The text 'Repeat Password: [User Input]' is displayed in a light blue monospace font. The text is positioned on the left side of the terminal, with a small gap between 'Repeat Password:' and '[User Input]'.

```
Repeat Password: [User Input]
```

Figura 11.14: Interface de repetição da password.

A terminal window with a dark background. The text '!!!! Passwords Diferentes !!!!!' is displayed in a light blue monospace font. The text is centered horizontally in the terminal.

```
!!!! Passwords Diferentes !!!!!
```

Figura 11.15: Interface passwords inválidas.

## 11.4 Menu Principal

```
***** Menu *****
*
*
* 1 - Depositar Fundos
*
*
*
*
* 2 - Mercados de Negociação
*
*
*
*
* 3 - Portfólio
*
*
*
*
* 4 - Histórico
*
*
*
*
* 5 - Levantar Fundos
*
*
*
*
* 0 - Terminar sessão
*
*
*****
```

Figura 11.16: Interface menu principal.

## 11.5 Alterar Perfil

```
***** Dados Pessoais *****
*
*
*
Nome: Luís Braga
Username: luisbraga23
Email: a82088@alunos.uminho.pt
NIF: 42527632
*
* 1 - Alterar Nome
* 2 - Alterar Username
* 3 - Alterar Password
* 4 - Alterar Email
* 5 - Alterar NIF
*
* 0 - Sair
*
*****
```

Figura 11.17: Interface email inválido.

```
Nome: [User Input]
```

Figura 11.18: Interface de inserção do nome.

```
Username: [User Input]
```

Figura 11.19: Interface de inserção do username.

```
!!!! Username Inválido !!!!
```

Figura 11.20: Interface username inválido.

```
Email: [User Input]
```

Figura 11.21: Interface de inserção do email.

```
!!!! Email Inválido !!!!!
```

Figura 11.22: Interface email inválido.

```
NIF: [User Input]
```

Figura 11.23: Interface de inserção do NIF.

```
Password: [User Name]
```

Figura 11.24: Interface de inserção da password.

```
Repeat Password: [User Input]
```

Figura 11.25: Interface de repetição da password.

```
!!!! Passwords Diferentes !!!!!
```

Figura 11.26: Interface passwords inválidas.

## 11.6 Histórico

***** Histórico *****				
*				*
*				*
*				*
*	Opção	Activo	T/Contrato	Data Término
	1	Netflix Inc.	Compra	21/10/2019
		.		
		.		
		.		
*				*
*				*
*	0 - Sair			*
*				*
*****				

Figura 11.27: Interface histórico.

## 11.7 Consulta de Contrato Encerrado

```
***** Contrato *****
*
*
*
*
*   Netflix Inc:
*
*   Tipo:   Venda
*
*   Unidades: 54.23      Leverage: X1      Take Profit:999$      Stop Loss:-999$
*
*   Valor de Abertura: 277.79$      Valor Término: 274.05$      Investido: 2000$
*
*   Data de início: 21/10/2019      Data de Término: 22/10/2019
*
*   Balanço Total: 2025.45$
*
*
*
*   0 - Sair
*
*****
```

Figura 11.28: Interface contrato encerrado.

## 11.8 Consulta de Activos Disponíveis

```
***** Mercado Financeiro *****
*
*
*
*
*   Opção   Activo      Vender      Comprar
*
*   1       Netflix Inc  266.31      266.82
*
*   2       Facebook    182.10      183.43
*
*
*   .
*   .
*   .
*
*
*   0 - Sair
*
*****
```

Figura 11.29: Interface activos disponíveis.



## 11.9 Realização de um Contrato

```
***** Contrato *****
*
*
*
  Netflix Inc:

  Valor de Abertura:  277.79$

  Data de início: 21/10/2019

  Tipo:
    1 - Venda
    2 - Compra
*
*
*
  0 - Sair
*
*****
```

Figura 11.30: Interface realização de contrato, definição do tipo de contrato.

```
***** Contrato *****
*
*
*
  Netflix Inc:

  Tipo: Venda

  Valor de Abertura:  277.79$

  Data de início: 21/10/2019

  Leverage:
    1 - X1 (Sem leverage)
    2 - X2
    3 - X5
*
*
*
  0 - Retroceder
*
*****
```

Figura 11.31: Interface realização de contrato, definição de leverage.

```
***** Contrato *****
*
*
*
*
*   Netflix Inc:
*
*   Tipo: Venda
*
*   Leverage: X1
*
*   Valor de Abertura: 277.79$
*
*   Data de início: 21/10/2019
*
*   1 - Montante
*
*
*   0 - Retroceder
*
*****
```

Figura 11.32: Interface realização de contrato, definição do montante.

```
Montante: [User Input]
```

Figura 11.33: Interface inserção de montante inválido.

```
!!!! Montante Inválido !!!!
```

Figura 11.34: Interface montante inválido.

```

***** Contrato *****
*
*
*
*
Netflix Inc:

    Tipo: Venda

    Unidades: 54.23   Leverage: X1   Take Profit:999$   Stop Loss:-999$

    Valor de Abertura: 277.79$   Investido: 2000$

    Data de início: 21/10/2019

    1 - Prosseguir
    2 - Take Profit
    3 - Stop Loss

*
*
*
*
0 - Retroceder
*
*
*****

```

Figura 11.35: Interface realização de contrato, definição de Take Profit e Stop Loss.

```
Take Profit: [User Input]
```

Figura 11.36: Interface inserção de Take Profit.

```
Stop Loss: [User Input]
```

Figura 11.37: Interface inserção de Stop Loss.

```
!!!! Stop Loss Inválido !!!!
```

Figura 11.38: Interface Stop Loss Inválido.

```

***** Contrato *****
*
*
*
*
*   Netflix Inc:
*
*   Tipo: Venda
*
*   Unidades:  54.23   Leverage: X1   Take Profit:999$   Stop Loss:-999$
*
*   Valor de Abertura: 277.79$   Investido: 2000$
*
*   Data de início: 21/10/2019
*
*   1 - Confirmar
*   2 - Cancelar
*
*
*   0 - Retroceder
*
*****

```

Figura 11.39: Interface confirmação de contrato.

## 11.10 Consulta do Portfólio

```

***** Portfólio *****
*
*
*
*   Opção      Activo      T/Contrato      Data Início
*
*   1          Netflix Inc.  Compra          21/10/2019
*   |          |          |
*   |          |          |
*   |          |          |
*
*
*   0 - Sair
*
*****

```

Figura 11.40: Interface consulta do portfólio.

## 11.11 Consulta de um Contrato do Portfólio

```
***** Contrato *****
*
*
*
*
Netflix Inc:
*
Tipo: Venda
*
Unidades: 54.23      Leverage: X1      Take Profit:999$      Stop Loss:-999$
*
Valor de Abertura: 277.79$      Valor Atual: 274.05$      Investido: 2000$
*
Data de início: 21/10/201
*
Balanco Total: 2025.45$
*
*
*      1 - Encerrar
*
*      2 - Alterar Take Profit
*
*      3 - Alterar Stop Loss
*
*      0 - Sair
*
*****
```

Figura 11.41: Interface consulta contrato do portfólio.

```
Take Profit: [User Input]
```

Figura 11.42: Interface consulta contrato do portfólio.

```
Stop Loss: [User Input]
```

Figura 11.43: Interface alteração do Take Profit.

```
!!!! Stop Loss Inválido !!!!
```

Figura 11.44: Interface alteração do Stop Loss.