



**Universidade do Minho**

Escola de Engenharia

Licenciatura em Engenharia informática

## **Unidade Curricular de Laboratórios de Informática IV**

Ano Letivo de 2023/2024

### **Belíssimos Collectibles**

**Gonçalo Costa, José Oliveira, Marta Rodrigues, Pedro Lopes**

Janeiro, 2024

**LI4**

**16**

|                  |  |
|------------------|--|
| Data de Recepção |  |
| Responsável      |  |
| Avaliação        |  |
| Observações      |  |

## **Belíssimos Collectibles**

**Gonçalo Costa, José Oliveira, Marta Rodrigues, Pedro Lopes**

Janeiro, 2024

## Resumo

O seguinte relatório foi elaborado no âmbito da unidade curricular de Laboratórios de Informática IV, para a qual foi-nos proposto fazer a modelação e o desenvolvimento de uma plataforma que fosse capaz de suportar Leilões Online numa área de negócio à escolha.

O trabalho foi separado em duas etapas e neste relatório é apresentada as fases que lhes fazem parte, a primeira, segunda e terceira fase do desenvolvimento desta solução.

A primeira fase começou pelo processo de recolha de ideias e a decisão de um tópico de trabalho, cuja materialização levou à contextualização, fundamentação e objetivos que o projeto se basearia.

A segunda fase do projeto foi marcada pela recolha e análise de requisitos funcionais e não funcionais. Após isso, esses mesmos requisitos foram desdobrados em casos de uso, que quando apresentados e especificados e em junção com o modelo domínio, servirão como fundação do sistema futuramente a desenvolver na segunda etapa. De seguida, foi concebido um diagrama de classes, resultante da análise dos aspetos a desenvolver, juntamente com diagramas de sequência e diagramas de atividade.

Para que os dados, das diferentes componentes e entidades se permanecessem sempre intactos após serem sujeitos a alterações e desta forma disponíveis para consulta sempre que desejado, foi desenhado o modelo lógico da base de dados, cuja qual servirá como base para implementação da mesma na fase seguinte.

Por último, foram feitos esboços da interface do sistema que representam as expectativas da equipa quanto à aparência do sistema a desenvolver, de forma a ter uma utilização o mais simples possível e intuitiva, assim como agradável.

Na terceira, e última, fase do desenvolvimento deste projeto, foram implementadas todas as componentes do mesmo, expondo as abordagens utilizadas na execução das funcionalidades estabelecidas.

**Área de Aplicação:** Serviço de leilões online para colecionáveis.

**Palavras-Chave:** Engenharia de *Software*, “Belíssimos Collectibles”, Leilões, Colecionáveis, *Microsoft*, *Blazor*, Fundamentação, Especificação, Bases de Dados Relacionais.

# Índice

|  |    |
|--|----|
| <a href="#">Resumo</a>   | i  |
| <a href="#">Índice</a>   | ii |
| <a href="#">Índice de Figuras</a>  | iv |
| <a href="#">Índice de Tabelas</a>  | v  |
| <a href="#">1. Definição do Sistema</a>  | 1  |
| <a href="#">1.1. Contextualização</a>  | 1  |
| <a href="#">1.2. Fundamentação</a>   | 1  |
| <a href="#">1.3. Objetivos</a>   | 2  |
| <a href="#">1.4. Viabilidade</a>   | 2  |
| <a href="#">1.5. Recursos a utilizar</a>   | 3  |
| <a href="#">1.6. Equipa de trabalho</a>  | 3  |
| <a href="#">1.7. Plano de execução do trabalho</a>   | 4  |
| <a href="#">2. Definição de Requisitos</a>   | 6  |
| <a href="#">2.1. Apresentação da estratégia e método</a>                                     | 6  |
| <a href="#">2.2. Descrição geral dos requisitos (funcionais e não funcionais) levantados</a> | 6  |
| <a href="#">Requisitos Funcionais:</a>   | 6  |
| <a href="#">2.2.1. Registo do Utilizador</a>   | 6  |
| <a href="#">2.2.2. Autenticação do Utilizador</a>  | 7  |
| <a href="#">2.2.3. Eliminar Conta</a>  | 7  |
| <a href="#">2.2.4. Alterar informações da conta</a>  | 7  |
| <a href="#">2.2.5. Iniciar um leilão para um produto</a>                                     | 8  |
| <a href="#">2.2.6. Interromper/Parar um leilão</a>   | 8  |
| <a href="#">2.2.7. Consultar o estado dos leilões</a>  | 9  |
| <a href="#">2.2.8. Alterar informações dos leilões</a>                                       | 9  |
| <a href="#">2.2.9. Pesquisar produtos</a>  | 9  |
| <a href="#">2.2.10. Fazer uma licitação</a>  | 10 |
| <a href="#">2.2.11. Consultar a conta de um utilizador</a>                                   | 10 |
| <a href="#">2.2.12. Inserir dinheiro na conta</a>  | 10 |
| <a href="#">2.2.13. Avaliar um vendedor</a>  | 11 |
| <a href="#">2.2.14. Terminar Sessão</a>  | 11 |
| <a href="#">Requisitos não funcionais:</a>   | 11 |
| <a href="#">2.3. Validação dos requisitos estabelecidos</a>                                  | 12 |
| <a href="#">3. Especificação e Modelação do Software</a>                                     | 13 |
| <a href="#">3.1. Apresentação geral da especificação</a>                                     | 13 |
| <a href="#">3.2. Aspetos estruturais</a>   | 13 |
| <a href="#">3.2.1. Modelo de Domínio</a>   | 13 |
| <a href="#">3.3. Aspetos Comportamentais</a>   | 14 |
| <a href="#">3.3.1 Diagrama de Use Cases</a>  | 14 |

|  |    |
|--|----|
| <a href="#"><u>3.3.2. Diagramas de Atividade</u></a>   | 23 |
| <a href="#"><u>3.3.3. Diagramas de Sequência</u></a>   | 26 |
| <a href="#"><u>3.3.4. Diagrama de Classes</u></a>  | 35 |
| <a href="#"><u>4. Conceção do Sistema de Dados</u></a>   | 36 |
| <a href="#"><u>4.1. Apresentação geral da estrutura (esquema) do sistema de dados</u></a>            | 36 |
| <a href="#"><u>4.2. Descrição detalhada dos vários elementos de dados e seus relacionamentos</u></a> | 37 |
| <a href="#"><u>5. Esboço dos Interfaces do Sistema</u></a>   | 39 |
| <a href="#"><u>5.1. Estrutura geral das interfaces do sistema</u></a>                                | 39 |
| <a href="#"><u>5.2. Caracterização das interfaces</u></a>  | 39 |
| <a href="#"><u>6. Implementação</u></a>  | 44 |
| <a href="#"><u>6.1. Ferramentas utilizadas</u></a>   | 44 |
| <a href="#"><u>7. Conclusões e Trabalho Futuro</u></a>   | 45 |
| <a href="#"><u>Referências</u></a>   | 47 |
| <a href="#"><u>Lista de Siglas e Acrónimos</u></a>   | 48 |
| <a href="#"><u>Anexos</u></a>  | 49 |
| <a href="#"><u>I. Anexo 1</u></a>  | 49 |

# Índice de Figuras

|   |    |
|---|----|
| <a href="#"><u>Figura 1 - Diagrama de Gantt parte 1.</u></a>                  | 4  |
| <a href="#"><u>Figura 2 - Diagrama de Gantt parte 2.</u></a>                  | 4  |
| <a href="#"><u>Figura 3 - Modelo de Domínio</u></a>                           | 14 |
| <a href="#"><u>Figura 4 - Diagrama de Use Cases</u></a>                       | 15 |
| <a href="#"><u>Figura 5 - Diagrama de Atividade "Iniciar um Leilão"</u></a>   | 24 |
| <a href="#"><u>Figura 6 - Diagrama de Atividade "Fazer uma Licitação"</u></a> | 25 |
| <a href="#"><u>Figura 7 - Diagrama de Sequência "registrar"</u></a>           | 27 |
| <a href="#"><u>Figura 8 - Diagrama de Sequência "autenticar"</u></a>          | 27 |
| <a href="#"><u>Figura 9 - Diagrama de Sequência "alterarConta"</u></a>        | 28 |
| <a href="#"><u>Figura 10 - Diagrama de Sequência "eliminarConta"</u></a>      | 28 |
| <a href="#"><u>Figura 11 - Diagrama de Sequência "iniciarLeilao"</u></a>      | 29 |
| <a href="#"><u>Figura 12 - Diagrama de Sequência "interromperLeilao"</u></a>  | 29 |
| <a href="#"><u>Figura 13 - Diagrama de Sequência "consultarLeilao"</u></a>    | 30 |
| <a href="#"><u>Figura 14 - Diagrama de Sequência "alterarLeilao"</u></a>      | 30 |
| <a href="#"><u>Figura 15 - Diagrama de Sequência "fazerLicitacao"</u></a>     | 31 |
| <a href="#"><u>Figura 16 - Diagrama de Sequência "terminarLeilao"</u></a>     | 31 |
| <a href="#"><u>Figura 17 - Diagrama de Sequência "pesquisarProdutos"</u></a>  | 32 |
| <a href="#"><u>Figura 18 - Diagrama de Sequência "consultarConta"</u></a>     | 33 |
| <a href="#"><u>Figura 19 - Diagrama de Sequência "avaliarVendedor"</u></a>    | 33 |
| <a href="#"><u>Figura 20 - Diagrama de Sequência "inserirDinheiro"</u></a>    | 34 |
| <a href="#"><u>Figura 21 - Diagrama de Classes</u></a>                        | 35 |
| <a href="#"><u>Figura 22 - Modelo Lógico da Base de Dados.</u></a>            | 36 |
| <a href="#"><u>Figura 23 - Página Inicial.</u></a>                            | 39 |
| <a href="#"><u>Figura 24 - Página de criação de conta.</u></a>                | 40 |
| <a href="#"><u>Figura 25 - Página para autenticação da conta.</u></a>         | 40 |
| <a href="#"><u>Figura 26 - Página de procura de leilões.</u></a>              | 41 |
| <a href="#"><u>Figura 27 - Página de perfil.</u></a>                          | 41 |
| <a href="#"><u>Figura 28 - Página de leilão ao vivo (vista CLIENTE).</u></a>  | 42 |
| <a href="#"><u>Figura 29 - Página de leilão ao vivo (vista VENDEDOR).</u></a> | 42 |
| <a href="#"><u>Figura 30 - Página de criação de leilão.</u></a>               | 43 |

# Índice de Tabelas

|   |    |
|---|----|
| <a href="#"><u>Tabela 1 - Use Case “Registo do Utilizador”.</u></a>         | 16 |
| <a href="#"><u>Tabela 2 - Use Case “Autenticar”.</u></a>                    | 16 |
| <a href="#"><u>Tabela 3 - Use Case “Alterar informações da conta”.</u></a>  | 17 |
| <a href="#"><u>Tabela 4 - Use Case “Eliminar Conta”.</u></a>                | 17 |
| <a href="#"><u>Tabela 5 - Use Case “Iniciar Leilão”.</u></a>                | 18 |
| <a href="#"><u>Tabela 6 - Use Case “Interromper Leilão”.</u></a>            | 18 |
| <a href="#"><u>Tabela 7 - Use Case “Consultar estado do leilão”.</u></a>    | 19 |
| <a href="#"><u>Tabela 8 - Use Case “Alterar informações do leilão”.</u></a> | 19 |
| <a href="#"><u>Tabela 9 - Use Case “Fazer uma licitação”.</u></a>           | 20 |
| <a href="#"><u>Tabela 10 - Use Case “Terminar leilão”.</u></a>              | 21 |
| <a href="#"><u>Tabela 11 - Use Case “Pesquisar produto”.</u></a>            | 22 |
| <a href="#"><u>Tabela 12 Use Case “Consultar conta utilizador”.</u></a>     | 22 |
| <a href="#"><u>Tabela 13 - Use Case “Avaliar vendedor”.</u></a>             | 22 |
| <a href="#"><u>Tabela 14 - Use Case “Inserir dinheiro na conta”.</u></a>    | 23 |
| <a href="#"><u>Tabela 15 - Use Case “Terminar sessão”.</u></a>              | 23 |

# **1. Definição do Sistema**

## **1.1. Contextualização**

Rui Afonso da Constança Mesquita Rouxinol era um jovem apaixonado pelo mundo fictício de séries, filmes e outros meios de entretenimento. Ao longo dos anos construiu uma vasta coleção de diversos tipos de merchandising das suas franquias favoritas.

Um dia, a mãe de Rui sofreu um acidente, levando a mesma a requisitar uma baixa de trabalho por tempo indefinido. Eventualmente, a família de Rui viu-se com dificuldade para pagar as contas mínimas mensais, o que impulsionou o filho a tentar arranjar uma solução para, pelo menos, aliviar a pressão dos pais. Como o mesmo ainda estudava e não tinha disponibilidade para arranjar um trabalho, foi estudando outras alternativas e foi assim que Rui Afonso, enquanto admirava a sua coleção, reparou que alguns dos seus itens mais antigos, além de raros, possuíam valores bastante elevados.

Para se aproveitar de tal, abriu uma venda de garagem aos fins de semana e aos poucos ganhou o suficiente para cobrir algumas das suas pequenas despesas. Através das redes sociais publicitou o seu humilde projeto e gradualmente, devido à raridade de alguns colecionáveis, a adesão foi aumentando, tornando-se assim excessivo para a sua pequena garagem, acrescentando à enorme quantidade de mensagens privadas e propostas que ia recebendo durante a semana e, além disso, reparou que havia um grande interesse pelos mesmos artigos por parte de vários clientes. Rui chegou então à conclusão que abrir apenas ao fim de semana na sua pobre garagem e vender os produtos ao primeiro interessado a chegar estava a impedir o acréscimo dos seus lucros. O negócio tornou-se bastante difícil de gerir sozinho devido à alta procura e tudo isso levou-o a criar um humilde site de leilões online, o que exponenciou ainda mais os lances e as pessoas interessadas, pois podia listar os seus diversos produtos e os vender ao maior preço possível.

## **1.2. Fundamentação**

À medida que ia conhecendo mais colecionadores e pessoas no mundo de leilões, Rui Afonso viu que havia uma demanda significativa por esse tipo de produtos e também de uma plataforma de leilões online e ele não estava sozinho na sua necessidade de vender ou comprar



itens raros. Como o seu site só deixava que apenas ele vendesse um produto de cada vez, Rui decidiu então criar a sua própria aplicação de leilões chamada "Belíssimos Collectibles", que permitiria várias pessoas leiloar os seus múltiplos produtos ao mesmo tempo. No entanto, criar tal plataforma sozinho seria impossível, por isso, pediu ajuda ao seu primo, aluno de Engenharia Informática na Universidade do Minho, que pediu aos seus colegas para o ajudarem no projeto.

### **1.3. Objetivos**

Com o aumento da procura de um local seguro para a realização de leilões, o principal objetivo desta plataforma é possibilitar, a qualquer utilizador, a venda dos seus colecionáveis através da criação de leilões e a compra, pela parte do mesmo, de artigos que lhe interesse, participando em leilões de terceiros simultaneamente.

Pretende-se que a venda de itens seja mais fácil e intuitiva para todos aqueles que pretendam criar um leilão, tornando-se apenas necessário preencher determinadas informações sobre o produto que o utilizador queira vender, e que se possa escolher entre uma vasta seleção de modelos de leilões que queira realizar. Com um leilão iniciado, o sistema garantirá o seu bom funcionamento, sem permitir manipulações por parte de outros, mas dará oportunidade ao seu criador de o alterar consoante as suas vontades, podendo este editá-lo ou até interrompê-lo.

Os utilizadores devem poder aceder a qualquer produto que pretenda através do seu nome, marca, categoria ou até por tipo de leilão em que se encontram. Tem também possibilidade de participar num leilão e de avaliar a fiabilidade dos restantes vendedores.

### **1.4. Viabilidade**

A criação da aplicação visa a expansão deste mundo de negócios a um mercado mais abrangente, esperando-se assim atrair mais clientes.

Possibilita, assim, também a introdução de novos produtos previamente indisponíveis nos leilões presenciais. Adicionalmente, fará com que o acesso a diferentes e novos leilões e a colecionáveis variados seja mais fácil do que nunca, oferecendo assim aos seus antigos e novos utilizadores um vasto catálogo de produtos nunca antes vistos.

A plataforma permitirá ainda a realização de diversos leilões ao mesmo tempo e que os seus clientes apostem e vendam vários produtos em diferentes leilões simultaneamente, o que não era possível até ao momento.

## **1.5. Recursos a utilizar**

Para garantir uma boa prática no desenvolvimento da aplicação foi fundamental assegurar todos os recursos necessários, uma vez que sem estes estaríamos a comprometer a sua implementação e funcionamento, garantidos monetariamente por recursos financeiros suficientes para a realização do projeto.

Antes de mais, ao nível de recursos humanos, foi necessário definir uma equipa de desenvolvimento qualificada e motivada que garanta a gestão das tarefas e a satisfação de todas as necessidades pretendidas, precisando, para tal, da colaboração da “Belíssimos Collectibles” e dos seus utilizadores.

Além disso, são necessários alguns recursos materiais para possibilitar a implementação da plataforma a nível de software e hardware, como por exemplo, a nível de software, de serviços fornecidos pelo Microsoft Office, a plataforma de desenvolvimento e execução de sistemas e aplicações, Microsoft .NET, a linguagem de programação Microsoft C# e, finalmente, um sistema de gestão de bases de dados para armazenar a informação recolhida e gerir esses dados, Microsoft SQL Server. Já a nível de hardware, serão utilizados diversos computadores e servidores, para manter o projeto em funcionamento remoto.

## **1.6. Equipa de trabalho**

Como mencionado previamente, de modo a garantir o bom progresso da aplicação foi estabelecida uma equipa de desenvolvimento com capacidades na área para fazer o levantamento de requisitos, especificar, modelar e implementar o sistema pedido.

Assim, para garantir que todas as etapas da aplicação, desde as especificações até à construção, são realizadas, definiu-se que seriam necessários 2 engenheiros informáticos experientes. Dada a complexidade desta aplicação, foi imprescindível a presença de 1 gestor de projetos que garanta a gestão das tarefas dentro dos prazos previamente estabelecidos e o bom funcionamento da equipa e da aplicação. Estimamos também a necessidade de 1 designer encarregue de dar à aplicação um aspeto agradável e garantir a sua simplicidade, a fim de facilitar a sua utilização e tornar perceptível e intuitivo as diversas funcionalidades para qualquer utilizador.

De modo a assegurar o melhor resultado possível, a representação e colaboração da “Belíssimos Collectibles” foi essencial para a equipa de trabalho, para transmitir à equipa de desenvolvimento informações relevantes sobre o produto a realizar e as suas necessidades. Neste caso, apenas o Rui Afonso representa a empresa.

A prol do sucesso da aplicação, foi fundamental o contacto com os futuros utilizadores do software, o público-alvo da aplicação, para se poder perceber como se poderia melhorar a plataforma sobre a interação que possuem com o sistema e permitir identificar falhas e a forma como podem ser resolvidas.

# 1.7. Plano de execução do trabalho

O projeto encontra-se dividido em duas partes distintas. A fundamentação (fase 1) e a especificação (fase 2) têm como objetivo analisar e projetar todos os requisitos e use cases operacionais e funcionais. Para a fase da especificação foi dada especial atenção ao levantamento de requisitos, a construção de modelos e diagramas que vão servir de fundamentação para a implementação do software pretendido.

A construção foca-se no desenvolvimento do software que engloba a lógica de negócios, base de dados e interface UI. Concluída esta fase obtemos o produto final da aplicação pronto a ser usado pelo público-alvo.

Efetivamente, o diagrama de *Gantt* é uma ferramenta que permite a pôr em perspetiva o tempo disponível para o desenvolvimento de todo o projeto. Por isso, optamos por construir um diagrama de *Gantt* de forma a preparar e planear o projeto, respeitando as datas propostas.

O primeiro diagrama (primeira imagem) corresponde à parte um e o segundo diagrama (segunda imagem) corresponde à parte dois.

## Plano de Execução do Trabalho

Belíssimos Collectibles

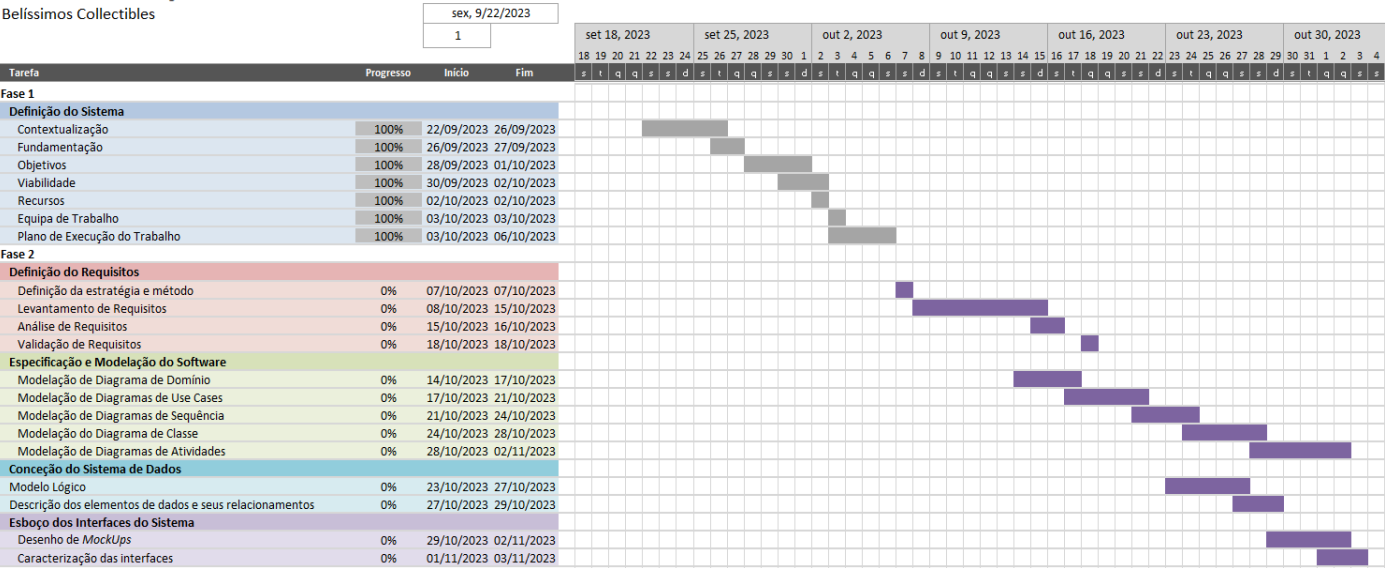


Figura 1 - Diagrama de Gantt parte 1.

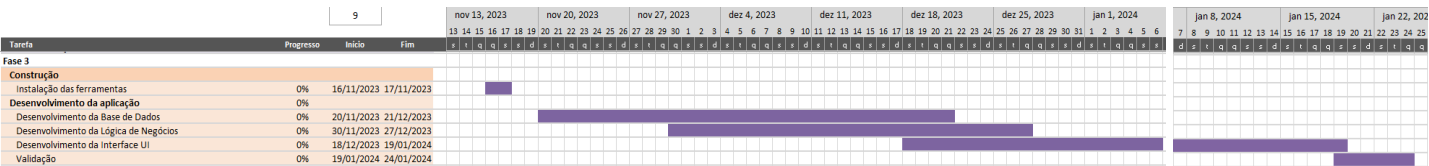


Figura 2 - Diagrama de Gantt parte 2.

A parte um tem uma duração de mais de um mês, com o objetivo de ficar concluída até ao dia 3 de novembro, enquanto a parte dois tem a duração de aproximadamente dois meses. Na primeira parte, decidimos dedicar mais tempo a etapa do Levantamento de Requisitos, comparado com as outras tarefas desta fase, visto que seria uma tarefa que precisaria ser bem estruturada e pensada, pois define todos os passos a tomar na restante execução do projeto e o formato da aplicação e implica bastante atenção. Adicionalmente, na fase de especificação foram definidos, aproximadamente, cerca de 4 a 6 dias para cada diagrama para ficarem bem estabelecidos e concluídos.

Para a seguinte parte, tomamos a decisão de alargar os prazos previstos, atribuindo mais tempo em cada etapa, tendo em consideração a disponibilidade da equipa na época definida. Como boa prática, consideramos reservar alguns dias no final da segunda fase, antes do prazo limite, para eventuais atrasos e caso exista necessidade de compensar algum imprevisto que ocorra no desenvolvimento.

Durante a primeira parte, este diagrama sofreu alterações, pois, após uma análise mais aprofundada, reparamos que os tempos não teriam sido atribuídos de forma completamente prática, visto que ponderamos que a execução de algumas tarefas iriam demorar mais tempo do que aquele que foi inicialmente atribuído.

Tal como foi mencionado anteriormente, estava previsto a primeira parte tivesse uma duração de pouco mais de um mês, enquanto a parte dois a duração de aproximadamente dois meses. No entanto, na primeira fase, foram planeados a realização de alguns diagramas que não foram totalmente concluídos a tempo, pois consideramos que estes necessitariam de extra revisão e seria melhor apresentá-los quando tivéssemos completa certeza dos mesmos. Assim, uma porção do tempo, no início da realização da segunda parte, foi aproveitada para rever os diagramas em falta, antes de então prosseguirmos para a implementação do projeto, tornando assim, na realidade, a duração da realização da fase 3 um pouco menor do que aquela que estávamos inicialmente à espera.

## **2. Definição de Requisitos**

### **2.1. Apresentação da estratégia e método**

Após definida a ideia geral do projeto que se pretende desenvolver e de esboçar algumas funcionalidades do programa, planeou-se algumas reuniões com o cliente, no caso será o representante da “Belíssimos Collectibles”, o Rui Afonso, com o objetivo de descobrir como atuam os leilões na área de negócio pretendida. A ata da primeira reunião encontra-se nos Anexos deste documento. Excluindo esta primeira, foram realizadas algumas outras para definir bem o caso de estudo a ter em conta.

Posteriormente, realizou-se alguns inquéritos aos típicos clientes dos leilões presenciais, aos utilizadores do pacato site e à restante comunidade que se mostrou interessada no projeto, para fundamentar como este programa seria usufruída pelos seus futuros utilizadores, as suas necessidades e como se poderia satisfazer ao máximo todas as entidades envolvidas. Foram ainda analisadas funcionalidades e características de algumas plataformas de leilão para identificar possíveis abordagens para a resolução dos problemas e objetivos levantados.

Assim, tendo em consideração os resultados deste inquérito, obtivemos um conjunto completo de funcionalidades que seria interessante adicionar para atrair a atenção dos consumidores.

### **2.2. Descrição geral dos requisitos (funcionais e não funcionais) levantados**

#### **Requisitos Funcionais:**

#### **2.2.1. Registo do Utilizador**

##### **Requisitos de Utilizador:**

- O utilizador terá de se registar na aplicação para poder utilizar a mesma.
- Efetuado o registo, o utilizador deverá poder alterar a sua informação conforme desejar.

##### **Requisitos de Sistema:**

- O sistema recolhe o primeiro e último nome, um username, email, password, número telefónico e morada.

- O sistema verifica se o username já consta na base de dados, nesse caso o sistema rejeita o registo e pede ao cliente para escolher um outro username.
- O sistema verifica se o e-mail já está associado a uma conta, nesse caso o sistema rejeita o registo e avisa o cliente que o email introduzido já está associado a uma conta.
- O sistema deve ainda associar um saldo inicial de 0€ à conta.
- O sistema deve armazenar os dados na Base de Dados.
- O sistema deve informar em caso de sucesso ou insucesso do registo.

### **2.2.2. Autenticação do Utilizador**

#### **Requisitos de Utilizador:**

- O utilizador deve poder iniciar a sessão na aplicação.

#### **Requisitos de Sistema:**

- O sistema solicita os dados de acesso do utilizador (email/username e password).
- O sistema verifica se o username/email consta na base dados e se a password introduzida está associada ao mesmo.

### **2.2.3. Eliminar Conta**

#### **Requisitos de Utilizador:**

- O utilizador deve conseguir apagar a sua conta assim como os dados associados a ela.

#### **Requisitos de Sistema:**

- O sistema deve enviar uma mensagem de confirmação após o utilizador requisitar a eliminação da conta.
- Após a confirmação, o sistema deve apagar todos os dados associados a ela da base de dados.

### **2.2.4. Alterar informações da conta**

#### **Requisitos de Utilizador:**

- O utilizador deve ser capaz de alterar as informações do seu perfil.

#### **Requisitos de Sistema:**

- O sistema não deve permitir a alteração do username ou do email.
- O sistema deve permitir a alteração de todas as outras informações.
- O sistema deve alterar e armazenar as informações novas na base de dados.

### **2.2.5. Iniciar um leilão para um produto**

#### **Requisitos de Utilizador:**

- O utilizador deve ter uma forma de colocar produtos que deseja em leilão.

#### **Requisitos de Sistema:**

- O sistema irá pedir ao vendedor para seleccionar o tipo de leilão que deseja.
- O sistema deve permitir inserir diversas informações, como uma descrição do produto, estado, fotos do produto e que comprovem a autenticação do mesmo.
- O sistema requisita também que seja definido em qual tipo de produto ele se encaixa e a marca do mesmo.
- Será preciso ainda o utilizador definir um preço inicial e um limite de tempo para o leilão acabar.
- O sistema deverá colocar o leilão no ar, e acrescentar o mesmo à base de dados após todas as informações serem preenchidas e caso nenhuma anomalia seja detetada.

### **2.2.6. Interromper/Parar um leilão**

#### **Requisitos de Utilizador:**

- O usuário deve ser capaz de cancelar ou interromper de imediato um leilão.

#### **Requisitos de Sistema:**

- O sistema deverá apresentar uma mensagem com as opções de cancelar o leilão, interromper no exato momento ou continuar.
- Caso selecione a opção de cancelar totalmente o leilão o produto não será vendido e o leilão será imediatamente cancelado.
- Na circunstância do vendedor querer interromper antes do tempo limite terminar, o leilão para de imediato e o cliente com o lance mais alto do momento fica com o produto.
- Caso seja selecionada a opção de continuar, o leilão irá prosseguir normalmente.
- O sistema deve notificar os participantes da ação tomada e caso haja um vencedor, avisar o mesmo.
- O sistema deve informar o vendedor que a sua ação selecionada foi concluída com sucesso.
- O sistema deve ainda retirar o leilão da base de dados e acrescentar o saldo pendente à conta.

### **2.2.7. Consultar o estado dos leilões**

#### **Requisitos de Utilizador:**

- O vendedor e os clientes devem ser capazes de consultar o estado a que se encontra o leilão num preciso momento.

**Requisitos de Sistema:**

- O sistema deve apresentar os últimos 5 maiores lances e o horário que ocorreram.
- O sistema deve apresentar o tempo que falta até o leilão ser encerrado.

## **2.2.8. Alterar informações dos leilões**

**Requisitos de Utilizador:**

- O vendedor deve ser capaz de alterar ou acrescentar informações enquanto um leilão ainda ocorre.

**Requisitos de Sistema:**

- O sistema deve apresentar exclusivamente para o vendedor um botão de editar.
- Após selecionar a opção de editar o sistema deve apresentar a página com as informações inicialmente introduzidas pelo utilizador.
- O sistema não deve permitir que o preço inicial e o tempo limite sejam alterados.
- O sistema deve permitir que todas as outras informações sejam alteradas ou mais informação seja acrescentada à descrição do produto e fotos.
- O sistema deve acrescentar e alterar todas essas informações à base de dados.

## **2.2.9. Pesquisar produtos**

**Requisitos de Utilizador:**

- O utilizador deve poder pesquisar produtos em leilão à sua escolha pelo seu nome.
- O utilizador deve poder filtrar os resultados consoante o tipo de leilão que procura, a categoria e a marca do produto.

**Requisitos de Sistema:**

- O sistema deve disponibilizar uma caixa de texto de pesquisa ao usuário e um botão de pesquisar que permita aplicar filtros.
- O sistema deve recolher todos os leilões na base de dados que coincidem com esses critérios.
- O sistema deve apresentar uma página com todos os resultados relacionados com o pedido feito.

## **2.2.10. Fazer uma licitação**

**Requisitos de Utilizador:**



- O utilizador deve ser capaz de participar num leilão dependendo do tipo do mesmo.
  - Leilão inglês → Um lance deve ter um valor superior ao maior lance atual.
  - Leilão holandês → O preço é descendente e o primeiro lance leva o produto.

**Requisitos de Sistema:**

- O sistema deverá conter um botão de lance, tanto no modelo inglês como no holandês.
- Em caso de leilão inglês, o sistema deverá apresentar uma caixa de texto em que o utilizador colocará o valor (em euros) do lance.
- O sistema deve verificar se o valor do lance é igual ou inferior ao saldo disponível da conta. Se tiver feito outras licitações em leilões ativos, passa a verifica o seu “saldo fantasma”.
- Caso o saldo do utilizador seja inferior ao lance, o sistema não possibilita que o lance seja concluído.
- O sistema deve informar que a operação foi concluída com sucesso, ou em caso de alguma anomalia ser detetada, que a operação não pode ser concluída.
- O sistema não deve permitir que o utilizador que coloca o leilão ativo faça uma licitação

## **2.2.11. Consultar a conta de um utilizador**

**Requisitos de Utilizador:**

- O utilizador deverá poder consultar o seu perfil como o de outros utilizadores.

**Requisitos de Sistema:**

- O sistema deverá mostrar uma página com todos os leilões ao vivo no momento de um certo usuário, assim como as suas avaliações e a sua avaliação média.

## **2.2.12. Inserir dinheiro na conta**

**Requisitos de Utilizador:**

- O utilizador deverá poder adicionar dinheiro ao saldo da conta.

**Requisitos de Sistema:**

- O sistema deverá conter um botão no perfil do mesmo com a opção de adicionar saldo.
- O sistema deve informar que a operação foi concluída com sucesso, ou em caso de alguma anomalia ser detetada, que a operação não pode ser concluída.

## **2.2.13. Avaliar um vendedor**

**Requisitos de Utilizador:**

- O utilizador após ganhar um leilão e receber o produto deve poder avaliar o vendedor atribuindo de 1 até 5 estrelas mediante o quão fiel à descrição o produto era e o estado da embalagem (se chegou danificado ou não).

**Requisitos de Sistema:**

- O sistema deverá ter a opção para o utilizador notificar que já recebeu o produto.
- Após o sistema deve apresentar uma caixa com 5 estrelas a ser seleccionadas e uma caixa de texto para descrição.
- O sistema deve adicionar a nova avaliação ao perfil do vendedor.
- O sistema deve atualizar a classificação média do vendedor.
- As novas informações têm de ser guardadas na base de dados.

## 2.2.14. Terminar Sessão

**Requisitos de Utilizador:**

- O utilizador deverá poder terminar a sessão da conta.

**Requisitos de Sistema:**

- O sistema deve disponibilizar um botão para terminar a sessão.
- O sistema deve terminar a sessão do utilizador.

## Requisitos não funcionais:

- A aplicação deverá ser de fácil uso, com um design simples e o mais acessível e intuitivo possível para todos os utilizadores.
- A aplicação deverá estar disponível durante 7 dias por semana, 24 horas por dia.
- O sistema deve ser suportado por vários browsers (Google Chrome, Microsoft Edge, Firefox, Safari, Opera, entre outros...).
- A aplicação deve destacar os leilões ativos no momento.
- A privacidade dos dados dos utilizadores deve ser garantida.

## 2.3. Validação dos requisitos estabelecidos

Antes de passar à próxima fase de especificação, para assegurar que os requisitos estabelecidos eram relevantes e estavam corretamente identificados, foram estabelecidas novas reuniões entre o cliente, a equipa de desenvolvimento e alguns potenciais novos utilizadores. Face à aprovação obtida pelo cliente no final das reuniões, foi possível avançar as próximas fases do projeto.

## 3. Especificação e Modelação do Software

### 3.1. Apresentação geral da especificação

Após a conclusão o levantamento e análise de requisitos e a sua validação, demos início à modelação do software que pretendemos criar e para tal desenvolveu-se alguns diagramas UML que irão de servir de representação abstrata das funcionalidades que se retirou dos requisitos levantados previamente.

É de elevada importância salientar que, na nossa aplicação, apenas o utilizador é capaz de interagir com o sistema, no entanto foram feitas algumas alterações nos diagramas apresentados para demonstrar certas especificações. Também definimos nesta etapa que grande parte das funcionalidades implementadas só poderiam ser usufruídas caso o utilizador se encontrasse autenticado, deste modo, caso o cliente não proceda ao registo e à autenticação ver-se-á limitado.

### 3.2. Aspetos estruturais

#### 3.2.1. Modelo de Domínio

Optamos por desenvolver modelos de domínio pois estes apresentam de uma forma muito abstrata os objetos que consideramos essenciais para a implementação do software.

Assim, desenvolvemos o modelo de domínio que melhor representa o nosso projeto, a área de negócio que estamos a trabalhar e as entidades que pretendemos representar. Com base nos requisitos levantados, concluímos que seria necessário definir duas entidades principais, o **Utilizador** e o **Leilão**, visto que o sistema funciona a partir das interações entre essas entidades. O Utilizador tem associado um **Nome Completo**, isto é, o primeiro e último nome do mesmo, um **Username**, **Email**, **Password**, **Número Telefónico**, **Morada** e **Saldo** da conta, que se encontra a 0€ quando cria a conta e a qual pode adicionar mais dinheiro. Já o Leilão tem de possuir um **Tipo**, que poderá ser **Inglês** ou **Holandês** e implementará regras diferentes no Leilão criado consoante o seu tipo, possui também um **Preço Inicial**, **Tempo Limite** que define quanto tempo terá o Leilão, e um **Produto** que por sua vez tem atribuído **Nome**, **Fotos**, **Marca**, **Descrição**, **Categoria** e um **Estado**.

O Utilizador tanto pode ser um **Vendedor** ou um **Cliente** de um Leilão, quando cria um leilão ou assim que participa noutros leilões, respetivamente. O Cliente pode alternativamente fazer uma **Licitação** com um **Preço**, no tipo de leilão inglês e holandês. Consideramos também que o Cliente deverá, posteriormente, avaliar o Vendedor dos leilões que ganhou.

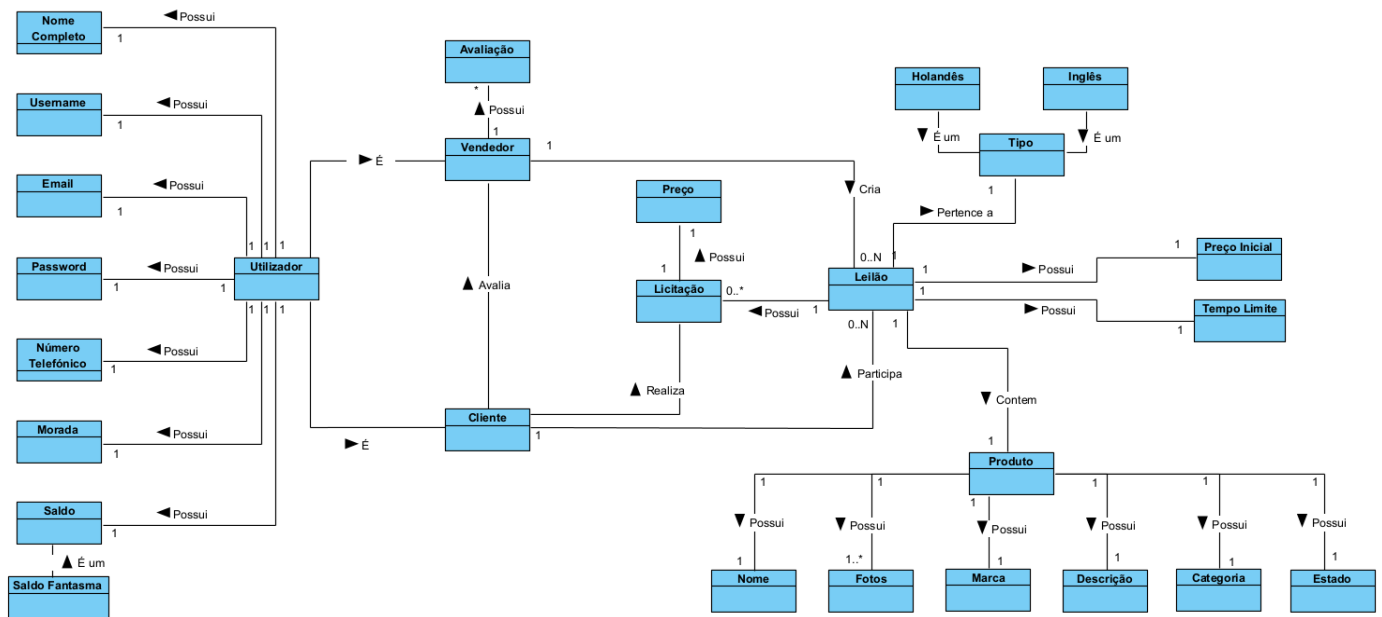


Figura 3 - Modelo de Domínio

### 3.3. Aspectos Comportamentais

#### 3.3.1 Diagrama de Use Cases

Os diagramas de use cases permitem-nos definir como o utilizador poderá interagir com o software e, consequente, o comportamento do sistema deve ter face aos pedidos do utilizador. Deste modo, estes diagramas representam de forma estruturada os requisitos levantados e permitem analisar com mais detalhe que funcionalidades devemos implementar.

No caso desta aplicação, existe apenas um Ator, o Utilizador, visto que este é o único que irá interagir com o sistema. No entanto, como ficou definido nos use cases que grande parte das funcionalidades do sistema apenas podem ser utilizadas caso o utilizador se encontre autenticado na aplicação, no diagrama a baixo existirão dois Atores, o **Utilizador autenticado** e o **Utilizador não autenticado**, de forma a distinguir que funcionalidades podem ser utilizadas quando o utilizador se encontra autenticado ou não.

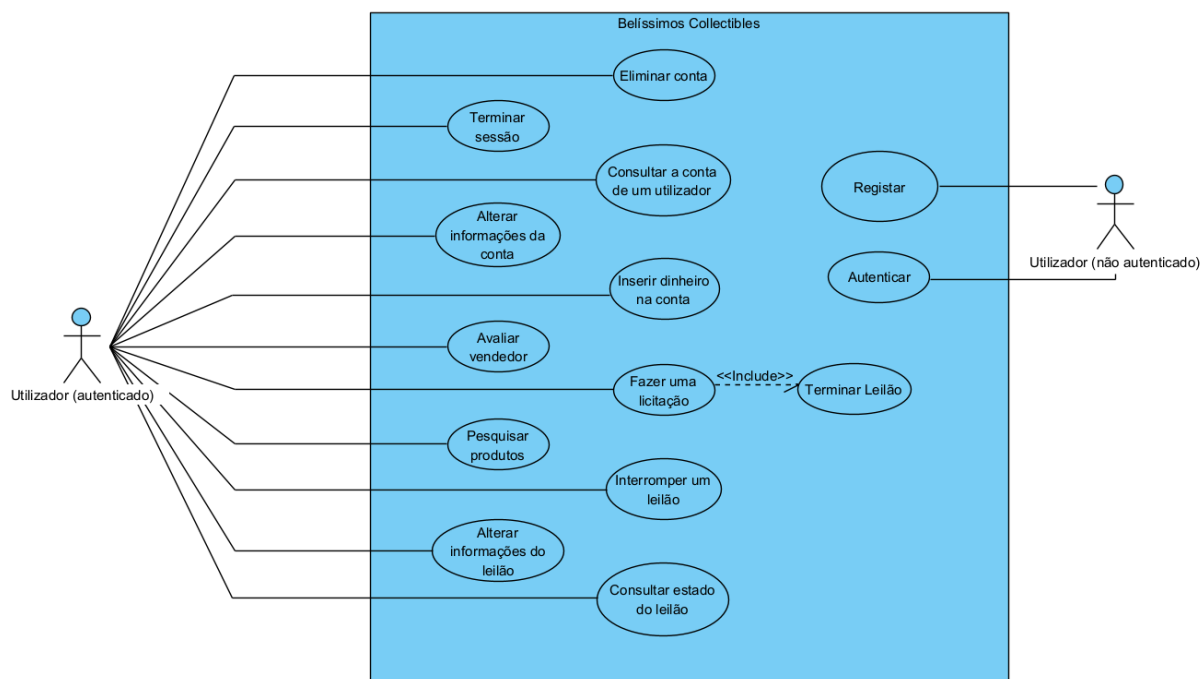


Figura 4 - Diagrama de Use Cases

Segue-se então, abaixo, as especificações de cada uma das funcionalidades referidas do sistema:

| USE CASE:         | Registar Utilizador   |
|-------------------|---|
| DESCRIÇÃO:        | É registado um utilizador na aplicação  |
| PRÉ-CONDIÇÃO:     | O Utilizador não possui nenhuma conta na aplicação  |
| PÓS-CONDIÇÃO:     | Utilizador é registado na aplicação e a sua conta é criada.   |
| FLUXO NORMAL:     | 1. Utilizador introduz um Username<br>2. Sistema valida que Username não existe.<br>3. Utilizador preenche restantes campos necessários para o registo.<br>4. Sistema recolhe informações do Nome (Primeiro e Último Nome), Email, Password, Número telefónico e Morada.<br>5. Sistema valida que Email não está associado a outra conta.<br>6. Sistema valida os restantes dados fornecidos.<br>7. Sistema associa um ID e um saldo de 0€ à conta.<br>8. Sistema regista Utilizador.<br>9. Sistema informa que a conta foi criada com sucesso. |
| FLUXO DE EXCEÇÃO: | <b>(1) [Username já utilizado] (passo 2)</b><br>2.1. Sistema informa que já existe uma conta com Username inserido.   |
| FLUXO DE EXCEÇÃO: | <b>(2) [Email já utilizado] (passo 5)</b>   |

|                   |   |
|-------------------|---|
|                   | 5.1. Sistema informa que já existe uma conta com o email fornecido.               |
| FLUXO DE EXCEÇÃO: | <b>(3) [Sistema recebe um número telefónico ou uma morada inválida] (passo 6)</b> |
|                   | 6.1. Sistema informa que dados fornecidos são inválidos.                          |

Tabela 1 - Use Case “Registo do Utilizador”.

O registo do utilizador é um use case, que permite aos utilizadores que não se encontram no sistema, de criar uma conta e se adicionarem ao sistema para poderem autenticar-se posteriormente.

|                   |   |
|-------------------|---|
| <b>USE CASE:</b>  | <b>Autenticar</b>   |
| DESCRIÇÃO:        | Utilizador autentica-se na aplicação com a sua conta            |
| PRÉ-CONDIÇÃO:     | Nenhuma conta encontra-se autenticada na aplicação              |
| PÓS-CONDIÇÃO:     | Utilizador está autenticado                                     |
| FLUXO NORMAL:     | 1. Utilizador insere o seu email ou username, e a palavra-passe |
|                   | 2. Sistema verifica se o email ou username existem              |
|                   | 3. Sistema verifica a validade da palavra-passe                 |
|                   | 4. Sistema autentica o Utilizador com sucesso                   |
| FLUXO DE EXCEÇÃO: | <b>(1) [A conta não existe no Sistema] (passo 2)</b>            |
|                   | 2.1. Sistema que a conta não existe                             |
| FLUXO DE EXCEÇÃO: | <b>(2) [A palavra-passe é inválida] (passo 3)</b>               |
|                   | 3.1. Sistema informa que a palavra-passe fornecida é inválida   |

Tabela 2 - Use Case “Autenticar”.

O utilizador deve realizar este processo para poder utilizar a aplicação, visto que existem processos que só podem ser realizados com a sessão iniciada. Para tal, indica o seu email ou username (nome de utilizador) e a sua password (palavra-passe) e, caso estejam corretos, passa a poder utilizar a aplicação.

|                  |   |
|------------------|---|
| <b>USE CASE:</b> | <b>Alterar informações da conta</b>   |
| DESCRIÇÃO:       | Utilizador altera as informações da sua conta   |
| PRÉ-CONDIÇÃO:    | Utilizador está autenticado   |
| PÓS-CONDIÇÃO:    | As informações da conta são alteradas   |
| FLUXO NORMAL:    | 1. Utilizador seleciona a opção de editar conta.  |
|                  | 2. Sistema apresenta os dados que permite alterar, isto é, todos os dados menos username e email do utilizador. |
|                  | 3. Utilizador insere os dados que deseja alterar.   |

|                   |   |
|-------------------|---|
|                   | 4. Utilizador guarda as alterações.   |
|                   | 5. Sistema atualiza os dados na base de dados e retorna a página do perfil. |
| FLUXO DE EXCEÇÃO: | <b>(1) [Utilizador cancela alteração] (passo 4)</b>                         |
|                   | 4.1. Sistema retorna à página do perfil                                     |

Tabela 3 - Use Case “Alterar informações da conta”.

Caso o utilizador pretenda, poderá editar algumas das suas informações pessoais, isto é, todas menos o seu username e email pois estas irão identificar o utilizador no sistema. No fim pode guardar ou cancelar as alterações feitas.

|                   |   |
|-------------------|---|
| <b>USE CASE:</b>  | <b>Eliminar Conta</b>   |
| DESCRIÇÃO:        | Um utilizador apaga a sua conta da aplicação                                      |
| PRÉ-CONDIÇÃO:     | Utilizador está autenticado   |
| PÓS-CONDIÇÃO:     | Utilizador não tem conta registada e os dados associados a ela deixam de existir. |
| FLUXO NORMAL:     | 1. Utilizador seleciona a opção de eliminar conta.                                |
|                   | 2. Sistema apresenta uma mensagem de confirmação da ação.                         |
|                   | 3. Utilizador confirma a ação.  |
|                   | 4. Sistema apaga todos os dados associados á conta.                               |
|                   | 5. Sistema retorna a tela inicial.  |
| FLUXO DE EXCEÇÃO: | <b>(1) [Utilizador cancela a eliminação] (passo 3)</b>                            |
|                   | 3.1. Sistema retorna a página do perfil.  |

Tabela 4 - Use Case “Eliminar Conta”.

Na eventualidade do utilizador se encontrar insatisfeito ou já não tenha utilidade para ela, pode eliminar a sua conta e consequentemente apagar todos os dados que se possui no sistema pode pedir ao sistema que o faça. Deste modo, todos os seus dados serão permanentemente perdidos e não poderá voltar a ter acesso a estes. Assim, como é uma funcionalidade com muito peso, o utilizador deve confirmar a ação no final.

|                  |   |
|------------------|---|
| <b>USE CASE:</b> | <b>Iniciar um leilão</b>  |
| DESCRIÇÃO:       | Um utilizador coloca um produto para leilão   |
| PRÉ-CONDIÇÃO:    | Utilizador está autenticado   |
| PÓS-CONDIÇÃO:    | Um leilão fica ativo  |
| FLUXO NORMAL:    | 1. Utilizador seleciona o tipo de leilão (leilão inglês ou holandês) que quer realizar. |

|                   |  |
|-------------------|--|
|                   | 2. Sistema regista o tipo de leilão selecionado.   |
|                   | 3. Utilizador introduz as informações necessárias.   |
|                   | 4. Sistema recolhe informações para o leilão, isto é, o nome do produto, a descrição do mesmo, estado, fotos, tipo de produto, marca, preço inicial (em euros) e tempo limite. |
|                   | 5. Sistema valida os dados fornecidos.   |
|                   | 6. Sistema acrescenta as informações à base de dados.  |
|                   | 7. Sistema inicia leilão e torna-o público.  |
| FLUXO DE EXCEÇÃO: | <b>(1) [Dados não são válidos] (passo 5)</b>   |
|                   | 5.1. Sistema informa que dados não são válidos.  |

Tabela 5 - Use Case "Iniciar Leilão".

|                    |  |
|--------------------|--|
| <b>USE CASE:</b>   | <b>Interromper um leilão</b>   |
| DESCRIÇÃO:         | Um utilizador pretende parar um leilão.  |
| PRÉ-CONDIÇÃO:      | Leilão foi iniciado pelo utilizador e ainda se encontra ativo.                               |
| PÓS-CONDIÇÃO:      | Leilão termina.  |
| FLUXO NORMAL:      | 1. Sistema apresenta as opções "Interromper Leilão", "Cancelar Leilão" e "Continuar Leilão". |
|                    | 2. Utilizador seleciona a opção "Interromper Leilão".  |
|                    | 3. Sistema para o leilão.  |
|                    | 4. Sistema regista que o utilizador no topo na lista de lances ganha o leilão.               |
|                    | 5. Sistema notifica o vencedor que ganhou.   |
|                    | 6. Sistema notifica participantes da ação feita.   |
|                    | 7. Sistema acrescenta saldo pendente às contas dos utilizadores que fizeram lances.          |
|                    | 8. Sistema informa que ação foi realizada com sucesso.                                       |
|                    | 9. Sistema retira o leilão e as suas informações da base de dados.                           |
| FLUXO ALTERNATIVO: | <b>(1) [Utilizador seleciona a opção "Cancelar Leilão"] (passo 2)</b>                        |
|                    | 2.1. Sistema para o leilão.  |
|                    | 2.2. Regressa a 6.   |
| FLUXO DE EXCEÇÃO:  | <b>(2) [Utilizador seleciona a opção "Continuar Leilão"] (passo 2)</b>                       |
|                    | 2.1. Sistema informa que o leilão continua ativo.  |

Tabela 6 - Use Case "Interromper Leilão".



Decidimos que deveríamos permitir ao utilizador parar os seus leilões antes do fim do tempo limite previamente definido e habilitar que defina o que acontece quando realiza esta funcionalidade. Deste modo, pode “Interromper Leilão” e, caso existam lances, o utilizador com maior lance ganha, e poderá também “Cancelar Leilão” onde ninguém ganha. A funcionalidade dá insucesso caso escolha “Continuar Leilão” visto que o leilão continua ativo.

| USE CASE:     | Consultar estado do leilão  |
|---------------|---|
| DESCRIÇÃO:    | Utilizador, seja este tanto o vendedor do leilão ou outro cliente, consulta o estado de um leilão.                  |
| PRÉ-CONDIÇÃO: | O utilizador tem de estar autenticado e o leilão a consultar a ocorrer.   |
| PÓS-CONDIÇÃO: | Um leilão foi consultado.   |
| FLUXO NORMAL: | 1. Utilizador seleciona o leilão que pretende consultar   |
|               | 2. Sistema apresenta os 5 maiores lances até ao momento, as horas a que esses ocorreram e as informações do leilão. |
|               | 3. Sistema apresenta o tempo que falta até ao leilão encerrar.  |

Tabela 7 - Use Case “Consultar estado do leilão”.

| USE CASE:     | Alterar informações do leilão   |
|---------------|---|
| DESCRIÇÃO:    | Um vendedor edita as informações do seu leilão  |
| PRÉ-CONDIÇÃO: | Utilizador está autenticado e é o vendedor do leilão.   |
| PÓS-CONDIÇÃO: | A informação do leilão é alterada   |
| FLUXO NORMAL: | 1. Sistema apresenta os campos editáveis do leilão, isto é, descrição do produto, estado e fotos. |
|               | 2. Utilizador altera campos que pretende.   |
|               | 3. Sistema atualiza as informações do leilão.   |

Tabela 8 - Use Case “Alterar informações do leilão”.

O utilizador tem a capacidade de alterar os campos editáveis dos seus leilões, isto é, a descrição do produto, o seu estado ou as fotos definidas.

| USE CASE:     | Fazer uma licitação  |
|---------------|--|
| DESCRIÇÃO:    | Utilizador participa num leilão inglês ou holandês que se encontra a decorrer. |
| PRÉ-CONDIÇÃO: | Utilizador está autenticado e o leilão é do tipo inglês ou holandês.           |
| PÓS-CONDIÇÃO: | O lance do utilizador fica registado.  |

|                    |   |
|--------------------|---|
| FLUXO NORMAL:      | 1. Utilizador seleciona a opção de Lance no leilão que pretende participar.   |
|                    | 2. Sendo o leilão do tipo inglês, o Sistema apresenta uma caixa de texto para receber o valor.                          |
|                    | 3. Utilizador introduz o valor do lance (em euros).   |
|                    | 4. Sistema valida se o valor do lance é superior ao maior lance feito até ao momento.                                   |
|                    | 5. Sistema valida se o lance é igual ou superior ao saldo da conta do utilizador.                                       |
|                    | 6. Sistema coloca o valor do lance como saldo pendente.   |
|                    | 7. Sistema valida que ainda existe mais de 1 minuto no tempo limite no momento.   |
|                    | 8. Sistema atualiza a lista das licitações, colocando o lance em primeiro lugar com o username, valor e hora do lance.  |
| FLUXO ALTERNATIVO: | <b>(1) [Leilão é do tipo holandês] (passo 2)</b>  |
|                    | 2.1. Sistema altera o tempo limite para 0.  |
|                    | 2.2. Sistema atualiza a lista de licitações, colocando o lance em primeiro lugar com o username, valor e hora do lance. |
|                    | 2.2. <<include>> Terminar Leilão.   |
| FLUXO DE EXCEÇÃO:  | <b>(2) [Lance é inferior ou igual ao maior lance atual] (passo 4)</b>   |
|                    | 4.1. Sistema informa que lance é inválido.  |
| FLUXO DE EXCEÇÃO:  | <b>(3) [Saldo da conta é inválido] (passo 5)</b>  |
|                    | 5.1. Sistema informa que saldo da conta do Utilizador não é suficiente para fazer lance.                                |
| FLUXO ALTERNATIVO: | <b>(4) [O tempo limite no momento é igual ou inferior a 1 minuto] (passo 7)</b>   |
|                    | 7.1. Sistema atualiza o tempo limite e adiciona-lhe mais 5 minutos.   |
|                    | 7.2. Regressa a 8.  |

Tabela 9 - Use Case “Fazer uma licitação”.

Na realização do último Use Case surgiram questões sobre o funcionamento dos leilões em casos de exceção, deste modo decidiu-se criar o Use Case “Terminar Leilão” para que se pudesse tratar do fluxo de exceção de um modo mais geral. Esta funcionalidade realiza-se assim que o tempo limite do leilão acaba, e é responsável por verificar a existência de vencedores, notificar os utilizadores envolvidos e, finalmente, retirar todas as informações do sistema. Consoante o tipo de leilão, a pré-condição para este novo caso de uso, ou seja, o final do leilão, é obtida de diferentes maneiras. Num Leilão Inglês o tempo terá de chegar naturalmente ao fim. No caso de ser um Leilão Holandês, assim que um determinado utilizador fizer uma licitação, o tempo passa a ser zero e o leilão acaba imediatamente. Finalmente, no tipo Holandês sentiu-se a

necessidade de utilizar e atualizar a lista de melhores lances no leilão pois assim o sistema necessitaria de apenas uma forma para verificar a existência de vencedores.

A criação deste Use Case permitiu implementar uma forma de o leilão acabar mesmo que não exista participantes de todo ou que terminem o leilão antes do tempo previsto, deste modo, não sentimos que seria correto colocar estes passos nos fluxos normais das funcionalidades mencionadas.

|                           |   |
|---------------------------|---|
| <b>USE CASE:</b>          | <b>Terminar Leilão</b>  |
| <b>DESCRIÇÃO:</b>         | Um leilão dá-se por terminado   |
| <b>PRÉ-CONDIÇÃO:</b>      | Leilão tem de estar ativo e o tempo limite acaba (tempo limite é igual a 0).  |
| <b>PÓS-CONDIÇÃO:</b>      | Leilão é apagado do sistema.  |
| <b>FLUXO NORMAL:</b>      | 1. Sistema para leilão.   |
|                           | 2. Sistema verifica que há um vencedor, ou seja, o vendedor que se encontra no topo da lista de lances.             |
|                           | 3. Sistema notifica utilizador que ganhou leilão.   |
|                           | 4. Sistema notifica restantes utilizadores que participaram do final do leilão.                                     |
|                           | 5. Sistema acrescenta saldo pendente, isto é o valor das licitações, às contas dos utilizadores que fizeram lances. |
|                           | 6. Sistema notifica o vendedor sobre o final do leilão e o seu estado (top 5 melhores lances).                      |
|                           | 7. Sistema retira o leilão e as suas informações da base de dados.  |
| <b>FLUXO ALTERNATIVO:</b> | <b>(1) [Não há utilizadores na lista de lances] (passo 3)</b>   |
|                           | 3.1. Sistema informa vendedor sobre o final do leilão e que não houve lances realizados.                            |
|                           | 3.2. Regressa a 7   |

Tabela 10 - Use Case "Terminar leilão".

|                      |  |
|----------------------|--|
| <b>USE CASE:</b>     | <b>Pesquisar Produto</b>   |
| <b>DESCRIÇÃO:</b>    | Um utilizador pesquisa produtos em leilão.                                 |
| <b>PRÉ-CONDIÇÃO:</b> | Utilizador está autenticado.   |
| <b>PÓS-CONDIÇÃO:</b> | Sistema disponibiliza vários leilões que contêm o produto inserido.        |
| <b>FLUXO NORMAL:</b> | 1. Utilizador indica o nome do produto que procura.                        |
|                      | 2. Sistema recolhe os leilões que possuem um produto com o nome fornecido. |

|                       |  |
|-----------------------|--|
|                       | 3. Sistema apresenta os leilões recolhidos.                                      |
| FLUXO<br>ALTERNATIVO: | <b>(1) [Utilizador filtra a pesquisa por tipo de leilão] (passo 1)</b>           |
|                       | 1.1. Sistema recolhe os leilões do tipo selecionado.                             |
|                       | 1.2. Regressa a 3.   |
| FLUXO<br>ALTERNATIVO: | <b>(2) [Utilizador filtra a pesquisa por categoria do produto] (passo 1)</b>     |
|                       | 1.1. Sistema recolhe os leilões que possuem um produto da categoria selecionada. |
|                       | 1.2. Regressa a 3.   |
| FLUXO<br>ALTERNATIVO: | <b>(3) [Utilizador filtra a pesquisa por marca do produto] (passo 1)</b>         |
|                       | 2. Sistema recolhe os leilões que possuem um produto da marca selecionada.       |
|                       | 1.3. Regressa a 3.   |

Tabela 11 - Use Case “Pesquisar produto”.

|                  |   |
|------------------|---|
| <b>USE CASE:</b> | <b>Consultar conta utilizador</b>   |
| DESCRIÇÃO:       | Utilizador consulta o seu perfil ou de outros utilizadores.   |
| PRÉ-CONDIÇÃO:    | Utilizador está autenticado   |
| PÓS-CONDIÇÃO:    | A conta desejada é consultada   |
| FLUXO NORMAL:    | 1. Sistema apresenta o perfil ao utilizador (com informações sobre o seu username e email, a sua avaliação média, as suas avaliações e todos os leilões que tem a decorrer no momento). |

Tabela 12 Use Case "Consultar conta utilizador”.

|                  |  |
|------------------|--|
| <b>USE CASE:</b> | <b>Avaliar vendedor</b>  |
| DESCRIÇÃO:       | O vencedor do leilão avalia o vendedor do leilão   |
| PRÉ-CONDIÇÃO:    | Utilizador tem de ganhar o leilão e as fases de pagamento/entrega têm de estar concluídas  |
| PÓS-CONDIÇÃO:    | O vendedor recebe uma avaliação  |
| FLUXO NORMAL:    | 1. Utilizador (vencedor) notifica que já recebeu o produto.  |
|                  | 2. Sistema apresenta a opção de avaliar o vendedor do produto.   |
|                  | 3. Utilizador seleciona a classificação de 1 a 5 estrelas e insere um comentário sobre a descrição do produto e o estado da embalagem. |
|                  | 4. Sistema calcula a classificação média do vendedor.  |
|                  | 5. Sistema atualiza as avaliações no perfil do vendedor.   |

|                       |  |
|-----------------------|--|
| FLUXO<br>ALTERNATIVO: | <b>(1) [Utilizador não deseja inserir um comentário] (passo 3)</b> |
|                       | 3.1. Utilizador seleciona a classificação de 1 a 5 estrelas.       |
|                       | 3.2. Regressa a 4.   |

Tabela 13 - Use Case "Avaliar vendedor".

|                  |   |
|------------------|---|
| <b>USE CASE:</b> | <b>Inserir dinheiro na conta</b>  |
| DESCRIÇÃO:       | O utilizador insere dinheiro na sua conta   |
| PRÉ-CONDIÇÃO:    | O utilizador está autenticado.  |
| PÓS-CONDIÇÃO:    | O saldo da conta do utilizador aumenta.   |
| FLUXO NORMAL:    | 1. Utilizador insere o valor (em euros) que pretende adicionar ao saldo da conta. |
|                  | 2. Sistema adiciona o valor inserido ao saldo existente na conta do utilizador.   |
|                  | 3. Sistema informa o sucesso do pedido.   |

Tabela 14 - Use Case "Inserir dinheiro na conta".

|                  |  |
|------------------|--|
| <b>USE CASE:</b> | <b>Terminar Sessão</b>   |
| DESCRIÇÃO:       | Utilizador termina sessão da sua conta da aplicação                                      |
| PRÉ-CONDIÇÃO:    | Utilizador está autenticado.   |
| PÓS-CONDIÇÃO:    | Utilizador não está autenticado.   |
| FLUXO NORMAL:    | 1. Utilizador seleciona a opção de terminar sessão.                                      |
|                  | 2. Sistema termina a sessão do utilizador e utilizador deixa assim de estar autenticado. |

Tabela 15 - Use Case "Terminar sessão".

### 3.3.2. Diagramas de Atividade

Os Diagramas de Atividade são utilizados para ilustrar o fluxo de controlo num sistema, referindo os passos envolvidos no seu processo, permitindo também facilitar a visualização do funcionamento dos Use Case e de todas as alternativas e estados envolvidos, através do encadeamento das ações que os compõem.

Decidimos representar apenas neste tipo de diagramas as funcionalidades **Iniciar um Leilão** e **Fazer uma Licitação**, pois representam as funcionalidades principais que este projeto necessita ter e porque sentimos que as restantes funcionalidades são de fácil compreensão. Esses diagramas encontram-se a seguir:

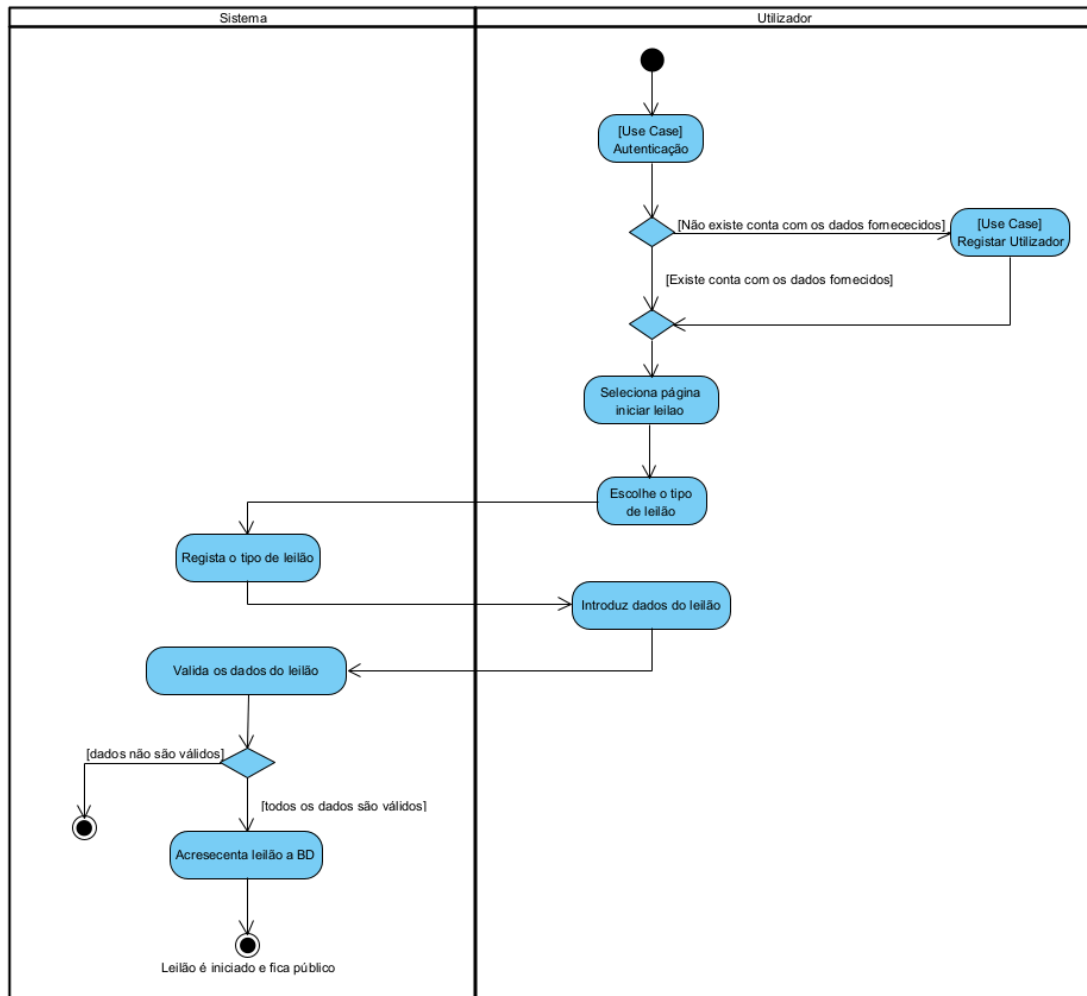


Figura 5 - Diagrama de Atividade "Iniciar um Leilão"

Neste diagrama de Atividade, decidimos representar o processo de um utilizador assumindo que irá entrar na aplicação no momento, apesar de no Use Case "Iniciar um Leilão" a pré-condição é o "Utilizador está autenticado", para podermos representar esse fluxo. Deste modo, inicialmente o utilizador deverá autenticar-se, caso já exista no sistema procede, caso contrário necessita registar-se para poder continuar.

De seguida, o utilizador deverá preencher as informações necessárias para o leilão e o sistema estará encarregue de verificar todos esses dados. Nesse instante, se algum dos dados não for válido a funcionalidade terminará, se não a funcionalidade irá terminar com sucesso.

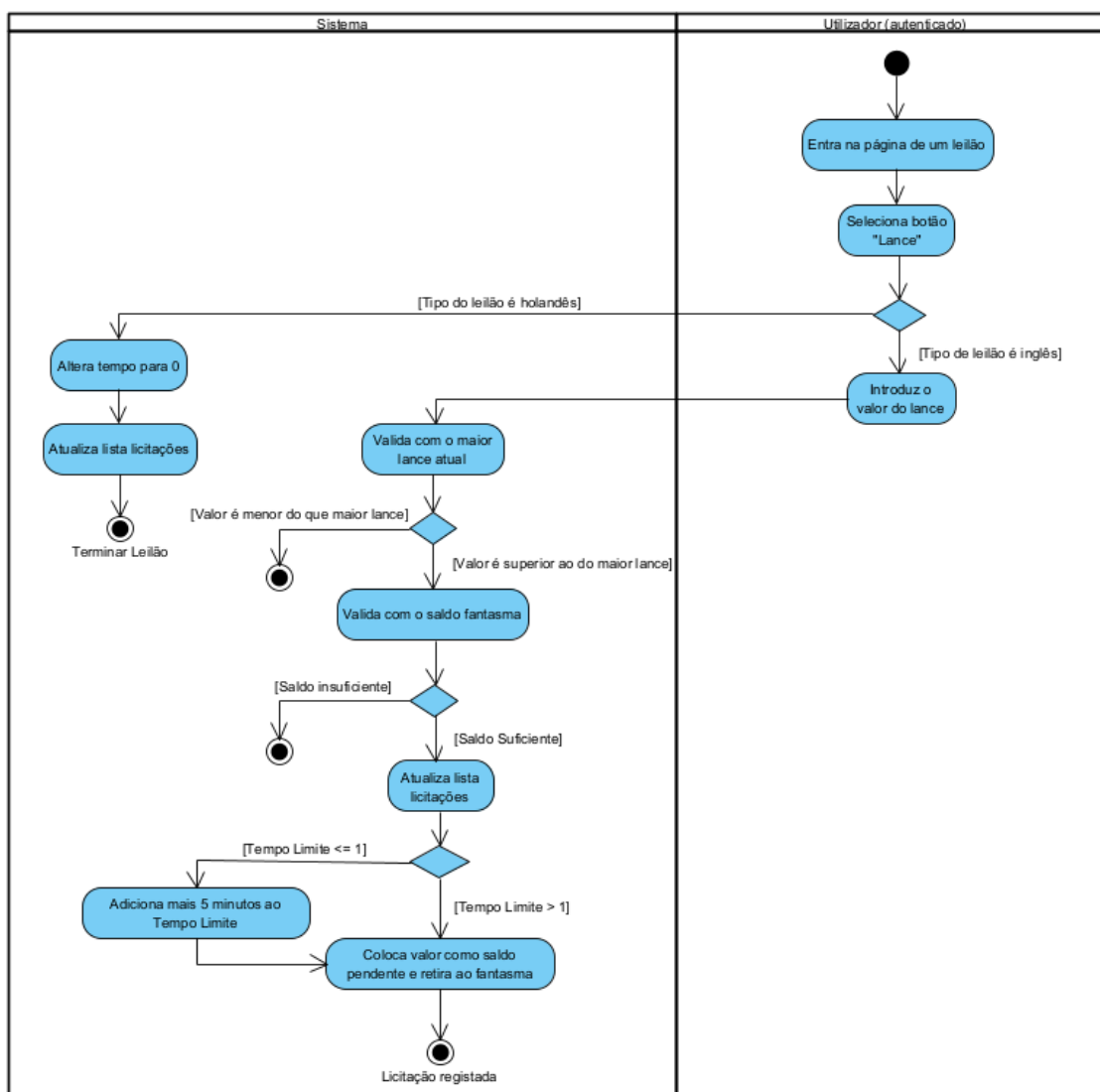


Figura 6 - Diagrama de Atividade "Fazer uma Licitação"

Para este diagrama, decidimos considerar que o utilizador foi autenticado com sucesso de modo a não repetir o que foi demonstrado anteriormente. Com isto, o utilizador necessita entrar na página do leilão que pretende e daí poderá realizar o seu lance.

Caso o leilão seja do tipo Holandês, atualizará a lista de licitações para esse leilão, tendo o lance o valor que o leilão tiver no momento e vai dar início ao Use Case "Terminar Leilão". No entanto, se o leilão for do tipo Inglês, é pedido um valor para o lance e o Sistema irá validá-lo, atualizar a lista, colocar o valor de lance fornecido como saldo pendente e irá, assim, terminar com sucesso a operação. Contudo, se esse valor não cumprir com os critérios necessários a operação vai terminar com insucesso.

### 3.3.3. Diagramas de Sequência

Para completar os modelos de comportamento apresentados anteriormente, desenvolvemos os Diagramas de Sequência. Optamos por definir estes diagramas dado que são uma ótima forma de complementar os Diagramas de Use Cases, uma vez que representa como o utilizador interage com o sistema e também observar como os diferentes objetos interagem entre si tendo por base algo mais semelhante com a arquitetura aplicacional proposta para este sistema de software, facilitando o trabalho da implementação.

Antes de começar o desenvolvimento dos diagramas de classes, definimos algumas entidades do software com base no Modelo de Domínio e acrescentamos uma entidade “BelissimosCollectibles” que se irá comportar como uma entidade principal. Iremos recorrer a esta para executar as principais funcionalidades e terá conhecimentos de alguns dados essenciais.

É de elevada importância mencionar que a estrutura dos diagramas de sequência desenvolvidos toma por base a estrutura lecionada na unidade curricular de Desenvolvimento de Sistemas Software, isto é, utilizam termos semelhantes a UML e da linguagem Java, tanto nas entidades como nos métodos chamados, o que poderá não se verificar futuramente na implementação do projeto, visto que, serão utilizadas outras ferramentas.

#### 1. **Registar Utilizador:**

Tal como mencionado no Use Case em que este diagrama se baseia, o registo deverá verificar que o username e o email fornecidos não estão a ser utilizados por qualquer outro utilizador. Assim, é imediatamente necessário, sendo que os utilizadores se irão identificar pelo username, ver se este pertence a outro. Caso ainda não, haverá outra verificação onde terá de percorrer todos os utilizadores no sistema para constatar se o email se encontra disponível.

De seguida, quando consegue cumprir com todos esses requisitos, cria um utilizador e coloca-o no sistema, o que na implementação será equivalente a adicioná-lo na base de dados.



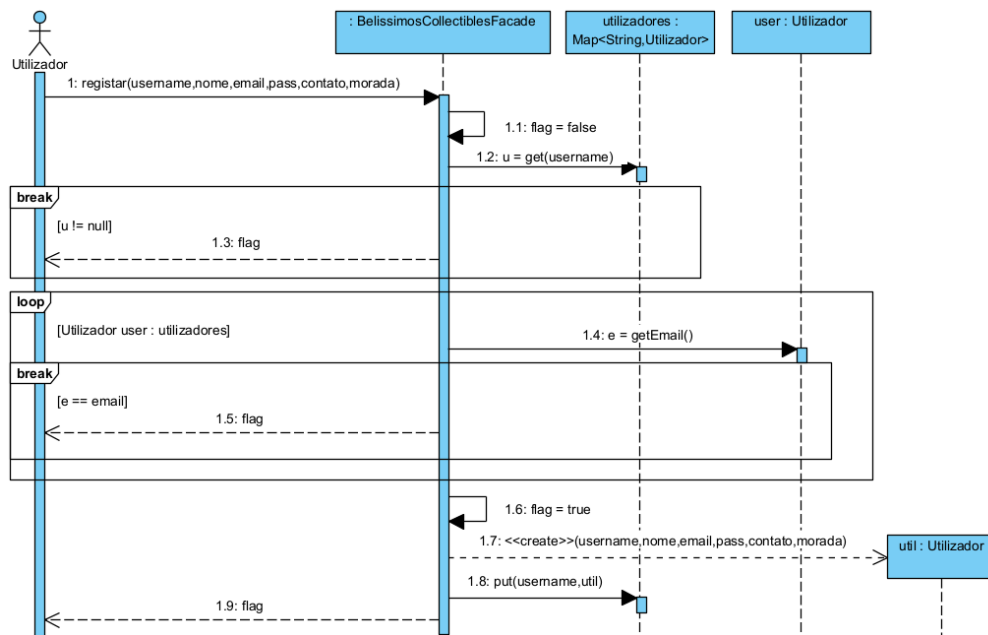


Figura 7 - Diagrama de Sequência "registrar"

## 2. Autenticar:

Para poder usufruir das funcionalidades do programa, um utilizador precisa de se autenticar e, para tal, deve indicar o seu username, ou o seu email, e a sua palavra-passe (*password*).

Se indicar o seu username, este processo torna-se muito mais eficiente. No entanto, como também é possível autenticar-se através do email, o sistema irá percorrer todos os utilizadores. Caso encontre um utilizador que corresponda com os dados fornecidos, compara as palavras-passe e determina se o utilizador pode ser autenticado.

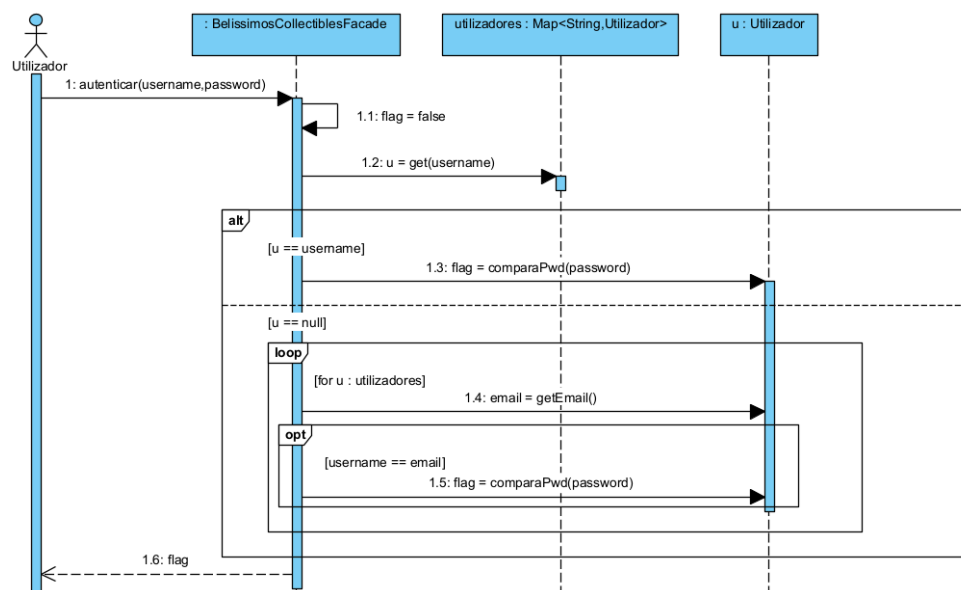


Figura 8 - Diagrama de Sequência "autenticar"

### 3. Alterar Informações da Conta:

Com o utilizador autenticado, se este pretender alterar as suas informações, o sistema irá primeiro tratar de lhe apresentar os dados que pode alterar e consoante os campos alterados, deve alterar a informação nesses mesmos campos. Essa nova informação será então alterada na base de dados.

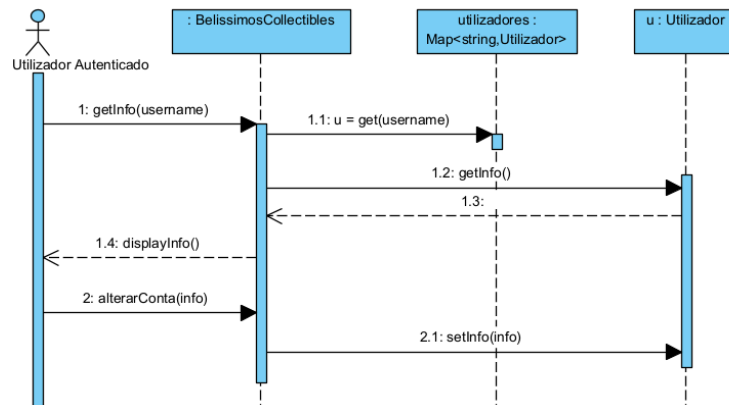


Figura 9 - Diagrama de Sequência "alterarConta"

### 4. Eliminar Conta:

Esta simples funcionalidade deve apenas remover o utilizador da base de dados. Visto que o utilizador precisa de se autenticado para realizar este processo, vamos assumir que todos os passos de verificação já foram feitos.

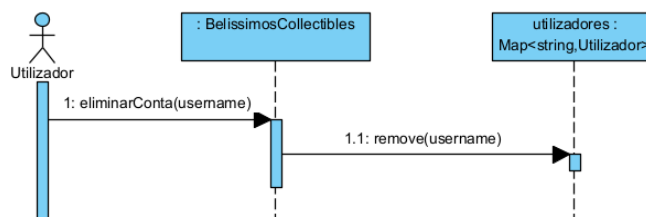


Figura 10 - Diagrama de Sequência "eliminarConta"

### 5. Iniciar Leilão:

Quando um utilizador pretende iniciar um novo leilão, este deve garantidamente autenticar-se para podermos associar o leilão a esse utilizador. Adicionalmente, visto que, as categorias são “restritas”, isto é, não podem necessariamente ser criadas categorias, esta funcionalidade trata de, inicialmente, procurar as categorias que o utilizador forneceu. Assim, é instanciado um objeto Leilão com os dados que apenas eles têm e, para esse, é lhe atribuído um Produto com as características que possui, referidas no modelo de domínio.

De seguida, é lhe atribuído um id e o leilão é inserido no sistema.

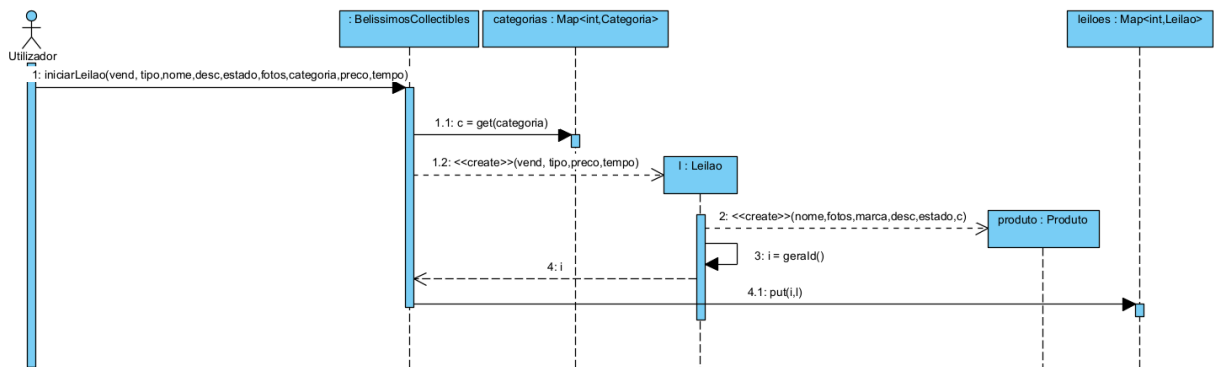


Figura 11 - Diagrama de Sequência "iniciarLeilao"

## 6. Interromper Leilão:

O processo de interromper um leilão é um pouco mais complexo, visto que, tal como é mencionado nos requisitos e no Use case do mesmo, esta funcionalidade possui 3 fluxos possíveis.

Assim, é atribuído um número segundo a opção escolhida pelo utilizador para ser mais fácil as verificações. Caso a operação seja cancelada ( $t = 0$ ), a função terminará com Erro e o leilão continua ativo. Se o leilão for interrompido ( $t = 1$ ), chama a função terminarLeilão que será explicada mais adiante. Finalmente, se cancelar o leilão ( $t = 2$ ), o processo será semelhante ao terminarLeilao, mas não vão existir vencedores, por isso o sistema vai devolver a todos os utilizadores que realizaram licitações o valor licitado ao seu saldo fantasma, isto é o saldo total que têm pendente para todos os leilões em que participaram.

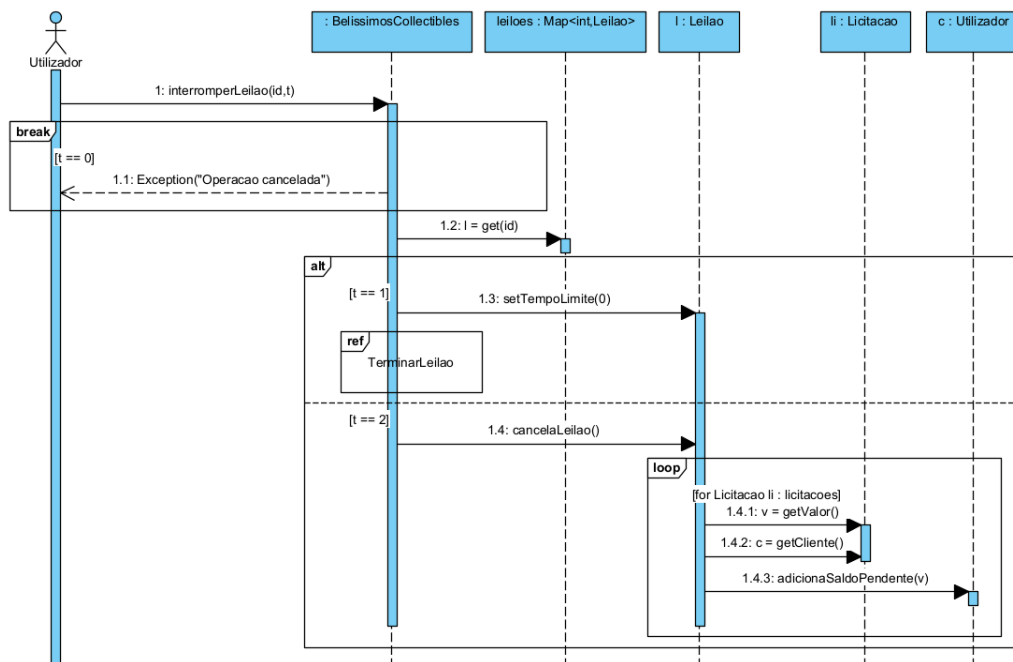


Figura 12 - Diagrama de Sequência "interromperLeilao"

## 7. Consultar Estado do Leilão:

Este Use Case permite visualizar a informação de um leilão pretendido, fornecendo assim o tempo restante até ao final do leilão e as 5 melhores licitações até ao momento. Tendo esses dados, o sistema deverá imprimi-los na interface do utilizador, o que já não faz parte das funcionalidades da lógica de negócios.

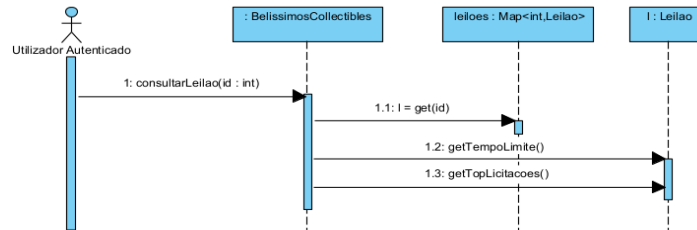


Figura 13 - Diagrama de Sequência "consultarLeilao"

## 8. Alterar Informações do Leilão:

Mesmo que o utilizador esteja autenticado, é necessário garantir que é de facto o vendedor desse leilão para poder alterar as informações, apesar de na implementação essa verificação ser automática. Assim, o vendedor poderá alterar os campos que pretende e o sistema irá adicionar essas informações na base de dados.

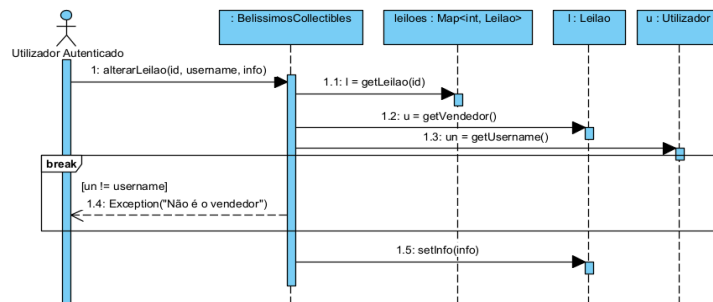


Figura 14 - Diagrama de Sequência "alterarLeilao"

## 9. Fazer uma Licitação:

Esta funcionalidade foi demonstrada de forma mais visual num Diagrama de Atividades anteriormente e seguirá passos semelhantes, no entanto, aqui iremos demonstrar o processo mais específico para um leilão Inglês.

Inicialmente, a licitação será validada, ou seja, vai ser verificado se o valor fornecido é superior à maior licitação feita até ao momento, se o utilizador tem saldo suficiente para realizar a licitação e se não é de facto o vendedor do leilão. Apenas de todos estes critérios forem

cumpridos é que o utilizador poderá prosseguir. O sistema deverá, então, atualizar o índice da maior licitação (anterior maior + 1) e criar uma instância para essa licitação, e posteriormente colocá-la na base de dados. No final, deverá fazer a verificação do tempo limite, isto é, se quando a licitação for feita faltar menos de 1 minuto para acabar o tempo, o sistema irá adicionar mais 5 minutos para ser mais justo para com os outros utilizadores.

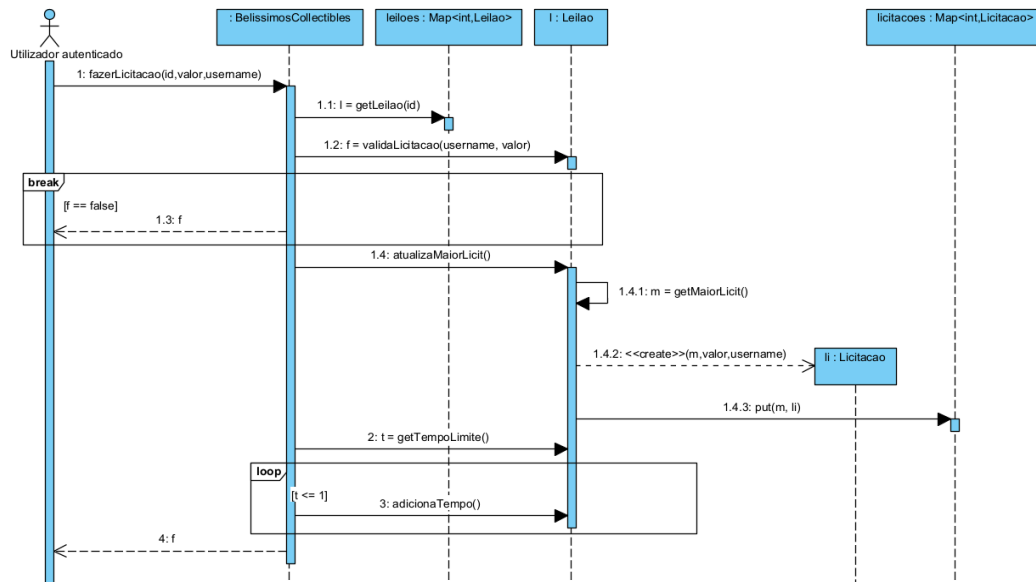


Figura 15 - Diagrama de Sequência "fazerLicitacao"

#### 10. Terminar Leilão:

Quando um leilão acabar, é necessário identificar o seu vencedor, caso este exista (se não existir maior licitação é igual a -1 e não é necessário continuar este método). Para tal, ao cliente vencedor será atualizado o saldo, processo que é explicado no diagrama de sequência de Inserir Dinheiro na Conta, ou seja, será retirado dinheiro no Saldo e irá tratar do saldo fantasma, e para todos os outros participantes devolve apenas ao saldo fantasma o saldo pendente que possuem para esse leilão.

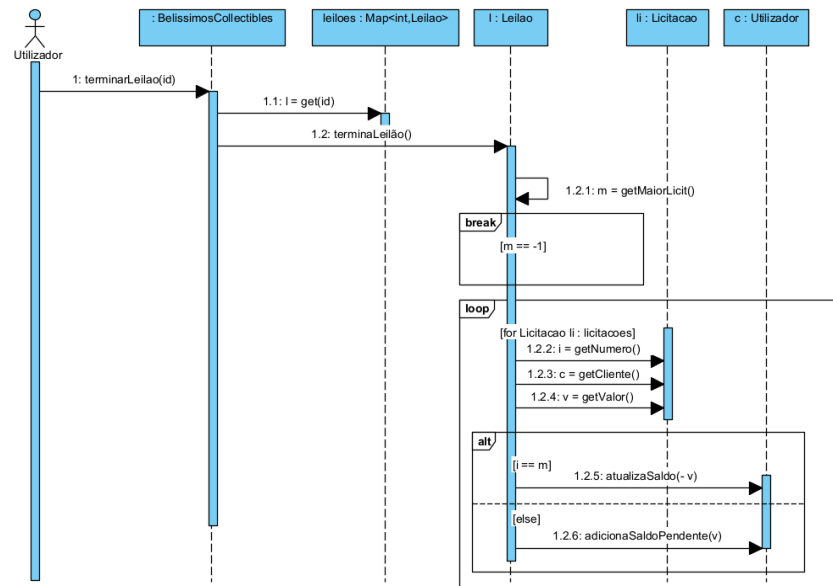


Figura 16 - Diagrama de Sequência "terminarLeilao"

## 11. Pesquisar Produtos:

Nesta funcionalidade é importante relembrar que a pesquisa de produtos em leilão pode ser feita tanto como por pesquisa do nome do produto, como por aplicação de filtros, isto é, por categoria e marca do produto e por tipo de leilão. Deste modo, é importante verificar quais destes campos se encontra ou não preenchido.

Assim, devemos percorrer todos os leilões existentes para os filtrar através das suas características. Caso cumpram com pelo menos um dos critérios, são colocados numa estrutura de dados "Set", para evitar que se devolvam repetidos.

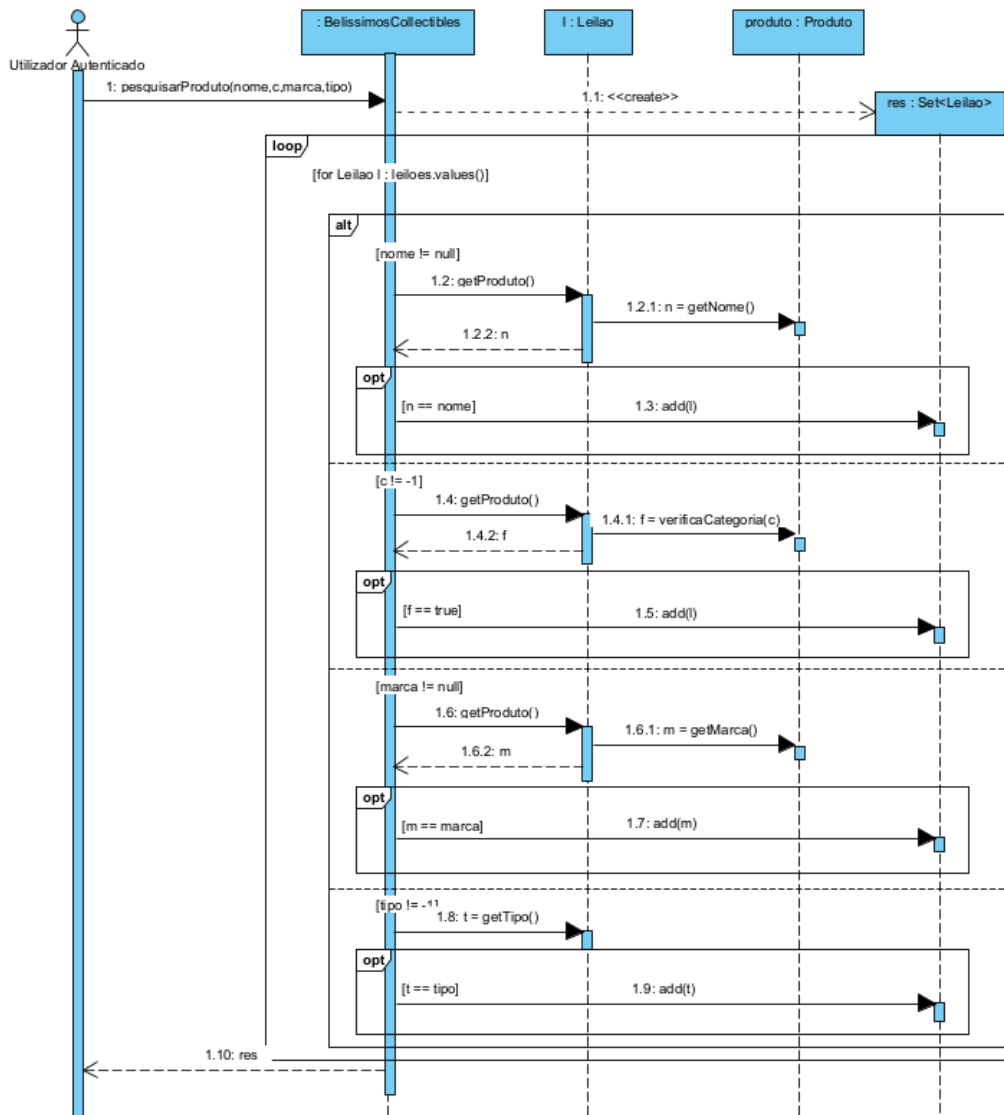


Figura 17 - Diagrama de Sequência "pesquisarProdutos"

## 12. Consultar Conta Utilizador:

Semelhante ao Use Case Consulta Leilão, um utilizador deverá por visualizar contas dele ou de outros utilizadores, indicando o utilizador que pretende visualizar. Esta funcionalidade está encarregue de selecionar as informações sobre o username, email, média das classificações e as avaliações individuais que possui, mostrando, para cada utilizador que o avaliou, o username, a classificação (em estrelas) e comentário, caso este exista.

Posteriormente, a interface do utilizador está encarregue de imprimir essas informações.

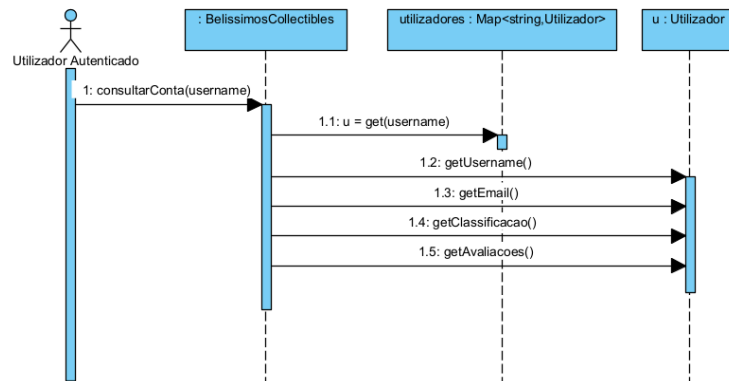


Figura 18 - Diagrama de Sequência "consultarConta"

## 13. Avaliar Vendedor

Para que o utilizador possa avaliar outro vendedor, deve indicar que utilizador pretende avaliar, atribuir uma classificação de 1 a 5, e, se pretender, adicionar um comentário. O sistema deve então procurar o utilizador, e caso essa conta não exista mais, podendo ter sido apagada anteriormente, a funcionalidade não pode ser comprida, caso contrário adiciona a esse utilizador a nova avaliação. A avaliação tem associada o username do utilizador que a fez.

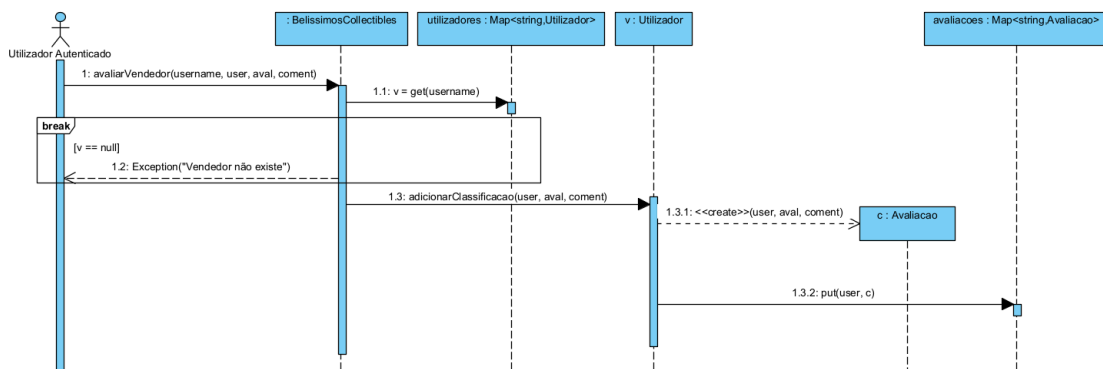


Figura 19 - Diagrama de Sequência "avaliarVendedor"



#### 14. Inserir Dinheiro na Conta:

Esta funcionalidade permite a um utilizador, já autenticado, inserir dinheiro no seu saldo dentro do sistema, para garantir segurança nas licitações. Deste modo, o utilizador indica o montante que quer acrescentar e o sistema irá atualizar o seu saldo e o seu saldo fantasma, quer este último seja igual ou inferior ao primeiro, e devolve a nova quantidade de saldo com que o utilizador fica.

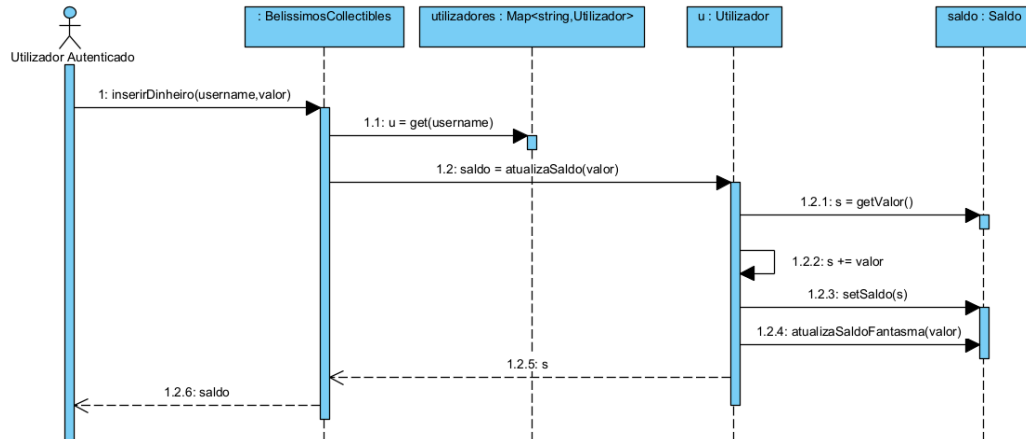


Figura 20 - Diagrama de Sequência "inserirDinheiro"

### 3.3.4. Diagrama de Classes

Tendo por base a visão geral do sistema idealizada no Modelo de Domínio, tal como os Use Cases apresentados e os Diagramas de Sequência realizados, desenvolvemos um Diagrama de Classes que se encontra a seguir, pois permite-nos visualizar em detalhe como as classes irão interagir entre elas, assim como, definir alguns dos métodos que consideramos necessários e uteis para a futura implementação, representando aproximadamente, assim, a camada de negócio do sistema. Adicionalmente as entidades principais, também identificadas no modelo de domínio, foi também criado um *facade* para mediar a interação com toda a camada de forma que a solução apresentada fosse modular.

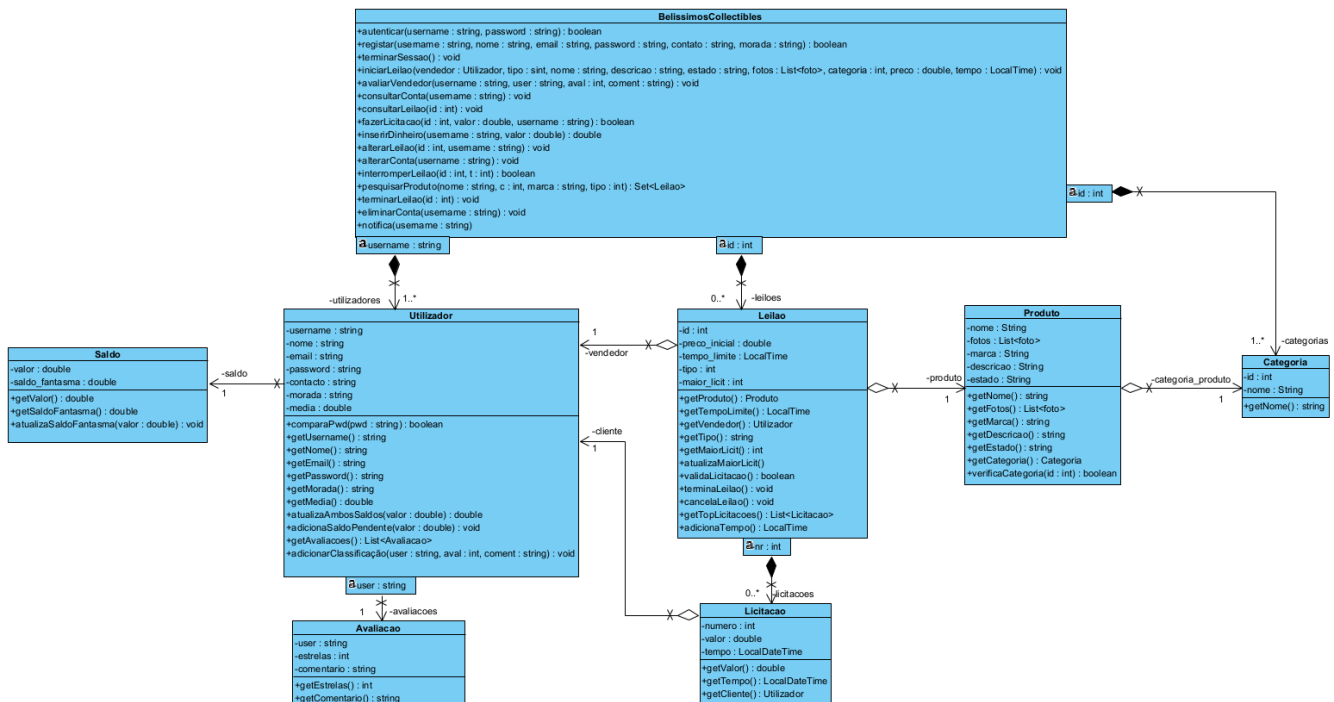


Figura 21 - Diagrama de Classes

## 4. Conceção do Sistema de Dados

### 4.1. Apresentação geral da estrutura (esquema) do sistema de dados

Com a análise dos requisitos do sistema completa, seguiu-se o desenvolvimento da base de dados. Decidiu-se implementar um Sistema de Gestão de Bases de Dados Relacional visto que é uma ferramenta essencial para o armazenamento de informação, aceder a dados relacionados uns com os outros o que é extremamente útil para o desenvolvimento do projeto, assegurar a segurança, consistência e integridade dos dados. Posto isto, com a implementação da nossa base de dados, pretendemos armazenar toda a informação relevante ao nosso problema de modo a garantir um melhor funcionamento da nossa aplicação.

Deste modo, desenvolveu-se o seguinte Modelo Lógico para representar a estruturação da informação pertinente a guardar, e a forma como as tabelas presentes no modelo se relacionam entre si.

É importante apontar que este modelo foi realizado em SQL Server, e não foi possível representar graficamente as chaves estrangeiras por isso, em todas as ligações referentes aos relacionamentos, tentamos colocá-las nos atributos que representam chaves estrangeiras, inclusivamente são também referidas na descrição dos elementos abaixo.

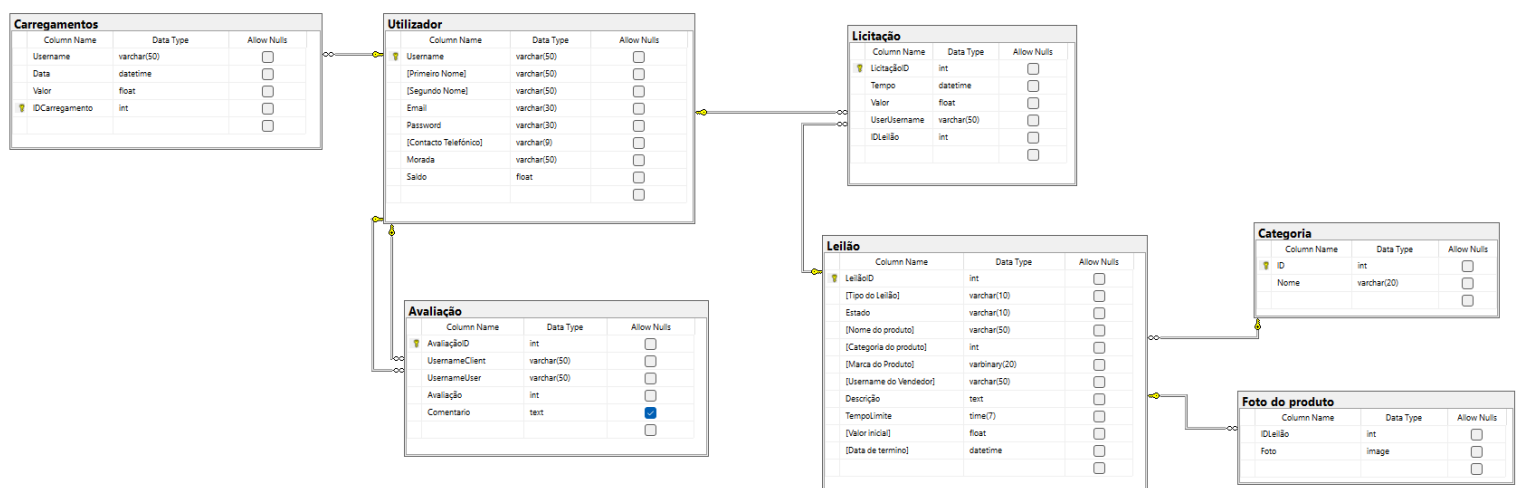


Figura 22 - Modelo Lógico da Base de Dados.

## 4.2. Descrição detalhada dos vários elementos de dados e seus relacionamentos

No desenvolvimento do Modelo Lógico apresentado identificamos as seguintes entidades:

- **Utilizador:** Um utilizador é identificado pelo seu Username. Possui também obrigatoriamente um Primeiro e Último Nome, um Email, uma Password, um Contacto Telefónico, uma Morada e um Saldo. Um utilizador pode ou não conter Avaliações. O mesmo aplica-se aos leilões e a carregamentos.
- **Carregamentos:** Representam as inserções de dinheiro feitas para o saldo de cada utilizador. São identificados por um IDCarregamento, e possuem uma data, um valor e uma chave estrangeira Username que indica para que Utilizador fez o carregamento.
- **Avaliação:** Cada avaliação contém obrigatoriamente um ID como chave primária, o Username do Utilizador a quem esta classifica, o Username do Utilizador que realizou a classificação, sendo estes dois últimos atributos chaves estrangeiras para diferentes Utilizadores. Adicionalmente contem uma Avaliação de 1 a 5. Pode também conter um Comentário escrito.
- **Licitação:** Uma licitação possui um ID que o identifica (LicitaçãoID), o Tempo no qual um utilizador a realizou, a quem ela se associa através do seu Username (chave estrangeira para Utilizador), um Valor da licitação em Euros, e o IDLeilão que identifica o leilão a quem esta se associa (chave estrangeira para Leilão).
- **Leilão:** Um leilão tem obrigatoriamente como chave primária o LeilaoID. Possui também como atributos o Tipo do Leilão, o Estado, o Nome do Produto, a Marca do Produto, a Descrição, um Tempo Limite, que define o tempo até ao fim do leilão, Valor Inicial e Data de Terminio. Possui também duas chaves estrangeiras, a CategoriaProduto que lhe associa a categoria em que o produto se encontra e UsernameVendedor que identifica o utilizador que coloca o leilão no ar.
- **Categoria:** Como mencionado acima, a categoria representa as categorias nos quais os produtos para leilão se encontram. Deste modo, uma Categoria é identificada pelo seu ID e possui o Nome da mesma.

Esta entidade surgiu com o intuito de limitar as categorias existentes no sistema. Caso não se limitasse pequenas alterações na introdução da categoria, a filtragem de produtos seria muito mais difícil.

- **Foto do Produto:** É um atributo multivariado, pois um leilão pode conter uma ou várias fotos, por isso, possui como chave estrangeira o IDLeilao, que representa o leilão a qual se encontra associado e como chave primária a Foto.

As entidades identificadas relacionam-se da seguinte forma:

Utilizador (1) “guarda” (N) Avaliação **ou** Utilizador (1) “faz” (N) Avaliação;

Utilizador (1) “realiza” (N) Carregamentos;

Utilizador (1) “licita” (N) Licitações;

Leilão (1) “tem” (N) Licitações;

Categoria (1) “identifica” (N) Leilão;

Leilão (1) “possui” (N) Foto do produto.

## 5. Esboço dos Interfaces do Sistema

### 5.1. Estrutura geral das interfaces do sistema

Na grande parte dos sistemas de software orientados ao usuário público e a uma utilização quotidiana, um dos aspetos essenciais para o sucesso do produto é a sua interface com o utilizador. Deste modo, de forma a prever a parte gráfica da aplicação, procedeu-se à criação de *mockups*, isto é, imagens que representem o *design*, esquema de cores, estrutura e outros aspetos visuais do programa.

Para além disso, os *mockups*, devem recriar instâncias em que o utilizador poderá interagir com o sistema. Com isto, para poder alcançar o sucesso esperado, é necessário garantir que a construção dos *mockups* ajude o planeamento, de forma a representar uma aplicação com uma interface simples, fácil e intuitiva para atingirem acessibilidade para todo o tipo de utilizadores.

### 5.2. Caracterização das interfaces

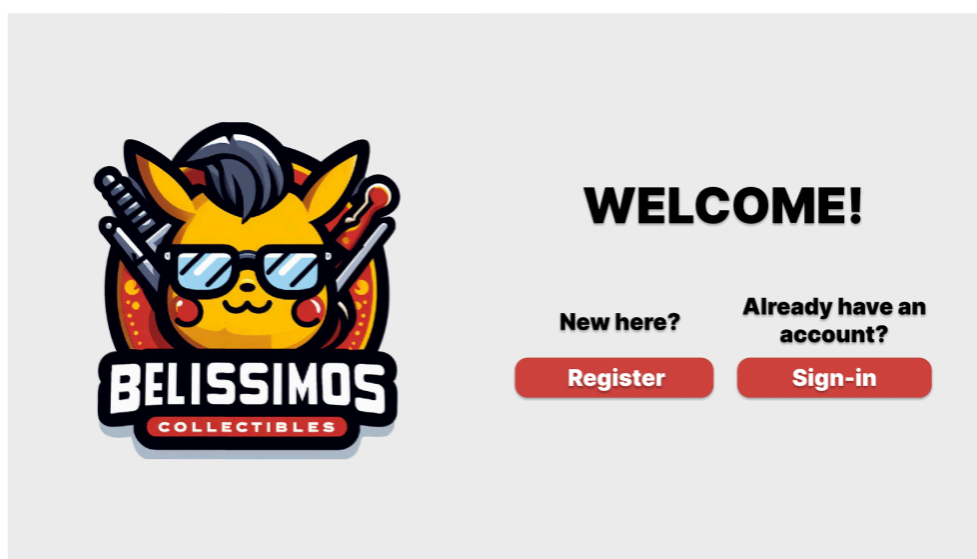
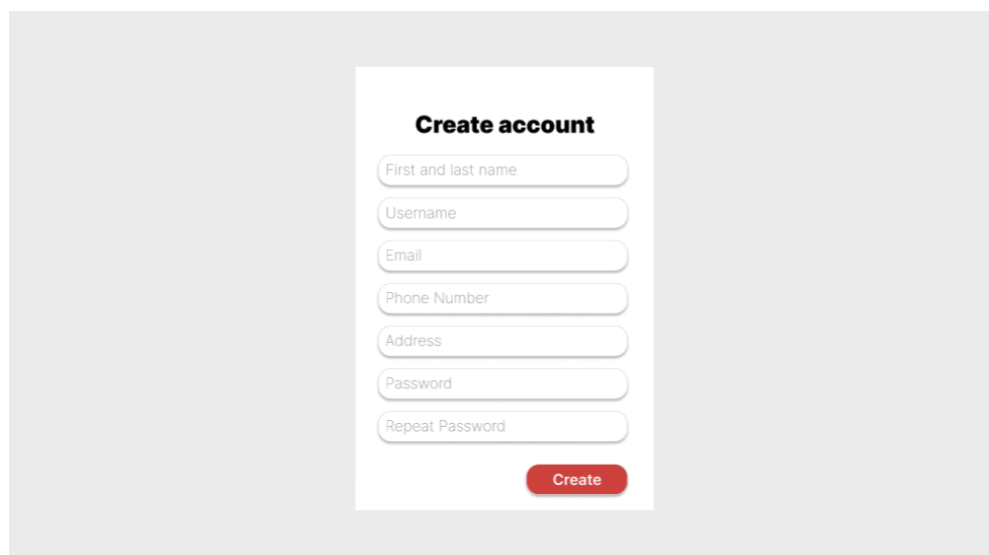
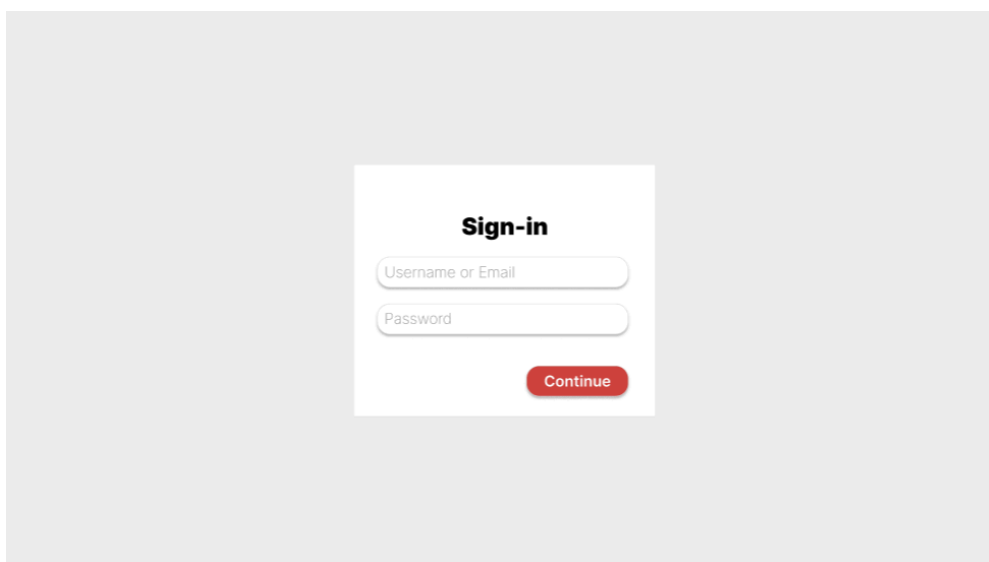


Figura 23 - Página Inicial.



The image shows a 'Create account' form centered on a light gray background. The form is a white rectangle with rounded corners. At the top, the title 'Create account' is displayed in bold black text. Below the title are seven input fields, each with a light gray border and rounded ends. The labels for these fields are 'First and last name', 'Username', 'Email', 'Phone Number', 'Address', 'Password', and 'Repeat Password', all in a small, light gray font. At the bottom right of the form is a red button with the word 'Create' in white text.

Figura 24 - Página de criação de conta.



The image shows a 'Sign-in' form centered on a light gray background. The form is a white rectangle with rounded corners. At the top, the title 'Sign-in' is displayed in bold black text. Below the title are two input fields, each with a light gray border and rounded ends. The labels for these fields are 'Username or Email' and 'Password', both in a small, light gray font. At the bottom right of the form is a red button with the word 'Continue' in white text.

Figura 25 - Página para autenticação da conta.

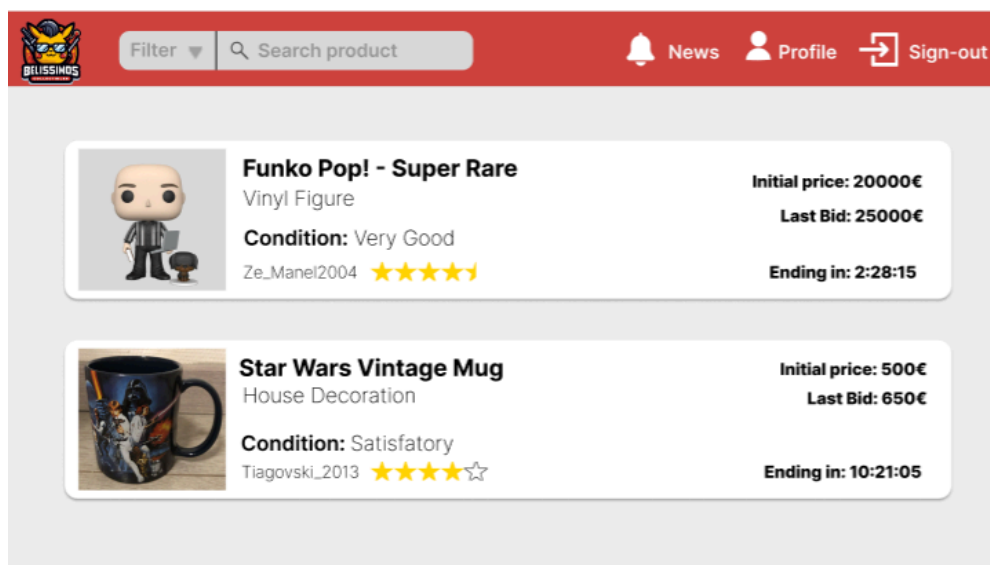


Figura 26 - Página de procura de leilões.

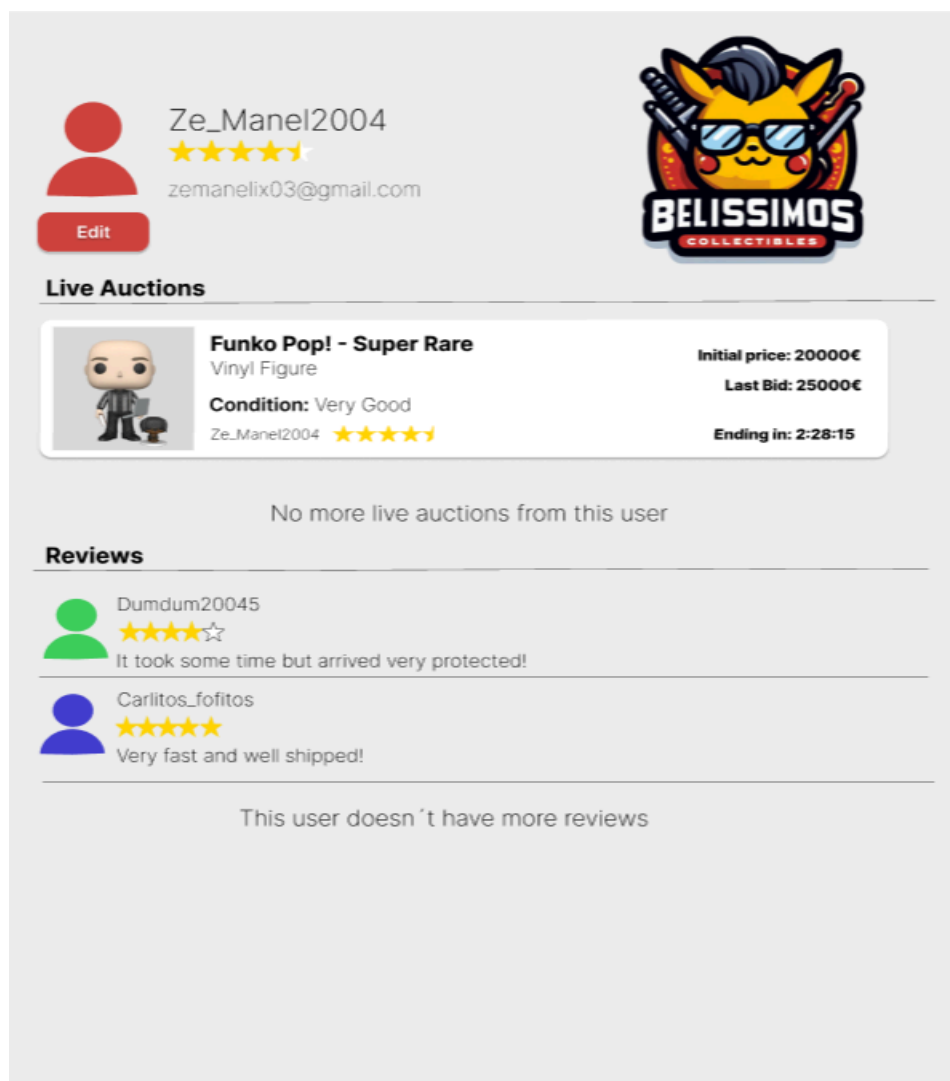






Figura 27 - Página de perfil.

**Funko Pop! - Super Rare**  
Vinyl Figure  
**Initial price:** 20000€  
**Condition:** Very Good



**Ze\_Mane!2004**  
★★★★★  
zemanelix03@gmail.com

**Description**


Super rare and vintage Funko Pop!  
From a very well know celebrity!


With the original box and sticker.  
10/10 condition!


I promise to ship it very carefull and as soon as possible :)


**Ending in: 2:28:15**

**Winning**

 Dumdum20045  
**25000€**

 Dumdum20045  
**21000€**

 Rui\_paulo9  
**22500€**


 Dumdum20045  
**25000€**


Value

Bid

Figura 28 - Página de leilão ao vivo (vista CLIENTE).

**Funko Pop! - Super Rare**  
Vinyl Figure  
**Initial price:** 20000€  
**Condition:** Very Good



**Ze\_Mane!2004**  
★★★★★  
zemanelix03@gmail.com

**Description**


Super rare and vintage Funko Pop!  
From a very well know celebrity!


With the original box and sticker.  
10/10 condition!


I promise to ship it very carefull and as soon as possible :)


**Ending in: 2:28:15**

**Winning**

 Dumdum20045  
**25000€**

 Dumdum20045  
**21000€**

 Rui\_paulo9  
**22500€**

 Dumdum20045  
**25000€**

Edit

Stop

Cancel

Figura 29 - Página de leilão ao vivo (vista VENDEDOR).

42

+ Load Photos

Frame  
Title

Frame  
Initial Price

Frame  
Category ▼

Frame  
Condition ▼

Frame  
Brand

Frame  
Auction Type ▼

Write a description of the product you are going to auction

Cancel Confirm

Figura 30 - Página de criação de leilão.

## 6. Implementação

### 6.1. Metodologia de Implementação

O padrão arquitetural escolhido foi o *Model, View, Controller* (MVC), devido à necessidade de obter um conjunto de componentes modulares que facilitariam o processo de desenvolvimento em paralelo.

Assim, a necessidade de implementação de uma aplicação multiplataforma foi muito mais eficaz, visto que, a estrutura interna dos dados e procedimentos (*Model*) torna-se independente da interface com o utilizador (*View*), encontrando-se apenas ligados por um mediador (*Controller*).

Posteriormente, a componente do modelo foi decomposta numa camada de negócios e de dados. Esta decisão foi inteiramente desencadeada pela necessidade de proporcionar a integridade, consistência e disponibilidade dos dados de toda a aplicação de uma forma totalmente independente da tecnologia de base de dados em uso.

### 6.2. Ferramentas utilizadas

A principal ferramenta usada na implementação da solução de software desenvolvida foi a framework de desenvolvimento ASP.NET e Entity Framework. Adicionalmente, utilizamos várias ferramentas destas *frameworks*, nomeadamente:

- Blazor: Plataforma utilizada para desenvolver o cliente para a nossa aplicação;
- Microsoft SQL Server: Permitiu fazer toda a gestão da base de dados. Inclusive, foi a ferramenta utilizada para definir o modelo lógico;
- Visual Studio 2022: o IDE é a principal ferramenta utilizada para desenvolver toda a aplicação;
- Figma: Utilizada para definir os *mockups* necessários. Decidimos utilizar esta ferramenta por vários motivos. Em primeiro lugar, permite que vários elementos trabalhem ao mesmo tempo. Adicionalmente, é intuitiva e contém funcionalidades que facilitam o desenvolvimento das interfaces do utilizador criadas;
- Browsers de Internet: Para avaliar a interface visual do produto final, compreender de melhor forma e testar o meio de interação dos futuros utilizadores.

### 6.3. Desenvolvimento

Para guardar todos os dados e fazer a gestão dos mesmos foi essencial a criação e conexão a um Base de Dados. Logo, depois de instalado e devidamente configurado o motor de

base de dados SQL Server da Microsoft, foi criada a base de dados para este serviço. O processo de conexão à Base de Dados foi gerido através da utilização da ferramenta EntityFramework da própria *framework* ASP.NET através de alguns ficheiros de configuração que definem como aceder à BD. As tabelas da BD, juntamente com os seus atributos, foram transformadas em classes *Model* e existirá, adicionalmente, uma outra classe responsável por realizar *queries* à BD, para remover, alterar e inserir dados. O *backend* foi desenvolvido com recurso à linguagem de programação C#.

O Blazor, também da autoria da Microsoft, permite construir interfaces de utilizador interativas web. Assim, com apoio nesta ferramenta e outras, como o Figma e outras fontes, conseguimos transformar os *mockups* previamente criados em páginas interativas (extremamente semelhantes a esses esboços) através de código, estabelecendo, deste modo, interfaces com que o utilizador irá interagir para cada uma das páginas do sistema. O código que traduz estas páginas foi desenvolvido em várias “*Pages*”, ficheiros que utilizam as linguagens de programação HTML e CSS para alcançar o aspeto pretendido. De modo a que possamos interagir com a BD através do *frontend*, são chamadas funções da classe, referida anteriormente, responsável pela BD, também, no próprio código das devidas “*Pages*”.

De modo geral, a maioria das funcionalidades definidas anteriormente, especialmente nos requisitos e nos Use Cases estabelecidos, foram implementadas, com o intuito da criação de uma plataforma funcional e intuitiva para os leilões de colecionáveis. Com exceção de alguns métodos “*navigate*” que por algum motivo desconhecido em certas páginas não funcionou, após consultas com colegas e revisão do código continuamos sem descobrir a origem do problema, sendo que os mesmos estavam implementados, segundo restantes colegas, “de forma correta”. Além disso, vale realçar que não conseguimos implementar a pesquisa para key words específicas, apenas aparecendo todos os leilões, assim como quando clicarmos num desses leilões ir para a página desse mesmo leilão em específico, apesar disso, as páginas dos leilões e todas as outras estão implementadas, sendo possível fazer licitações, apesar dessas não serem demonstradas no front-end, são atualizadas na tabela de “Licitações” na base de dados (para isso temos um modo de testes que demonstra que essas funcionalidades são executadas).

## 6.4. Validação da aplicação desenvolvida

Terminado o desenvolvimento da aplicação e a fase de testes que gerou a necessidade de efetuar ligeiras alterações a certos métodos desenvolvidos, a equipa apresentou o projeto a um grupo de pessoas pertencente ao público-alvo, com o objetivo de validar o sistema desenvolvido. Isto permitiu concluir que o projeto tem bastantes pontos positivos e correspondeu à maioria das expectativas.

Dito isto, da parte da equipa, consideramos que obtivemos um bom resultado e, de modo geral e dentro das nossas expectativas, a aplicação ficou boa, apesar das dificuldades existentes com a gestão

de tempo, visto que a elevada quantidade de projetos e testes que cada elemento da equipa possui levaram a falhas que também necessitam da sua atenção, e com algumas das ferramentas utilizadas.

## 7. Conclusões e Trabalho Futuro

Dado por concluído o desenvolvimento do projeto presente, pudemos presenciar todo o desenvolvimento de software desde a fundamentação do sistema até ao desenvolvimento do mesmo, contendo assim três fases principais. Para todas estas fases, a abordagem foi minuciosa e cuidada com o intuito de garantir o cumprimento de todos os objetivos estabelecidos.

Na fundamentação, desenvolvemos de uma forma clara o que se pretendia implementar e quais eram os objetivos a alcançar com a aplicação, decidindo-se desde cedo uma estratégia para o desenvolvimento do sistema e enumerando-se os requisitos que este deveria ser capaz de cumprir, exigindo da nossa parte uma investigação mais profunda do mercado de colecionáveis e dos leilões, particularmente os online. Assim, sentimos que a fundamentação do nosso projeto ficou ilustrada de forma clara e objetiva, o que se tornou extremamente útil na evolução do desenvolvimento.

Na segunda fase do projeto, a especificação, tentamos expor e analisar com mais detalhe aquilo que queríamos desenvolver, assim como os desejos dos utilizadores e interessados no sucesso da aplicação, partindo assim para a definição da estrutura e dos aspetos comportamentais do sistema, criando-se assim o modelo de domínio e as especificações dos casos de uso, diagramas de atividades, sequência e um diagrama de classes, tomando por base os requisitos levantados. Todos estes diagramas permitiram antecipar problemas de insatisfação dos requisitos definidos e incoerências estruturais nesses mesmos diagramas, tal como ajudaram a prever que comportamentos e estrutura o que programa a desenvolver necessitaria.

Nesta fase, foram também concebidos o sistema de dados, isto é, o modelo lógico e esboços das interfaces. Consideramos estes últimos importantes para termos uma ideia do aspeto da aplicação que pretendíamos, o que facilitou a visualização dos relacionamentos entre os diferentes componentes do sistema e, posteriormente, em toda a implementação do projeto.

Sentimos algumas dificuldades nesta segunda fase do desenvolvimento, nomeadamente na concessão do sistema de dados, pois não tivemos certeza total da modelação tendo em conta a dimensão e as especificações feitas previamente, e no desenvolvimento de certos modelos, mais especificamente no de classes e nos de sequência, pois exigiram pensar muito mais antecipadamente sobre um pouco do que a implementação deveria cumprir e do que nos era esperado com estes diagramas para um projeto de Engenharia de Software. Contudo, consideramos que ultrapassamos estas dificuldades.

A implementação foi, para nós, a parte mais desafiante de todo este projeto, principalmente a adaptação e aprendizagem das ferramentas de desenvolvimento que deveríamos usar. Esse tempo despendido, bem como o tempo gasto a melhorar algumas partes da fase anterior revelaram-se penalizadoras para o produto final. Adicionalmente, é importante salientar que foi a primeira vez que toda a equipa explorou esta área de desenvolvimento de aplicações web, não tendo qualquer base e experiência em desenvolvimento de qualquer tipo de *frontend* ou como funciona a estrutura de uma aplicação web. O conglomerar de tantos novos paradigmas e

tecnologias diferentes levou a que o projeto se mostrasse bastante desafiador, o que impediu que todas as nossas expectativas iniciais da implementação fossem cumpridas.

No futuro, acreditamos que este projeto tem potencial para ser expandido, trazendo algumas funcionalidades que o aproximassem da realidade como, por exemplo, a implementação de pagamentos ou de registo de entregas e até disponibilizando mais tipos de leilão, como tínhamos previsto inicialmente.

Em suma, apesar de todas as adversidades, consideramos que o produto final se adequa às expectativas da equipa, uma vez que é um protótipo funcional e que demonstra a aplicação de conhecimentos das novas ferramentas com as quais a equipa teve de lidar, apesar de tudo o que foi aprendido mostrou-se totalmente diferente de tudo o que nos foi ensinado ao longo da licenciatura, tornando as aprendizagens feitas ao longo do desenvolvimento deste projeto, complementos essenciais para a mesma. Assim, avaliamos positivamente o desenvolvimento deste projeto apesar das dificuldades obtidas, apesar do tempo apertado que sofremos.

## Referências

Sommerville, I. (2016). Software Engineering. 10th ed. London: Pearson Education



## Lista de Siglas e Acrónimos

|     |                                  |
|-----|----------------------------------|
| BD  | Base de Dados                    |
| SQL | <i>Structured Query Language</i> |
| UML | <i>Unified Modeling Language</i> |

# Anexos

## I. Anexo 1

### Ata da Reunião 1

Equipa de desenvolvimento x Belíssimos Collectibles

#### Pontos da Reunião

1. Funcionamento geral dos leilões de colecionáveis;
2. Aspetos relevantes a integrar no projeto a desenvolver;
3. Outros assuntos.

#### Funcionamento geral dos leilões de colecionáveis

Os leilões deverão basear-se nas interações entre utilizadores. Esses mesmos poderão iniciar os seus próprios leilões para um certo produto, com as características necessárias e valorizadas na área de negócios a explorar. A partir desse momento, todos os utilizadores deverão poder explorar, aceder e participar em todos os leilões ativos, sendo que vendedores não poderão participar nos seus próprios leilões. Deve, contudo, existir também um método de verificação dos pagamentos, para garantir a validade dos licitadores, e avaliação dos vendedores.

#### Aspetos relevantes a integrar no projeto a desenvolver

- Registo e autenticação dos utilizadores;
- Colocação de produtos para leilão, tendo por base um formato escolhido;
- Participação nos leilões por forma de licitações
- Processo de inserção e remoção de dinheiro para um saldo virtual dentro do sistema;
- Atribuição de avaliações quantitativas e qualitativas a vendedores;
- Os vendedores poderão manipular os seus leilões de forma limitada;
- Pesquisar por leilões;
- Consultar leilões.

#### Outros assuntos

Não foram discutidos outros assuntos.

#### Presenças

Rui Afonso, proprietário e representante da “Belíssimos Collectibles”

Todos os membros da equipa de desenvolvimento do projeto

Braga, 7 de outubro de 2023

**Rui Afonso Rouxinol**