

**Universidade do Minho**

Escola de Engenharia

Licenciatura em Engenharia Informática

Mestrado Integrado em Engenharia Informática

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

Ano Letivo de 2023/2024

### **UMusic**

**Rodrigo Monteiro**  
a100706

**Miguel Gramoso**  
a100835

**Luís Figueiredo**  
a100549

**Diogo Abreu**  
a100646

Setembro, 2023

Data da Receção	
Responsável	
Avaliação	
Observações	

**UMusic**

**Rodrigo Monteiro**  
a100706

**Miguel Gramoso**  
a100835

**Luís Figueiredo**  
a100549

**Diogo Abreu**  
a100646

Setembro, 2023

## Resumo

O presente relatório apresenta o progresso do projeto UMusic até o momento, que tem como objetivo o desenvolvimento de uma aplicação para leilões online de memorabilia musical. A plataforma visa permitir aos utilizadores comprar e vender itens relacionados à música, como instrumentos, discos, autógrafos, entre outros.

Neste relatório, são apresentados os resultados da fase inicial do projeto, que incluiu a Definição do Sistema, Definição dos Requisitos, a Especificação e Modelação do Software, a Conceção do Sistema de Dados e o Esboço das Interfaces do Sistema. Além disso, são descritas as tarefas que serão executadas na próxima fase do projeto, que incluem a implementação e desenvolvimento do sistema. O relatório também inclui uma apreciação crítica sobre o trabalho realizado até o momento, apontando seus pontos fortes e fracos, bem como eventuais tarefas a realizar futuramente ou novas opções para estender o trabalho realizado

**Área de Aplicação:** Desenvolvimento de uma aplicação para um leilão online.

**Palavras-Chave:** Leilões Online, Music Memorabilia, SQL, C#, Engenharia de Software, Bases de Dados.

# Índice

<b>1. Definição do Sistema</b>	<b>1</b>
1.1. Contextualização	1
1.2. Fundamentação	1
1.3. Objetivos	1
1.4. Viabilidade	2
1.5. Recursos a utilizar	3
1.6. Equipa de trabalho	3
1.7. Plano de execução do trabalho	3
<b>2. Definição de Requisitos</b>	<b>5</b>
2.1. Apresentação da estratégia e método	5
2.2. Descrição geral dos requisitos levantados	5
2.2.1. Requisitos Funcionais	5
2.2.1.1. Registo de um utilizador	5
2.2.1.2. Adicionar um método de pagamento	5
2.2.1.3. Adicionar uma morada	6
2.2.1.4. Autenticação	6
2.2.1.5. Terminar sessão	6
2.2.1.6. Editar conta	6
2.2.1.7. Pesquisa	6
2.2.1.8. Consulta de um lote	7
2.2.1.9. Consulta de um leilão	7
2.2.1.10. Consulta da lista de favoritos	7
2.2.1.11. Consulta de notificações	7
2.2.1.12. Consulta das categorias	7
2.2.1.13. Consulta dos lotes ganhos	8
2.2.1.14. Submeter um lote	8
2.2.1.15. Consultar lista de lotes submetidos	8
2.2.1.16. Entrar num leilão	8
2.2.1.17. Funcionamento/ Sair do leilão	9
2.2.1.18. Página de gestão do sistema	9
2.2.1.19. Página de gestão de leilões	9
2.2.1.20. Criação de um leilão	9
2.2.1.21. Página de gestão de lotes	10
2.2.1.22. Aprovação de um lote	10
2.2.2. Chegada de um item (ou conjunto de itens)	10
2.2.2.1. Página de gestão de utilizadores	10
2.2.3. Requisitos não funcionais	10
2.3. Validação dos requisitos	11
<b>3. Especificação e Modelação do Software</b>	<b>12</b>
3.1. Apresentação geral da especificação	12
3.2. Aspetos estruturais	12
3.2.1. Modelo de domínio	12

3.3. Aspectos comportamentais	13
3.3.1. Diagramas de casos de uso	13
3.3.2. Casos de uso	15
3.3.2.1. Conta do utilizador	15
3.3.2.1.1. Criar conta	15
3.3.2.1.2. Login	16
3.3.2.1.3. Redefinir palavra-passe	16
3.3.2.1.4. Adicionar novo método de pagamento	16
3.3.2.1.5. Adicionar nova morada	16
3.3.2.2. Consultas	17
3.3.2.2.1. Pesquisar	17
3.3.2.2.2. Consulta de um lote	17
3.3.2.2.3. Consulta de um leilão	17
3.3.2.2.4. Consulta das listas de favoritos	17
3.3.2.2.5. Remoção de um item da lista de favoritos	18
3.3.2.2.6. Consulta das notificações	18
3.3.2.2.7. Consulta das categorias	18
3.3.2.2.8. Consulta de uma categoria	18
3.3.2.2.9. Consulta dos lotes ganhos	18
3.3.2.2.10. Consulta dos lotes submetidos	19
3.3.2.2.11. Consulta da página de gestão do sistema	19
3.3.2.3. Funcionamento dos lotes	19
3.3.2.3.1. Submeter um lote	19
3.3.2.3.2. Aprovar submissão	19
3.3.2.3.3. Criar um lote	20
3.3.2.3.4. Editar um lote	20
3.3.2.4. Funcionamento de um leilão	20
3.3.2.4.1. Entrar num leilão	20
3.3.2.4.2. Sair de um leilão	21
3.3.2.4.3. Ganhar um lote	21
3.3.2.4.4. Criar um leilão	21
3.3.2.4.5. Editar um leilão	22
3.3.2.4.6. Remover um leilão	22
<b>4. Conceção do Sistema de Dados</b>	<b>23</b>
4.1. Apresentação geral da estrutura do sistema de dados	23
4.2. Descrição detalhada dos vários elementos de dados e seus relacionamentos	23
4.2.1. Entidades	23
4.2.2. Atributos	24
4.2.3. Relacionamentos	25
<b>5. Esboço das Interfaces do Sistema</b>	<b>26</b>
5.1. Estrutura geral das interfaces do sistema	26
5.2. Caracterização das interfaces	28
<b>6. Implementação da Aplicação</b>	<b>29</b>
6.1. Estrutura do código	29
6.1.1. wwwroot	29
6.1.2. Areas	30
6.1.3. Data	30
6.1.4. Migrations	30
6.1.5. Models	30
6.1.6. Repository	30
6.1.7. Utility	31

6.1.8. Views	31
6.1.9. appsettings.json	31
6.1.10. program.cs	31
6.1.11. Exemplo e explicação de um <i>controller method</i>	32
6.2. Ferramentas utilizadas	32
6.3. Serviços implementados	33
6.3.1. Página de registo	33
6.3.2. Página de login	34
6.3.3. Página de configurações da conta	34
6.3.4. Management: Lot	34
6.3.5. Management: Auction	36
6.3.6. Home Page	37
6.3.7. Privacy	37
6.3.8. Auction Details	38
6.3.9. Waiting	39
6.3.10. Bidding	39
6.4. Análise e avaliação da aplicação desenvolvida	39
<b>7. Conclusões</b>	<b>40</b>
<b>8. Trabalho Futuro</b>	<b>41</b>
<b>Referências</b>	<b>42</b>
<b>Lista de Siglas e Acrónimos</b>	<b>43</b>
<b>Anexos</b>	<b>44</b>

## Lista de Figuras

Figura 1: Diagrama de Gantt - Plano de execução	4
Figura 2: Task-Manager Bot	4
Figura 3: Modelo de domínio	12
Figura 4: Diagrama de casos de uso - Conta do utilizador	13
Figura 5: Diagrama de casos de uso - Consultas	14
Figura 6: Diagrama de casos de uso - Funcionamento dos lotes	14
Figura 7: Diagrama de casos de uso - Funcionamento dos leilões	15
Figura 8: Modelo lógico da base de dados	23
Figura 9: Interface Home page	26
Figura 10: Interface de consulta de lote	26
Figura 11: Interface para o leilão	27
Figura 12: Interface de registo	27
Figura 13: Interface Log in	28
Figura 14: Padrão MVC	29
Figura 15: Estrutura do código	29
Figura 16: Migração e update	30
Figura 17: Bootstrap CSS	32
Figura 18: Mensagem de sucesso	33
Figura 19: Página de registo	33
Figura 20: Página de login	34
Figura 21: Página de configurações da conta	34
Figura 22: Lista de lotes	35
Figura 24: Update de lote	35
Figura 25: Remover um lote	36
Figura 26: Lista de auctions	36
Figura 27: Update auction	37
Figura 28: Home page	37
Figura 29: Privacy policy page	38
Figura 30: Auction details (1)	38
Figura 31: Auction details (2) - Lot description	39
Figura 32: Waiting page	39

## Lista de Tabelas

Tabela 1: Identificação das entidades	24
Tabela 2: Identificação dos atributos das entidades	25
Tabela 3: Relacionamento entre entidades	25



# 1. Definição do Sistema

## 1.1. Contextualização

Alexandre Sousa, um empreendedor calculoso, começou a sua carreira como um contabilista certificado, e com o passar do tempo ficou reconhecido na sua área de trabalho. Possui uma postura calma; no entanto, o seu entusiasmo por vezes interrompe essa serenidade. Estas características, juntamente com a sua capacidade de liderança, levou-o a formar uma empresa que, posteriormente, obteve bastante sucesso a nível internacional.

Décadas mais tarde, em 2018, vendeu a sua parte da empresa, e deixou o cargo de diretor. Assim, com 51 anos de idade, decidiu afastar-se da vida profissional. Apesar disso, continuou a direcionar a sua atenção para o surgimento de novas empresas e modelos de negócio da indústria. Afinal, almeja sempre aprender e empenhar-se nas áreas que mais admira.

Em 2021, encontrou um domínio que instigou particularmente o seu interesse: leilões. Levado por essa vontade, estudou teoria de leilões, começando com os livros e artigos de Paul Klemperer, até chegar, por fim, ao estudo e análise do desenvolvimento e design de leilões. Passado um ano, decidiu começar uma nova empresa, desta vez de leilões online. Determinou, também, que a área de negócio seria “music memorabilia & vinyls”, inspirada pelo seu hobby de colecionar vinis exclusivos, e pela sua afeição à música.

Agora, em 2023, Alexandre Sousa reúne uma equipa: um gestor, um administrador de leilões, um especialista em vinis, entre outros. Para além disso, para o desenvolvimento do sistema, contrata uma empresa de desenvolvimento de software.

## 1.2. Fundamentação

Depois de ter obtido conhecimento teórico acerca de leilões, o seu interesse não terminou. Então, após falar com diversos especialistas na área, determinou que seria uma boa decisão colocar uma das suas ideias em prática. Estando já familiarizado com o mercado de leilões relacionados a música, tradicionais e online, e possuindo ele próprio uma extensa coleção de vinis, encarou esta decisão como uma oportunidade para juntar duas das suas áreas de interesse e para voltar à vida profissional, que tivera anos atrás, com entusiasmo. Além disso, reconheceu e concluiu que um leilão online traria muitas vantagens em comparação a um tradicional.

## 1.3. Objetivos

Portanto, definiu um conjunto de objetivos, alinhados com a sua ética de trabalho, que pretende atingir com a sua nova empresa.

- Estabelecer um valor baixo para a comissão do comprador (“buyer’s premium”) e para a comissão de venda (“sales commission”) [1].

- Fazer um sistema honesto e seguro através de diversos mecanismos: utilização de um tipo de leilão que siga esses princípios; manutenção de uma base de dados acerca dos produtos vendidos, ou à venda no mercado, atualizada por assistentes e até por alguns utilizadores selecionados; verificação da autenticidade dos vendedores e dos compradores, e verificação da sua identidade.
- Obter uma comunidade distribuída geograficamente.
- Desencorajar comportamentos colusivos, e predatórios [2].
- Preservar e valorizar a história musical dos artigos.
- Vender artigos assinados ou com dedicatórias, vinis de edições exclusivas, antigas ou limitadas, itens pessoais, adereços, roupas ou instrumentos utilizados pelos artistas em vídeos ou concertos, prêmios musicais, entre outros.
- Obter um sistema capaz de suportar interação com um grande número de clientes em simultâneo, sem latência ou outros problemas técnicos.
- Obter um sistema que permita a gestão dos eventos por parte de administradores e assistentes.

## 1.4. Viabilidade

Como a ideia visiona um leilão online, inúmeras vantagens provêm desse facto: os custos de transação para os compradores e vendedores diminuem, podem ser aceites diversos métodos de pagamento, podem existir leilões de duração ininterrupta, a exibição e descrição detalhada de produtos torna-se simples e facilmente acessível, e uma plataforma digital facilita a realização de leilões complexos e simultâneos, e amplia a participação, sendo que a localização geográfica dos participantes deixa de ser um obstáculo — uma grande mudança em relação aos leilões tradicionais.

Para além disso, há uma oferta limitada de leilões online nesta área de negócio, o que é surpreendente, visto que a música é uma forma de entretenimento consumida pela maioria das pessoas, desde a rádio até às trilhas sonoras dos filmes. Por consequência desta magnitude, o conjunto de pessoas com uma forte conexão ao mundo da música é significativo, e, portanto, origina a necessidade de existirem empresas nesta área de negócio.

Por fim, como a maioria dos clientes prefere métodos justos e honestos, será adotado um tipo de leilão que na teoria possui essas características, e este será customizado para se alinhar com o ambiente digital. Assim, não será utilizado, por exemplo, o “english auction”, mas sim um “ascending clock auction”, também chamado de leilão japonês.

**Ascending clock auction** *In the model most commonly used by auction theorists, the price rises continuously while bidders gradually quit the auction. Bidders observe when their competitors quit, and once someone quits they are not let back in. There is no possibility for one bidder to preempt the process by making a large “jump bid” [3].*

Ou seja, o preço aumenta continuamente, possivelmente por intervalos de tempo predeterminados, a partir de um preço inicial, e os participantes podem escolher sair. Esta ideia, apesar de tradicional, adequa-se ao ambiente digital pois o processo é muito mais simples do que, por exemplo, o de um “ascending-bid auction” normal, possuindo ainda assim características de um “general model”<sup>1</sup>. Também serão utilizados alguns aspetos da “second-price sealed-bid”, de uma forma semelhante ao leilão “Anglo-Dutch”, que consegue capturar as vantagens do “sealed-bid auction” [2].

---

<sup>1</sup>Também existem outros tipos básicos de leilões como o “descending-bid”, “first-price sealed-bid”, e “second-price sealed-bid”. Em teoria, mas tendo mais em consideração um “private-value model”, o “descending-bid” é semelhante a um “first-price sealed-bid”, sendo por vezes chamado de “open first-price auction”. Da mesma forma, o “ascending-bid” (o leilão japonês) é semelhante a um “second-price auction”, tal como enuncia o “revenue equivalence theorem”. [4]

## 1.5. Recursos a utilizar

### Ferramentas

- Microsoft Visual Studio e Microsoft Visual Studio Code
- Microsoft .NET C#
- Microsoft SQL Server
- Visual Paradigm
- Typst

### Recursos Humanos

- Levantadores de requisitos
- Utilizadores
- Patrocinadores

## 1.6. Equipa de trabalho

### Pessoal Interno

- Um gestor
- Um administrador de leilões
- Um especialista em vinis
- Um especialista em marketing
- Fotógrafos
- Assistentes
- Catalogadores

### Pessoal Externo

- Equipa de desenvolvimento do sistema:
  - Rodrigo Monteiro, Luís Figueiredo, Diogo Abreu e Miguel Gramoso.

### Outros

- Pessoas selecionadas para testar do sistema.

## 1.7. Plano de execução do trabalho

Antes de ser iniciada a execução do projeto, foi definido um plano de execução da fase inicial, através de um diagrama de Gantt (Figura 1), e cujas tarefas estão distribuídas ao longo do tempo e por diferentes grupos de tarefas.

É possível verificar que foi alocada uma quantidade de tempo significativa para a pesquisa e estudo acerca do tema, visto que achamos que influencia a qualidade do trabalho futuro. Por fim, definimos que algumas tarefas têm dependências, i.e., têm uma sequência fixa, e que algumas podem ser executadas em paralelo, de modo a que se aumente a eficiência do grupo de trabalho face às restrições de tempo. As restantes tarefas são relativas à implementação e desenvolvimento do sistema e serão planeadas no fim da fase inicial.

Para facilitar este desenvolvimento em paralelo e o cumprimento atempado das tarefas, foram planeadas diversas reuniões, e foi desenvolvido um *bot* de gestão de tarefas (Figura 2).

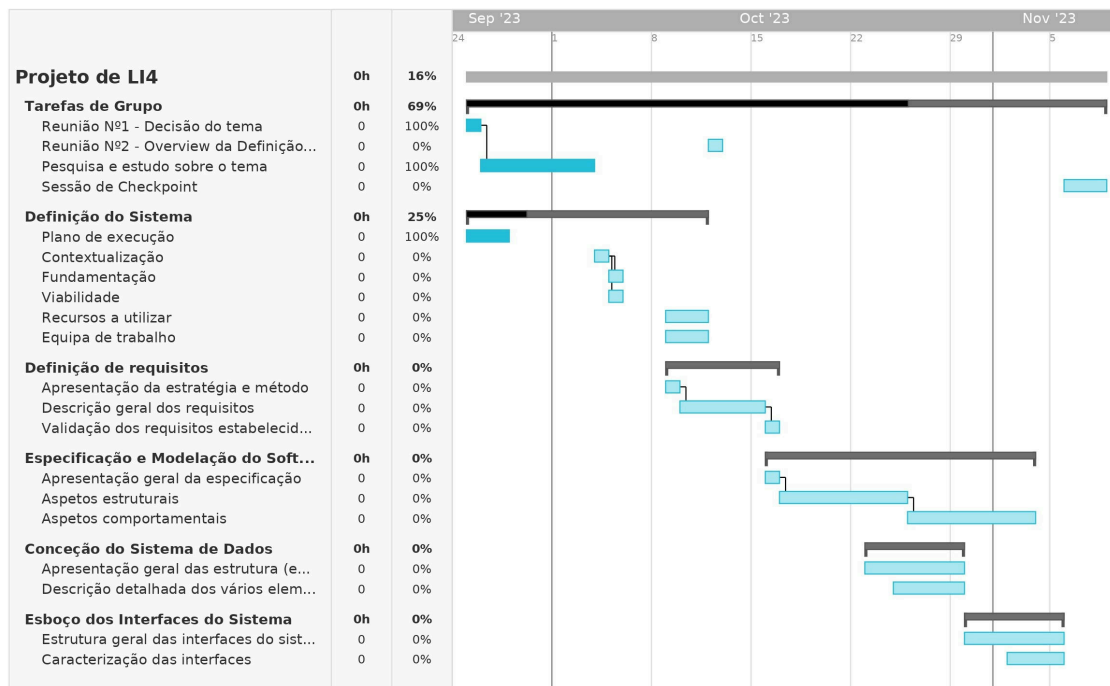


Figura 1: Diagrama de Gantt - Plano de execução

.get group 2 tasks

task-manager BOT 10/09/2023 3:32 PM

### Tarefas - Grupo 2

Task ID: 3  
 Name: **Plano de execução**  
 Begin Date: Mon Sep 25 2023  
 End Date: Wed Sep 27 2023  
 Completion: 100%  
 Assigned Team Members: 1  
 Group: 2

Message #bot-room

Figura 2: Task-Manager Bot

## 2. Definição de Requisitos

### 2.1. Apresentação da estratégia e método

Iniciámos o processo de definição de requisitos com uma análise aprofundada do mercado de leilões e colecionismo na área de “music memorabilia” e vinis. A pesquisa abrangeu desde as melhores práticas em plataformas semelhantes<sup>2</sup> até às expectativas dos colecionadores e vendedores neste domínio específico.

Para além disso, realizamos reuniões regulares com a equipa, incluindo Alexandre Sousa, para agregar perspectivas significativas no levantamento de requisitos, procurando manter uma aprovação contínua. Na organização de requisitos, estes foram divididos em requisitos funcionais e não funcionais. [5]

Por fim, a estratégia também incluiu entrevistas, inquéritos e questionários direcionados a potenciais utilizadores da plataforma. Esta abordagem proporcionou uma compreensão abrangente das expectativas dos utilizadores, permitindo ajustes iterativos nos requisitos.

### 2.2. Descrição geral dos requisitos levantados

#### 2.2.1. Requisitos Funcionais

##### 2.2.1.1. Registo de um utilizador

Requisitos do utilizador:

1. O utilizador só pode participar nos leilões se estiver registado. [Fonte: reunião com a equipa]
2. O utilizador escolhe criar uma conta na plataforma. [Fonte: reunião com a equipa]
3. Para que um utilizador seja administrador, tal terá de ser assinalado manualmente por outros administradores ou por responsáveis pelo sistema/ base de dados do sistema. [Fonte: reunião com a equipa]

Requisitos do sistema:

1. O sistema pede o nome completo, email e palavra-passe. [Fonte: reunião com a equipa]
2. Opcionalmente, o cliente pode escolher definir um método de pagamento, uma morada de envio, e uma autenticação de dois fatores introduzindo o número de telemóvel. (Requisitos 2 e 3.) [Fonte: reunião com a equipa]
3. Não é permitida a existência de utilizadores com o mesmo email. [Fonte: reunião com a equipa]
4. As informações são guardadas. [Fonte: reunião com a equipa]
5. É enviado um email para verificação da conta. [Fonte: reunião com a equipa]

##### 2.2.1.2. Adicionar um método de pagamento

Requisitos do utilizador:

1. O utilizador escolhe introduzir os seus dados de pagamento. [Fonte: reunião com a equipa]

Requisitos do sistema:

---

<sup>2</sup>Por exemplo, <https://www.omegaauctions.co.uk>, <https://www.liveauctioneers.com>, <https://www.artsy.net/auctions>, etc.

1. O utilizador fornece o nome associado ao cartão, o número do cartão, a data de validade, e o CVV/CVC. [Fonte: reunião com a equipa]
2. Pode introduzir uma morada de faturação ou escolher que esta seja a mesma da morada de envio. [Fonte: reunião com a equipa]
3. Também pode definir ACH como método de pagamento. [Fonte: reunião com a equipa]
4. O sistema valida o método de pagamento. [Fonte: reunião com a equipa]
5. Caso o método de pagamento não seja válido, o utilizador é informado e os dados não são aceites. [Fonte: reunião com a equipa]

#### **2.2.1.3. Adicionar uma morada**

Requisitos do utilizador:

1. O utilizador escolhe introduzir uma morada de faturação ou de envio. [Fonte: reunião com a equipa]

Requisitos do sistema:

1. O utilizador fornece a morada, país, estado/província/distrito, cidade e código-postal. [Fonte: reunião com a equipa]

#### **2.2.1.4. Autenticação**

Requisitos do utilizador:

1. O utilizador pode aceder à sua conta e pode participar nos leilões depois de estar autenticado. [Fonte: reunião com a equipa]

Requisitos do sistema:

1. O sistema pede o email e a palavra-passe do utilizador. [Fonte: reunião com a equipa]
2. O email tem de estar registado, caso contrário a autenticação é rejeitada. [Fonte: reunião com a equipa]
3. Caso esteja registado, e a palavra-passe encriptada corresponda à fornecida, a autenticação é feita com sucesso. [Fonte: reunião com a equipa]

#### **2.2.1.5. Terminar sessão**

Requisitos do utilizador:

1. O utilizador escolhe terminar a sessão, tendo uma sessão iniciada. [Fonte: reunião com a equipa]

Requisitos do sistema:

1. O sistema termina a sessão em curso. [Fonte: reunião com a equipa]
2. Para entrar na conta, terá de ser feita uma autenticação novamente. [Fonte: reunião com a equipa]

#### **2.2.1.6. Editar conta**

Requisitos do utilizador:

1. O utilizador acede ao seu perfil para editar os seus dados. [Fonte: reunião com a equipa]

Requisitos do sistema:

1. O email e a palavra-passe só podem ser editados se for efetuada uma nova autenticação (introdução da palavra-passe). [Fonte: reunião com a equipa]
2. Podem ser adicionadas até 5 moradas de faturação e de envio, ou editadas. [Fonte: entrevista a potenciais utilizadores]
3. Pode ser adicionado ou removido um método de pagamento. [Fonte: reunião com a equipa]
4. As alterações são guardadas, após o utilizador submeter as alterações. [Fonte: reunião com a equipa]

#### **2.2.1.7. Pesquisa**

Requisitos do utilizador:

1. Pesquisa de um lote, artista, ou leilão. [Fonte: reunião com a equipa]

2. O utilizador pode adicionar filtros e opções de ordenação dos resultados. [Fonte: entrevista a potenciais utilizadores]
3. O utilizador pode mudar a página de resultados caso exista mais do que uma. [Fonte: reunião com a equipa]

Requisitos do sistema:

1. O sistema recebe uma *string* de pesquisa com *keywords*, e, opcionalmente, determinados filtros (por exemplo, categoria, estado, etc.) e opções de sorting (por exemplo, preço, popularidade, relevância, *latest*, etc.). [Fonte: reunião com a equipa]
2. Retorna os resultados correspondentes (com paginação, se necessário). [Fonte: reunião com a equipa]

#### **2.2.1.8. Consulta de um lote**

Requisitos do utilizador:

1. O utilizador escolhe a visualização específica de um lote. [Fonte: reunião com a equipa]
2. Pode escolher visualizar todos os lotes do leilão a que o lote pertence. [Fonte: reunião com a equipa]
3. Pode adicionar ou remover o lote da sua lista de favoritos. [Fonte: inquérito a potenciais utilizadores]

Requisitos do sistema:

1. O sistema apresenta as imagens disponíveis, o nome, a descrição, o valor estimado, o preço inicial, as percentagens da comissão do comprador e a data do leilão. [Fonte: reunião com a equipa]

#### **2.2.1.9. Consulta de um leilão**

Requisitos do utilizador:

1. O utilizador escolhe visualizar os detalhes de um leilão. [Fonte: reunião com a equipa]
2. Pode escolher ser notificado quando este começar. [Fonte: reunião com a equipa]
3. Pode adicionar o leilão aos favoritos. [Fonte: inquérito a potenciais utilizadores]

Requisitos do sistema:

1. Fornece o nome/ tema do leilão, descrição, a data e uma listagem do(s) lote(s). [Fonte: reunião com a equipa]
2. Fornece opções de filtragem e ordenação dos lotes. [Fonte: reunião com a equipa]

#### **2.2.1.10. Consulta da lista de favoritos**

Requisitos do utilizador:

1. O utilizador pode escolher verificar a sua lista de leilões, lotes, ou de categorias favoritas. [Fonte: inquérito a potenciais utilizadores]

Requisitos do sistema:

1. O sistema mostra a lista de favoritos do utilizador com base na data de adição e com paginação, se necessário. [Fonte: inquérito a potenciais utilizadores]

#### **2.2.1.11. Consulta de notificações**

Requisitos do utilizador:

1. O utilizador escolhe visualizar as suas notificações (alertas/ avisos). [Fonte: reunião com a equipa]
2. Pode selecionar um filtro: “Todas as mensagens”/ “Mensagens Lidas”/ “Mensagens não lidas”. [Fonte: reunião com a equipa]
3. Seleciona uma ou mais notificações e marca como lida ou não lida. [Fonte: reunião com a equipa]

Requisitos do sistema:

1. Sistema mostra uma listagem das notificações. [Fonte: reunião com a equipa]

#### **2.2.1.12. Consulta das categorias**

Requisitos do utilizador:

1. O utilizador escolhe visualizar as categorias disponíveis. [Fonte: reunião com a equipa]
2. Pode adicionar uma categoria aos favoritos. [Fonte: inquérito a potenciais utilizadores]
3. Pode escolher visualizar os lotes associados a essa categoria. [Fonte: reunião com a equipa]

Requisitos do sistema:

1. O sistema apresenta uma lista das categorias, cada uma representada pelo seu nome e por uma imagem. [Fonte: reunião com a equipa]

#### **2.2.1.13. Consulta dos lotes ganhos**

Requisitos do utilizador:

1. O utilizador escolhe visualizar os lotes que ganhou. [Fonte: reunião com a equipa]
2. Pode seleccionar e visualizar um lote em específico. [Fonte: reunião com a equipa]

Requisitos do sistema:

1. O sistema apresenta uma lista dos lotes. [Fonte: reunião com a equipa]

#### **2.2.1.14. Submeter um lote**

Requisitos do utilizador:

1. O utilizador escolhe submeter um lote seu para ser leilado com ajuda da empresa. [Fonte: reunião com a equipa]

Requisitos do sistema:

1. O sistema pede:
  - uma descrição detalhada: material, dono original, quem fez o item ou itens, quando foi feito, condição, assinaturas, etc.; [Fonte: reunião com a equipa]
  - uma seleção de como é que foi obtido: comprado, herdado, encontrado, ou oferecido; [Fonte: reunião com a equipa]
  - uma seleção das categorias a que pertence (presentes no sistema); [Fonte: reunião com a equipa]
  - informações e documentos adicionais; [Fonte: reunião com a equipa]
  - e imagens. [Fonte: reunião com a equipa]
2. A informação submetida é guardada e fica numa fila de espera para verificação e aprovação por parte de especialistas. [Fonte: reunião com a equipa]

#### **2.2.1.15. Consultar lista de lotes submetidos**

Requisitos do utilizador:

1. O utilizador escolhe visualizar a lista dos próprios lotes, podendo filtrar os resultados por “Todos os lotes”/ “Vendidos”/ “Em leilão”/ “Sem leilão”/ “Em verificação”/ “Em espera”. [Fonte: reunião com a equipa]
2. Um utilizador não pode entrar no leilão relativo ao seu lote. [Fonte: reunião com a equipa]

Requisitos do sistema:

1. O sistema apresenta a lista de lotes de acordo com os filtros e com paginação, se necessário. [Fonte: reunião com a equipa]

#### **2.2.1.16. Entrar num leilão**

Requisitos do utilizador:

1. Quando a data de início é atingida, o leilão começa e é feita uma fila de espera para os utilizadores escolherem entrar no leilão. [Fonte: reunião com a equipa]

Requisitos do sistema:

1. O sistema regista que o utilizador irá participar no leilão e é adicionado a uma lista visível para outros utilizadores. [Fonte: reunião com a equipa]
2. O número de utilizadores a participar no leilão é incrementado – este número também é visível para todos os utilizadores. [Fonte: reunião com a equipa]



#### 2.2.1.17. Funcionamento/ Sair do leilão

Requisitos do utilizador:

1. Durante o intervalo de tempo definido, o utilizador escolhe ou não sair do leilão. [Fonte: reunião com a equipa]

Requisitos do sistema:

1. Caso o utilizador queira continuar no leilão, passa para a próxima ronda. [Fonte: reunião com a equipa]
2. Caso o utilizador não queira continuar no leilão, insere um valor que esteja no intervalo atual (por exemplo, caso queiram sair numa ronda em que o valor seja 300 e o valor de incremento 50, têm de escolher um valor entre 250 e 299), i.e., uma *sealed-bid* final. [Fonte: reunião com a equipa]
3. Caso um utilizador não escolha, sairá do leilão automaticamente com uma *sealed-bid* de valor mínimo. [Fonte: reunião com a equipa]
4. Se restar apenas uma pessoa numa ronda, essa pessoa ganha e paga o segundo valor mais alto das *sealed-bids* da ronda anterior. [Fonte: reunião com a equipa]
5. Se todas as pessoas restantes saírem numa ronda, ganha a que tiver o maior valor na sua *sealed-bid* final, e paga o segundo valor mais alto. Neste caso, o desempate é feito com base no tempo de submissão da *sealed-bid* final. [Fonte: reunião com a equipa]
6. O sistema utiliza um ascending-clock que a cada determinado intervalo de tempo (por exemplo, 3 minutos), incrementa um determinado valor. [Fonte: reunião com a equipa]
7. O sistema disponibiliza a lista de utilizadores a participar, e o seu número total, em tempo real. [Fonte: reunião com a equipa]
8. O sistema fornece uma opção de sair do leilão. [Fonte: reunião com a equipa]
9. O sistema regista as entradas e saídas do leilão, tal como os valores submetidos na saída e o respetivo *timestamp*. [Fonte: reunião com a equipa]
10. Todos os acontecimentos do leilão são guardados num *log* automaticamente pelo sistema. [Fonte: reunião com a equipa]

#### 2.2.1.18. Página de gestão do sistema

Requisitos do utilizador:

1. O utilizador tem de ser administrador para ter acesso a estas funcionalidades de gestão do sistema. [Fonte: reunião com a equipa]

Requisitos do sistema:

1. O sistema disponibiliza opções de gestão de leilões, de lotes e de utilizadores. (Requisitos 18, 21, 23.) [Fonte: reunião com a equipa]

#### 2.2.1.19. Página de gestão de leilões

Requisitos do utilizador:

1. O administrador escolhe visualizar a página de gestão de leilões a partir da página de gestão do sistema principal. [Fonte: reunião com a equipa]

Requisitos do sistema:

1. O sistema fornece uma opção de criação de um novo leilão. [Fonte: reunião com a equipa]
2. O sistema fornece uma lista dos leilões, os quais podem ser editados ou removidos. [Fonte: reunião com a equipa]
3. O sistema fornece uma lista de leilões a decorrer. [Fonte: reunião com a equipa]

#### 2.2.1.20. Criação de um leilão

Requisitos do utilizador:

1. O administrador escolhe criar um leilão na página de gestão de leilões. [Fonte: reunião com a equipa]

Requisitos do sistema:

1. O sistema pede o nome e o tema do leilão, uma descrição, uma data de início, uma escolha entre “início automático” ou “início manual”, e o intervalo de tempo do *ascending clock*. [Fonte: reunião com a equipa]
2. O sistema apresenta um conjunto de lotes que não pertencem a outros leilões, os quais podem ser filtrados (por exemplo, por categoria) e que podem ser selecionados para fazerem parte do leilão a ser criado. [Fonte: reunião com a equipa]

#### **2.2.1.21. Página de gestão de lotes**

Requisitos do utilizador:

1. O administrador pode escolher visualizar a lista de lotes submetidos por verificar, e de lotes aprovados com e sem leilões. [Fonte: reunião com a equipa]
2. Pode criar, editar (por exemplo, associando-o a um leilão) ou remover um lote. [Fonte: reunião com a equipa]

Requisitos do sistema:

1. O sistema disponibiliza a lista de lotes por verificar e verificados. [Fonte: reunião com a equipa]
2. O sistema permite adicionar ou remover lotes verificados de um leilão. [Fonte: reunião com a equipa]
3. O sistema permite a verificação e aprovação de um lote. [Fonte: reunião com a equipa]

#### **2.2.1.22. Aprovação de um lote**

Requisitos do utilizador:

1. O administrador seleciona um lote. [Fonte: reunião com a equipa]
2. O administrador pode aprovar ou não aprovar. [Fonte: reunião com a equipa]

Requisitos do sistema:

1. O sistema envia uma notificação ao utilizador a informar se o lote foi aprovado ou não. [Fonte: reunião com a equipa]
2. Caso tenha sido aprovado, são enviadas informações acerca do envio do seu lote. [Fonte: reunião com a equipa]
3. O lote passa a estar com estado de “espera” até chegar ao armazém da empresa. [Fonte: reunião com a equipa]

#### **2.2.2. Chegada de um item (ou conjunto de itens)**

Requisitos do utilizador:

1. Funcionário ou administrador indicam que um lote já pode ser leilado, estando presente no armazém e catalogado. [Fonte: reunião com a equipa]

Requisitos do sistema:

1. O sistema muda o estado do lote de “em espera” para “disponível”. [Fonte: reunião com a equipa]

#### **2.2.2.1. Página de gestão de utilizadores**

Requisitos do utilizador:

1. O administrador escolhe gerir os utilizadores do sistema. [Fonte: reunião com a equipa]

Requisitos do sistema:

1. O sistema fornece uma lista de todos os utilizadores (com paginação, se necessário), e uma opção de pesquisa e de filtragem de resultados. [Fonte: reunião com a equipa]
2. O sistema permite notificar, editar, banir e remover utilizadores. [Fonte: reunião com a equipa]

#### **2.2.3. Requisitos não funcionais**

1. O sistema, e, portanto, o servidor, deverá estar operacional 24 horas por dia, ou seja, sempre em funcionamento. [Fonte: reunião com a equipa]

2. O sistema deverá possuir documentação adequada acerca do seu funcionamento e implementação.  
[Fonte: reunião com a equipa]

## **2.3. Validação dos requisitos**

Após a execução dos métodos de levantamento estabelecidos, procedemos à organização e análise detalhada de todos os requisitos identificados. Este processo teve como objetivo eliminar quaisquer erros, inconsistências, redundâncias ou ambiguidades presentes no documento final, garantindo assim a sua integridade e completude.

Na reunião subsequente, todos os membros participaram ativamente na discussão detalhada, validando e aprovando coletivamente os requisitos em conformidade com as expectativas do projeto.

## 3. Especificação e Modelação do Software

### 3.1. Apresentação geral da especificação

Esta secção pretende formar uma visão abrangente do sistema, detalhando tanto os aspetos estruturais como os aspetos comportamentais. Para isso, é utilizado um modelo de domínio, um diagrama de casos de uso, os respetivos casos de uso estruturados, e diagramas de atividade. Verificar-se-á que o sistema tem três elementos fundamentais: o utilizador, o leilão e o lote.

### 3.2. Aspetos estruturais

#### 3.2.1. Modelo de domínio

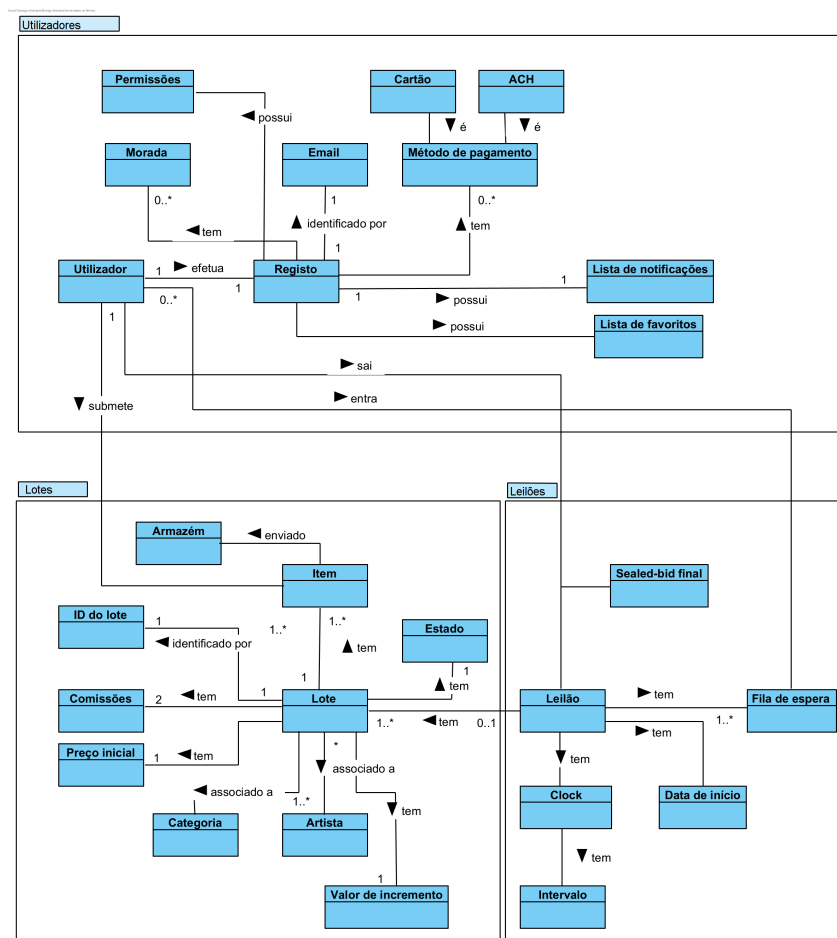


Figura 3: Modelo de domínio

Este modelo descreve o problema, não uma solução, sendo o seu objetivo explicitar o contexto em que o sistema irá agir, a estrutura essencial do problema tendo em conta os requisitos recolhidos e organizados. Assim, é descrito que os utilizadores possuem um registo, o qual é identificado por um email único. A este registo pode estar associado um ou mais métodos de pagamento, e uma ou mais moradas de faturação e de envio. Um registo pode ter permissões comuns ou de administrador, e possui uma lista de notificações (acerca de leilões, de atualizações dos estados das submissões, de lotes ganhos, etc.) e uma lista de favoritos (leilões, categorias, artistas, e lotes).

Um utilizador pode submeter um ou mais itens, que será considerado um lote, e quando é verificado e aprovado, pode ser enviado para um armazém da empresa e depois devidamente catalogado, passando a estar disponível para pertencer a um leilão. Um lote, tal como já foi descrito antes com mais detalhe, está associado a uma ou mais categorias e artistas, possui os valores das comissões de comprador e de vendedor, um número identificador, um preço inicial, um valor de incremento intervalar, e um estado (vendido, em leilão, disponível, em espera, e em verificação).

Por fim, o leilão tem uma data de início, e uma fila de espera com uma determinada duração de tempo (para cada lote). Como é um *ascending clock auction*, possui um relógio que incrementa o valor do lote de acordo com um intervalo de tempo predeterminado. Num leilão podem ser leiloados um ou mais lotes. Quando um utilizador quer sair do leilão de um lote, faz uma *sealed bid* final.

### 3.3. Aspetos comportamentais

#### 3.3.1. Diagramas de casos de uso

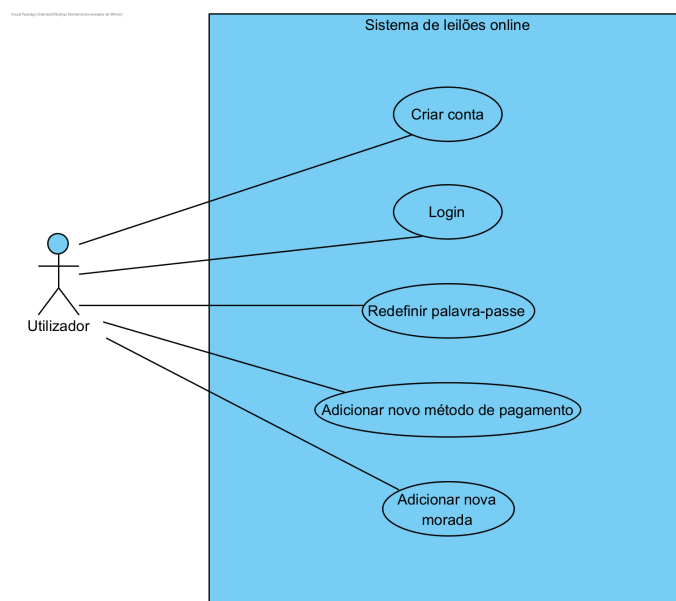


Figura 4: Diagrama de casos de uso - Conta do utilizador



Figura 5: Diagrama de casos de uso - Consultas

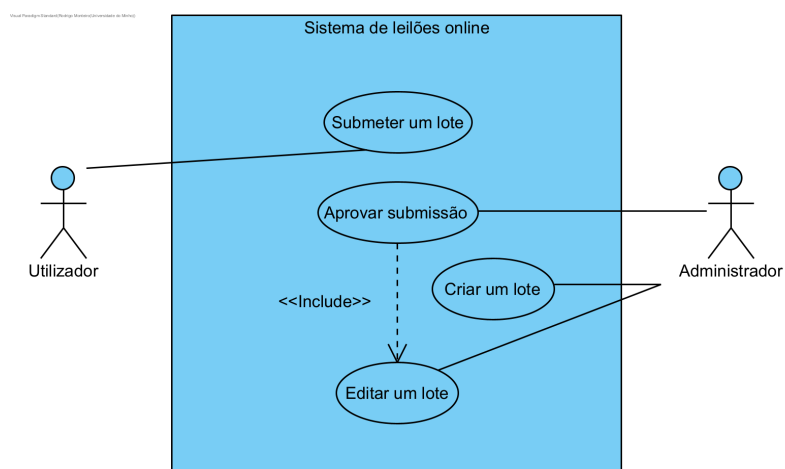


Figura 6: Diagrama de casos de uso - Funcionamento dos lotes

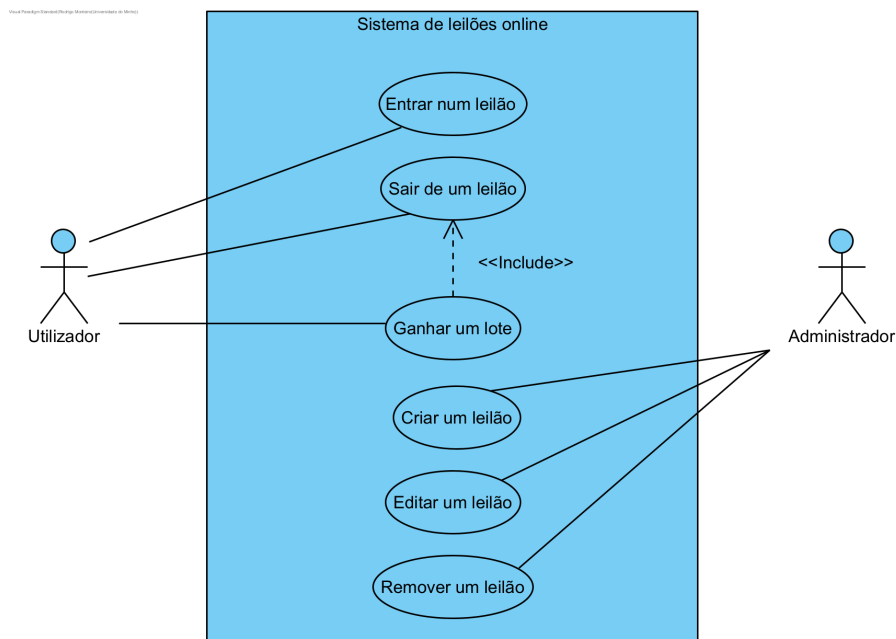


Figura 7: Diagrama de casos de uso - Funcionamento dos leilões

### 3.3.2. Casos de uso

#### 3.3.2.1. Conta do utilizador

##### 3.3.2.1.1. Criar conta

- **Descrição:** Cria uma nova conta de utilizador.
- **Pré-condição:** True.
- **Pós-condição:** O sistema regista a conta do utilizador.
- **Fluxo normal:**
  1. O sistema pede para introduzir o seu nome completo, email e palavra-passe.
  2. O cliente escreve essas informações.
  3. O sistema verifica se as informações são válidas.
  4. O sistema pergunta ao utilizador se quer inserir um método de pagamento e morada.
  5. O cliente escolhe inserir esses dados.
  6. O sistema pede ao cliente para introduzir um método de pagamento e uma morada de envio e de faturação.
  7. O cliente indica um método de pagamento e uma morada de envio e de faturação.
  8. O sistema valida o método de pagamento e as moradas.
  9. O cliente é registado no sistema.
- **Fluxo alternativo (1):** [Email já registado no sistema] (passo 3)
  - 3.1. O sistema avisa que o email já está registado.
  - 3.2. Regressa ao passo 1.
- **Fluxo alternativo (2):** [Palavra-passe inválida] (passo 3)
  - 3.1. O sistema avisa que a palavra-passe é inválida e o motivo.
  - 3.2. Regressa ao passo 1.
- **Fluxo alternativo (3):** [Cliente escolhe não inserir método de pagamento] (passo 5)
  - 4.1. Avança para o passo 9.
- **Fluxo alternativo (4):** [Método de pagamento ou morada inválida] (passo 8)
  - 8.1. O sistema avisa que os dados são inválidos.
  - 8.2. Regressa ao passo 6.

#### 3.3.2.1.2. Login

- **Descrição:** Utilizador entra no sistema.
- **Pré-condição:** O utilizador está registado no sistema.
- **Pós-condição:** O utilizador fica autenticado, podendo utilizar mais funcionalidades do sistema.
- **Fluxo normal:**
  1. O utilizador fornece o seu email e a palavra-passe.
  2. O sistema verifica se o email e palavra-passe são válidos.
  3. O utilizador fica autenticado.
- **Fluxo alternativo (1):** [Email ou palavra-passe inválida] (passo 2)
  - 2.1. O sistema verifica que o email não está registado ou que o email está registado e a palavra-passe é inválida.
  - 2.2. Regressa ao passo 1.

#### 3.3.2.1.3. Redefinir palavra-passe

- **Descrição:** O utilizador redefine a sua palavra-passe.
- **Pré-condição:** O utilizador está registado e autenticado.
- **Pós-condição:** As mudanças são guardadas no sistema.
- **Fluxo normal:**
  1. O utilizador acede ao seu perfil e escolhe redefinir a sua palavra-passe.
  2. O sistema pede que a sua palavra passe atual e a nova sejam inseridas.
  3. O cliente insere esses dados.
  4. O sistema valida os dados.
  5. A palavra-passe do utilizador é atualizada no sistema.
- **Fluxo alternativo (1):** [Dados inválidos] (passo 4)
  - 4.1. O sistema avisa que a palavra-passe é inválida.
  - 4.2. Regressa ao passo 2.

#### 3.3.2.1.4. Adicionar novo método de pagamento

- **Descrição:** O utilizador adiciona um novo método de pagamento.
- **Pré-condição:** O utilizador está registado e autenticado.
- **Pós-condição:** As mudanças são guardadas no sistema.
- **Fluxo normal:**
  1. O utilizador acede ao seu perfil e escolhe adicionar um novo método de pagamento.
  2. O sistema pede para escolher o tipo de método de pagamento (cartão ou ACH).
  3. O utilizador escolhe o tipo e insere os parâmetros necessários para esse tipo.
  4. O sistema valida os dados inseridos.
  5. O sistema guarda o método de pagamento adicionado.
- **Fluxo alternativo (1):** [Dados inválidos] (passo 4)
  - 4.1. O sistema avisa que os dados inseridos são inválidos.
  - 4.2. Regressa ao passo 2.

#### 3.3.2.1.5. Adicionar nova morada

- **Descrição:** O utilizador adiciona uma nova morada.
- **Pré-condição:** O utilizador está registado e autenticado.
- **Pós-condição:** As mudanças são guardadas no sistema.
- **Fluxo normal:**
  1. O utilizador acede ao seu perfil e escolhe adicionar uma morada.
  2. O sistema pede o tipo de morada, e os seus dados.
  3. O utilizador escolhe o tipo de morada, de envio e/ou de faturação, e insere os respetivos dados.
  4. O sistema valida a morada.
  5. A morada é adicionada ao registo do utilizador.



- **Fluxo alternativo (1):** [Dados inválidos] (passo 4)
  - 4.1. O sistema avisa que um ou mais dos parâmetros é inválido.
  - 4.2. Regressa ao passo 2.

### 3.3.2.2. Consultas

#### 3.3.2.2.1. Pesquisar

- **Descrição:** O utilizador utiliza a funcionalidade de pesquisa do sistema.
- **Pré-condição:** True.
- **Pós-condição:** O sistema mostra um resultado, se existir.
- **Fluxo normal:**
  1. O utilizador insere os termos de pesquisa.
  2. Opcionalmente, adiciona filtros (artista, lote, categoria ou leilão).
  3. Opcionalmente, adiciona opções de ordenação (preço, popularidade, relevância, início mais próximo, adicionados recentemente).
  4. O sistema processa a consulta e procura por correspondências nos dados armazenados.
  5. O sistema apresenta uma lista de resultados.
- **Fluxo alternativo (1):** [Não existem resultados] (passo 5)
  - 5.1. O sistema informa que não existem resultados correspondentes à pesquisa.

#### 3.3.2.2.2. Consulta de um lote

- **Descrição:** O utilizador escolhe consultar um lote.
- **Pré-condição:** True.
- **Pós-condição:** O sistema apresenta os dados relativos ao lote.
- **Fluxo normal:**
  1. O utilizador seleciona um lote.
  2. O sistema mostra os dados relativos ao lote.
  3. Opcionalmente, o utilizador adiciona o lote à sua lista de favoritos.
  4. Opcionalmente, o utilizador decide consultar o leilão desse lote.
- **Fluxo alternativo (1):** [Utilizador não registado] (passo 3)
  - 3.1. O sistema avisa que o utilizador precisa de estar registado para adicionar o lote à lista de favoritos.
  - 3.2. «include» Criar conta

#### 3.3.2.2.3. Consulta de um leilão

- **Descrição:** O utilizador escolhe consultar um leilão.
- **Pré-condição:** True.
- **Pós-condição:** O sistema apresenta os dados relativos ao leilão.
- **Fluxo normal:**
  1. O utilizador seleciona um leilão.
  2. O sistema apresenta os dados relativos ao leilão, e uma listagem dos seus itens com opções de filtragem e de ordenação.
- **Fluxo alternativo (1):** [Utilizador decide consultar um lote] (passo 2)
  - 2.1. «include» Consulta de um lote

#### 3.3.2.2.4. Consulta das listas de favoritos

- **Descrição:** O utilizador escolhe consultar a sua lista de favoritos.
- **Pré-condição:** O utilizador está registado e autenticado no sistema.
- **Pós-condição:** O sistema apresenta a lista de favoritos solicitada.
- **Fluxo normal:**
  1. O utilizador seleciona a opção de aceder às suas listas de favoritos.

2. O sistema apresenta as diferentes listas: de lotes, de leilões, de artistas e de categorias.
3. O utilizador seleciona uma das listas.
4. O sistema apresenta essa lista.

#### 3.3.2.2.5. Remoção de um item da lista de favoritos

- **Descrição:** O utilizador escolhe remover um item da lista de favoritos.
- **Pré-condição:** O utilizador está registado e autenticado no sistema.
- **Pós-condição:** O item é retirado da lista.
- **Fluxo normal:**
  1. «include» Consulta das listas de favoritos
  2. O utilizador seleciona o item que quer remover e indica a ação de remoção.
  3. O sistema remove o item da lista.
- **Fluxo de exceção (1):** [Não existem itens para remover] (passo 1)
  - 1.1. O sistema não fornece a opção de remoção de itens.

#### 3.3.2.2.6. Consulta das notificações

- **Descrição:** O utilizador consulta as notificações disponíveis.
- **Pré-condição:** O utilizador está registado e autenticado no sistema.
- **Pós-condição:** O sistema apresenta as notificações, se existirem.
- **Fluxo normal:**
  1. O utilizador seleciona a opção de visualizar as suas notificações.
  2. O sistema apresenta as notificações.
  3. O utilizador seleciona uma notificação.
  4. O sistema mostra os dados relativos a essa notificação.
- **Fluxo alternativo 1:** [Não existem notificações] (passo 2)
  - 2.1. O sistema informa que não existem notificações.

#### 3.3.2.2.7. Consulta das categorias

- **Descrição:** O utilizador consulta as categorias presentes no sistema.
- **Pré-condição:** True.
- **Pós-condição:** O sistema apresenta a lista de categorias.
- **Fluxo normal:**
  1. O utilizador escolhe consultar as categorias.
  2. O sistema apresenta uma listagem das categorias, identificadas pelo seu nome.
- **Fluxo alternativo (1):** [Utilizador escolhe consultar uma categoria] (passo 2)
  - 2.1. «include» Consulta de uma categoria

#### 3.3.2.2.8. Consulta de uma categoria

- **Descrição:** O utilizador consulta uma categoria.
- **Pré-condição:** True.
- **Pós-condição:** O sistema apresenta os dados da categoria.
- **Fluxo normal:**
  1. O utilizador seleciona uma categoria.
  2. O sistema mostra uma listagem dos lotes nessa categoria (uma pesquisa com filtragem por essa categoria).

#### 3.3.2.2.9. Consulta dos lotes ganhos

- **Descrição:** O utilizador consulta os seus lotes ganhos.
- **Pré-condição:** O utilizador está registado e autenticado no sistema.
- **Pós-condição:** O sistema apresenta os lotes ganhos do utilizador.

- **Fluxo normal:**
  1. O sistema apresenta uma listagem dos lotes ganhos.
- **Fluxo alternativo (1):** [Consulta de um lote ganho] (passo 1)
  - 1.1. «inclui» Consulta de um lote
- **Fluxo de exceção (1):** [Não existem lotes ganhos] (passo 1)
  - 1.1. O sistema informa que não existem lotes ganhos.

#### 3.3.2.2.10. Consulta dos lotes submetidos

- **Descrição:** O utilizador consulta os lotes que submeteu.
- **Pré-condição:** O utilizador está registado e autenticado no sistema.
- **Pós-condição:** O sistema apresenta os lotes que o utilizador submeteu.
- **Fluxo normal:**
  1. O sistema apresenta uma listagem dos lotes submetidos e o seu estado.
- **Fluxo alternativo (1):** [Utilizador seleciona um lote] (passo 1)
  - 1.1. O sistema mostra a informação que o utilizador forneceu acerca desse lote, o seu estado e outras informações (em verificação, em espera, se pertence a um leilão, histórico, utilizador que ganhou o lote, etc.).
- **Fluxo de exceção (1):** [Não existem lotes submetidos] (passo 1)
  - 1.1. O sistema informa que não existem lotes submetidos.

#### 3.3.2.2.11. Consulta da página de gestão do sistema

- **Descrição:** O administrador acede à área de gestão do sistema.
- **Pré-condição:** O utilizador está registado e autenticado como administrador.
- **Pós-condição:** True.
- **Fluxo normal:**
  1. O sistema fornece uma área de gestão de leilões.
  2. O sistema fornece uma área de gestão de utilizadores.
  3. O sistema fornece uma área de gestão de lotes (incluindo submissões).

### 3.3.2.3. Funcionamento dos lotes

#### 3.3.2.3.1. Submeter um lote

- **Descrição:** O utilizador submete um lote, fornecendo uma série de informações necessárias, e que depois serão verificadas.
- **Pré-condição:** O utilizador está registado e autenticado no sistema.
- **Pós-condição:** A submissão é feita com sucesso, e colocada numa fila de submissões do sistema.
- **Fluxo normal:**
  1. O sistema pede uma descrição detalhada.
  2. O sistema apresenta uma seleção obrigatória de como é que o item foi obtido.
  3. O sistema pede uma seleção de categorias a que o lote pertence.
  4. O sistema pede informações e documentos adicionais, e imagens.
  5. O cliente fornece os dados necessários.
  6. O sistema valida os dados.
  7. O sistema guarda a submissão, e é marcada como “em verificação.”
- **Fluxo alternativo (1):** [Dados inválidos ou em falta] (passo 6)
  - 6.1. O sistema informa que um ou mais dados são inválidos ou estão em falta.
  - 6.2. Regressa ao passo 1.

#### 3.3.2.3.2. Aprovar submissão

- **Descrição:** Um administrador aprova uma submissão.

- **Pré-condição:** O utilizador está registado e autenticado como administrador.
- **Pós-condição:** A submissão é aprovada.
- **Fluxo normal:**
  1. O administrador aprova uma submissão selecionada.
  2. O sistema envia uma notificação ao utilizador a informar a aprovação.
  3. O sistema cria o lote automaticamente, permitindo ao administrador de seguida editá-lo. («include» Editar um lote)
  4. O sistema coloca o lote com estado “em espera”, e com visibilidade “privado”.
- **Fluxo de exceção (1):** [O administrador não aprova a submissão] (passo 1)
  - 1.1. O administrador não aprova o lote selecionado.
  - 1.2. O sistema envia uma notificação ao utilizador a informar que lote não foi aprovado.
  - 1.3. O sistema marca o lote como “recusado”.

#### 3.3.2.3.3. Criar um lote

- **Descrição:** Um administrador cria um lote.
- **Pré-condição:** O utilizador está registado e autenticado como administrador.
- **Pós-condição:** O lote é criado e guardado no sistema.
- **Fluxo normal:**
  1. O sistema adiciona um número identificador ao lote automaticamente.
  2. O sistema pede um nome, uma descrição, e imagens (opcional).
  3. O sistema pede uma seleção de categorias.
  4. O sistema pede uma seleção de estado e de visibilidade (com *default*: “em espera” e “privado”).
  5. O sistema pede para associar a um leilão (opcional).
  6. O administrador fornece esses dados.
  7. O sistema valida os dados.
  8. O lote é guardado no sistema.
- **Fluxo alternativo (1):** [Dados inválidos ou em falta] (passo 7)
  - 7.1. O sistema informa que um ou mais dados são inválidos ou estão em falta.
  - 7.2. Regressa ao passo 1.

#### 3.3.2.3.4. Editar um lote

- **Descrição:** Um administrador edita um lote.
- **Pré-condição:** O utilizador está registado e autenticado como administrador.
- **Pós-condição:** O alterações feitas no lote são guardadas no sistema.
- **Fluxo normal:**
  1. O sistema fornece os campos de edição do lote.
  2. O administrador edita os campos que achar necessário.
  3. O administrador submete as alterações.
  4. O sistema valida as alterações.
  5. O sistema guarda as alterações.
- **Fluxo alternativo (1):** [Alterações inválidas] (passo 4)
  - 4.1. O sistema informa que as alterações feitas são inválidas.
  - 4.2. Regressa ao passo 1.

#### 3.3.2.4. Funcionamento de um leilão

##### 3.3.2.4.1. Entrar num leilão

- **Descrição:** O utilizador entra num leilão.
- **Pré-condição:** O utilizador não está a participar noutro leilão ao mesmo tempo. O utilizador está registado e autenticado no sistema.
- **Pós-condição:** O utilizador é adicionado à fila de espera.

- **Fluxo normal:**
  1. O utilizador indica que quer entrar no leilão.
  2. O sistema adiciona o utilizador à fila de espera.
  3. O sistema informa que o utilizador entrou no leilão.
  4. O sistema apresenta a lista de utilizadores na fila de espera e a sua contagem.
- **Fluxo de exceção (1):** [Sistema não permite entrada] (passo 2)
  - 2.1. O sistema informa que a fila está cheia, ou que o tempo de entrada já passou.

#### 3.3.2.4.2. Sair de um leilão

- **Descrição:** O utilizador decide sair do leilão de um lote.
- **Pré-condição:** O utilizador está registado e autenticado no sistema. O utilizador está na fila de espera do leilão desse lote.
- **Pós-condição:** O utilizador sai do leilão.
- **Fluxo normal:**
  1. O sistema apresenta a fila dos participantes em tempo real.
  2. A cada intervalo de tempo predefinido, o sistema aumenta o valor do lote de acordo com o seu valor de incremento.
  3. O utilizador escolhe sair nesse intervalo de tempo.
  4. O sistema pede que seja inserida uma licitação privada final que esteja no intervalo do valor atual.
  5. O utilizador insere um valor.
  6. O sistema regista a saída do utilizador.
- **Fluxo alternativo (1):** [O utilizador não insere um valor no intervalo de tempo devido] (passo 5)
  - 5.1. O sistema retira o utilizador do leilão, registando a saída com uma licitação final de valor mínimo.

#### 3.3.2.4.3. Ganhar um lote

- **Descrição:** O utilizador ganha um lote num leilão.
- **Pré-condição:** O utilizador está registado e autenticado no sistema. O utilizador está registado como participante no leilão.
- **Pós-condição:** O utilizador ganha o lote.
- **Fluxo normal:**
  1. O utilizador escolhe sair do leilão. («include» Sair de um leilão)
  2. Sendo o último utilizador restante, ganha o lote e paga o segundo valor mais alto das licitações privadas e finais da ronda anterior.
- **Fluxo alternativo (1):** [Todos os utilizadores restantes também escolhem sair] (passo 2)
  - 2.1. O utilizador ganha por ter o maior valor de licitação privada e final, pagando o segundo valor mais alto das licitações feitas nessa ronda.
- **Fluxo alternativo (1.2):** [Empate nas licitações finais] (passo 2.1)
  - 2.1.1. O utilizador ganha por ter o menor tempo de saída registado.

#### 3.3.2.4.4. Criar um leilão

- **Descrição:** Um administrador cria um leilão.
- **Pré-condição:** O utilizador está registado e autenticado como administrador.
- **Pós-condição:** A criação do leilão é efetuada no sistema.
- **Fluxo normal:**
  1. O sistema pede o nome do leilão, o tema, a data de início, o tipo de inicialização (automática ou manual), o intervalo de tempo de cada ronda, e a visibilidade.
  2. O administrador insere esses dados.
  3. O sistema valida os dados inseridos.
  4. O sistema fornece uma lista de lotes disponíveis/ sem leilão, com opções de filtragem.
  5. O administrador seleciona um ou mais lotes, e submete.
  6. O sistema valida a submissão.
  7. O leilão é criado e guardado no sistema.

- **Fluxo alternativo (1):** [Dados inválidos] (passo 3)
  - 3.1. O sistema informa que os dados inseridos são inválidos/ estão em falta.
  - 3.2. Regressa ao passo 1.
- **Fluxo alternativo (2):** [Administrador não selecionou] (passo 5)
  - 5.1. O sistema informa que é necessário selecionar um ou mais itens para criar um leilão.
  - 5.2. Regressa ao passo 4.

#### 3.3.2.4.5. Editar um leilão

- **Descrição:** Um administrador edita os dados de um leilão.
- **Pré-condição:** O utilizador está registado e autenticado como administrador.
- **Pós-condição:** As alterações feitas são guardadas.
- **Fluxo normal:**
  1. O sistema apresenta os parâmetros que podem ser editados.
  2. O sistema fornece uma lista dos lotes do leilão, podendo ser removidos.
  3. O sistema fornece uma opção de selecionar novos lotes ao leilão, fornecendo uma lista dos lotes disponíveis, com opções de filtragem.
  4. O administrador efetua e submete as suas edições.
  5. O sistema valida as edições feitas.
  6. O sistema guarda as alterações, e envia uma notificação acerca das alterações feitas aos utilizadores que têm o leilão na lista de favoritos.
- **Fluxo alternativo (1):** [As edições não são válidas] (passo 6)
  - 6.1. O sistema informa que as edições feitas não são válidas.
  - 6.2. Regressa ao passo 1.

#### 3.3.2.4.6. Remover um leilão

- **Descrição:** Um administrador remove um leilão.
- **Pré-condição:** O utilizador está registado e autenticado como administrador.
- **Pós-condição:** O leilão é removido do sistema.
- **Fluxo normal:**
  1. O administrador seleciona um leilão, e escolhe removê-lo.
  2. O sistema pede a sua palavra-passe.
  3. O administrador insere a sua palavra-passe.
  4. O sistema valida a palavra-passe.
  5. O leilão é removido do sistema.
  6. O sistema envia notificações para os utilizadores que têm o leilão na lista de favoritos acerca da sua remoção.
  7. Os dados do leilão removido são guardados em ficheiros *backup*.
- **Fluxo de exceção (1):** [Palavra-passe não é válida] (passo 4)
  - 4.1. O sistema informa que a palavra-passe não é válida.

## 4. Conceção do Sistema de Dados

### 4.1. Apresentação geral das estrutura do sistema de dados

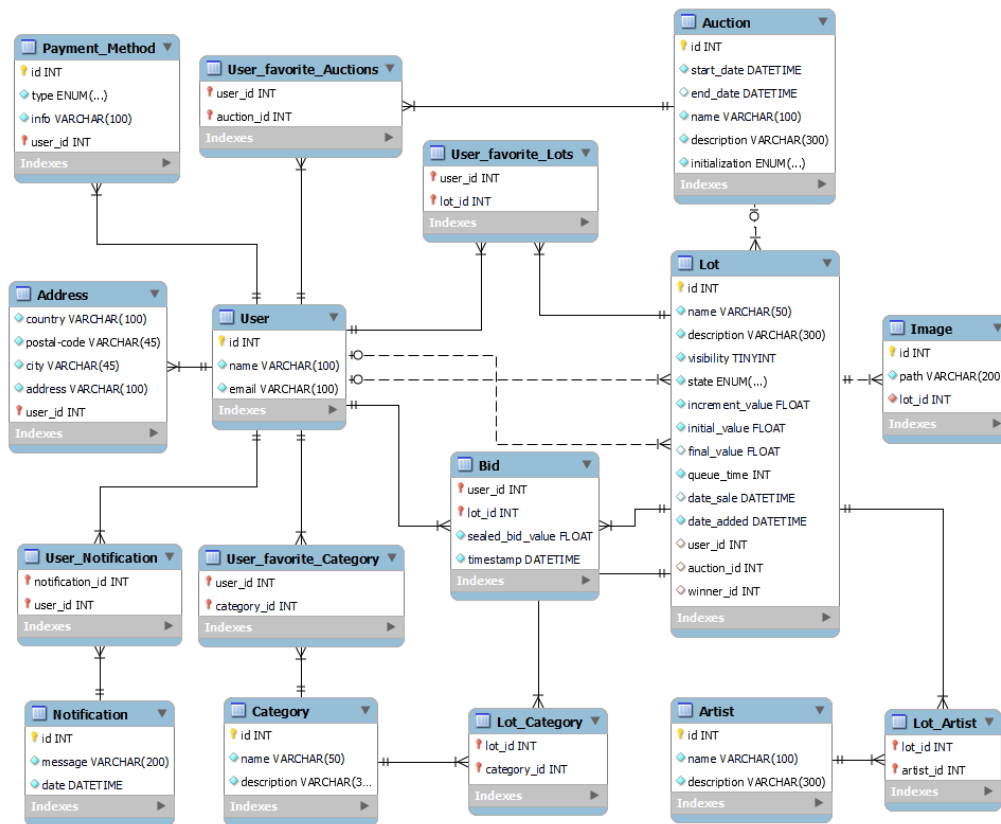


Figura 8: Modelo lógico da base de dados

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

#### 4.2.1. Entidades

Entidade	Descrição	Ocorrência
Utilizador	Termo geral para os que usam a aplicação com o objetivo de comprar ou vender	Cada utilizador poderá visitar o site para comprar ou vender um ou mais itens num leilão
Lote	Termo geral para o item ou conjunto de itens submetidos pelos utilizadores para posterior avaliação	Os utilizadores submetem um item ou conjunto de itens. Estas submissões são avaliadas e, caso aprovadas, ficam em estado de espera até se colocarem disponíveis para leilão
Leilão	Termo geral para o processo de compra e venda de um ou mais itens	Cada utilizador pode participar num leilão quer como comprador quer como vendedor
Lance	Termo geral para a quantia de dinheiro oferecida por um utilizador a participar num leilão	Um utilizador ao entrar num leilão pode introduzir uma quantia de dinheiro que deseja oferecer pelo lote
Artista	Termo geral para o artista musical	Cada item estará associado a um artista musical
Categoria	Termo geral para a categoria referente a um item ou lote	Cada categoria será exibida na aplicação, sendo possível um utilizador pesquisar por uma determinada categoria durante a procura de itens

Tabela 1: Identificação das entidades

#### 4.2.2. Atributos



Entidade	Atributos	Descrição	Opcional
Utilizador	<ul style="list-style-type: none"> <li>ID</li> <li>Nome</li> <li>Email</li> <li>Leilões Favoritos</li> <li>Categorias Favoritas</li> <li>Lotes Favoritos</li> </ul>	<ul style="list-style-type: none"> <li>Identificador do Utilizador</li> <li>Nome do Utilizador</li> <li>Email do Utilizador</li> <li>Leilões Favoritos do Utilizador</li> <li>Categorias Favoritas do Utilizador</li> <li>Lotes Favoritos do Utilizador</li> </ul>	Não Não Não Sim Sim Sim
Lote	<ul style="list-style-type: none"> <li>ID</li> <li>Nome</li> <li>Descrição</li> <li>Visibilidade</li> <li>Estado</li> <li>Valor Inicial</li> <li>Valor Final</li> <li>Fila de Espera</li> <li>Data de Adição</li> </ul>	<ul style="list-style-type: none"> <li>Identificador do Lote</li> <li>Nome do Lote</li> <li>Descrição do Lote</li> <li>Visibilidade do Lote</li> <li>Estado do Lote</li> <li>Valor Inicial do Lote</li> <li>Valor Final do Lote</li> <li>Fila de Espera para o Lote</li> <li>Data de Adição do Lote</li> </ul>	Não Não Não Não Não Não Não Não
Leilão	<ul style="list-style-type: none"> <li>ID</li> <li>Data de Início</li> <li>Nome</li> <li>Descrição</li> <li>Inicialização</li> </ul>	<ul style="list-style-type: none"> <li>Identificador do Leilão</li> <li>Data de Início do Leilão</li> <li>Nome do Leilão</li> <li>Descrição do Leilão</li> <li>Modo de Inicialização do Leilão</li> </ul>	Não Não Não Não Não
Lance	<ul style="list-style-type: none"> <li>ID do Utilizador</li> <li>ID do Lote</li> <li>Preço do Lance Final</li> <li>Registo Data / Hora</li> </ul>	<ul style="list-style-type: none"> <li>Identificação do Utilizador</li> <li>Identificação do Lote</li> <li>Preço do Lance Final</li> <li>Registo Data / Hora</li> </ul>	Não Não Não Não

Tabela 2: Identificação dos atributos das entidades

#### 4.2.3. Relacionamentos

Entidade	Multiplicidade	Relacionamento	Multiplicidade	Entidade
Leilão	(1,1)	tem	(1,n)	Lote
Utilizador	(1,n)	participa	(1,n)	Leilão
Categoria	(1,1)	identifica	(1,n)	Lote
Artista	(1,1)	identifica	(1,n)	Lote
Utilizador	(1,n)	dá	(1,n)	Lance
Lote	(1,n)	possuem	(1,n)	Lance

Tabela 3: Relacionamento entre entidades

## 5. Esboço das Interfaces do Sistema

### 5.1. Estrutura geral das interfaces do sistema

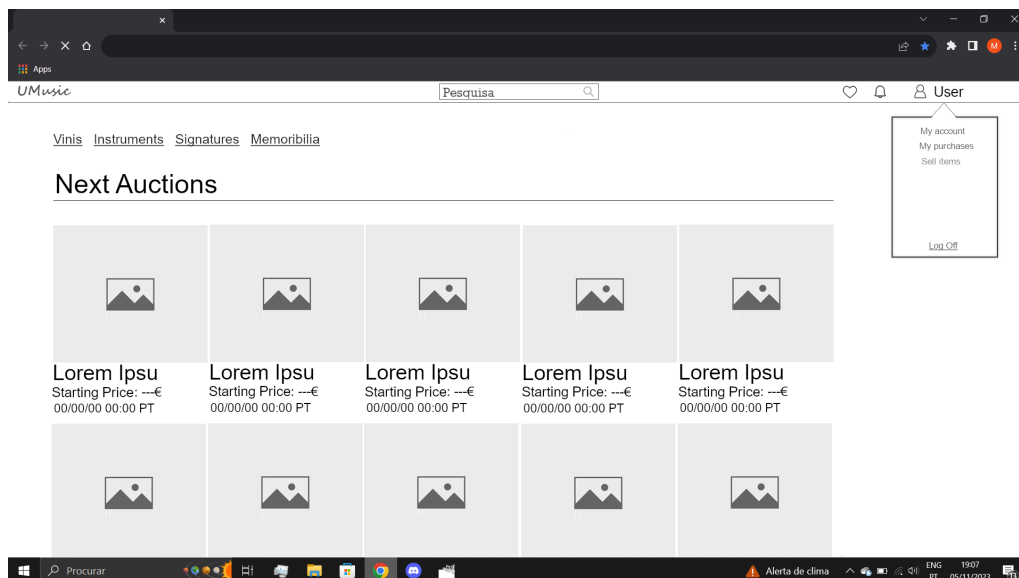


Figura 9: Interface Home page

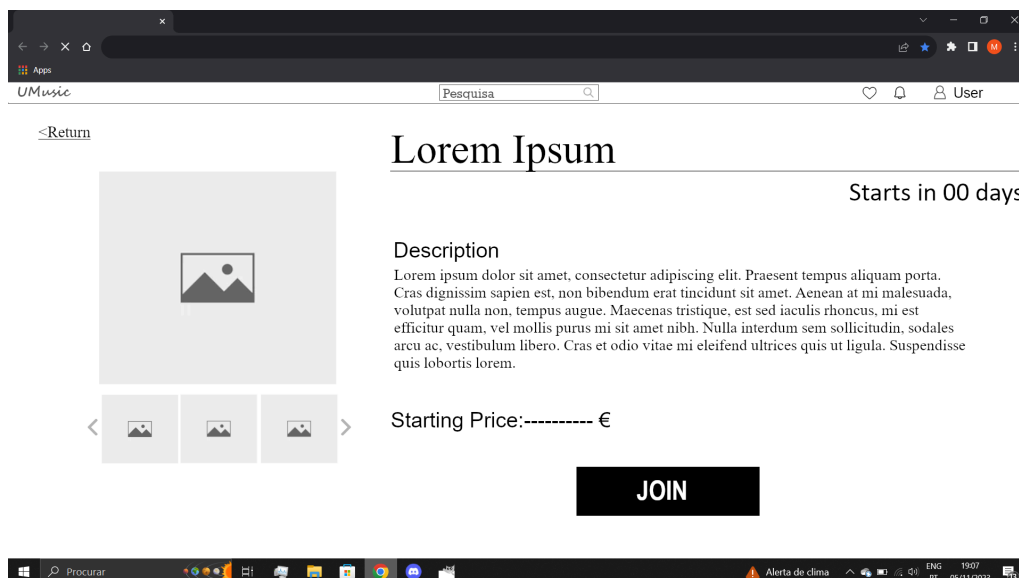


Figura 10: Interface de consulta de lote

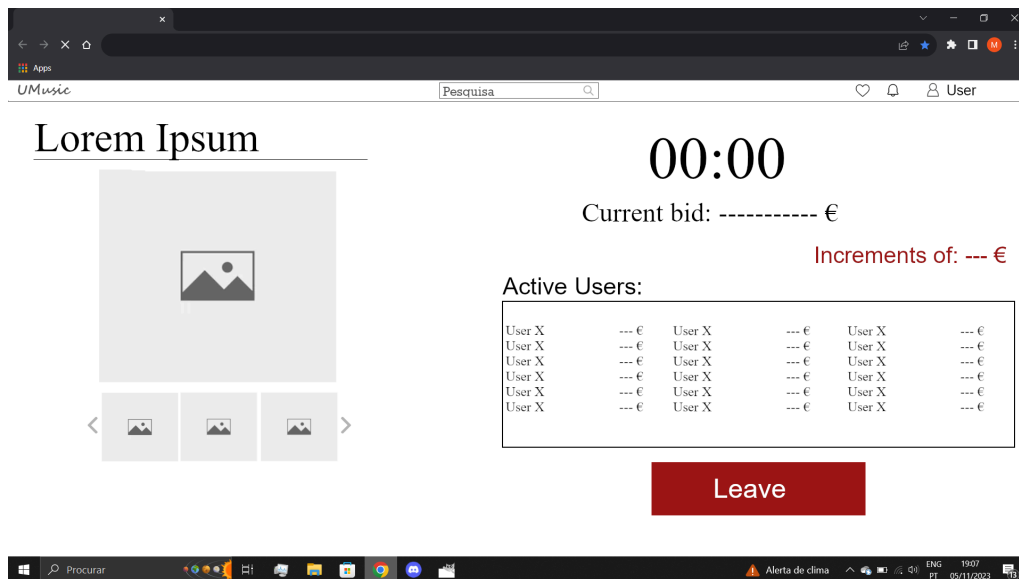


Figura 11: Interface para o leilão

Figura 12: Interface de registo

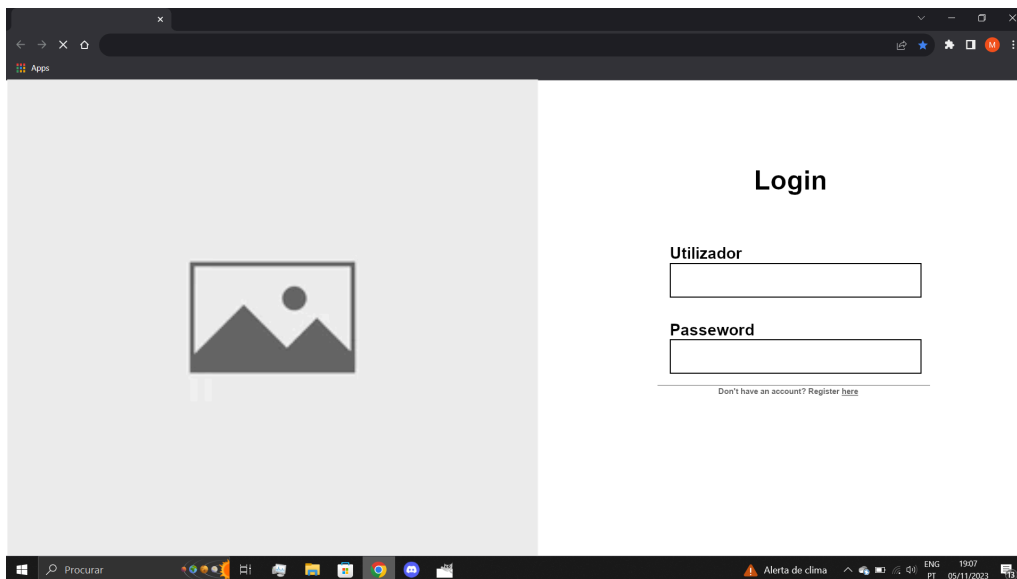


Figura 13: Interface Log in

## 5.2. Caracterização das interfaces

As Interfaces a serem utilizadas para este projeto pretendem cumprir com todos os casos de uso analisados, mantendo ao mesmo tempo uma plataforma simples, intuitiva e de fácil utilização.

Tendo em conta os casos de uso descritos anteriormente, foram criadas as seguintes demonstradas no ponto anterior.

A interface Home page (Figura 9), é a primeira interface que o utilizador irá ver ao aceder ao nosso site. Esta interface tem de ser capaz de mostrar os leilões que vão começar, os leilões que já acabaram e os leilões que o utilizador possa ter interesse baseado na sua lista de favoritos. Como esta interface agrupa todas as consultas, consideramos que satisfaz o caso de uso 3.3.2.2 (Consultas)

Da mesma maneira, a interface de consulta de lote (Figura 10), caso de uso 3.3.2.2.2 (Consulta de um lote), tem de ser capaz de mostrar a informação do lote que o utilizador escolheu, desde a hora de início do leilão, até ao artista que o lote representa.

A interface para o leilão (Figura 11), caso de uso 3.3.2.2.3 (Consulta de um leilão), tem de ser capaz de mostrar a informação do leilão, licitação atual, um temporizador de ronda, o incremento por ronda, e a lista de utilizadores que estão a participar no leilão.

Para que o utilizador possa ter acesso aos leilões, é necessário que este esteja registado no site, sendo assim preciso a página de registo (Figura 12), caso de uso 3.3.2.1.1 (Criar conta), e a página de login (Figura 13), caso de uso 3.3.2.1.2 (Login). Estas páginas têm de ser capazes de receber os dados do utilizador e de os validar, para que este possa ter acesso ao site.

## 6. Implementação da Aplicação

Para a implementação da aplicação, optamos por utilizar o ASP.NET Core MVC como o principal framework. O ASP.NET Core MVC destaca-se pela sua arquitetura modular e flexível, permitindo-nos criar uma aplicação robusta e escalável. Através do padrão MVC, organizamos a lógica de negócios de forma coerente, melhorando a legibilidade e a extensibilidade do software.

Para além disso, é importante destacar a compatibilidade multiplataforma, pois ao garantir que a aplicação pode ser executada em diversos sistemas operativos, asseguramos uma maior flexibilidade no deployment e uma maior amplitude de utilização, o que é essencial para atender às necessidades de um público diversificado.

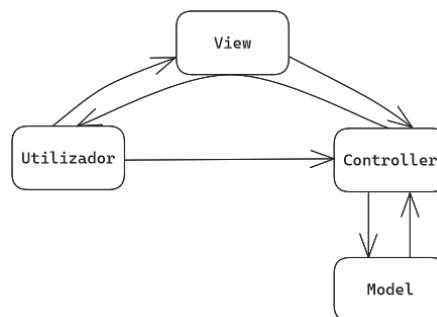


Figura 14: Padrão MVC

### 6.1. Estrutura do código

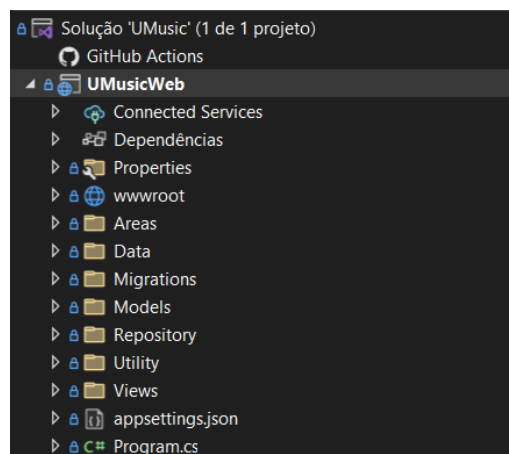


Figura 15: Estrutura do código

#### 6.1.1. wwwroot

A pasta `wwwroot` é o local onde são armazenados os recursos estáticos da aplicação (ficheiros CSS, JS, HTML, imagens, etc.), os quais são acessíveis publicamente. Assim, a aparência geral do website é definida a partir de ficheiros contidos nesta pasta, que inclui também imagens, por exemplo, dos lotes.

### 6.1.2. Areas

Foram definidas três áreas: Admin, Customer, e Identity. Cada área tem o seu conjunto específico de *controllers* e as respetivas *views*. Esta distinção é feita de modo a que as áreas sejam acessíveis por meio de permissões/ *roles*.

A área Admin permite gerir as entidades da base de dados, i.e., permite que um utilizador com *role* Admin possa efetuar operações CRUD sobre qualquer tipo de dados da aplicação. A área Customer permite aceder à *home page*, visualizar leilões e lotes, entrar num leilão de um lote, etc. A área Identity corresponde aos meios de autenticação, de autorização, e gestão de dados do utilizador (login, registo, logout, etc.)

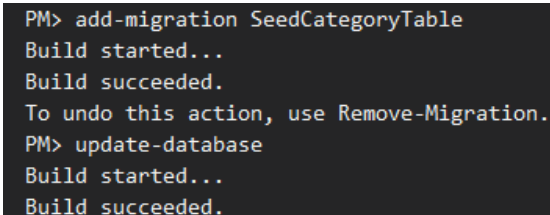
### 6.1.3. Data

A pasta Data possui apenas um ficheiro: *ApplicationDbContext* que é uma subclasse de *IdentityDbContext<IdentityUser>*, possui *DbSets* (conjuntos de entidades que representam tabelas na base de dados) e faz uma pequena população inicial das tabelas (*OnModelCreating*).

```
public class ApplicationDbContext : IdentityDbContext<IdentityUser>
{
    public ApplicationDbContext(DbContextOptions<ApplicationDbContext> op) : base(op) {}
    public DbSet<Lot> Lots { get; set; }
    public DbSet<AppUser> AppUsers { get; set; }
    // ...
    protected override void OnModelCreating(ModelBuilder modelBuilder)
    { /* ... */ }
}
```

### 6.1.4. Migrations

Dentro desta, encontram-se ficheiros que representam as diversas migrações ao longo do desenvolvimento da aplicação – representam uma forma de versionar o esquema da base de dados ao longo do tempo. Quando são feitas modificações na classe *ApplicationDbContext*, ou em classes na pasta *Models*, é necessário adicionar uma nova migração, por exemplo:



```
PM> add-migration SeedCategoryTable
Build started...
Build succeeded.
To undo this action, use Remove-Migration.
PM> update-database
Build started...
Build succeeded.
```

Figura 16: Migração e update

### 6.1.5. Models

Esta pasta é utilizada para armazenar as classes que representam os modelos de dados da aplicação. Estes modelos são mapeados para entidades na base de dados e são utilizados para transportar dados entre a aplicação e a base de dados.

### 6.1.6. Repository

Nesta pasta são definidas interfaces e classes para os repositórios dos tipos de dados definidos nas classes da pasta *Models*. Assim, cada repositório tem de ter certas operações (*Add*, *Get*, *GetAll*, *Remove*, *RemoveRange*), sendo que cada interface de um repositório específico (por exemplo, *IAuctionRepository*) pode ter os seus próprios métodos de *Update*, ou outros, conforme necessário.

Classe *Repository*:

```
public class Repository<T> : IRepository<T> where T : class
{
```

```

private readonly ApplicationDbContext _db;
internal DbSet<T> dbSet;
public Repository(ApplicationDbContext db)
{
    _db = db;
    this.dbSet = _db.Set<T>();
    _db.Lots.Include(u => u.Auction);
}

public void Add(T entity) { dbSet.Add(entity); }
public T Get(Expression<Func<T, bool>> filter, string? includeProperties = null)
{ /* ... */ }
// ...
}

```

Exemplo de uma interface:

```

public interface IAppUserRepository : IRepository<AppUser>
{
    void Update(AppUser appUser);
    public void UpdateJoinedAuction(AppUser user, int? auctionId);
    // ...
}

```

Para além disso, também contém uma classe bastante importante: `UnitOfWork` que tem instâncias de todos os repositórios definidos, e uma função `Save`, pois é comum a todos os repositórios.

```

public interface IUnitOfWork
{
    ILotRepository Lot { get; }
    IAppUserRepository AppUser { get; }
    // ...
    void Save();
}

```

Assim, caso se queira executar alguma operação sobre um tipo de dados, basta utilizar esta classe: `_unitOfWork.«model».«operação»`, por exemplo, `Lot obj = _unitOfWork.Lot.Get(u => u.Id == id);`.

### 6.1.7. Utility

Para as classes que fornecem funcionalidades genéricas e reutilizáveis em toda a aplicação. Contém uma classe chamada `SD`, para *static details* (ex.: nomes dos *roles*).

### 6.1.8. Views

Pasta para as *views* partilhadas, como, por exemplo, o *layout* do website, que está sempre presente.

### 6.1.9. appsettings.json

É um ficheiro de configuração, que foi utilizado para definir a conexão com a base de dados (SQL server).

### 6.1.10. program.cs

Componente central do projeto, que serve como ponto de inicialização da aplicação. Portanto, é responsável pela inicialização de serviços: conexão com a base de dados, *Identity*, *cookies*, *scoped services* como a `UnitOfWork`, *controller routes*, autenticação, autorização, etc.

### 6.1.11. Exemplo e explicação de um *controller method*

```
[Authorize(Roles = SD.Role_Customer + "," + SD.Role_Admin)]
public IActionResult Waiting(int id)
{
    Lot lot = _unitOfWork.Lot.Get(u => u.Id == id);

    DateTime now = DateTime.Now;
    DateTime endDate = lot.DateStart.AddMinutes(lot.QueueTime);

    if (now >= lot.DateStart && now <= endDate)
    {
        string? userId = User.FindFirstValue(ClaimTypes.NameIdentifier);
        if (!string.IsNullOrEmpty(userId))
        {
            AppUser appUser = _unitOfWork.AppUser.Get(u => u.Id == userId);
            if (appUser != null && appUser.JoinedAuctionId == null)
            {
                _unitOfWork.AppUser.UpdateJoinedAuction(appUser, lot.AucionId);
                _unitOfWork.Save();
                return View(lot);
            }
        }
    }
    TempData["error"] = "Failed to join the queue";
    return RedirectToAction("Index");
}
```

Achamos útil explicar o funcionamento deste método do HomeController pois utiliza diversas partes e recursos do programa. Assim, serve também como uma boa explicação do programa em geral.

Primeiramente, é utilizada uma data annotation de modo a restringir a utilização do método, i.e., apenas utilizadores com login efetuado o podem usar. A partir do argumento id e da IUnitOfWork obtém-se o objeto lot. Depois, verifica-se se é possível entrar na *queue* com base em algumas restrições. Caso todas as restrições sejam cumpridas, o appUser é atualizado na base de dados – especificamente, o atributo JoinedAuctionId. Caso contrário, será gerada uma mensagem de erro (com ajuda da ferramenta toastr, mencionada posteriormente).

## 6.2. Ferramentas utilizadas

- <https://bootswatch.com/> - CSS para o aspeto geral da WebApp.

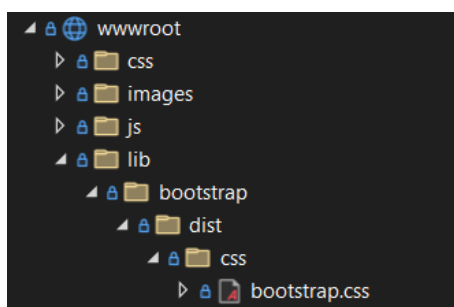


Figura 17: Bootstrap CSS

- <https://icons.getbootstrap.com/> - Icons utilizados.
- <https://codeseven.github.io/toastr/> - Para mensagens de sucesso, erro, e aviso.



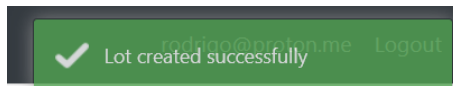


Figura 18: Mensagem de sucesso

- <https://datatables.net/> - Utilizado para gerar tabelas com paginação. (Ver Secção 6.3.4).

```
function loadDataTable() {  
    dataTable = $('#tblData').DataTable({  
        "ajax": { url: '/admin/lot/getall' },  
        "columns": [  
            { data: 'name' },  
            // ...  
        ]  
    });  
}
```

- <https://www.tiny.cloud/> - Utilizado para gerar área de edição de texto (substituindo o elemento textarea de HTML). (Ver Secção 6.3.4).
- Microsoft.EntityFrameworkCore
- Microsoft.EntityFrameworkCore.SqlServer
- Microsoft.EntityFrameworkCore.Tools
- Microsoft.AspNetCore.Identity.EntityFrameworkCore
- Microsoft.AspNetCore.Identity.UI

## 6.3. Serviços implementados

### 6.3.1. Página de registo

## Register

### Create a new account

Email

Password

Confirm Password

Name

Phone Number

Address

City

Postal Code

-Select Role-

Register

Figura 19: Página de registo

6.3.2. Página de login

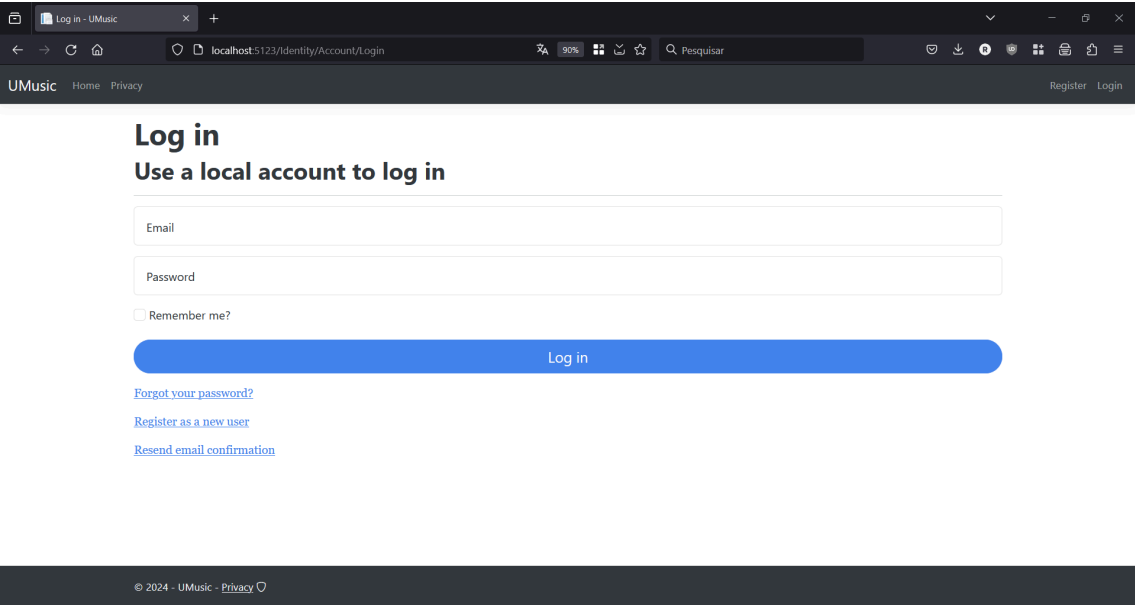


Figura 20: Página de login

6.3.3. Página de configurações da conta

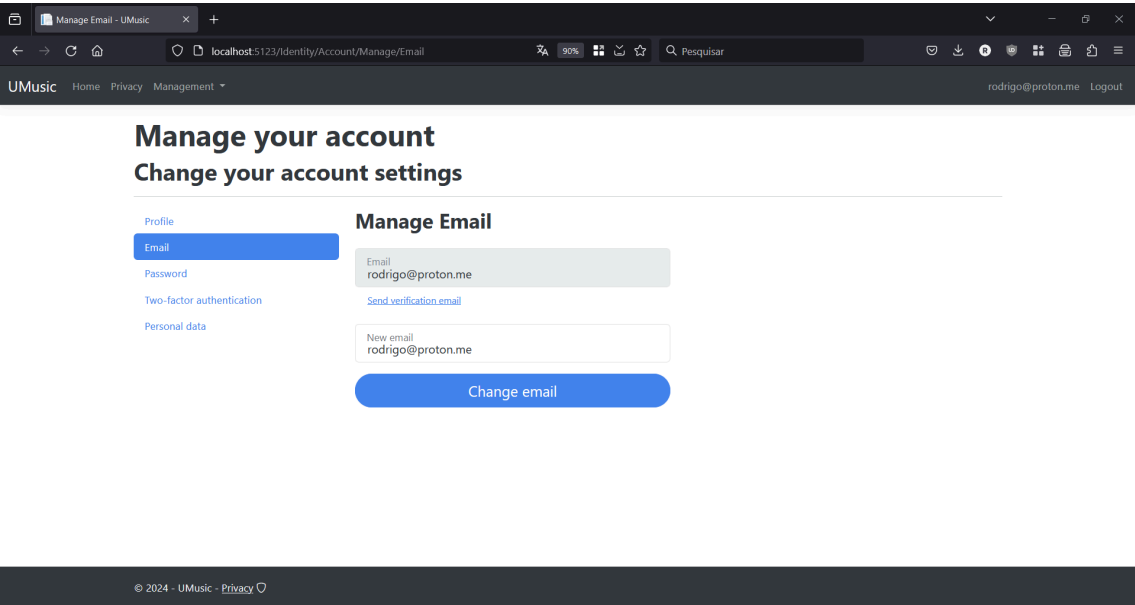


Figura 21: Página de configurações da conta

6.3.4. Management: Lot

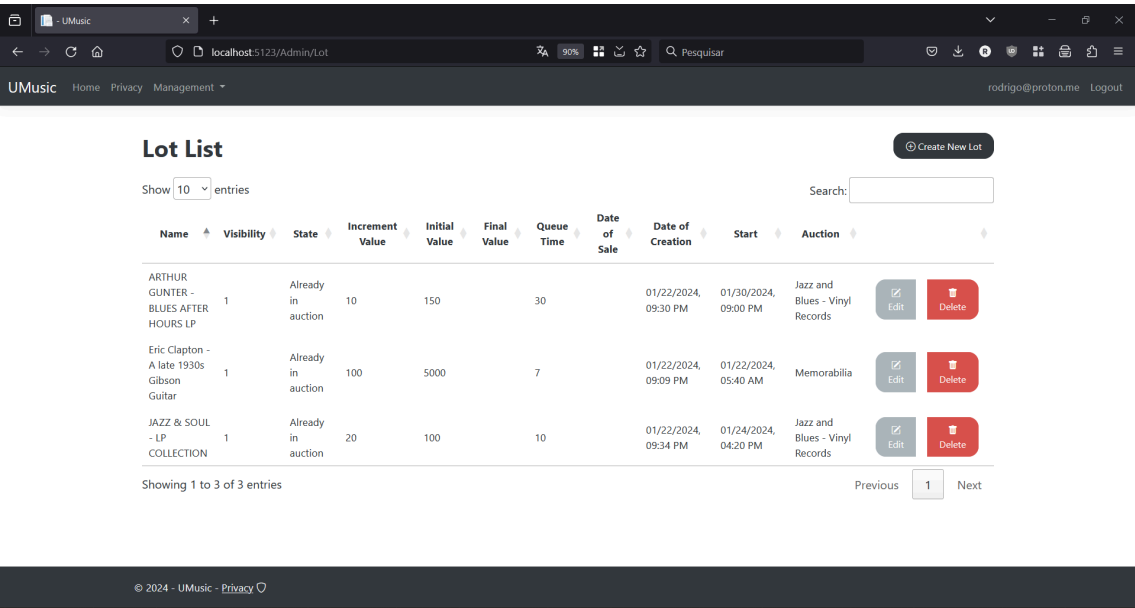


Figura 22: Lista de lotes

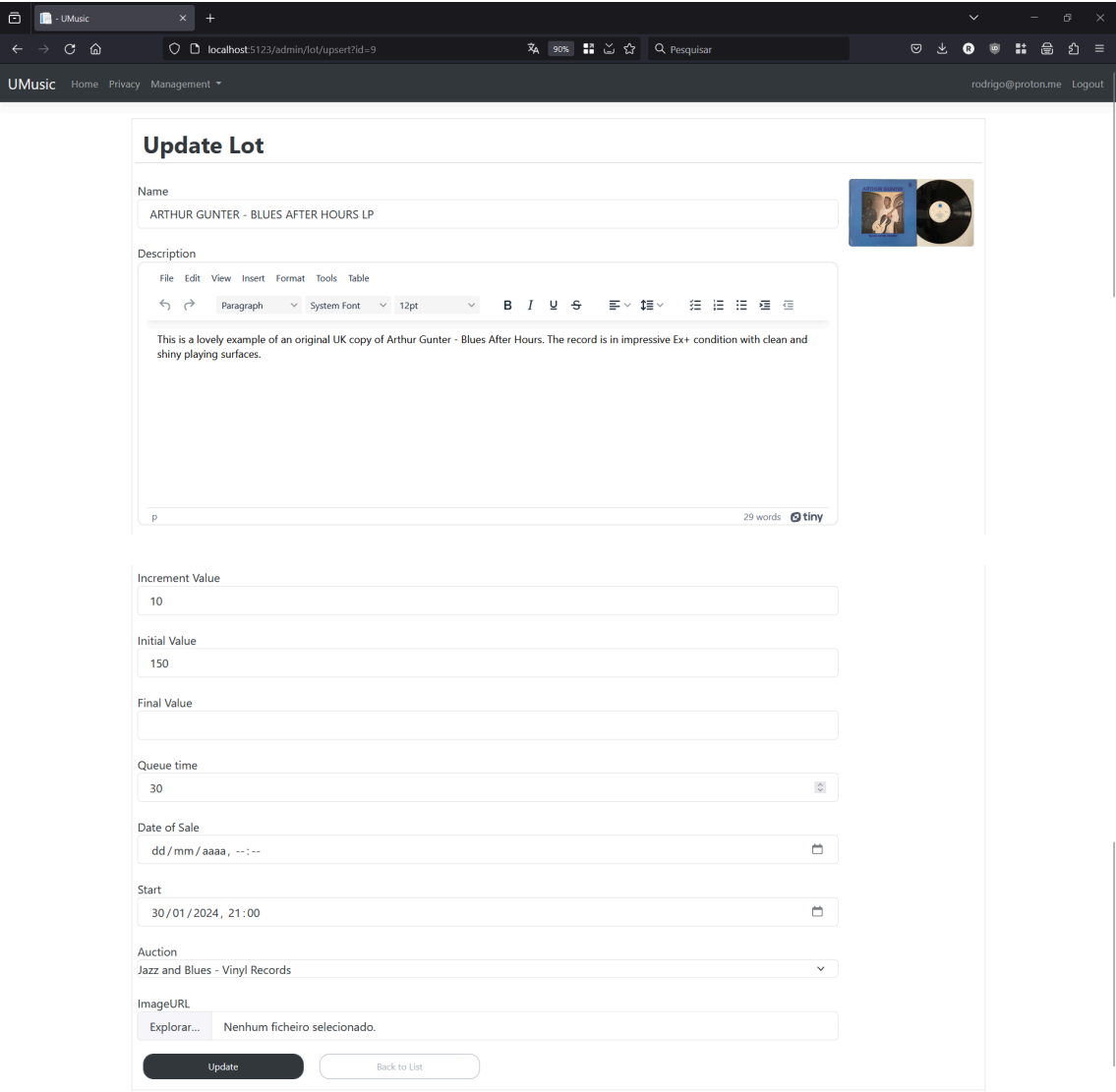


Figura 24: Update de lote

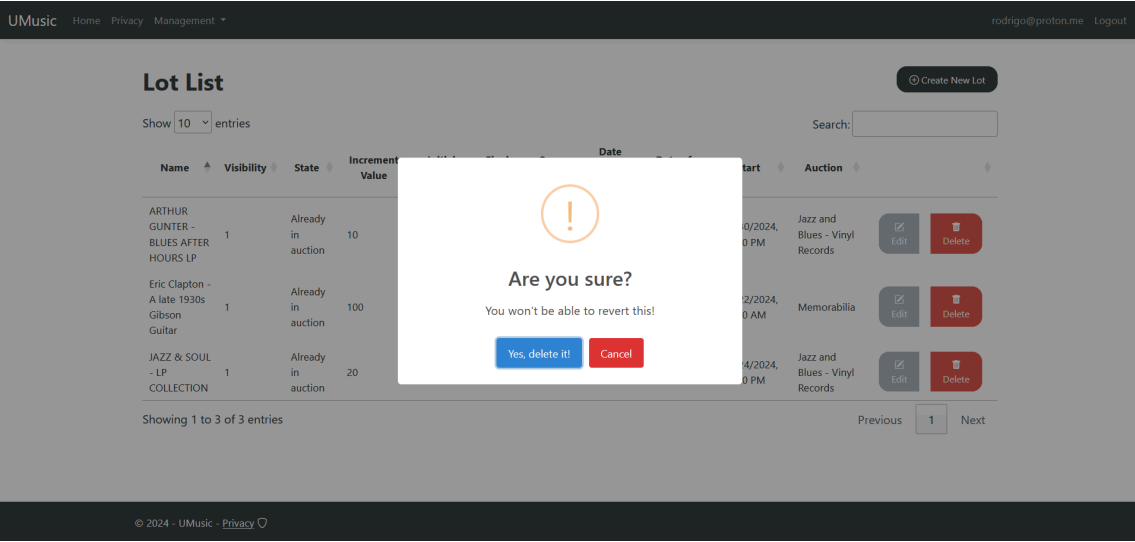


Figura 25: Remover um lote

6.3.5. Management: Auction

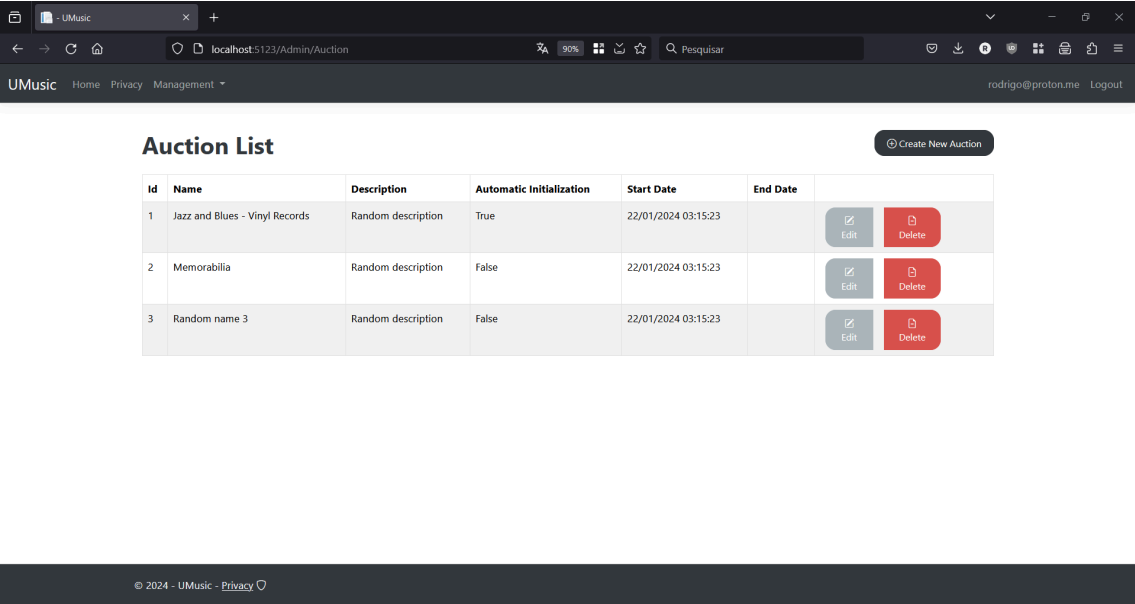


Figura 26: Lista de auctions

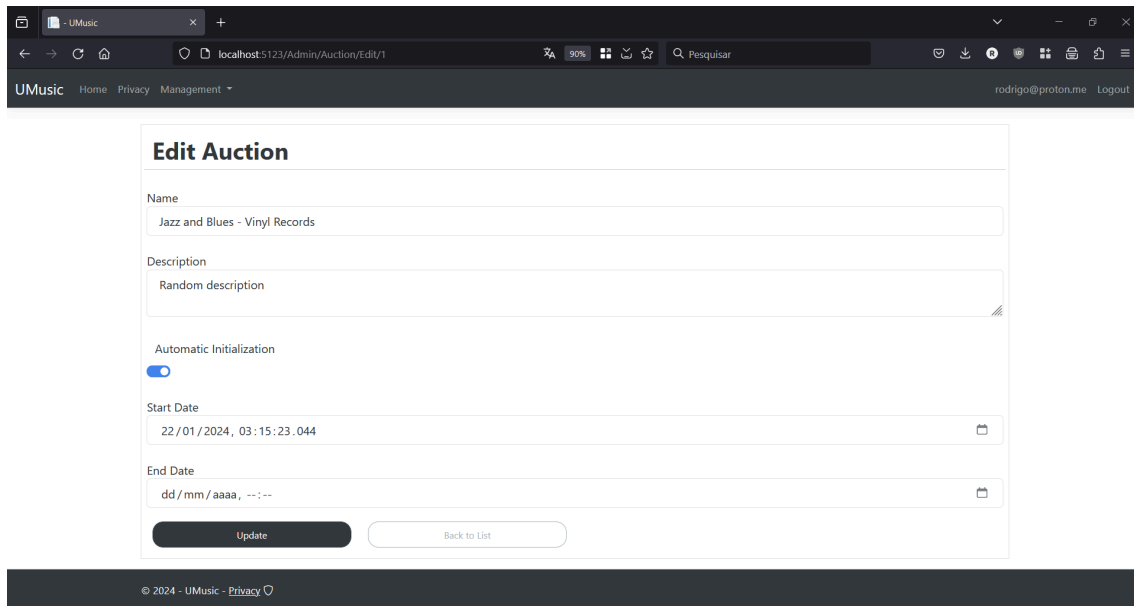


Figura 27: Update auction

### 6.3.6. Home Page

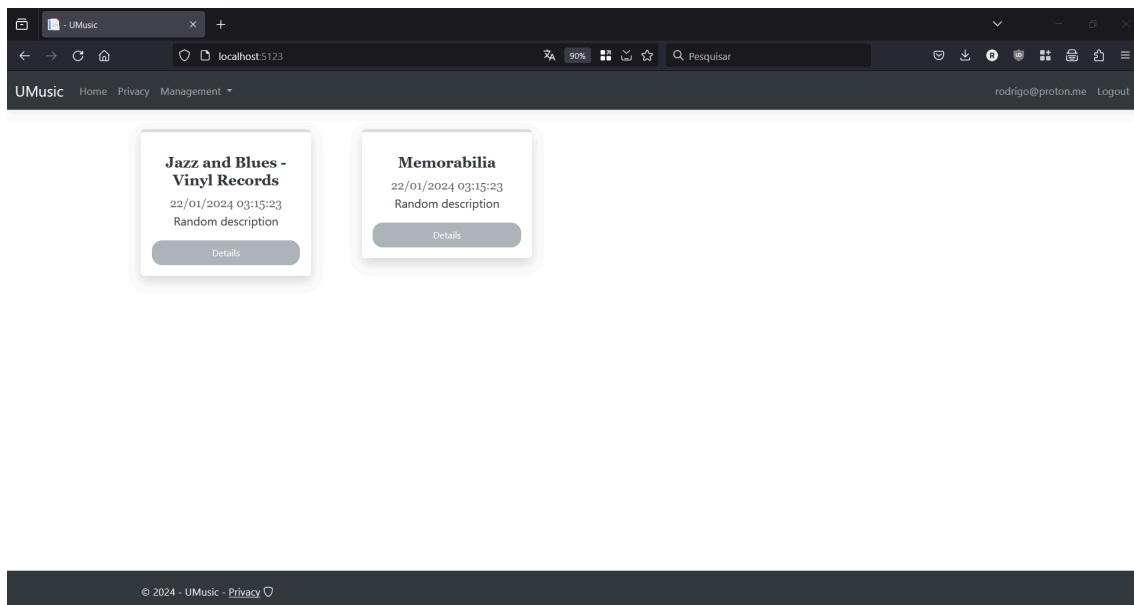


Figura 28: Home page

### 6.3.7. Privacy

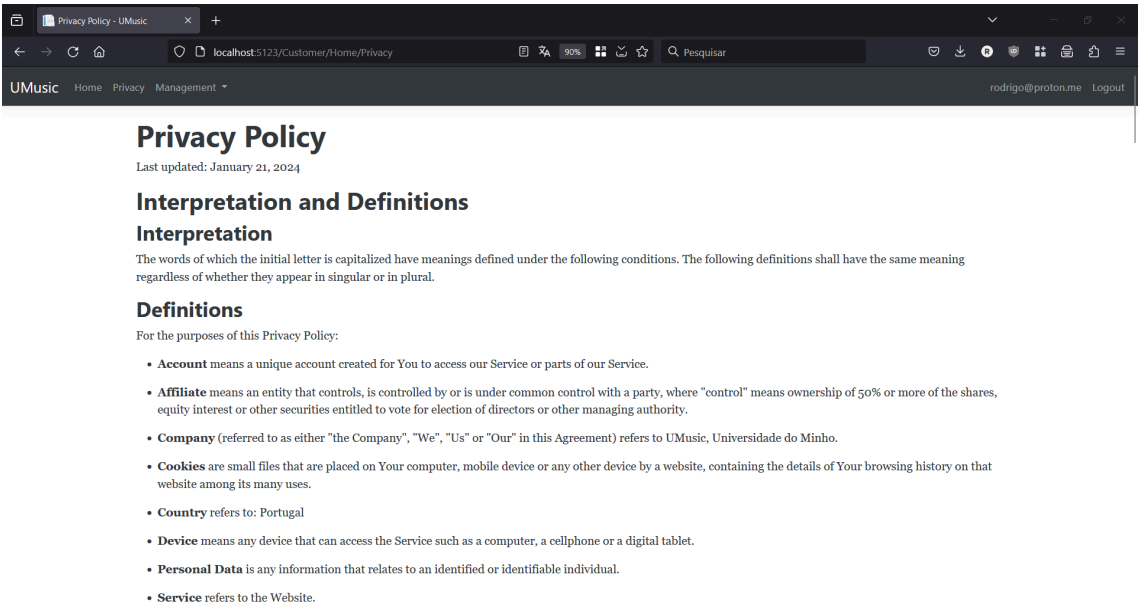


Figura 29: Privacy policy page

6.3.8. Auction Details

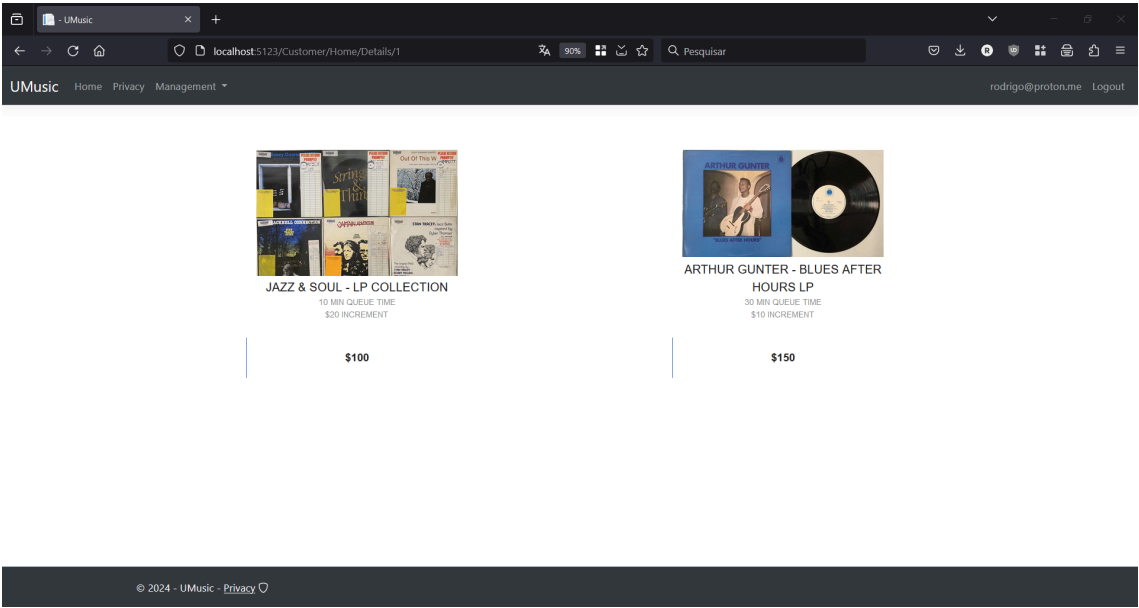


Figura 30: Auction details (1)

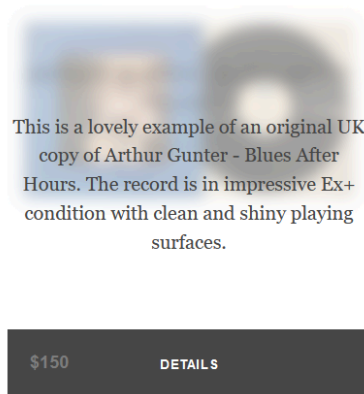


Figura 31: Auction details (2) - Lot description

### 6.3.9. Waiting

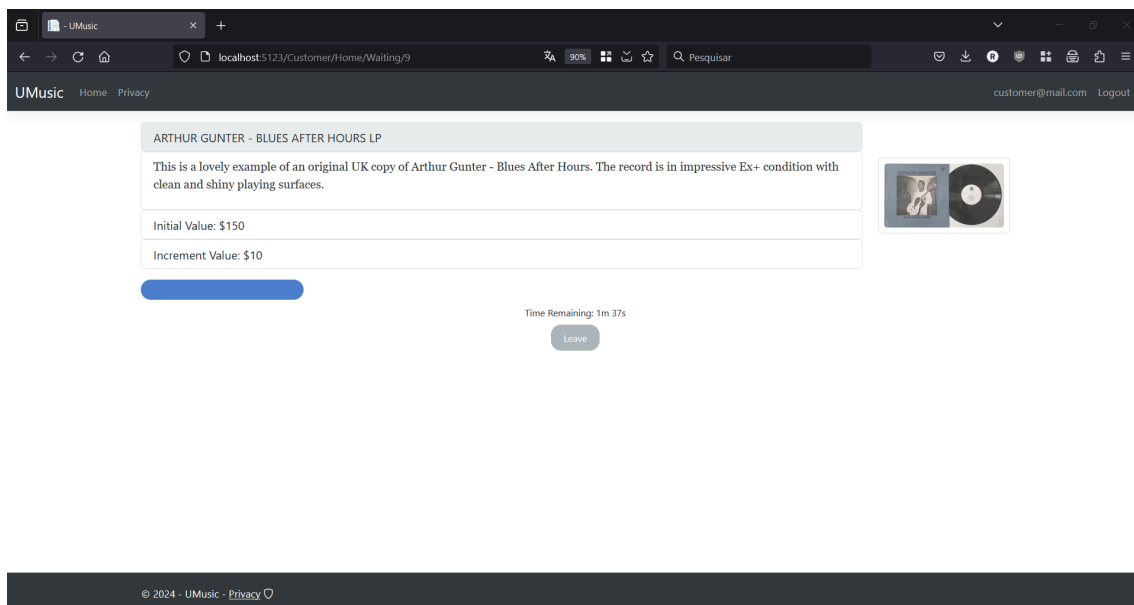


Figura 32: Waiting page

### 6.3.10. Bidding

Não foi implementada.

## 6.4. Análise e avaliação da aplicação desenvolvida

## 7. Conclusões

Numa fase inicial começamos por nos contextualizar no tema do nosso projeto de forma a conseguirmos entender, primeiramente, se o tema do nosso projeto de facto era algo que devesse ser explorado, o que levou a uma contextualização, fundamentação e análise de viabilidade.

Em paralelo com a definição dos nossos objetivos, fomos também construindo um plano de desenvolvimento, de forma a conseguirmos organizar-nos. Esta fase provou-se crucial, uma vez que permitiu que nós ficassemos a saber mais acerca do projeto e trabalhá-lo de uma forma mais lógica e consistente.

Já na parte da implementação, procurámos seguir o plano anteriormente definido de modo a garantir um desenvolvimento progressivo e para serem atingidos os nossos objetivos. Apesar disto, reconhecemos que algumas partes do projeto não foram totalmente implementadas. Os obstáculos que nos apareceram durante a implementação foram, em grande parte, devido à nossa inexperiência com as ferramentas selecionadas e não por causa de deficiências nas fases anteriores do projeto, que foram cruciais para a concepção sólida do trabalho.



## 8. Trabalho Futuro

No futuro temos as seguintes intenções:

- Implementar o que não foi feito até agora, nomeadamente a funcionalidade de licitação;
- Melhorar várias páginas e funcionalidades porque, apesar de considerarmos que fizemos um bom trabalho na implementação da interface de utilizador no geral, entendemos que esta é uma parte fundamental do projeto;
- Adicionar suporte de várias linguagens de modo a tornar a aplicação mais inclusiva e acessível a uma audiência global, proporcionando uma experiência de usuário mais personalizada e adaptada às preferências linguísticas individuais.

## Referências

- [1] AM, «Comparison of online auction websites». [Em linha]. Disponível em: [https://auctionmemorabilia.com/media/images/Auction-Houses-Comparison-Table\\_2020.jpg](https://auctionmemorabilia.com/media/images/Auction-Houses-Comparison-Table_2020.jpg)
- [2] P. Klemperer, «What Really Matters in Auction Design», *Journal of Economic Perspectives*, vol. 16, n.º 1, 2002.
- [3] P. Klemperer, *Auctions: Theory and Practice*. Princeton University Press, 2004.
- [4] Wikipedia, «Revenue equivalence». [Em linha]. Disponível em: [https://en.wikipedia.org/wiki/Revenue\\_equivalence](https://en.wikipedia.org/wiki/Revenue_equivalence)
- [5] I. Sommerville, «Chapter 4 - requirements basics, the document and the requirements». 2015.

## **Lista de Siglas e Acrónimos**

**BD** Base de Dados

**DW** Data Warehouse

**OLTP** On-Line Analytical Processing

## **Anexos**