



Universidade do Minho
Escola de Engenharia
Licenciatura em Engenharia Informática

Unidade Curricular de Laboratórios de Informática IV

Ano Letivo de 2023/2024

Noitcua

Grupo 10:

Bernardo Lima	(A93258)	David Teixeira	(A100554)
João Pedro Pastore	(A100543)	Luís Ferreira	(A91672)

28 de janeiro de 2024

LI4

Data de Receção	
Responsável	
Avaliação	
Observações	

Noitcua

Grupo 10:

Bernardo Lima	(A93258)	David Teixeira	(A100554)
João Pedro Pastore	(A100543)	Luís Ferreira	(A91672)

28 de janeiro de 2024

Índice

1	Definição do Sistema	1
1.1	Contextualização	1
1.2	Fundamentação	1
1.3	Objetivos	1
1.4	Viabilidade	2
1.5	Recursos a utilizar	2
1.6	Equipa de Trabalho	3
1.6.1	Pessoal Interno	3
1.6.2	Pessoal Externo	3
1.7	Plano de Execução do Trabalho	3
2	Definição de Requisitos	5
2.1	Apresentação da Estratégia e Método	5
2.2	Descrição Geral dos Requisitos (Funcionais e Não Funcionais) Levantados . . .	5
2.2.1	Requisitos Funcionais	5
2.2.2	Requisitos Não Funcionais	8
2.3	Validação dos Requisitos Estabelecidos	8
3	Especificação e Modelação do Software	9
3.1	Apresentação Geral da Especificação	9
3.2	Aspetos estruturais	9
3.3	Aspetos comportamentais	10
3.3.1	Diagrama de Use Cases	10
3.3.2	Especificação de Use Cases	12
4	Conceção do Sistema de Dados	21
4.1	Apresentação Geral da Estrutura (esquema) do Sistema de Dados	21
4.1.1	Modelo Lógico	21
4.2	Descrição detalhada dos vários elementos de dados e seus relacionamentos . .	22
4.2.1	Entidades:	22
4.2.2	Relacionamentos	23
5	Esboço das Interfaces do Sistema	25
5.1	Estrutura geral das interfaces do sistema	25
5.2	Caracterização das interfaces	26
5.2.1	Página de Entrada	26
5.2.2	Páginas de Login e Registo	27

5.2.3	Página de Vendas	28
5.2.4	Página de Criação de Sala	29
5.2.5	Página de Perfil	30
5.2.6	Página de Salas do Utilizador	31
5.2.7	Página do Histórico	32
6	Implementação da Aplicação	33
6.1	Descrição da Estrutura Final da Aplicação e Exposição dos Serviços Implementados	33
6.1.1	Descrição da Estrutura Final da Aplicação	34
6.1.2	Exposição dos Serviços Implementados	35
6.2	Ferramentas utilizadas	44
6.3	Análise Crítica e Avaliação da Aplicação Desenvolvida	45
7	Conclusões e Trabalho Futuro	48
7.1	Conclusões no Checkpoint	48
7.2	Conclusões na Entrega Final	48
Anexos		50
7.1	Instruções SQL	50
7.1.1	Drop/Create All	50

Lista de Figuras

1.1	Diagrama de Gantt	4
3.1	Modelo de Domínio	10
3.2	Diagrama de Use Cases	12
4.1	Modelo Lógico	21
5.1	Estrutura geral das interfaces do sistema	25
5.2	Mockup da página inicial	26
5.3	Mockup da página de Login	27
5.4	Mockup da página de Registo	27
5.5	Mockup da página de Vendas	28
5.6	Mockup da página de Criação de Sala	29
5.7	Mockup da página de Perfil	30
5.8	Mockup da página de Salas do Utilizador	31
5.9	Mockup da página do Histórico	32
6.1	Diagrama de Gantt Atualizado	33
6.2	Estrutura Final da Aplicação	34
6.3	Página de Entrada	36
6.4	Página de Login	37
6.5	Página de Recuperação de Palavra Chave	37
6.6	Página de Registo	38
6.7	Tamanho Mínimo de Password	38
6.8	Página de Perfil	39
6.9	Página de Mudança de Handle	40
6.10	Página de Mudança de Palavra Passe	40
6.11	Página de Vendas	41
6.12	Página de Criar Sala	42
6.13	"Nova Sala" apresentada na Página de Vendas	42
6.14	Página da Salas do Utilizador	43
6.15	Página do Histórico	43
6.16	Produto Entregue	44

1 Definição do Sistema

1.1 Contextualização

Desde 1996, a Noitcua tem se destacado como especialista em leilões presenciais, solidificando a sua presença em Braga. No entanto, prevendo as oportunidades inerentes ao dinâmico mercado, o Sr. Augusto, proprietário da empresa, está comprometido em conduzir a Noitcua para uma expansão estratégica no mundo online.

O reconhecimento do mercado como uma arena propícia para lucros extraordinários motiva a decisão do Sr. Augusto. Como parte integrante dessa visão, propõe-se a criação de um site abrangente, adaptado para uso em computadores e dispositivos móveis.

1.2 Fundamentação

O cerne desse empreendimento é direcionar-se prioritariamente às necessidades dos clientes e licitantes, agilizando processos e proporcionando eficiência aos potenciais vendedores através de uma modalidade distinta de leilão, conhecida como leilão invertido.

Deste formato inovador, destaca-se a liberdade concedida aos licitantes, uma vez que os leilões se desencadeariam exclusivamente mediante solicitação específica dos compradores. Este processo não apenas modernizaria a Noitcua, como também a orientaria de uma maneira mais acentuada às necessidades e conveniências dos clientes, proporcionando uma base sólida para a exploração de novas oportunidades de negócios no cenário em expansão.

Ao adotar esta abordagem estratégica e centrada no cliente, a Noitcua não só se posicionará como uma empresa líder no segmento de leilões online, como também estará preparada para capitalizar as vantagens competitivas e as oportunidades emergentes neste ambiente dinâmico.

1.3 Objetivos

O Sr. Augusto elaborou uma lista de metas para a implementação do sistema de leilões online:

- Introduzir o conceito de leilão invertido.

- Ampliar demograficamente a base de clientes, aproveitando os benefícios tecnológicos para alcançar um público mais amplo.
- Facilitar integralmente o processo de leilão para todas as partes envolvidas, incluindo comprador (cliente) e vendedor/licitante.

1.4 Viabilidade

- **Viabilidade Técnica:** A Noitcua possui infraestruturas adequadas para implementar o sistema de leilões online. Uma análise dos membros da equipa foi realizada para garantir a competência necessária, resultando no recrutamento de profissionais aptos para iniciar a informatização do serviço.
- **Viabilidade Financeira:** A equipa da Noitcua estima um custo de 100.000€ para a implementação do sistema online de leilões. A sólida reputação da empresa no cenário leiloeiro sustenta a viabilidade financeira, projetando-se a cobertura do investimento em um ano e meio com os lucros da loja. Mesmo diante de desafios, a capacidade de compensar prejuízos é contemplada, incluindo alternativas como a venda de dados dos clientes.
- **Viabilidade Operacional:** As operações e funcionalidades da plataforma de leilões foram avaliadas como suficientes para garantir uma transição segura e sustentável para este novo tipo de mercado.

1.5 Recursos a utilizar

Humanos:

- Compradores, vendedores e funcionários da empresa de desenvolvimento de *software*.

Materiais:

- Software:
 - Microsoft Visual Studio
 - Microsoft .NET C#
 - Microsoft SQL Server
 - Microsoft Office
- Hardware

- 1 servidor
- 4 portáteis

1.6 Equipa de Trabalho

A implementação bem-sucedida da plataforma de leilões online da Noitcua requer uma equipa qualificada e comprometida. A seguir, apresentamos os membros envolvidos nesse processo.

1.6.1 Pessoal Interno

- Sr. Augusto: Encarregado da gestão empresarial, trazendo uma vasta experiência na área, juntamente com um foco dedicado ao atendimento aos clientes.

1.6.2 Pessoal Externo

- Equipa da empresa de desenvolvimento de software. Responsável por (numa primeira fase) :
 1. Definição de requisitos.
 2. Especificação e modelação do software.
 3. Conceção do sistema de dados.
 4. Esboço das interfaces do sistema.
- Implementação do sistema (numa segunda fase)

Esta colaboração entre recursos internos e uma equipa especializada externa visa assegurar uma implementação eficiente e bem-sucedida da plataforma, alinhando-se com os padrões profissionais e exigências do projeto.

1.7 Plano de Execução do Trabalho

Para guiar de uma forma eficaz a implementação da plataforma de leilões online da Noitcua, a equipa de trabalho adotará uma abordagem estruturada, incluindo a utilização de diagramas de Gantt. Este recurso oferece uma representação visual do cronograma de atividades, permitindo um acompanhamento claro e preciso do progresso.

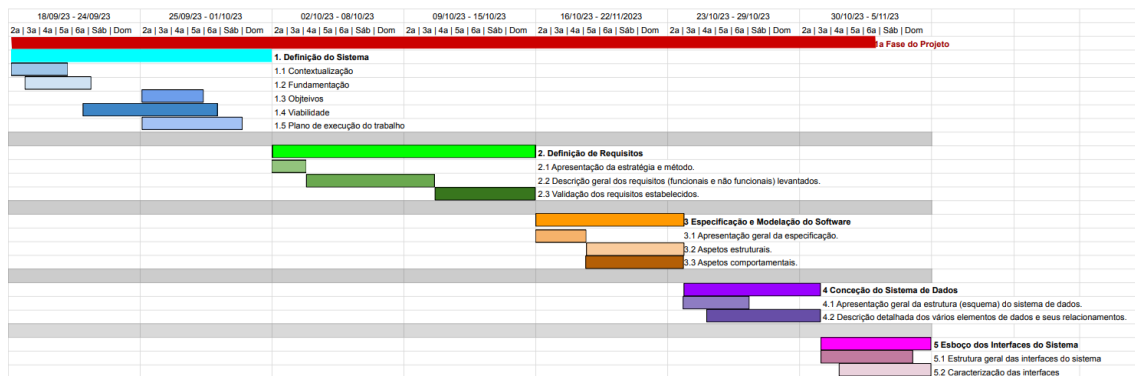


Figura 1.1: Diagrama de Gantt

A imagem do Diagrama de Gantt específico para este projeto está disponível para referência, fornecendo uma representação visual clara do planeamento e execução das atividades.

2 Definição de Requisitos

2.1 Apresentação da Estratégia e Método

A fase de levantamento de requisitos desempenha um papel crucial no desenvolvimento de *software*, estabelecendo metas essenciais para o êxito da aplicação. O método empregado envolveu uma série de reuniões entre a equipa de desenvolvimento de *software* e o Sr. Augusto, proprietário da empresa. Essas interações foram fundamentais para alinhar as expectativas, identificar requisitos específicos e garantir a integração das necessidades da empresa no processo de desenvolvimento. Os requisitos resultantes destas interações foram categorizados como não funcionais e funcionais.

2.2 Descrição Geral dos Requisitos (Funcionais e Não Funcionais) Levantados

2.2.1 Requisitos Funcionais

Histórico de Compras

- **[RF1]** O histórico de compras deverá apresentar uma janela contendo todas as compras realizadas pelo cliente. Cada compra incluirá informações como nome do vendedor, título da sala, data da compra, preço e estado da compra. No caso de uma compra pendente, será disponibilizado um botão para conclusão do seu estado.

Página Inicial

- **[RF2]** A página deve exibir o nome do *site*, proporcionando uma identificação fácil e rápida para os utilizadores.
- **[RF3]** Deverá ser disponibilizada uma área de *login* na página, permitindo que os utilizadores tenham acesso às suas contas de uma forma segura e eficiente.
- **[RF4]** Após o *login* bem-sucedido, o utilizador terá acesso a um conjunto de funciona-

lidades específicas:

1. Área do Perfil:

O utilizador poderá visualizar e editar as informações do seu perfil, incluindo *handle* a usar no *website* e configurações da conta. Para além disso, também terá acesso a duas novas páginas embutidas no perfil, o "Histórico" e "As Minhas Salas".

2. Vendas:

Uma secção que apresenta uma lista de salas disponíveis para um utilizador participar e encontrar compradores para os seus produtos.

- [RF5] Após o *login* bem-sucedido, o utilizador tem a opção de sair da sua conta ao premir a opção *logout*.

Minhas Salas

- [RF6] As salas podem-se encontrar nos estados *open*, *pending* ou *closed*, sendo que aquelas no estado *pending* ou *closed* não se encontram acessíveis a ninguém.
- [RF7] Quando um comprador chega a um acordo com um vendedor, tem a capacidade de expulsar todos os utilizadores através do comando `'/sold@handle__vendedor/preço/método'`. Neste momento o estado da sala transita de *open* para *pending*, impedindo novas mensagens e entradas.
- [RF8] O utilizador pode trocar de forma livre entre as salas em que interagiu/criou.
- [RF9] O vendedor pode escolher sair de uma sala onde anteriormente interagiu (com o auxílio do comando `/exit`).
- [RF10] O comprador pode apagar qualquer sala que criou.
- [RF11] Os utilizadores podem interagir uns com os outros escrevendo mensagens no chat das salas.

Criar Sala

- [RF12] Os compradores têm a permissão de criar até um máximo de três salas simultaneamente.
- [RF13] Ao criar uma sala, é apenas necessário fornecer um título, podendo também o comprador deixar uma descrição de modo a descrever o que este pretende encontrar, como: tipo do produto, especificações, valor máximo e a sua condição.

Perfil

- [RF14] Em cada perfil é possível visualizar as salas que foram criadas na secção "Minhas Salas".
- [RF15] No perfil é possível visualizar o histórico de compras e vendas na secção "Histórico".
- [RF16] Na área de perfil é possível modificar o *handle* através da opção "Mudar Handle".
- [RF17] É possível no perfil modificar a *password* através da opção "Mudar Password".
- [RF18] É possível reentrar nas salas que já foram interagidas através da opção "Minhas Salas".

Utilizador

- [RF19] Os utilizadores têm a capacidade de alterar as suas informações de perfil a qualquer momento, garantindo flexibilidade e atualização contínua das informações pessoais.

Autenticação e Criação de Conta

- [RF20] Quando a opção Login/Registar-se é seleccionada, o visitante (utilizador não autenticado) deve escolher se deseja fazer a autenticação (*login*) ou se registar no *site* (criar uma conta).
- [RF21] Se o visitante escolher autenticar-se, deve fornecer o seu e-mail e *password*.
- [RF22] Para criar uma conta no *site*, os utilizadores devem fornecer as seguintes informações durante o processo de registo: nome de utilizador (*handle*), data de nascimento, nacionalidade, *email* e *password*.

Página de vendas

- [RF23] Os utilizadores podem pesquisar por salas, escrevendo o nome do que procuram na barra de pesquisa, sendo que lhe serão apresentados resultados consoante aquilo que eles escreverem.
- [RF24] Para se juntar a uma sala, o utilizador necessita clicar no botão abaixo do nome da mesma, sendo redirecionado para o chat da própria.

2.2.2 Requisitos Não Funcionais

- **[RNF1]** O site deve ser desenvolvido exclusivamente utilizando as seguintes ferramentas: *MS Visual Studio* para a programação, *MS Project* para a gestão do projeto, *MS SQL Server* para a base de dados, *Microsoft .NET* para a *framework* de desenvolvimento, e *Overleaf* para a documentação e colaboração.
- **[RNF2]** O projeto passará por uma avaliação intermediária até o dia 8 de novembro de 2023, na qual as etapas a ser cumpridas pelo pessoal externo (em particular, equipa de desenvolvimento de software) devem estar concluídas.
- **[RNF3]** A entrega final do projeto ao cliente está programada para o dia 30 de janeiro de 2024. Nessa data, o sistema deve estar totalmente testado e incluir todas as funcionalidades solicitadas pelo cliente, com possíveis alterações previamente discutidas e acordadas.
- **[RNF4]** Após a conclusão da fase de Definição e Fundamentação, o projeto deverá apresentar uma identificação clara e caracterização geral da aplicação a ser desenvolvida. Isso inclui uma fundamentação robusta para o seu desenvolvimento. Além disso, devem ser estabelecidas medidas de sucesso, acompanhadas por um diagrama de desenvolvimento que ilustre o fluxo do projeto.
- **[RNF5]** Na fase de Requisitos, é esperado que o projeto inclua uma análise abrangente dos requisitos da aplicação, abordando tanto os funcionais quanto os não funcionais. A documentação inicial deve incluir uma lista de *Use Cases*, juntamente com um Modelo de Domínio, para proporcionar uma visão clara da arquitetura proposta.
- **[RNF6]** Após a conclusão da fase de Implementação e Avaliação, o projeto deve ter passado pelo processo de instalação das ferramentas de desenvolvimento. Além disso, a aplicação em si deve estar totalmente desenvolvida e integrada de forma eficaz com os sistemas adjacentes propostos. Este estágio marca a finalização do desenvolvimento e a prontidão para a avaliação completa do sistema.

2.3 Validação dos Requisitos Estabelecidos

Após a identificação, análise e organização dos requisitos, uma reunião de validação foi agendada com todos os envolvidos no processo. Durante esse encontro, todos os requisitos foram minuciosamente analisados e ajustes necessários foram efetuados. A versão final do documento de requisitos recebeu aprovação unânime dos participantes.

3 Especificação e Modelação do Software

3.1 Apresentação Geral da Especificação

Na etapa de Especificação e Modelação do *Software*, adotamos abordagens rigorosas para garantir uma compreensão abrangente e clara do sistema em desenvolvimento. Dois instrumentos fundamentais foram empregados:

- Modelos de Domínio: Utilizamos modelos de domínio para mapear as principais entidades, relações e conceitos envolvidos no contexto da plataforma de leilões *online*. Esta representação visual proporcionou uma visão global do sistema, facilitando a identificação de componentes essenciais e as suas respectivas interconexões.
- Diagramas e Especificação de *Use Cases*: A criação e especificação de *use cases* permitiu uma análise detalhada das interações entre o sistema e os seus utilizadores. Identificamos *use cases* chave, delineando as funcionalidades oferecidas aos utilizadores finais. Esta abordagem orientada para o utilizador foi crucial para definir requisitos e garantir que as necessidades dos utilizadores fossem integralmente compreendidas.

3.2 Aspetos estruturais

Foi desenvolvido um Modelo de Domínio com base nos requisitos analisados. Este modelo, utilizando a notação **UML**, apresenta de maneira simples e clara as interações no sistema a ser implementado. A sua notação simplificada é ideal para comunicação com o cliente, permitindo uma compreensão intuitiva das entidades e relacionamentos cruciais para atender às necessidades identificadas. O Modelo de Domínio proporciona uma visão rápida e abrangente de todos os componentes essenciais na construção da aplicação, garantindo a conformidade com os requisitos estabelecidos.

A aplicação gira em torno do conceito central de um **Utilizador**, que é a entidade principal. Cada utilizador tem atributos essenciais como **nome**, **data de nascimento**, **password**, **email**, **nacionalidade**, e **possui** um **histórico** que, por sua vez, **possui** uma lista de **venda**.

A entidade **venda** é crucial, e cada uma **possui** um **Estado**, **Preço**, **ID**, e **Método de**

Há também a presença de **chats**, onde **vendedores** e **compradores** interagem. Cada participante **escreve** mensagens, com o **comprador** tendo a capacidade de declarar o vendedor que ganhou o leilão.

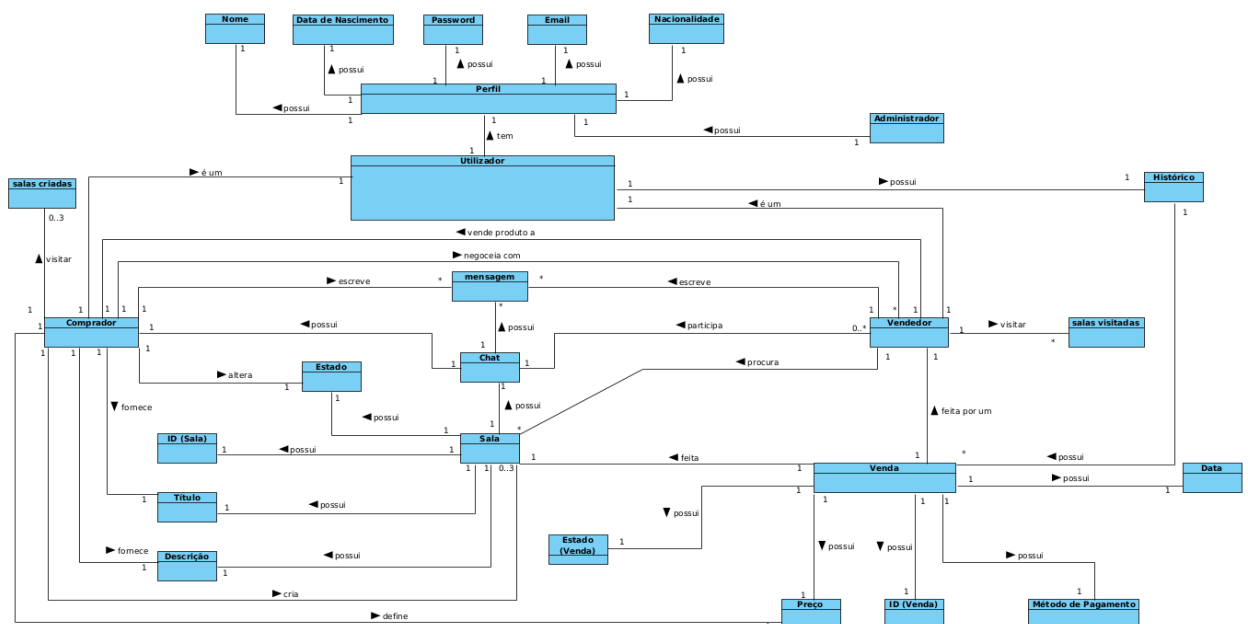


Figura 3.1: Modelo de Domínio

3.3 Aspetos comportamentais

3.3.1 Diagrama de Use Cases

Um *Use Case* descreve a interação dos atores com o sistema para concretizar cenários específicos. O Diagrama de *Use Cases* representa atores e entidades externas envolvidas, associando cada uma a casos de uso que atendem aos cenários propostos. Os atores identificados para a nossa aplicação incluem:

1. Visitante : Os visitantes são utilizadores não autenticados. Eles não possuem qualquer tipo de acesso ao *site* e apenas podem registar-se ou fazer *login*.
2. Comprador : Os Compradores são os utilizadores que não estão à procura de nenhum produto para vender, mas sim para comprar. Eles podem criar até ao máximo de 3 salas onde procuram por produtos. Nas salas eles funcionam como administradores, podendo apagar a sala ou fecha-la quando tiverem decidido com quem fazer negócio. Além disto, os Compradores podem (através do perfil) aceder diretamente às salas criadas sem as precisar de procurar pelo *site*.
3. Vendedor : Os Vendedores são utilizadores que estão à procura de compradores dos seus produtos. Eles procuram por salas de Compradores que queiram algum produto do Vendedor e negociam na sala com vendedor oferecendo o produto por um preço menor. Dentro da sala, o vendedor não tem liberdade para a manipular. Além disto, os Vendedores podem (através do perfil) aceder diretamente às salas em que interagiram sem as precisar de procurar pelo *site*.
4. Utilizador : As duas últimas entidades acima são consideradas utilizadores e, como tal, possuem benefícios partilhados, que incluem coisas como: editar o próprio perfil, escrever nas salas, fazer *logout* e aceder ao histórico de compras feito (se existir).
5. Administrador : Os Administradores são utilizadores invisíveis que representam os responsáveis pela manutenção do *site*. O objetivo deles é manter o *site* funcional e livre de problemas. Como tal, têm liberdade para fechar o *site* para manutenção, banir utilizadores que sejam denunciados, obter estatísticas do *site* e fechar salas que não estejam a cumprir os termos e condições. Para além disso, possuem um perfil especial de administrador que podem alterar caso seja necessário.

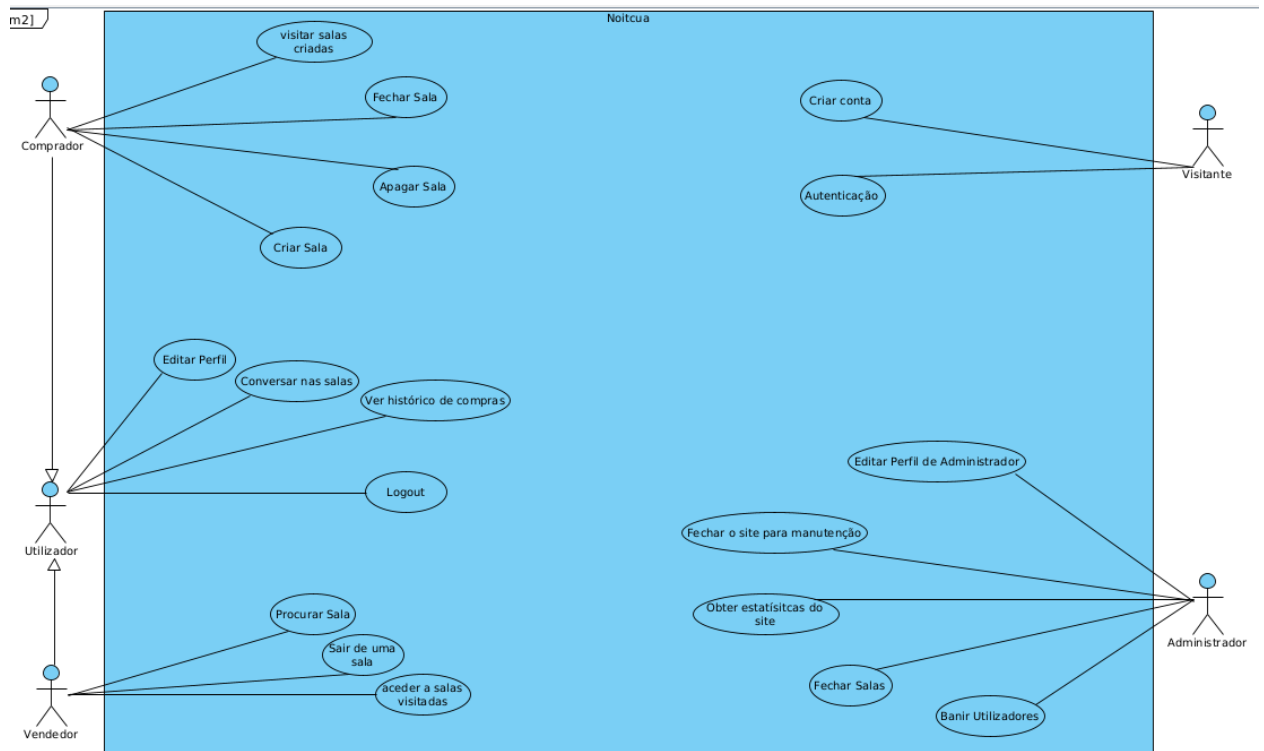


Figura 3.2: Diagrama de Use Cases

3.3.2 Especificação de Use Cases

Use Cases de Compradores

■ Use Case: Criar Sala

Ator: Comprador

Pré-Condição: N/A

Pós-Condição: O comprador possui uma sala associada

(Fluxo Normal)

1. O Comprador acede a secção "Vendas"
2. O Comprador seleciona a opção "Criar Sala"
3. O Comprador fornece título e descrição à sala
4. Sala é criada

(Fluxo Exceção - Limite de Salas)

2.1 O Sistema alerta sobre o limite de salas ativas

2.2 O Comprador não consegue criar a sala

Este *use case* satisfaz o requisito **RF12 e RF13**.

▪ Use Case: Apagar Sala

Ator: Comprador

Pré-Condição: Comprador tem sala criada

Pós-Condição: Comprador tem uma sala a menos associada

(Fluxo Normal)

1. Comprador escreve `"/exit"`

2. Sistema apaga a sala

Este *use case* satisfaz o requisito **RF10**.

▪ Use Case: Fechar Sala

Ator: Comprador

Pré-Condição: Comprador tem sala criada

Pós-Condição: Sala fica fechada e o comprador tem uma venda pendente no histórico

(Fluxo Normal)

1. Comprador escreve `"/sold@<nome do vendedor>/<preço>/<método>"`

2. Sistema cria venda pendente no histórico e fecha a sala

Este *use case* satisfaz o requisito **RF7**.

▪ Use Case: Ver Salas Criadas

Ator: Comprador

Pré-Condição: Comprador tem sala criada

Pós-Condição: Comprador acede diretamente à sua sala criada

(Fluxo Normal)

1. Comprador acede ao seu perfil
2. Comprador acede à secção "Minhas Salas" para ver as salas criadas e/ou visitadas recentemente
3. Comprador seleciona a sala desejada
4. Sistema leva o Comprador para o chat da sala

Este *use case* satisfaz o requisito **RF14**.

Use Cases de Vendedores

▪ **Use Case: Procurar sala**

Ator: Vendedor

Pré-Condição: N/A

Pós-Condição: O vendedor encontra sala com especificações de acordo com os seus critérios de pesquisa.

(Fluxo Normal)

1. O vendedor acede à zona de vendas da plataforma
2. O vendedor procura na lista de salas ou através da barra de pesquisa uma sala com o nome pretendido
3. É escolhida da lista uma sala para entrar

Este *use case* satisfaz o requisito **RF23**

▪ **Use Case: Ver salas já interagidas**

Ator: Vendedor

Pré-Condição: Vendedor já interagiu com alguma sala previamente criada.

Pós-Condição: O vendedor encontra lista de salas já interagidas.

(Fluxo Normal)

1. Vendedor acede à página de perfil
2. É escolhida a opção de “Minhas Salas”

Este *use case* satisfaz o requisito **RF18**

▪ **Use Case: Sair de uma sala**

Ator: Vendedor

Pré-Condição: Vendedor já está presente numa sala.

Pós-Condição: O vendedor sai da sala.

(Fluxo Normal)

1. Vendedor escreve o comando “/exit” no chat da sala
2. Vendedor sai da sala

Este *use case* satisfaz o requisito **RF9**

Use Cases de Visitantes

▪ **Use Case: Criar Conta**

Ator: Visitante

Pré-Condição: N/A

Pós-Condição: Visitante possui conta associada ao site

(Fluxo Normal)

1. Visitante seleciona a opção “Login”
2. Visitante seleciona a opção “Registar”
3. Sistema leva o Visitante a uma sala onde fornece Nome, E-mail, Senha, Data de Nascimento e Nacionalidade
4. Visitante torna-se Utilizador, com conta associada ao site

(Fluxo Exceção - Nome ou E-mail já em uso)

3.1 Sistema alerta sobre nome ou e-mail já em uso

3.2 Visitante altera o nome ou e-mail para algo único (avança para o Passo 4)

Este *use case* satisfaz o requisito **RF22**.

▪ Use Case: Autenticar-se

Ator: Visitante

Pré-Condição: Visitante possui conta no site

Pós-Condição: Visitante entra com a sua conta no site

(Fluxo Normal)

1. Visitante seleciona "Login"
2. Visitante seleciona "Login"
3. Sistema leva o Visitante a uma sala onde indica o seu E-mail e Senha
4. Visitante torna-se Utilizador

(Fluxo Exceção - E-mail ou Senha incorretos)

3.1 Sistema alerta sobre e-mail ou senha incorretos

3.2 Visitante altera e-mail ou senha para algo correto (avança para o Passo 4)

Este *use case* satisfaz o requisito **RF21**.

Use Cases de Utilizadores

▪ Use Case: Editar Perfil

Ator: Utilizador

Pré-Condição: N/A

Pós-Condição: O Utilizador altera informação no site

(Fluxo Normal)

1. O Utilizador acede ao seu perfil

2. O Utilizador escolhe a opção de alterar nome ou senha
3. O Utilizador muda o nome ou senha
4. O Sistema confirma e regista as mudanças

(Fluxo Exceção - Nome já em Uso)

- 3.1 O Sistema alerta o Utilizador sobre o nome já estar em uso
- 3.2 O Utilizador escolhe outro nome único

Este *use case* satisfaz o requisito **RF16 e RF17** .

▪ **Use Case: Conversar nas Salas**

Ator: Utilizador

Pré-Condição: Utilizador está numa sala

Pós-Condição: O Utilizador comunica no chat

(Fluxo Normal)

1. O Utilizador acede a uma sala pelo site
2. O Utilizador escreve uma mensagem no chat e publica-a

Este *use case* satisfaz o requisito **RF11**

▪ **Use Case: Ver Histórico de Compras**

Ator: Utilizador

Pré-Condição: N/A

Pós-Condição: O Utilizador consulta o seu histórico de compras

(Fluxo Normal)

1. O Utilizador acede ao seu perfil
2. O Utilizador acede a opção histórico
3. O Utilizador consulta as suas compras feitas pelo site (se existirem)

(Fluxo Alternativo - Alterar Estado de Compra)

4. O Utilizador clica no botão de uma venda no estado "pendente"
5. O Sistema muda o estado da venda para "recebido"
6. O Sistema apaga a sala associada à venda

Este *use case* satisfaz o requisito **RF15**

■ Use Case: Logout

Ator: Utilizador

Pré-Condição: N/A

Pós-Condição: O Utilizador passa a Visitante

(Fluxo Normal)

1. O Utilizador vai ao seu perfil
2. O Utilizador seleciona a opção "Logout"
3. O Utilizador deixa de estar autenticado no site e passa a ser um Visitante

Este *use case* satisfaz o requisito **RF5**

Use Cases de Administrador

■ Use Case: Editar Conta de Administrador

Ator: Administrador

Pré-Condição: Já existe uma conta de administrador

Pós-Condição: Um ou mais atributos da conta de administrador são editados

(Fluxo Normal)

1. O Administrador acede à opção "Perfil" no cabeçalho.
2. O Sistema exibe os atributos da conta de administrador, como nome, e-mail, etc.
3. O Administrador faz as edições necessárias nos atributos desejados.

4. O Sistema confirma as alterações e atualiza a conta de administrador com as informações editadas.

(Fluxo Exceção - Nome já em Uso)

- 3.1 O Sistema alerta o Administrador de que o nome já está em uso.
- 3.2 O Administrador escolhe outro nome que seja único.

■ Use Case: Fechar o Site para Manutenção

Ator: Administrador

Pré-Condição: O site está em funcionamento

Pós-Condição: O site é fechado para manutenção

(Fluxo Normal)

1. O Administrador seleciona a opção "Fecho para manutenção" no perfil.
2. O Administrador insere a palavra-passe após ser pedida como medida de segurança.
3. O Sistema exibe uma mensagem informando aos utilizadores que o site está em manutenção e não está disponível para acesso.

(Fluxo Alternativo – Administrador Reabre o Site após Manutenção)

4. O Administrador escolhe reabrir o site após a manutenção e clica no botão agora acessível para reabrir o acesso aos utilizadores.
5. O Sistema volta a ficar online e os utilizadores podem aceder livremente ao serviço.

■ Use Case: Obter Estatísticas do Site

Ator: Administrador

Pré-Condição: O Administrador está autenticado no sistema.

Pós-Condição: O Administrador obtém estatísticas atualizadas do site.

(Fluxo Normal)

1. O Administrador acede a opção "Estatísticas" no perfil.

2. O Sistema exibe informações estatísticas relevantes, como número de utilizadores, transações, salas ativas, etc.

▪ Use Case: Fechar Salas

Ator: Administrador

Pré-Condição: O Administrador recebe uma queixa de um utilizador.

Pós-Condição: Uma sala é fechada.

(Fluxo Normal)

1. O Administrador recebe uma queixa de um utilizador.
2. Ele acede à opção de gerir salas na página de perfil.
3. O Administrador seleciona as salas que deseja fechar.
4. O sistema fecha as salas selecionadas, impedindo novas interações devido ao estado da sala ter sido modificado para fechada.

▪ Use Case: Banir Utilizadores

Ator: Administrador

Pré-Condição: N/A

Pós-Condição: Um ou mais utilizadores são banidos do sistema.

(Fluxo Normal)

1. O Administrador acede a opção "Banir utilizadores" no perfil.
2. O Administrador insere o e-mail ou handle do utilizador na caixa de texto.
3. O Sistema muda o estado da conta inserida para banida, impedindo o seu acesso ao sistema.

(Fluxo Exceção - Ofensa Não Suficiente para Banir)

- 3.1 O Administrador informa ao utilizador que a ofensa não justifica a ação de banir.

4 Conceção do Sistema de Dados

4.1 Apresentação Geral da Estrutura (esquema) do Sistema de Dados

4.1.1 Modelo Lógico

O modelo lógico representa a estrutura do sistema de dados de forma abstrata, tendo como foco as relações entre entidades sem considerar a implementação física (que será feita numa fase posterior). Ele fornece uma visão conceptual das informações, identificando entidades, atributos e as interações entre elas. Esta abordagem facilita a compreensão do sistema, priorizando a lógica da informação em detrimento de detalhes técnicos.

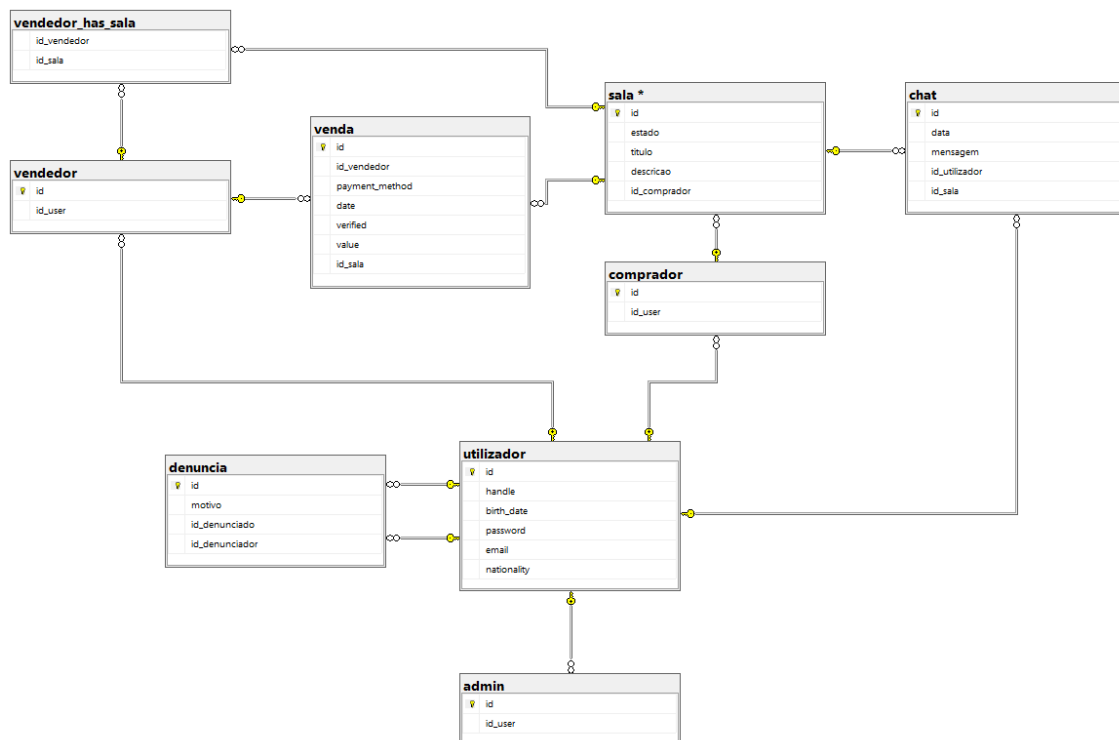


Figura 4.1: Modelo Lógico

4.2 Descrição detalhada dos vários elementos de dados e seus relacionamentos

4.2.1 Entidades:

- **Chat:**

- ID: INT
- ID_Sala: INT
- ID_Utilizador: INT
- Data: datetime
- Mensagem: nvarchar

Conforme especificado no **RF13**

- **Utilizador:**

- ID: INT
- Handle: nvarchar
- Birth_date: date
- Password: nvarchar
- Email: nvarchar
- Nationality: nvarchar

Conforme especificado no **RF22**

- **Comprador:**

- ID: INT
- ID_User: INT

- **Sala:**

- ID: INT
- Estado: INT
- Titulo: nvarchar

- Descrição: nvarchar
- ID_Cliente: INT
- ID_Produto: INT
- ID_Venda: INT

Conforme especificado no **RF6, RF13 e RF24**

▪ **Venda:**

- ID: INT
- Payment_Method: nvarchar
- Date: datetime
- Verified: bit

Conforme especificado no **RF1**

▪ **Vendedor:**

- ID: INT
- ID_User: INT

▪ **Admin:**

- ID: INT
- ID_User: INT

▪ **Denuncia:**

- ID: INT
- ID_Denunciado: INT
- ID_Denunciador: INT
- Motivo: nvarchar

4.2.2 Relacionamentos

- **Utilizador:** O utilizador tem um relacionamento **1:1** com as entidades Admin, Comprador e Vendedor, uma vez que são todos utilizadores.

- **Venda:** Uma venda tem um relacionamento **1:1** com a Sala, pois é o resultado de uma transação, levando posteriormente ao encerramento normal da sala.
- **Sala:** Uma sala tem um relacionamento **1:1**, para além da venda, com o comprador, sendo este o seu criador. Também possui um relacionamento **N:M** com o vendedor, podendo este estar presente em múltiplas salas, tal como vários vendedores poderão estar presentes na mesma sala.
- **Chat:** O chat tem um relacionamento **1:N** com o utilizador, pois representa a ligação de uma mensagem no chat ao utilizador que a escreveu, podendo este escrever várias mensagens no mesmo chat. Também tem outro relacionamento **1:1** com a sala, onde o chat se encontra.
- **Denuncia:** A denuncia tem dois relacionamentos **1:1**, sendo um deles com o utilizador que **denuncia** e o outro, com o utilizador **denunciado**.

5 Esboço das Interfaces do Sistema

Após uma análise detalhada dos requisitos, a criação de diagramas e a caracterização da base de dados, chegamos ao ponto de desenvolver *mockups* que retratam a interação dos utilizadores com o sistema. Este capítulo apresenta imagens elaboradas pelo grupo, acompanhadas de explicações concisas sobre o funcionamento dos principais elementos nelas contidos. Além disso, fornecemos uma estrutura simplificada para facilitar a interpretação das interfaces.

5.1 Estrutura geral das interfaces do sistema

Para compreender a transição entre as diferentes interfaces, foi desenvolvido o seguinte esquema:

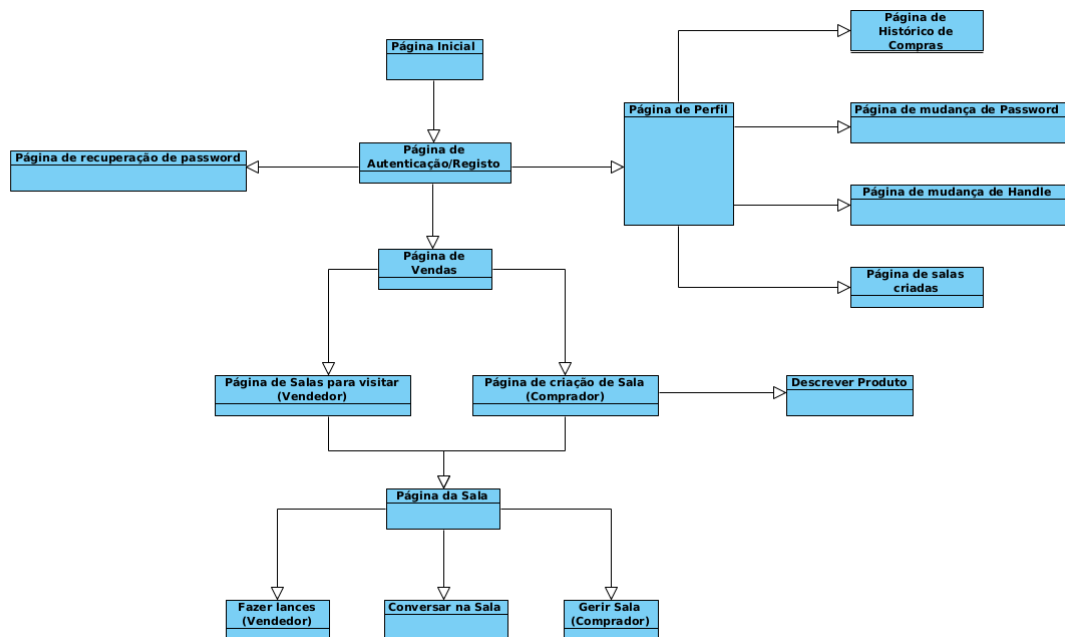


Figura 5.1: Estrutura geral das interfaces do sistema

5.2 Caracterização das interfaces

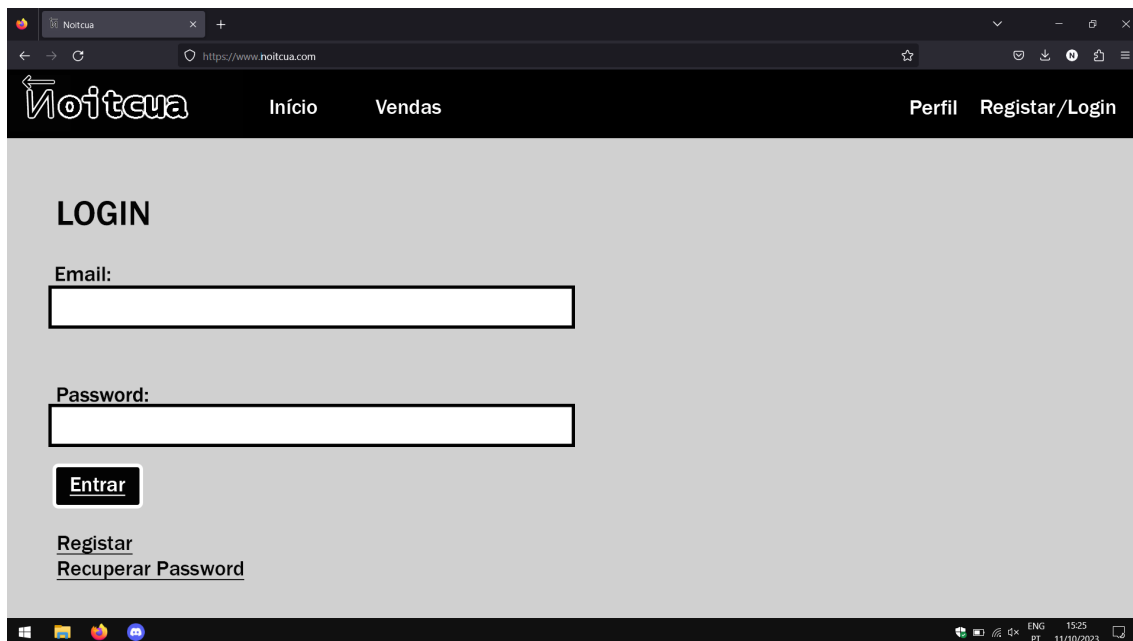
5.2.1 Página de Entrada



Figura 5.2: Mockup da página inicial

O sistema Noitcua apresenta uma interface principal apresentada a todos os visitantes. Na página inicial, além das boas-vindas, os utilizadores serão informados acerca das regras operacionais, com respostas às perguntas frequentes relacionadas com o uso do serviço.

5.2.2 Páginas de Login e Registo

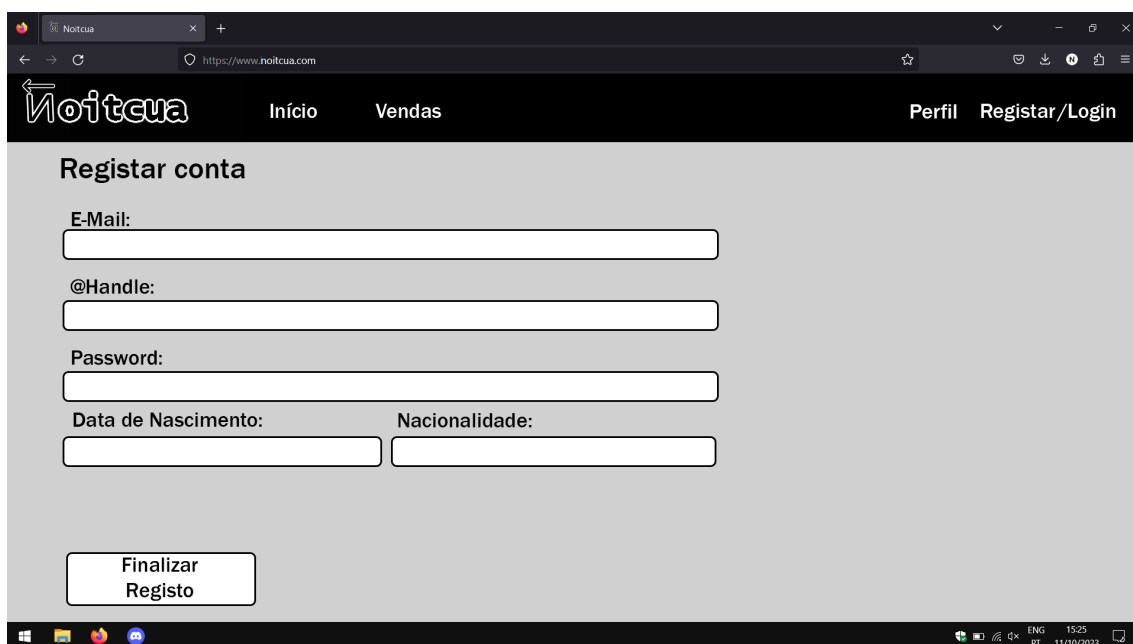


The image shows a web browser window with the URL <https://www.noitcua.com>. The page has a dark header with the logo "Noitcua" on the left and navigation links "Início", "Vendas", "Perfil", and "Registrar/Login" on the right. The main content area is titled "LOGIN" and contains the following elements:

- An "Email:" label followed by a text input field.
- A "Password:" label followed by a text input field.
- An "Entrar" button.
- Links for "Registrar" and "Recuperar Password".

The browser's taskbar at the bottom shows the Windows logo, application icons, and system tray information including language (ENG/PT) and date/time (15:25, 11/10/2023).

Figura 5.3: Mockup da página de Login



The image shows the same web browser window, but the page is titled "Registar conta". The registration form includes the following fields:

- "E-Mail:" followed by a text input field.
- "@Handle:" followed by a text input field.
- "Password:" followed by a text input field.
- "Data de Nascimento:" followed by a text input field.
- "Nacionalidade:" followed by a text input field.
- A "Finalizar Registo" button.

The header and browser taskbar are identical to the previous figure.

Figura 5.4: Mockup da página de Registo

Para aceder às restantes funcionalidades do *site*, o utilizador deve autenticar-se clicando no botão "Login/Registar". Será redirecionado para uma página onde poderá inserir o seu e-

mail e *password* se já possuir uma conta. Caso contrário, ao clicar no botão "Registar", será redirecionado para o formulário de registo, onde deverá fornecer e-mail, *handle* ou nome de utilizador, *password*, nacionalidade e data de nascimento.

5.2.3 Página de Vendas



Figura 5.5: Mockup da página de Vendas

Após a autenticação, a primeira opção disponível é a de aceder à secção de vendas. Os utilizadores podem interagir ao entrar em salas organizadas por data de criação, onde podem vender um produto correspondente ao interesse de outros utilizadores ou criar a sua própria sala para encontrar vendedores que possuam o item desejado.

5.2.4 Página de Criação de Sala

Nome da Sala:

Descrição:

Criar Sala

Salas Criadas 2/3

Figura 5.6: Mockup da página de Criação de Sala

Ao clicar no botão de criação de sala na página mencionada anteriormente (Figura 5.5), um menu é exibido solicitando a inserção do nome da sala e a sua descrição. Cada utilizador pode criar até 3 salas. Após o preenchimento desses campos, a sala é tornada pública na secção de vendas.

5.2.5 Página de Perfil

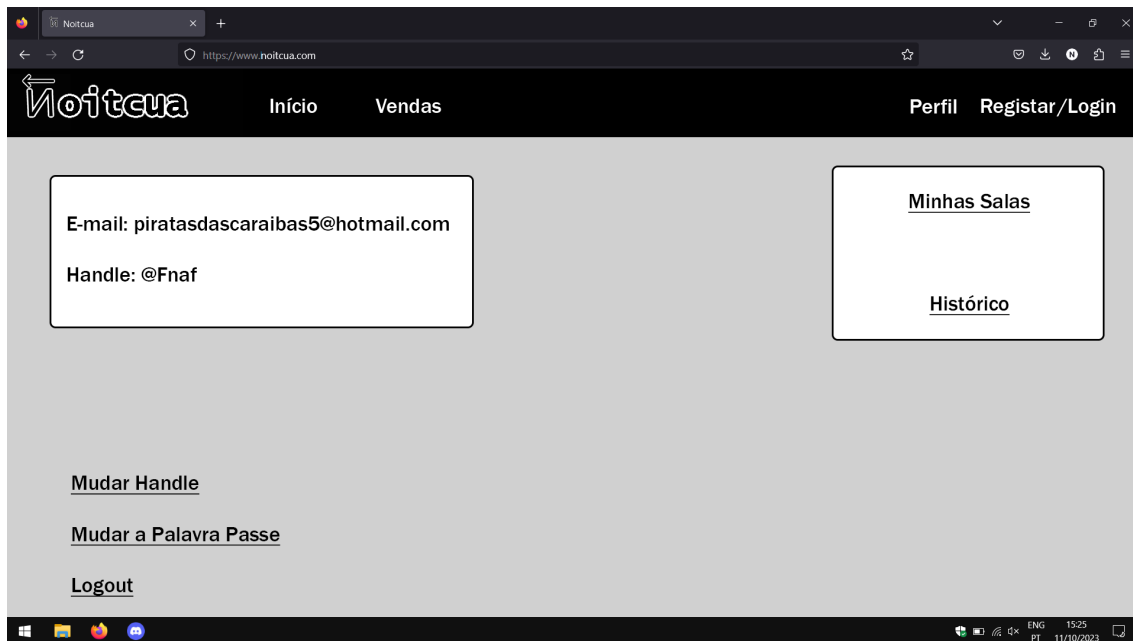


Figura 5.7: Mockup da página de Perfil

A opção seguinte é a de acesso ao perfil, onde o utilizador pode visualizar e modificar as suas informações, incluindo senha e *handle*. Além disso, existem duas opções adicionais: "Minhas Salas" e "Histórico".

5.2.6 Página de Salas do Utilizador

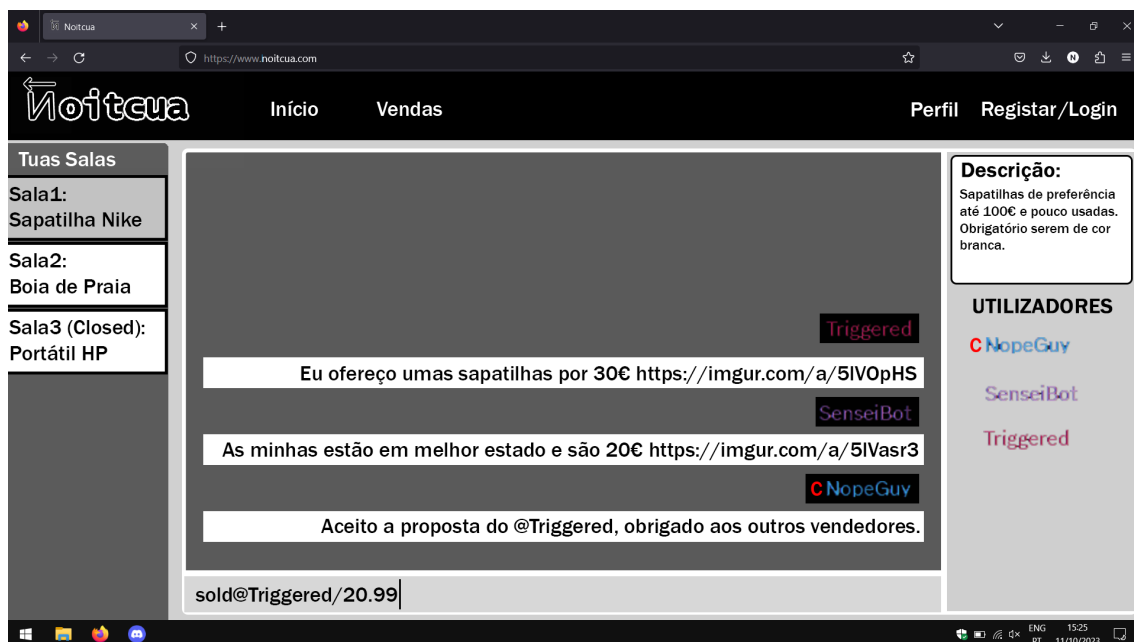


Figura 5.8: Mockup da página de Salas do Utilizador

Na opção "Minhas Salas", o utilizador terá acesso a uma página com todas as salas previamente interagidas, onde poderá seleccionar qualquer uma para se comunicar com outras pessoas. A página exibirá o histórico de mensagens trocadas, uma lista de utilizadores presentes (com o comprador marcado com um "C" ao lado do nome) e a descrição da sala acima dessa lista. O utilizador também terá acesso a uma lista de comandos para gerir e facilitar o uso desta funcionalidade.

5.2.7 Página do Histórico

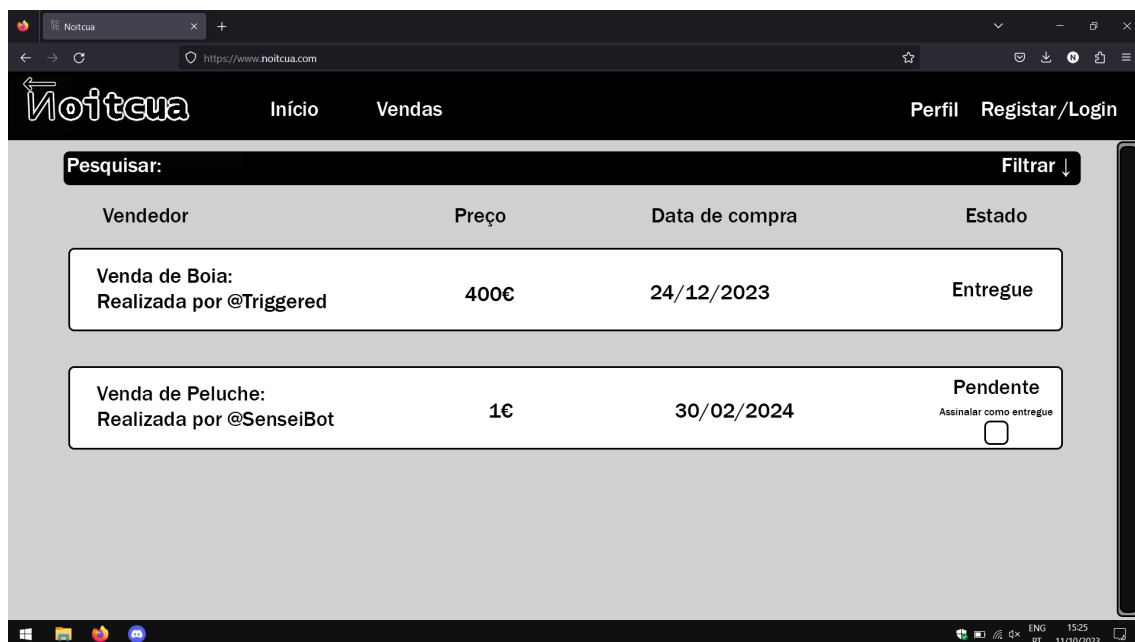


Figura 5.9: Mockup da página do Histórico

Finalmente, temos o histórico, onde cada utilizador terá acesso a uma lista de compras e vendas realizadas através da plataforma. Em caso de compra, o utilizador pode assinalar a receção do item para indicar que o vendedor enviou efetivamente o produto.

6 Implementação da Aplicação

Antes de dar início à última fase do projeto, a equipa de desenvolvimento de *software* realizou uma atualização no diagrama de Gantt, referido na Figura 1.1, de modo a facilitar todo o procedimento restante.

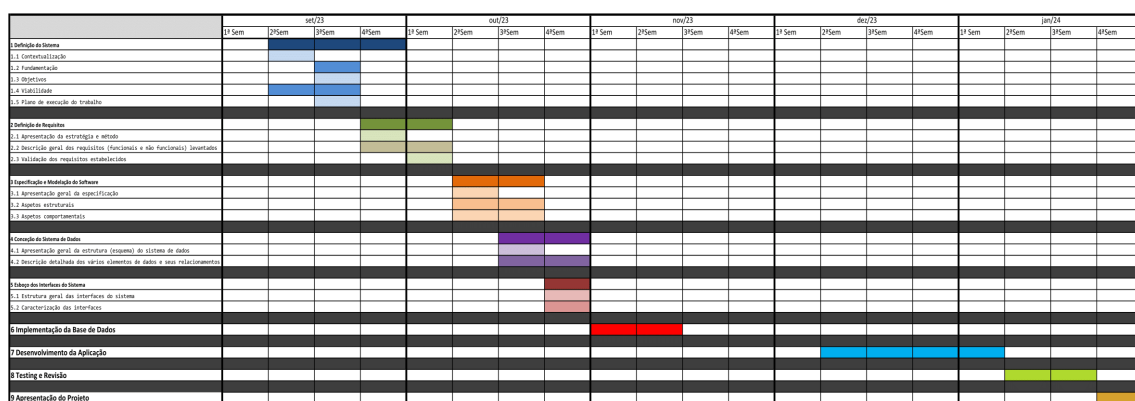


Figura 6.1: Diagrama de Gantt Atualizado

6.1 Descrição da Estrutura Final da Aplicação e Exposição dos Serviços Implementados

Durante a fase de implementação da aplicação Noitcua, a equipa de desenvolvimento procurou assegurar que, numa fase de conclusão, o *software* entregue estivesse totalmente alinhado com as visões cuidadosamente delineadas nas fases anteriores. A seguir, serão apresentados de forma abrangente todos os serviços implementados, acompanhados por uma descrição detalhada da estrutura final da aplicação.

6.1.1 Descrição da Estrutura Final da Aplicação

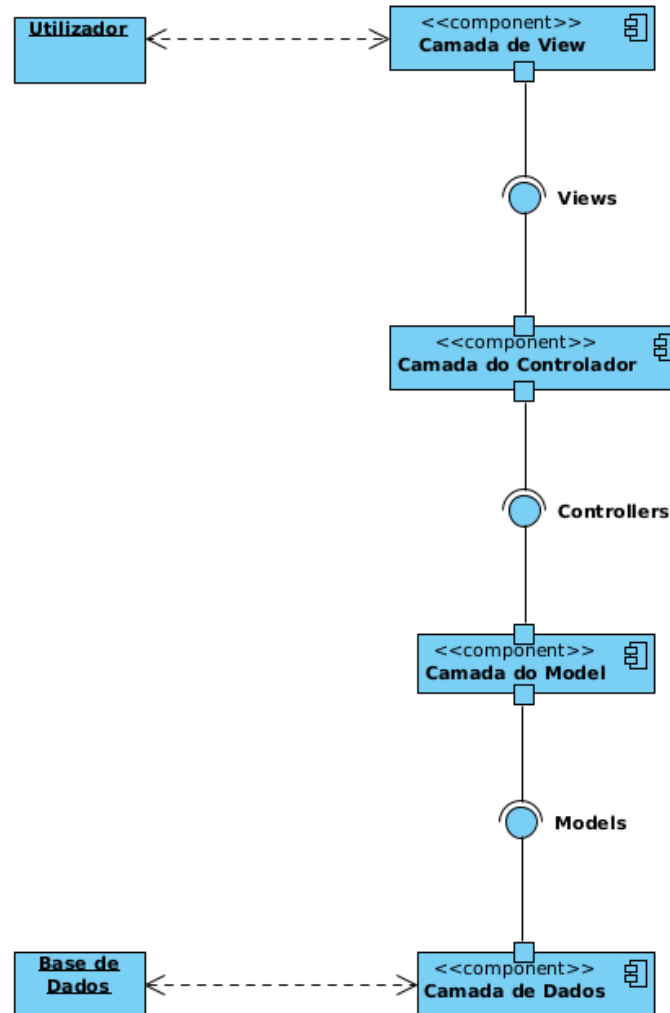


Figura 6.2: Estrutura Final da Aplicação

No desenvolvimento desta aplicação, foi adotado o padrão de arquitetura **MVC** (*Model-View-Controller*). Como ilustrado no esquema acima, cada componente está interconectado, desempenhando um papel específico no funcionamento do programa.

Os **Models** são responsáveis pelo acesso direto à Base de Dados. Na pasta *Models*, encontram-se diversos ficheiros, como *Vendedor.cs* (que define a estrutura de um vendedor na base de dados) e *ForgotPasswordModel.cs* (que contém os atributos a serem modificados ao alterar a senha de um utilizador).

Os **Controllers** têm a função de invocar as *views* e manipular os atributos dos *models*, pos-

sibilitando a modificação e manipulação de dados. Através dos *controllers*, processos como a mudança de *password* e o acesso às páginas *web* são viabilizados. Um exemplo de ficheiro presente nos *controllers* é *AccountController.cs*, responsável pela manipulação de dados relacionados com a conta do utilizador.

As **Views** contêm as páginas do *site* destinadas à visualização por parte dos utilizadores. Ficheiros na *view* incluem, por exemplo, *Profile.cshtml* (página de perfil de um utilizador) e *sales.cshtml* (página de procura de salas).

Desta forma, a arquitetura **MVC** permite uma organização eficiente do código, separando claramente as responsabilidades de modelação de dados, manipulação de lógica de negócios e apresentação de informações aos utilizadores.

6.1.2 Exposição dos Serviços Implementados

Página de Entrada

A página de entrada desempenha um papel crucial como ponto de partida e principal interface para os utilizadores ao acederem a aplicação. Analisemos a secção de **FAQ** (Perguntas Frequentes), presente nela:

- **FAQ** (Perguntas Frequentes): A secção de **FAQ** é uma componente vital da página inicial, projetada para fornecer respostas rápidas e claras às perguntas mais comuns dos utilizadores. Esta área tem como objetivo aprimorar a usabilidade e a compreensão da aplicação, abordando questões frequentemente levantadas.

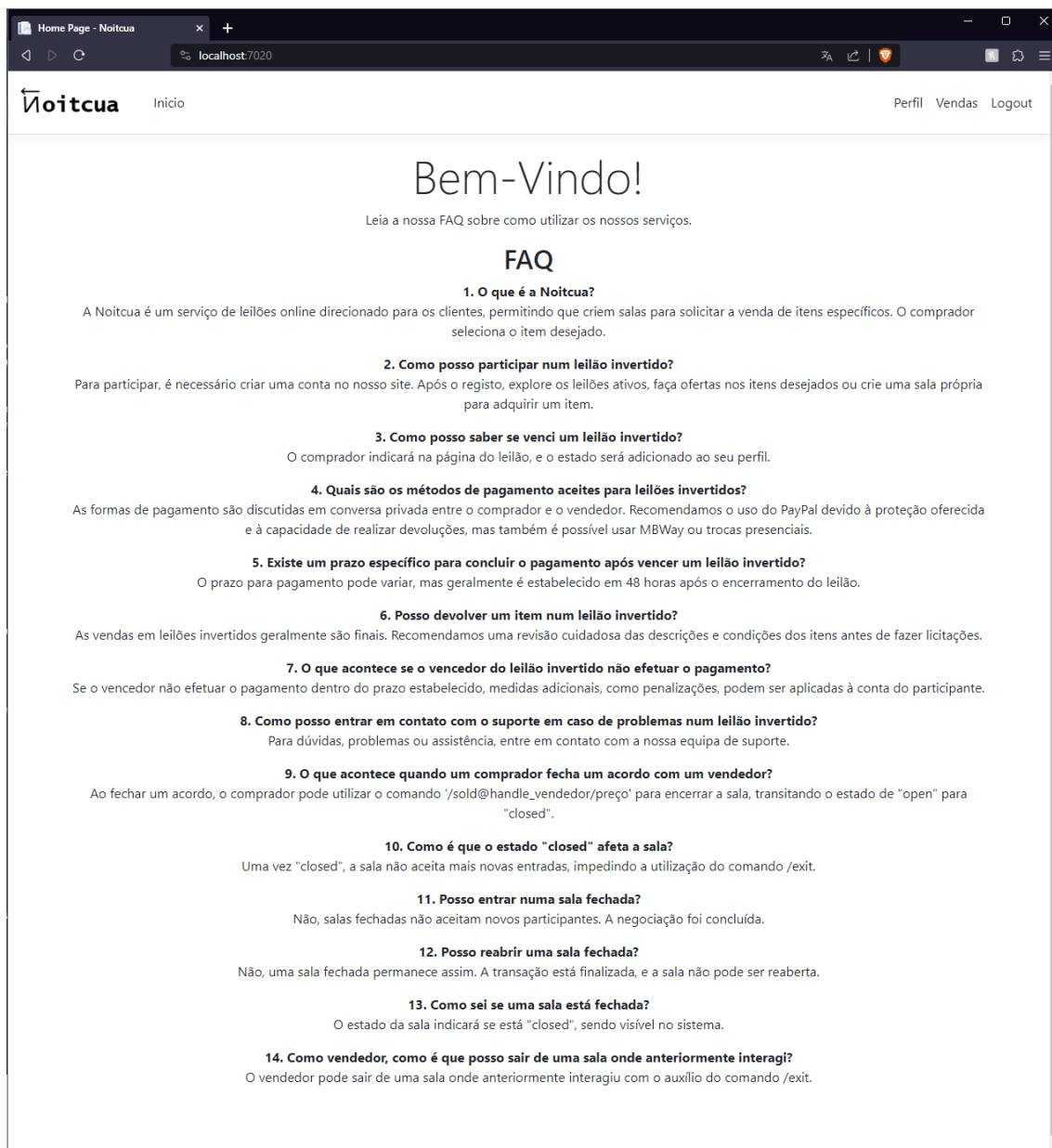
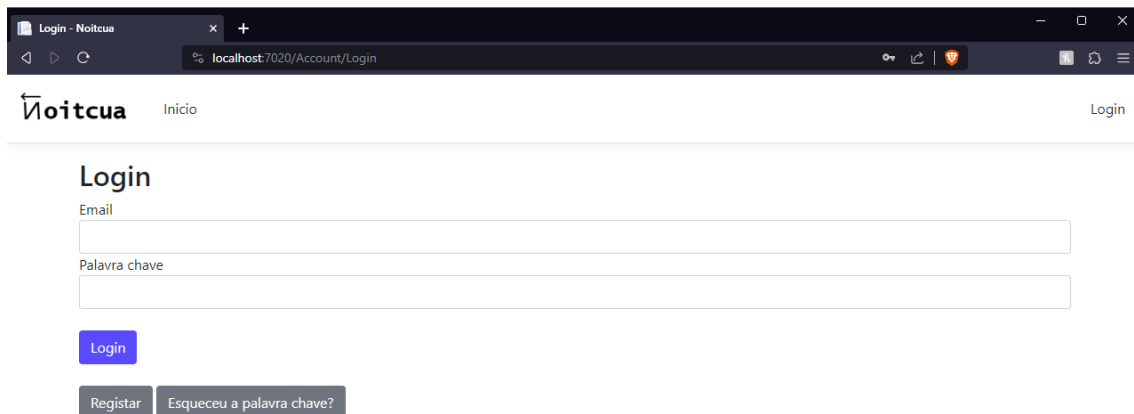


Figura 6.3: Página de Entrada

Página de Login

A partir do botão *login*, o utilizador pode realizar a sua autenticação. Nesta página, deve inserir o seu endereço de e-mail e a respectiva *password*. É a partir desta página, que o utilizador pode aceder às páginas de registo e de recuperação de palavra chave.

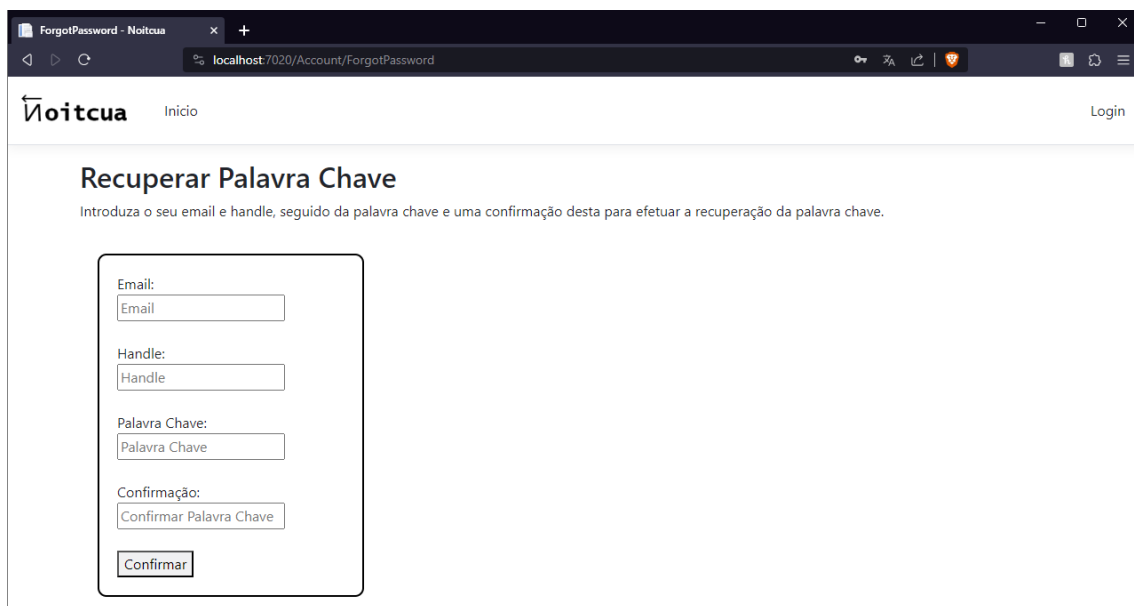


The screenshot shows a web browser window with the address bar displaying 'localhost:7020/Account/Login'. The page title is 'Login - Noitcua'. The header includes the 'Noitcua' logo, a link to 'Início', and a 'Login' link. The main content area is titled 'Login' and contains a form with two input fields: 'Email' and 'Palavra chave'. Below the fields is a blue 'Login' button. At the bottom of the form are two links: 'Registar' and 'Esqueceu a palavra chave?'.

Figura 6.4: Página de Login

Página de Recuperação de Palavra Chave

Nesta página, o utilizador tem a opção de recuperar a sua *password*. Para tal, precisa de inserir dados como: e-mail, *handle*, *password* e confirmação da mesma. Esta funcionalidade é útil em casos onde o utilizador se esqueceu da sua *password*.



The screenshot shows a web browser window with the address bar displaying 'localhost:7020/Account/ForgotPassword'. The page title is 'ForgotPassword - Noitcua'. The header includes the 'Noitcua' logo, a link to 'Início', and a 'Login' link. The main content area is titled 'Recuperar Palavra Chave' and includes a sub-header: 'Introduza o seu email e handle, seguido da palavra chave e uma confirmação desta para efetuar a recuperação da palavra chave.' Below this is a form with four input fields: 'Email:', 'Handle:', 'Palavra Chave:', and 'Confirmação:'. Each field has a placeholder text matching its label. At the bottom of the form is a 'Confirmar' button.

Figura 6.5: Página de Recuperação de Palavra Chave

Página de Registo

A partir da página de registo, é possível criar uma conta na aplicação. Para realizar o registo, é necessário fornecer os seguintes dados:

- *Handle*
- *Birth Date*
- *Password*
- *Confirm Password*
- *Email*
- *Nationality*

Figura 6.6: Página de Registo

Além disso, a aplicação assegura que, durante o processo de registo, o utilizador escolha uma *password* com um mínimo de 8 caracteres.

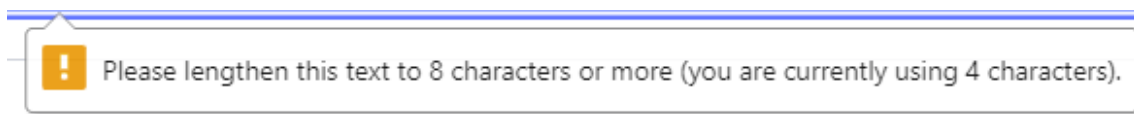


Figura 6.7: Tamanho Mínimo de Password

Página de Perfil

Após uma autenticação bem-sucedida, os utilizadores têm a oportunidade de explorar o menu de perfil, oferecendo uma visão abrangente da sua conta. Neste espaço personalizado, os seguintes elementos são destacados:

- **Detalhes do Perfil** : Os utilizadores podem aceder a informações cruciais, como o seu endereço de e-mail e o identificador único associado à sua conta, conhecido como *handle*. Estes detalhes são fundamentais para garantir uma identificação precisa e segura.

- **Navegação Rápida**

Além disso, são oferecidas opções de navegação rápida para áreas-chave da aplicação. Os utilizadores podem escolher entre:

- **Minhas Salas**: Garante acesso à página dedicada às salas onde o utilizador participa ou gere.
- **Histórico**: Garante acesso a uma página com um histórico detalhado de atividades, proporcionando uma visão abrangente das interações passadas.
- **Mudar Handle**: Garante acesso à página onde é possível ajustar e personalizar o identificador *handle* associado à conta.
- **Mudar Senha**: Garante acesso à página dedicada à alteração segura da *password* da conta, promovendo uma gestão eficaz da segurança.

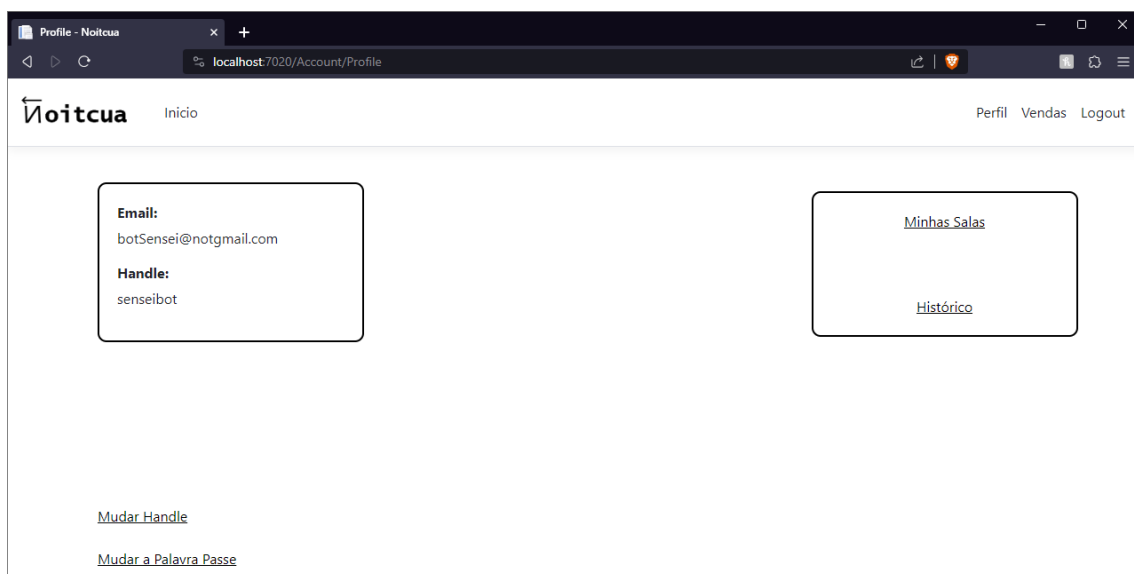
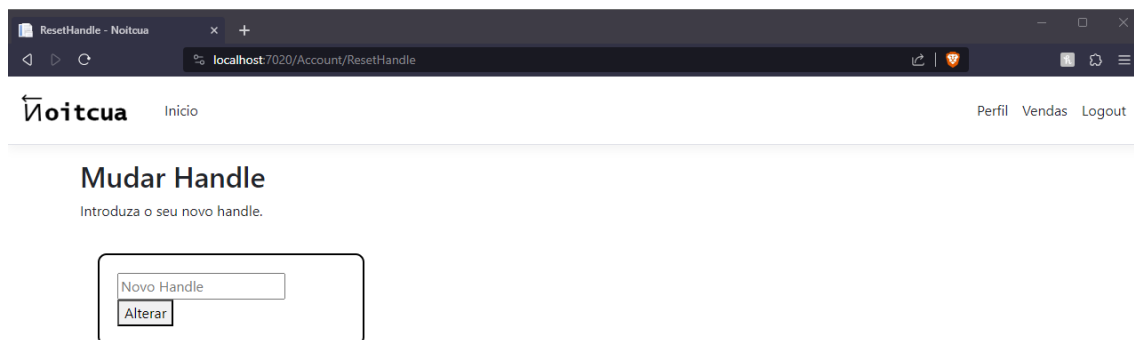


Figura 6.8: Página de Perfil

Esta abordagem intuitiva e centralizada permite que os utilizadores controlem e personalizem facilmente a sua experiência na plataforma, garantindo praticidade e segurança em cada interação.

Página de Mudança de Handle

Nesta página, os utilizadores têm a capacidade de modificar o seu *handle*.



The screenshot shows a web browser window with the address bar displaying 'localhost:7020/Account/ResetHandle'. The page header includes the 'Noitcua' logo, a 'Início' link, and user links for 'Perfil', 'Vendas', and 'Logout'. The main heading is 'Mudar Handle', followed by the instruction 'Introduza o seu novo handle.' Below this is a form with a text input field labeled 'Novo Handle' and a button labeled 'Alterar'.

Figura 6.9: Página de Mudança de Handle

Página de Mudança de Palavra-Passe

Esta página proporciona aos utilizadores a capacidade de alterar a sua *password*.



The screenshot shows a web browser window with the address bar displaying 'localhost:7020/Account/ResetPassword'. The page header is identical to the previous figure. The main heading is 'Alterar Palavra Chave', followed by the instruction 'Introduza a palavra chave e uma confirmação desta para efetuar a mudança da palavra chave.' Below this is a form with two text input fields: 'Palavra Chave:' and 'Confirmação:'. The first field contains the placeholder text 'Palavra Chave', and the second field contains the placeholder text 'Confirmar Palavra Chave'. A 'Confirmar' button is located at the bottom of the form.

Figura 6.10: Página de Mudança de Palavra Passe

Página de Vendas

A página de Vendas exibe todas as salas ativas, oferecendo aos utilizadores a possibilidade de participar em leilões em progresso. Os leilões ativos são listados, proporcionando uma visão fácil e acessível para os interessados. Os utilizadores têm a opção de procurar salas através de uma barra de pesquisa, entrar em salas existentes ao clicar no botão "Entrar na Sala", ou criar uma nova sala.

Esta abordagem intuitiva permite aos utilizadores uma navegação eficiente, tornando a experiência na plataforma mais amigável.

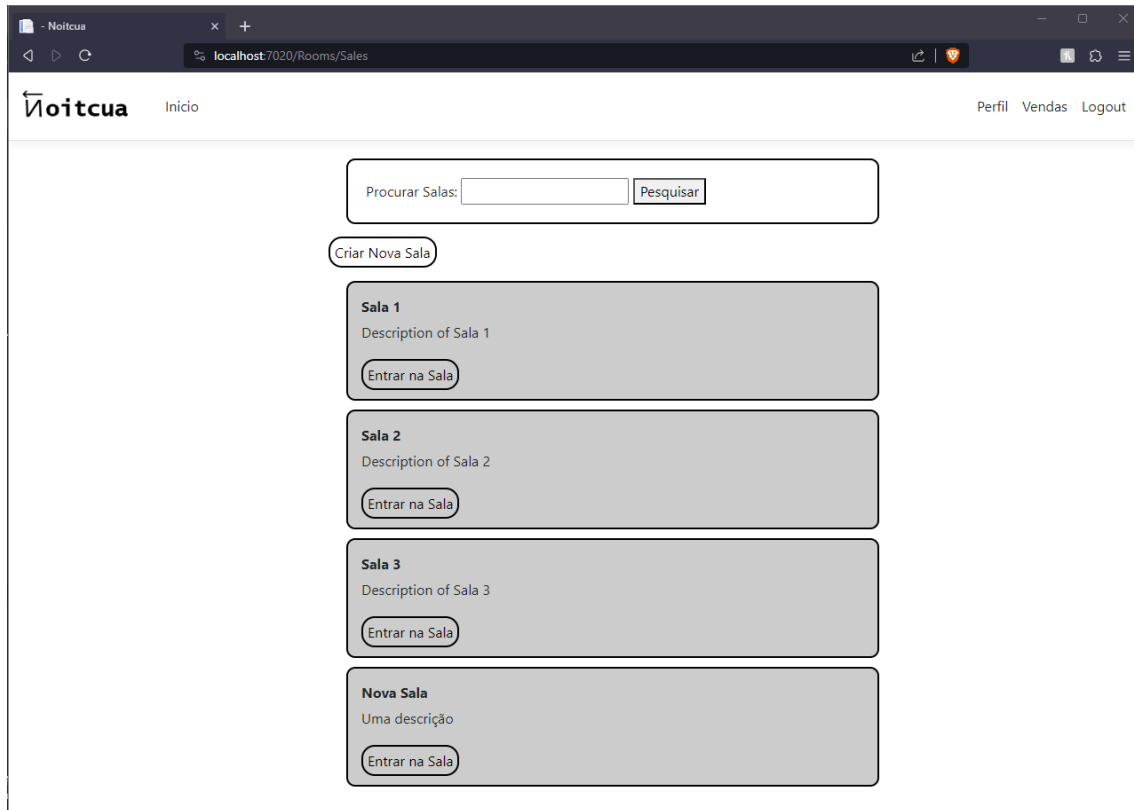


Figura 6.11: Página de Vendas

Página de Criação de uma Sala

A página de Criação de uma Sala permite ao utilizador criar uma sala, que depois será apresentada dentro da Página de Vendas.

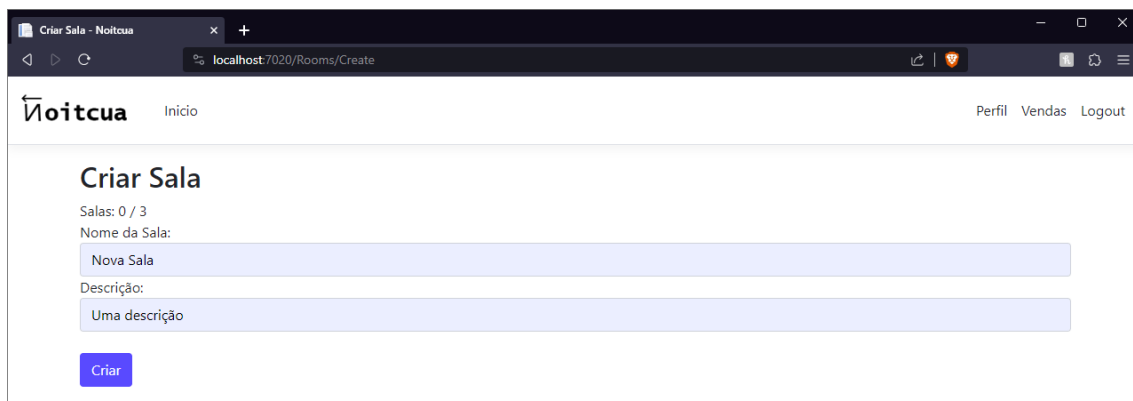


Figura 6.12: Página de Criar Sala

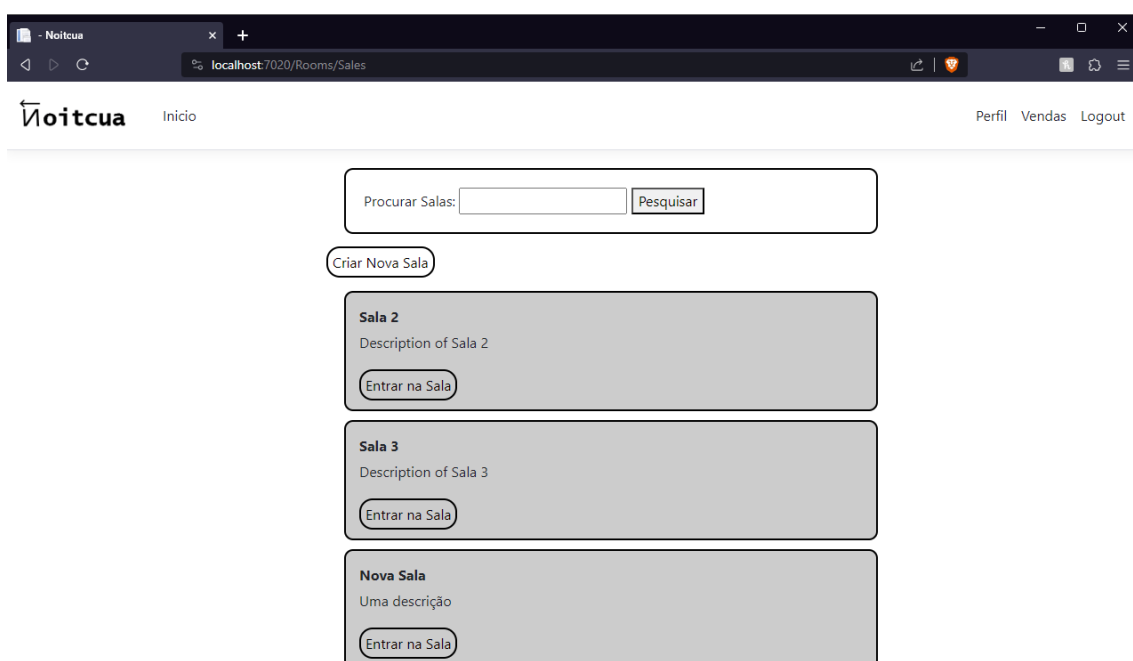


Figura 6.13: "Nova Sala" apresentada na Página de Vendas

Página de Salas do Utilizador

A página de salas do utilizador exibe o histórico de mensagens trocadas, uma lista de utilizadores presentes (com o comprador marcado com um "C" ao lado do nome) e a descrição da sala acima dessa lista. O utilizador também tem acesso a uma lista de comandos para gerir e facilitar o uso desta funcionalidade.

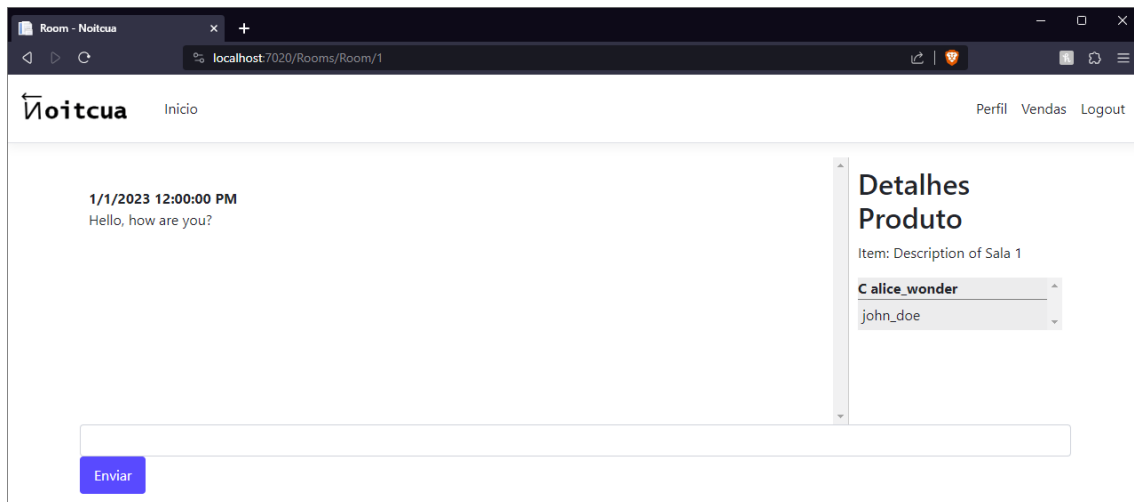


Figura 6.14: Página da Salas do Utilizador

Página do Histórico

Nesta página, cada utilizador tem acesso a uma lista de compras e vendas realizadas através da plataforma. Em caso de compra, o utilizador pode assinalar a receção do item para indicar que o vendedor enviou efetivamente o produto.

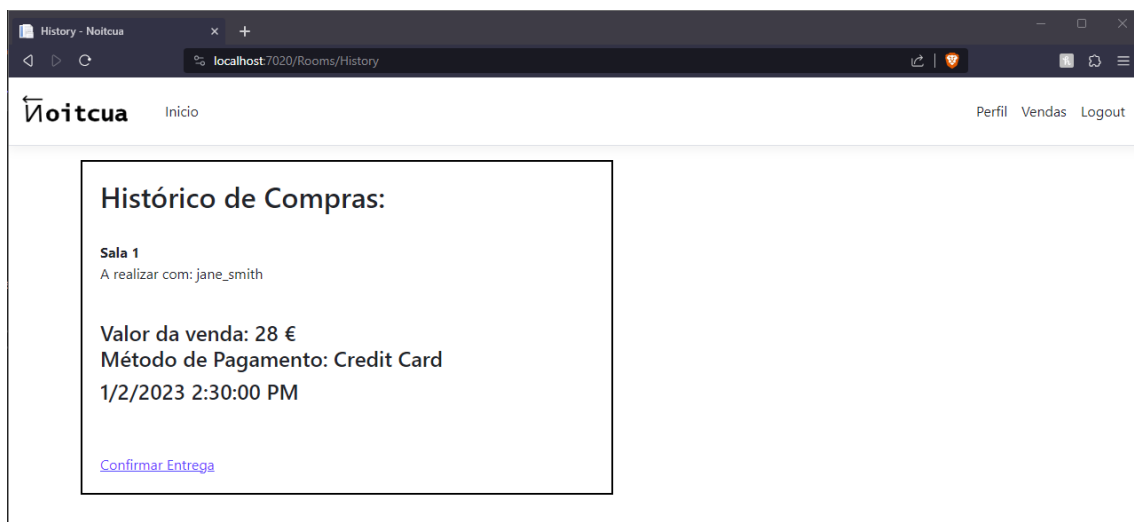


Figura 6.15: Página do Histórico

Após a confirmação da entrega, a página de histórico é atualizada, indicando que o produto foi entregue.

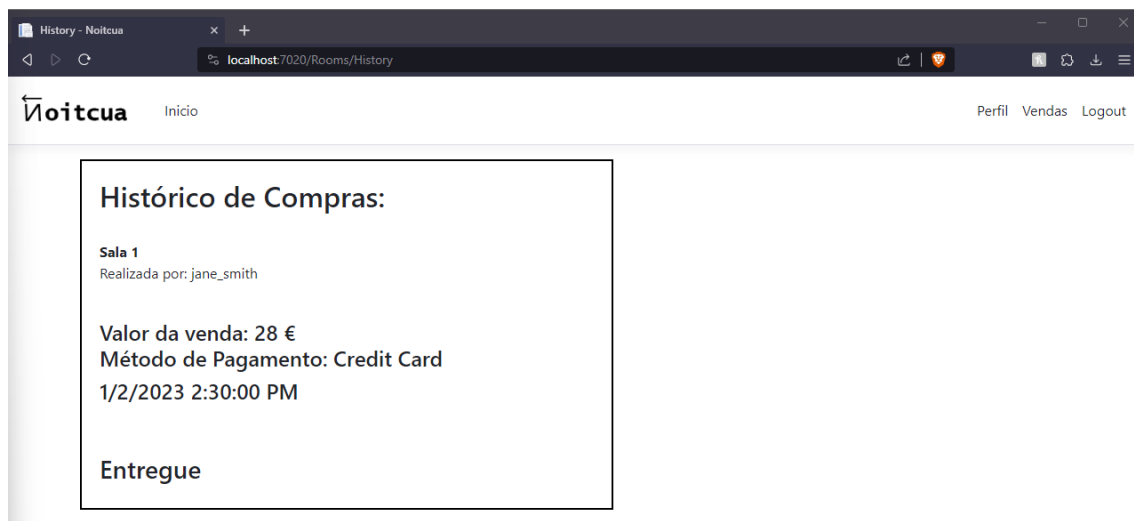


Figura 6.16: Produto Entregue

6.2 Ferramentas utilizadas

Numa fase introdutória do projeto, concentramo-nos nos recursos fundamentais, tanto humanos quanto materiais, necessários para a sua realização. No âmbito dos recursos materiais de *software*, identificamos as seguintes exigências essenciais:

Sistema de Gestão de Bases de Dados: Optamos por adotar o *Microsoft SQL Server* como a nossa solução para gestão de bases de dados. Esta decisão procurou proporcionar uma solução segura e de elevado desempenho para a administração de todas as informações pertinentes à aplicação.

Ambiente de Desenvolvimento: O ambiente de desenvolvimento utilizado trata-se do *Microsoft Visual Studio 2022*, uma plataforma integrada de desenvolvimento (**IDE**). Este ambiente revelou-se crucial na implementação do código da aplicação, oferecendo recursos robustos para facilitar o desenvolvimento orientado a testes. Adicionalmente, empregamos a plataforma **.NET** em conjunto com as *frameworks* **ASP.NET** para conceber a interface gráfica da aplicação.

Ferramentas de Apoio à Gestão do Projeto: No contexto da gestão do projeto, utilizamos ferramentas tais como o *GitHub*, uma plataforma essencial para o desenvolvimento colaborativo de *software*. Adicionalmente, contamos com a plataforma *Overleaf*, desempenhando um papel vital na elaboração da documentação do projeto. Para a modelação dos diagramas especificados, adotamos a linguagem **UML**. Na criação dos diagramas, confiamos na ferramenta *Visual Paradigm*, que oferece uma abordagem visual para a construção de diagramas **UML**.

Estas escolhas estratégicas proporcionaram uma base robusta para o desenvolvimento do projeto, assegurando eficiência, segurança e colaboração eficaz ao longo de todo o processo.

6.3 Análise Crítica e Avaliação da Aplicação Desenvolvida

Ao analisar de forma abrangente o desenvolvimento deste projeto, identificamos alguns requisitos, *use cases* e funcionalidades que não foram totalmente cumpridos ou implementados conforme o planeamento inicial. Aqui está uma lista desses pontos:

- Requisitos não implementados :
 - **[RF2]** O histórico deverá estar equipado com uma barra de pesquisa que permitirá ao utilizador pesquisar a janela de compras pelo nome do vendedor ou nome do artigo.
 - **[RF3]** O histórico de compras incluirá um campo de filtro, permitindo que o utilizador insira um intervalo de datas ou de preços para filtrar os resultados na janela de compras.
 - **[RF9]** O comprador tem a opção de remover vendedores da sala mediante o comando `'/kick @nome do vendedor'`.
 - **[RF10]** O comprador tem a capacidade de filtrar as mensagens trocadas com um vendedor na sala através do comando `'/filter @nome do vendedor'` e desfazer essa ação com o comando `'/unfilter'`.
 - **[RF21]** Os utilizadores têm permissão para aceder aos perfis uns dos outros, visualizando as informações públicas, mas não têm acesso ao histórico de atividades dos outros utilizadores.
 - **[RF22]** Os utilizadores têm a capacidade de denunciar outro utilizador, sendo obrigados a fornecer uma justificação para a denúncia. Esta funcionalidade visa promover a segurança e o cumprimento das normas da comunidade.
 - **[RF23]** Se uma conta for denunciada por um motivo justificável, a conta é sujeita a ser banida, como medida de aplicação de políticas e garantia da integridade da comunidade.
 - *Use Cases* não implementados :
 - **Use Case: Filtrar Mensagens**
 - Ator:* Comprador
 - Pré-Condição:* Sala criada com mais de um vendedor a interagir
 - Pós-Condição:* Comprador acede a mensagens de um vendedor específico
- (Fluxo Normal)**

1. Comprador escreve `"/filter @<nome do vendedor>"`
2. Sistema mostra mensagens do vendedor mencionado

(Fluxo Alternativo - Desfiltrar Mensagens)

3. Comprador escreve `"/unfilter"`
4. Sistema mostra mensagens de todos os vendedores presentes na sala

– Use Case: aceder Perfil de Outro Utilizador

Ator: Utilizador

Pré-Condição: Utilizador está numa sala com outros Utilizadores

Pós-Condição: O Utilizador consulta informações de outro Utilizador

(Fluxo Normal)

1. O Utilizador clica no nome de outro Utilizador presente na sala
2. O Sistema redireciona-o para o perfil que ele deseja consultar
3. O Utilizador tem acesso a informações como e-mail, data de nascimento e nacionalidade de outro utilizador

(Fluxo Alternativo - Reportar Outro Utilizador)

4. O Utilizador seleciona a opção "Reportar Utilizador"
5. O Utilizador fornece uma descrição do motivo
6. A queixa é enviada aos administradores do site, que avaliam o comportamento do sujeito e aplicam uma punição adequada

■ Features não implementadas :

- Filtrar mensagens
- Denúncias
- Comando de *kick*
- Administrador

Por outro lado, é importante destacar todo o tipo de alterações e adições que foram feitas às fases anteriores à de implementação :

- Requisitos Funcionais **[RF20]** e **[RF21]** adicionados
- Use case de sala de Vendas
- Atualização do diagrama de use cases
- Atualização do Modelo lógico
- Associação de cada *use case* e entidades do modelo lógico a um ou mais requisitos funcionais

7 Conclusões e Trabalho Futuro

7.1 Conclusões no Checkpoint

Na fase inicial do projeto, os órgãos responsáveis pelo desenvolvimento das cinco etapas iniciais de uma plataforma de leilões online expressam a satisfação com a efetiva conclusão de todas as fases. No entanto, observa-se a necessidade de uma abordagem mais aprofundada nas fases de Especificação e Modelação do Software, exigindo a criação de diagramas de classes, atividades e sequências.

Adicionalmente, é imperativo revisitar a fase 2 (Definição de Requisitos), uma vez que, ao longo do processo, as fases subsequentes (3, 4 e 5) devem estar estritamente alinhadas com os requisitos estabelecidos. Essa revisão visa assegurar a coesão e integridade do projeto, garantindo que cada etapa contribua de maneira consistente para os objetivos e exigências previamente definidos.

7.2 Conclusões na Entrega Final

A bem-sucedida implementação do nosso projeto foi facilitada pela adoção de um modelo de desenvolvimento estruturado, cuidadosamente dividido em fases distintas. Esta abordagem permitiu um planeamento preciso e uma implementação eficiente do *software*, tendo cada fase sido dedicada a aspectos cruciais do projeto: Definição do Sistema, Definição de Requisitos, Especificação e Modelação do Software, Conceção do Sistema de Dados, Esboço das Interfaces do Sistema e Implementação da Aplicação.

Na fase inicial de Definição do Sistema, estabelecemos as bases do projeto, desde a contextualização do problema até a uma definição dos objetivos finais do sistema. Uma análise de viabilidade abrangente foi conduzida, e a alocação estratégica de recursos foi realizada para todos os aspectos do desenvolvimento.

Nas fases seguintes, Definição de Requisitos e Especificação e Modelação do Software, detalhamos os aspectos estruturais e comportamentais do sistema, desde o levantamento dos requisitos até à elaboração de diagramas UML. Estas etapas foram cruciais para fornecer uma especificação clara do sistema.

Estas fases desempenharam um papel fundamental em garantir um planeamento robusto,

confiável e bem estruturado, proporcionando uma compreensão aprofundada do produto de *software* desejado e, conseqüentemente, facilitando a implementação.

Com uma especificação clara do sistema, avançamos para as fases de Conceção do Sistema de Dados e Esboço das Interfaces do Sistema. Na primeira, produzimos um modelo lógico da base de dados, com entidades e relacionamentos criados com base nas fases anteriores. Na segunda, analisamos as transições das interfaces e modelamos as páginas que surgiriam na fase de implementação.

Com todas as fases anteriores bem estruturadas e trabalhadas, entramos na fase de Implementação, focada no desenvolvimento do *software*. Devido à magnitude do projeto, restrições temporais e à necessidade de assimilar novas ferramentas, algumas funcionalidades planeadas não foram totalmente implementadas. Priorizamos tarefas e direcionamos esforços para áreas consideradas cruciais.

Apesar dos desafios, a implementação do sistema alinhou-se com a especificação das fases anteriores, resultando num desenvolvimento positivo. Estamos satisfeitos com a fase de planeamento, cuja importância foi enfatizada pela disciplina.

Para o futuro, planeamos concluir a implementação de funcionalidades pendentes, como a criação de um administrador. Considerando o crescimento esperado no número de clientes, planeamos tornar a aplicação escalável, suportando múltiplas conexões em tempo real. Isto implicará alterações arquiteturais significativas para garantir uma adaptação eficaz a este fenómeno.

Anexos

7.1 Instruções SQL

7.1.1 Drop/Create All

```
USE [li4]
GO
```

```
/***** USER *****/
```

```
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE TABLE [dbo].[utilizador](
[id] [int] IDENTITY(1,1) NOT NULL,
[handle] [nvarchar](20) NOT NULL,
[birth_date] [date] NOT NULL,
[password] [nvarchar](50) NOT NULL,
[email] [nvarchar](50) NOT NULL,
[nationality] [nchar](3) NOT NULL,
CONSTRAINT [PK_utilizador] PRIMARY KEY CLUSTERED
(
[id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = OFF, ALLOW_PAGE_LOCKS = OFF) ON [PRIMARY]
GO
```

```
/***** ADMIN *****/
```

```
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE TABLE [dbo].[admin](
[id] [int] IDENTITY(1,1) NOT NULL,
[id_user] [int] NOT NULL,
CONSTRAINT [PK_admin] PRIMARY KEY CLUSTERED
(
[id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = OFF, ALLOW_PAGE_LOCKS = OFF) ON [PRIMARY]
GO
```

```
ALTER TABLE [dbo].[admin] WITH CHECK ADD CONSTRAINT [FK_admin_utilizador] FOREIGN KEY([id])
REFERENCES [dbo].[utilizador] ([id])
GO
```

```
ALTER TABLE [dbo].[admin] CHECK CONSTRAINT [FK_admin_utilizador]
GO
```

```
/***** VENDEDOR *****/
```

```
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE TABLE [dbo].[vendedor](
[id] [int] IDENTITY(1,1) NOT NULL,
[id_user] [int] NOT NULL,
CONSTRAINT [PK_vendedor] PRIMARY KEY CLUSTERED
(
[id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = OFF, ALLOW_PAGE_LOCKS = OFF) ON [PRIMARY]
GO
```

```
ALTER TABLE [dbo].[vendedor] WITH CHECK ADD CONSTRAINT [FK_vendedor_utilizador] FOREIGN KEY([id])
REFERENCES [dbo].[utilizador] ([id])
GO
```



```
REFERENCES [dbo].[utilizador] ([id])
GO
```

```
ALTER TABLE [dbo].[vendedor] CHECK CONSTRAINT [FK_vendedor_utilizador]
GO
```

```
/***** COMPRADOR *****/
```

```
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE TABLE [dbo].[comprador](
[id] [int] IDENTITY(1,1) NOT NULL,
[id_user] [int] NOT NULL,
CONSTRAINT [PK_comprador] PRIMARY KEY CLUSTERED
(
[id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = OFF, ALLOW_PAGE_LOCKS = OFF) ON [PRIMARY]
GO
```

```
ALTER TABLE [dbo].[comprador] WITH CHECK ADD CONSTRAINT [FK_comprador_utilizador] FOREIGN KEY ([id_user]) REFERENCES [dbo].[utilizador] ([id])
GO
```

```
ALTER TABLE [dbo].[comprador] CHECK CONSTRAINT [FK_comprador_utilizador]
GO
```

```
/***** DENUNCIA *****/
```

```
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE TABLE [dbo].[denuncia](
```

```

[id] [int] IDENTITY(1,1) NOT NULL,
[motivo] [nvarchar](50) NOT NULL,
[id_denunciado] [int] NOT NULL,
[id_denunciador] [int] NOT NULL,
    CONSTRAINT [PK_denuncia] PRIMARY KEY CLUSTERED
(
[id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = OFF,
) ON [PRIMARY]
GO

```

```

ALTER TABLE [dbo].[denuncia] WITH CHECK ADD CONSTRAINT [FK_denunciado_utilizador] FOREIGN KEY
REFERENCES [dbo].[utilizador] ([id])
GO

```

```

ALTER TABLE [dbo].[denuncia] CHECK CONSTRAINT [FK_denunciado_utilizador]
GO

```

```

ALTER TABLE [dbo].[denuncia] WITH CHECK ADD CONSTRAINT [FK_denunciador_utilizador] FOREIGN KEY
REFERENCES [dbo].[utilizador] ([id])
GO

```

```

ALTER TABLE [dbo].[denuncia] CHECK CONSTRAINT [FK_denunciador_utilizador]
GO

```

```

/***** SALA *****/

```

```

SET ANSI_NULLS ON
GO

```

```

SET QUOTED_IDENTIFIER ON
GO

```

```

CREATE TABLE [dbo].[sala](
[id] [int] IDENTITY(1,1) NOT NULL,
[estado] [int] NOT NULL,
[titulo] [nvarchar](50) NOT NULL,
[descricao] [nvarchar](200) NULL,
[id_comprador] [int] NOT NULL,
    CONSTRAINT [PK_sala] PRIMARY KEY CLUSTERED
(
[id] ASC
)

```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = OFF)
ON [PRIMARY]
GO
```

```
ALTER TABLE [dbo].[sala] WITH CHECK ADD CONSTRAINT [FK_sala_comprador] FOREIGN KEY([id_sala])
REFERENCES [dbo].[comprador] ([id])
GO
```

```
ALTER TABLE [dbo].[sala] CHECK CONSTRAINT [FK_sala_comprador]
GO
```

```
/***** CHAT *****/
```

```
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE TABLE [dbo].[chat](
[id] [int] IDENTITY(1,1) NOT NULL,
[data] [datetime] NOT NULL,
[mensaje] [nvarchar](100) NOT NULL,
[id_utilizador] [int] NOT NULL,
[id_sala] [int] NOT NULL,
CONSTRAINT [PK_chat] PRIMARY KEY CLUSTERED
(
[id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = OFF)
ON [PRIMARY]
GO
```

```
ALTER TABLE [dbo].[chat] WITH CHECK ADD CONSTRAINT [FK_chat_sala] FOREIGN KEY([id_sala])
REFERENCES [dbo].[sala] ([id])
GO
```

```
ALTER TABLE [dbo].[chat] CHECK CONSTRAINT [FK_chat_sala]
GO
```

```
ALTER TABLE [dbo].[chat] WITH CHECK ADD CONSTRAINT [FK_chat_utilizador] FOREIGN KEY([id_utilizador])
```

```
REFERENCES [dbo].[utilizador] ([id])
GO
```

```
ALTER TABLE [dbo].[chat] CHECK CONSTRAINT [FK_chat_utilizador]
GO
```

```
/***** VENDA *****/
```

```
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE TABLE [dbo].[venda](
[id] [int] IDENTITY(1,1) NOT NULL,
[id_vendedor] [int] NOT NULL,
[payment_method] [nvarchar](20) NOT NULL,
[date] [datetime] NOT NULL,
[verified] [bit] NOT NULL,
[value] [float] NOT NULL,
[id_sala] [int] NOT NULL,
CONSTRAINT [PK_venda] PRIMARY KEY CLUSTERED
(
[id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = OFF, ALLOW_PAGE_LOCKS = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
```

```
ALTER TABLE [dbo].[venda] WITH CHECK ADD CONSTRAINT [FK_venda_sala] FOREIGN KEY([id_sala])
REFERENCES [dbo].[sala] ([id])
GO
```

```
ALTER TABLE [dbo].[venda] WITH CHECK ADD CONSTRAINT [FK_venda_vendedor] FOREIGN KEY([id_vendedor])
REFERENCES [dbo].[vendedor] ([id])
GO
```

```
ALTER TABLE [dbo].[venda] CHECK CONSTRAINT [FK_venda_sala]
GO
```

```
/****** VENDEDOR_HAS_SALA *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE TABLE [dbo].[vendedor_has_sala](
```

```
  [id_vendedor] [int] NOT NULL,
```

```
  [id_sala] [int] NOT NULL
```

```
) ON [PRIMARY]
```

```
GO
```

```
ALTER TABLE [dbo].[vendedor_has_sala] WITH CHECK ADD CONSTRAINT [FK_vendedor_has_sala_sala]  
REFERENCES [dbo].[sala] ([id])
```

```
GO
```

```
ALTER TABLE [dbo].[vendedor_has_sala] CHECK CONSTRAINT [FK_vendedor_has_sala_sala]
```

```
GO
```

```
ALTER TABLE [dbo].[vendedor_has_sala] WITH CHECK ADD CONSTRAINT [FK_vendedor_has_sala_vendedor]  
REFERENCES [dbo].[vendedor] ([id])
```

```
GO
```

```
ALTER TABLE [dbo].[vendedor_has_sala] CHECK CONSTRAINT [FK_vendedor_has_sala_vendedor]
```

```
GO
```