



Universidade do Minho
Escola de Engenharia
Licenciatura em Engenharia Informática

Unidade Curricular de Laboratórios de Informática IV

Ano Letivo de 2022/2023

UMarket

Gabriela Santos Ferreira da Cunha a97393
João António Redondo Martins a96215
João Pedro Antunes Gonçalves a95019
Miguel de Sousa Braga a97698
Nuno Guilherme Cruz Varela a96455

15 de janeiro de 2023

LI4

Data de Receção	
Responsável	
Avaliação	
Observações	

UMarket

Gabriela Santos Ferreira da Cunha a97393
João António Redondo Martins a96215
João Pedro Antunes Gonçalves a95019
Miguel de Sousa Braga a97698
Nuno Guilherme Cruz Varela a96455

15 de janeiro de 2023

Resumo

Este relatório foi elaborado no âmbito do desenvolvimento de uma aplicação para feiras *online*, capaz de acolher diversos tipos de certames comerciais e possibilitar ao utilizador a escolha do certame a visitar e compras a realizar de acordo com as suas necessidades e preferências. A aplicação permite definir o objetivo de cada feira, os seus diversos *stands* de vendas e os bens e serviços que neles estão a ser comercializados.

Numa primeira fase foi realizada a definição e fundamentação do projeto, onde fundamentamos e projetamos o desenvolvimento do sistema, esclarecendo os objetivos e a viabilidade do projeto.

Em seguida, procedemos à análise e levantamento de requisitos operacionais e funcionais, de forma a produzir uma especificação do sistema capaz de suportar o seu desenvolvimento. Foram, assim, identificados os aspetos estruturais e comportamentais do sistema, maioritariamente representados através de diagramas UML. Em paralelo, foi, ainda, descrito o modelo lógico da base de dados que acolherá toda a informação da plataforma. Com isto, foram também realizados alguns esboços das interfaces.

Por último, passamos à implementação do sistema previamente modelado, com recurso às *frameworks* .NET e Blazor e, ainda, Dapper para a conexão à base de dados que, por sua vez, foi implementada em SQL Server. Devido à sobrecarga de trabalho e ao tempo de execução reduzido, as funcionalidades propostas acabaram por não ser implementadas na totalidade.

Área de Aplicação: Engenharia de *Software*, Desenho e Arquitetura de Bases de Dados, Comércio

Palavras-Chave: Especificação de *Software*, Bases de Dados Relacionais, *Unified Modelling Language*, C#, HTML, CSS, SQL Server

Índice

1	Introdução	1
1.1	Contextualização	1
1.2	Fundamentação	1
1.3	Objetivos	2
1.4	Viabilidade	2
1.5	Recursos a utilizar	3
1.6	Equipa de Trabalho	3
1.7	Plano de Execução do Trabalho	3
1.8	Comentário Final	4
2	Levantamento e Análise de Requisitos	5
2.1	Apresentação da estratégia e método	5
2.2	Descrição geral dos requisitos (funcionais e não funcionais) levantados	6
2.2.1	Registar conta	6
2.2.2	Autenticar utilizador	7
2.2.3	Consultar feiras	7
2.2.4	Consultar catálogos	8
2.2.5	Avaliar produto	8
2.2.6	Consultar produto	9
2.2.7	Consultar lista de produtos favoritos	9
2.2.8	Adicionar produto ao carrinho de compras	10
2.2.9	Consultar carrinho de compras	11
2.2.10	Negociar produto	12
2.2.11	Finalizar compra	12
2.2.12	Fazer registo numa feira	13
2.2.13	Registar e colocar produto à venda	14
2.2.14	Consultar produtos à venda	14
2.2.15	Definir parâmetros de negociação	15
2.2.16	Consultar histórico de vendas	15
2.3	Validação dos requisitos estabelecidos	16
3	Especificação e Modelação do Software	17
3.1	Apresentação geral da especificação	17
3.2	Aspetos estruturais	18
3.2.1	Modelo de domínio	18
3.2.2	Diagrama de componentes	18

3.3	Aspetos comportamentais	19
3.3.1	Casos de uso	19
3.3.2	Diagramas de atividades	23
4	Conceção do Sistema de Dados	26
4.1	Apresentação geral da estrutura (esquema) do sistema de dados	26
4.2	Descrição detalhada dos vários elementos de dados e seus relacionamentos . . .	27
5	Esboço das Interfaces do Sistema	31
5.1	Estrutura geral das interfaces do sistema	31
5.2	Caracterização das interfaces	32
6	Implementação da Aplicação	33
6.1	Processo de implementação	33
6.2	Apresentação dos serviços implementados e estrutura final da aplicação	34
6.3	Análise e avaliação da aplicação desenvolvida	35
6.4	Ferramentas Utilizadas	39
7	Conclusões e Trabalho Futuro	40
	Lista de Siglas e Acrónimos	42
	Anexos	43
I.	Especificação de <i>use cases</i>	43
II.	Mockups adicionais	51
	Cliente	51
	Vendedor	55

Lista de Figuras

1.1	Diagrama de Gantt- Definição e Especificação do sistema.	4
1.2	Diagrama de Gantt-Implementação.	4
3.1	Arquitetura em camadas	17
3.2	Modelo de domínio.	18
3.3	Diagrama de componentes	19
3.4	Diagrama de <i>use cases</i>	20
3.5	<i>Use case</i> “Consultar Produto”	21
3.6	<i>Use case</i> “Adicionar Produto ao Carrinho de Compras”	22
3.7	<i>Use case</i> “Negociar Produto”	23
3.8	Diagrama de atividades de comprar produto(s)	24
3.9	Diagrama de atividades de realizar negociação	24
3.10	Diagrama de atividades de finalizar compra	25
4.1	Esquema lógico do sistema de dados.	26
4.2	Relacionamento favoritos	27
4.3	Relacionamento avaliações	28
4.4	Relacionamento carrinho de compras	28
4.5	Relacionamento entre cliente e compra	29
4.6	Relacionamento entre compra e produto	29
4.7	Relacionamento entre vendedor e feira	30
4.8	Relacionamento entre produto e registo de feira	30
5.1	Transições entre <i>mockups</i>	31
5.2	<i>Mockup</i> de apresentação de um produto	32
5.3	<i>Mockup</i> de negociação de um produto	32
6.1	Página inicial	35
6.2	Página de apresentação de feiras	35
6.3	Página de produtos de uma feira	36
6.4	Página de um produto	36
6.5	Página para registo em uma feira	37
6.6	Página para adicionar produto a uma feira	37
6.7	Página para efetuar o pagamento de uma compra	38
7.1	<i>Use case</i> “Criar feira”	43
7.2	<i>Use case</i> “Autenticar utilizador”	43

7.3	Use case “Realizar compra”	44
7.4	Use case “Registar utilizador”	45
7.5	Use case “Consultar detalhes de feira”	45
7.6	Use case “Atualizar informação do produto”	46
7.7	Use case “Registar produto”	47
7.8	Use case “Definir parâmetros de negociação”	48
7.9	Use case “Avaliar vendedor”	49
7.10	Use case “Fazer registo numa feira”	49
7.11	Use case “Consultar catálogo de uma feira”	50
7.12	Use case “Adicionar produto aos favoritos”	51
7.13	Mockup de registo	51
7.14	Mockup de autenticação	52
7.15	Mockup de apresentação das feiras	52
7.16	Mockup de apresentação do catálogo de uma feira	53
7.17	Mockup de apresentação do carrinho de compras	53
7.18	Mockup de finalização de compra	54
7.19	Mockup de apresentação dos favoritos	54
7.20	Mockup de seleção vendedor	55
7.21	Mockup de seleção de feira para o vendedor	55
7.22	Mockup de adição de produto para venda para o vendedor	56
7.23	Mockup de consulta de produto à venda para o vendedor	56
7.24	Mockup de edição de produto à venda para o vendedor	57
7.25	Mockup de consulta do relatório de vendas para o vendedor	57

Lista de Tabelas

1.1	Elementos que compõem a equipa de desenvolvimento.	3
6.1	Distribuição do grupo pelos componentes da implementação.	33

1 Introdução

1.1 Contextualização

A feira “Ponte dos Arcos Verdes” é um evento semanal com enorme tradição na região minhota de Portugal. Acontece às quartas-feiras e conta com a presença de milhares de pessoas de todo o mundo. Este acontecimento realiza-se alternadamente nos concelhos de Arcos de Valdevez, Ponte de Lima e Vila Verde. As suas origens remontam há cerca de um milénio e marcam o início de uma história sem precedentes no contexto das feiras nacionais. O ambiente é marcado pelas constantes e bem audíveis negociações entre clientes e vendedores. Todo o tipo de pessoas podem participar, quer passeando pelos longos corredores de barracas repletas de todo o tipo de produtos, quer montando o seu próprio posto de venda. Contudo, o surgimento da pandemia nos últimos 2 anos, reduziu de forma drástica a quantidade de pessoas que se reunia para celebrar e beneficiar deste grandioso evento. Efetivamente, o número médio de visitantes teve uma quebra de, aproximadamente, 50% nos últimos 2 anos, reduzindo de 20 mil para 10 mil o número de pessoas que normalmente visitam este espaço semanalmente. Apesar da situação se ter vindo a tornar mais estável, com a diminuição do número de infeções e, por consequência do número de confinamentos, os municípios receiam que a situação se possa voltar a agravar no futuro.

1.2 Fundamentação

Com o aumentar da popularidade deste evento no passado ano pós-pandémico, o número de pessoas que vêm participar tem crescido de forma exponencial, o que tem revelado falhas na organização e motivado queixas por parte dos utilizadores. Para além disso, o facto de se realizar em três espaços físicos diferentes tem criado bastantes problemas logísticos e tem aumentado as preocupações de alguns participantes, nomeadamente os vendedores que têm que deslocar os seus produtos pelos três concelhos e os clientes que perdem a facilidade de poderem comprar produtos num local mais próximo. Assim, de modo a sustentar o crescimento e resolver os diversos problemas existentes, os municípios chegaram à conclusão que um sistema *online* seria o caminho para o futuro. Este sistema deveria promover o bem-estar de todos os participantes, garantindo uma maior conveniência tanto na hora de vender como na hora de comprar produtos. Em particular, possibilitaria um meio alternativo aos clientes com problemas de locomoção, sejam aqueles mais debilitados fisicamente, sejam aqueles que vivem nas aldeias mais remotas para participar neste evento. Para além disso, este produto teria também como

alvo o público mais jovem, geralmente menos visto neste tipo de espaços físicos. Apesar da grande expectativa em torno deste novo sistema, os municípios não tencionam substituir o modelo presencial atualmente em prática.

1.3 Objetivos

Através de diversos contactos com as entidades superiores, estabeleceram-se os seguintes objetivos:

- Possibilidade dos participantes comprarem os produtos que necessitam, sem se terem de deslocar ao espaço físico.
- Aumentar o número de participantes na feira.
- Organizar o modelo de negócio, guardando registos de compras/vendas, melhorando a capacidade de gestão.
- Possibilitar aos clientes uma forma de saber qual a feira mais próxima onde os produtos que necessitam estão disponíveis.
- Aumentar a credibilidade da feira, dos vendedores e dos seus produtos, através da avaliação das compras por parte dos clientes.
- Tornar a aplicação acessível e fácil de usar para um utilizador comum.

1.4 Viabilidade

De acordo com os recursos a utilizar, a equipa de desenvolvimento acredita que o projeto estará concluído dentro dos prazos, pelo que o seu desenvolvimento será viável. Com este novo modelo de negócios, acredita-se que os municípios conseguirão recuperar, na época de pico de afluência, cerca de 10% do prejuízo de vendas do ano anterior, ano este em que a feira se realizou com limitações em parte do mesmo devido à situação pandémica. Os municípios poderão também ter acesso ao relatório de contas de cada dia de feira, de modo a definir estratégias de vendas que garantem maior retorno financeiro. De modo a aumentar o número de participantes, haverá uma melhor organização dos temas das feiras em função da época do ano e, na eventualidade de uma nova situação pandémica, esta aplicação poderá vir a ser o único meio de negociação.

1.5 Recursos a utilizar

De forma a que o processo de implementação, estruturação e desenvolvimento da aplicação seja bem sucedido, são necessários diversos recursos. A nível de recursos humanos serão necessários clientes e vendedores, para fazer uso da aplicação, bem como um analista de testes, um administrador de base de dados, dois programadores e engenheiros de *software* responsáveis pela criação e manutenção da aplicação. Como suporte do sistema a desenvolver, será necessário um recurso de *software* fundamental - uma base de dados relacional - e um servidor de teste (recurso material).

1.6 Equipa de Trabalho

A partir dos recursos referidos na secção anterior, apresentamos, assim, uma tabela representativa dos membros e respetivas funções da equipa responsável pelo desenvolvimento e manutenção do *software*.

Nomes	Área de Trabalho
Gabriela Santos Ferreira da Cunha	Front-end Developer
João António Redondo Martins	Administrador de Base de Dados
João Pedro Antunes Gonçalves	Analista de Testes
Miguel de Sousa Braga	Back-end Developer
Nuno Guilherme Cruz Varela	Engenheiro de Software

Tabela 1.1: Elementos que compõem a equipa de desenvolvimento.

A estes elementos da empresa de desenvolvimento adicionamos também uma amostra de clientes e vendedores, responsáveis por validar e testar as funcionalidades do sistema e um representante de cada município, responsável por acompanhar o desenvolvimento do projeto e validar cada uma das fases do seu desenvolvimento.

1.7 Plano de Execução do Trabalho

Este projeto será dividido em duas fases principais - a definição e especificação do sistema e a implementação deste. De forma a distribuir o trabalho por toda a equipa e organizar a execução deste projeto, apresentamos o plano de execução realizado no Diagrama de Gantt seguinte.

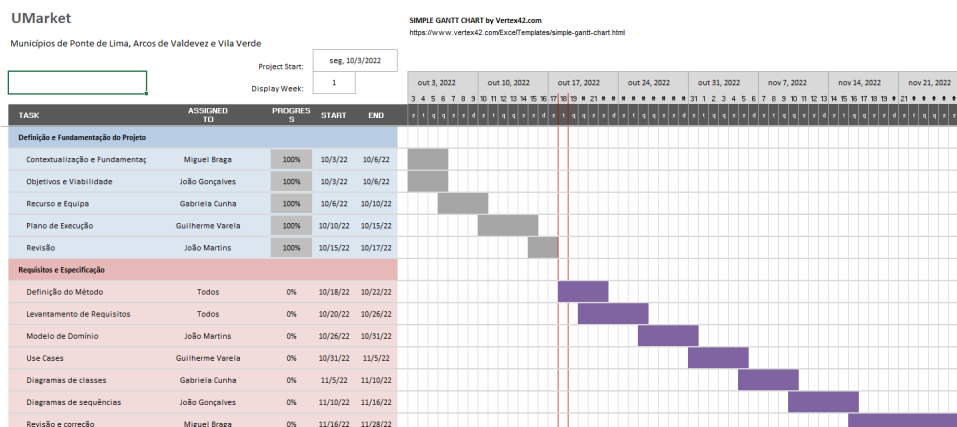


Figura 1.1: Diagrama de Gantt- Definição e Especificação do sistema.

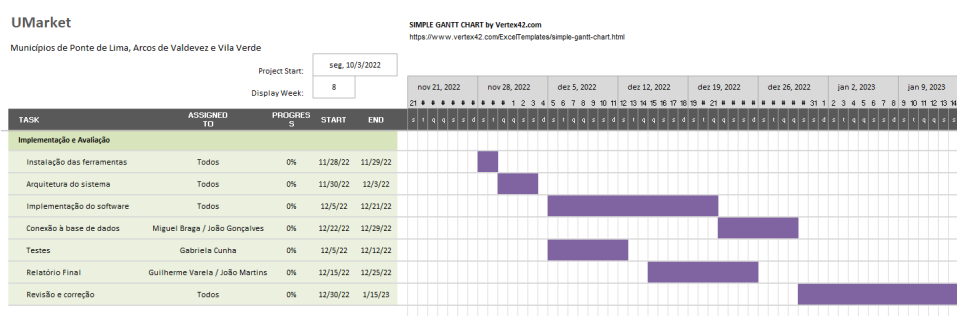


Figura 1.2: Diagrama de Gantt-Implementação.

1.8 Comentário Final

Após a validação do que foi dito com as entidades envolvidas (*stakeholders*), avançamos para a parte da especificação, onde definimos os requisitos que este sistema deverá suportar. Estes terão por base o conjunto de objetivos estipulados anteriormente, pelo que esta definição e fundamentação foi um passo importante e necessário tendo em vista o processo de desenvolvimento bem estruturado da nossa aplicação.

2 Levantamento e Análise de Requisitos

Nesta segunda fase de desenvolvimento do produto “UMarket“, começamos por realizar o levantamento e a análise de requisitos. Numa primeira parte, definimos a estratégia e o método de levantamento dos requisitos. Em seguida, procedemos ao levantamento propriamente dito destes requisitos, com base nos métodos estabelecidos.

2.1 Apresentação da estratégia e método

Assim, de modo a estudar as funcionalidades esperadas para este *software*, definimos as seguintes tarefas, essenciais para conhecermos em detalhe que operações a nossa aplicação deve suportar:

- Visita a um contexto de feira, com observação dos processos de compra e venda entre comerciantes e clientes;
- Entrevistas a vendedores e a clientes sobre quais os aspetos que poderiam ser melhorados no ambiente das feiras;
- Análise de *software* desenvolvido no âmbito de problemas semelhantes (Ebay, OLX, entre outros), uma vez que nos permite estar a par da concorrência;
- Reuniões com os 3 municípios de modo a apurar funcionalidades relevantes para a implementação da aplicação;
- Questionários de opinião aos clientes (sobretudo os mais jovens), procurando saber os seus interesses no contexto geral das feiras, de modo a estimular a sua participação neste tipo de eventos.

2.2 Descrição geral dos requisitos (funcionais e não funcionais) levantados

Através dos métodos referidos previamente, pudemos levantar vários requisitos funcionais. Um requisito funcional é uma declaração de como o sistema deve se comportar, isto é, define o que o sistema deve fazer para atender às necessidades ou expectativas do utilizador. Em contrapartida, um requisito não funcional explica as limitações e restrições do sistema, ou seja, descreve como o sistema deverá fazer para atender às necessidades ou expectativas do utilizador. Em seguida, apresentamos os requisitos levantados.

2.2.1 Registar conta

Utilizador

- Deve poder registar-se como cliente ou como vendedor.
- Insere o nome (Nome Próprio e Apelido), endereço de email, palavra-passe e NIF.
- Deve ser alertado quando tenta criar uma nova conta, cujo NIF já exista no sistema.
- Após ter sido efetuado o registo será imediatamente autenticado.

Sistema

- Deve apresentar as opções “Cliente” e “Vendedor” para o utilizador seleccionar o papel que pretende adquirir.
- Fornece uma operação de “Adicionar Utilizador”, em que é criada uma nova entrada contendo obrigatoriamente os atributos “Nome”, “Endereço de Email”, “NIF” e “Palavra-Passe”
- Deve verificar se o NIF do novo utilizador está já associado a uma conta registada.
- Deve guardar o role de cada utilizador como um atributo do registo ‘Utilizador’.
- Terá de encriptar a palavra-passe quando esta é armazenada na base de dados.

2.2.2 Autenticar utilizador

Utilizador

- Não estando autenticado, terá acesso a um botão 'Autenticar', que lhe apresenta uma nova página onde deverá inserir os dados de autenticação.
- Deve inserir os seus dados de *login* (*email* e palavra-passe).
- Estando autenticado como cliente, terá acesso aos seus dados de perfil assim como às funcionalidades de compra de produtos.
- Estando autenticado como vendedor, terá acesso às funcionalidades de registar-se numa feira e de registar produto numa feira.

Sistema

- Deve fornecer campos em que o utilizador insere o *email* e a *password* de acesso.
- Será capaz de validar se os dados fornecidos correspondem a alguma conta já registada.

2.2.3 Consultar feiras

Utilizador

- Pode consultar as feiras.
- Tem a possibilidade de consultar o catálogo de uma feira específica.
- Não necessita de estar autenticado para consultar as feiras.
- Pode consultar os dados (tema, descrição, vendedores) sobre as diversas feiras em funcionamento.

Sistema

- As informações dos detalhes das feiras estão armazenadas na base de dados.
- Deve ser possível consultar os registos da base de dados referentes à feira indicada.

2.2.4 Consultar catálogos

Utilizador

- Tem acesso aos produtos que constituem os diversos catálogos das feiras em funcionamento, não precisando de estar registado para ter esta funcionalidade.
- Pode filtrar os resultados por preço, categoria ou vendedor.

Sistema

- Deve ter acesso aos registos dos produtos de cada feira na base de dados.
- Para cada produto, deve ser possível saber o nome do produto, preço, quantidade em *stock*, descrição, categoria, avaliação e vendedor.

2.2.5 Avaliar produto

Utilizador

- Tem de estar autenticado para avaliar um produto.
- Se não tiver nenhuma compra registada nesse produto e tentar avaliá-lo, não deverá poder avaliar e recebe uma mensagem de aviso.
- Tem a possibilidade de avaliar um produto numa escala de 1 a 5.
- Tem a possibilidade de alterar/remover a sua avaliação.
- Só pode ter uma avaliação para cada produto.

Sistema

- Deve validar se o utilizador está autenticado para permitir o acesso a esta funcionalidade.
- Deve armazenar as avaliações realizadas por cada cliente, para cada produto na base de dados.
- Não deve permitir que o mesmo cliente realize várias avaliações ao mesmo produto.
- Deve restringir as avaliações a clientes que tenham comprado o produto.

- Deve permitir adicionar/atualizar avaliações a um dado produto.

2.2.6 Consultar produto

Utilizador

- Pode consultar o nome do produto, preço, quantidade em stock, descrição, categoria, avaliação, vendedor e a imagem do produto.
- Pode adicionar o produto ao carrinho de compras.
- Se já avaliou o produto, pode consultar a sua avaliação.
- Pode avaliar o produto.
- Se já adicionou o produto aos favoritos, deve ser informado disso.
- Se já adicionou o produto aos favoritos, deve poder removê-lo dessa lista.
- Se não adicionou o produto aos favoritos, pode adicionar o produto aos favoritos.

Sistema

- Os registos dos produtos têm de estar armazenados na base de dados.
- O sistema deverá saber o nome, preço, quantidade em *stock*, descrição, categoria, avaliação, imagem associada e identificação do vendedor de um produto.
- O sistema deverá saber a avaliação de cada cliente.
- O sistema deverá saber, para cada cliente, se esse produto está nos seus favoritos.
- Deve permitir adicionar e remover um novo produto à lista dos favoritos de cada utilizador.

2.2.7 Consultar lista de produtos favoritos

Utilizador

- Tem de estar autenticado para usufruir desta funcionalidade.
- Ao seleccionar a opção "Consultar favoritos" do menu de utilizador, tem acesso à lista dos seus produtos favoritos.

- Pode consultar a imagem, o nome, o preço e a descrição de cada produto favorito.
- Pode remover produtos da lista de favoritos.
- Ao clicar na imagem do produto será redirecionado para a página principal do produto.

Sistema

- Deve validar se o utilizador está autenticado para permitir o acesso a esta funcionalidade.
- Deve ter acesso à lista de produtos favoritos de um dado utilizador.
- Deve permitir remover produtos da lista de favoritos de cada utilizador.

2.2.8 Adicionar produto ao carrinho de compras

Utilizador

- Tem de estar autenticado para usufruir desta funcionalidade.
- Ao consultar um dado produto, pode adicioná-lo ao carrinho de compras.
- Tem de definir a quantidade desse produto que quer comprar.
- Caso produto já esteja no carrinho, o preço não pode ser novamente negociado.
- Caso o produto já esteja no carrinho, a quantidade desse produto é incrementada.
- O cliente deve saber se um dado produto pode ser ou não negociado.
- Pode escolher entre negociar ou não o preço unitário do produto antes de o adicionar ao carrinho de compras.

Sistema

- Deve validar se o utilizador está autenticado para permitir o acesso a esta funcionalidade.
- Para cada cliente, deve armazenar o produto e a quantidade desse produto guardados no carrinho de compras.

- Quando um cliente já tem esse produto adicionado ao carrinho de compras, o sistema deve atualizar a quantidade do produto.
- Pode adicionar registros de compra ao carrinho de compras de cada utilizador.
- Deve apenas permitir que sejam negociados produtos em que estejam definidos os parâmetros de negociação.
- Deve permitir que o utilizador negocie o preço de um produto.

2.2.9 Consultar carrinho de compras

Utilizador

- Tem de estar autenticado para usufruir desta funcionalidade.
- Pode consultar a lista de produtos do seu carrinho de compras.
- Para cada produto no carrinho, deve ser apresentado o nome, a descrição, o preço negociado e a quantidade.
- Pode remover um produto do carrinho de compras.
- Tem acesso ao preço total do carrinho.
- Estando satisfeito com os produtos que tem no carrinho, pode finalizar a compra.

Sistema

- Deve validar se o utilizador está autenticado para permitir o acesso a esta funcionalidade.
- Deve armazenar/ter acesso aos produtos do carrinho de compras de cada utilizador.
- Para cada produto, deve ser armazenado o preço negociado por cada cliente e a quantidade que este deseja comprar.
- Deve calcular o custo total da compra, somando os preços negociados de todos os produtos do carrinho.
- Deve permitir finalizar a compra do cliente.

2.2.10 Negociar produto

Utilizador

- Tem de ter escolhido adicionar um produto ao carrinho de compras para poder usufruir desta funcionalidade.
- Para cada iteração da negociação, o cliente deve saber o valor proposto pelo sistema (algoritmo) e o histórico dos valores propostos.
- Para cada iteração da negociação, o cliente indica o valor que está disposto a atribuir ao produto.
- Ao propor um valor demasiado baixo, o cliente sujeita-se a que a negociação fracasse.
- Ao propor um valor próximo da proposta do sistema, o cliente sujeita-se a que a negociação seja concluída, ficando encontrado o preço final para o produto.

Sistema

- O sistema deverá ter acesso aos parâmetros de negociação definidos pelo vendedor para esse produto.
- Para cada valor do cliente, o sistema deve gerar uma resposta.
- Essa resposta poderá indicar o cancelamento da negociação, o aceitar da negociação ou a proposta de um novo valor (limite superior do intervalo de negociação).
- Para cada iteração da negociação, deverá guardar os limites do intervalo de negociação.
- Sempre que o utilizador vai propor um novo valor, o sistema deve calcular o limite a partir do qual esse valor não será aceite (fator de tolerância).
- Sempre que o utilizador vai propor um novo valor, o sistema deve calcular o limite a partir do qual o valor proposto será aceite (fator de aceitação).

2.2.11 Finalizar compra

Utilizador

- Para finalizar compra, deverá ter criado um carrinho de compras e selecionado a opção "Finalizar compra".
- Tem de inserir o nome de faturação, a morada e o número de telemóvel.

- Deve escolher entre uma das diversas formas de pagamento disponíveis (MBWay, Transferência MB, Cartão de Crédito).
- Caso selecione a opção "MBWay", deverá inserir o seu número de telemóvel.
- Caso selecione a opção "Transferência MB", insere o nome e IBAN da conta de destino.
- Caso seleciona a opção "Cartão de Crédito", insere o número de cartão de crédito e o código de segurança.
- Deverá ser notificado se o stock disponível não permitir a compra.

Sistema

- Deve apresentar as diversas formas de pagamento (MBWay, Transferência MB, Cartão de Crédito).
- Caso o pagamento seja por MBWay, deverá pedir o número de telemóvel para pagamento.
- Caso o pagamento seja por "Transferência MB", deverá pedir o nome e o IBAN da conta de destino.
- Caso o pagamento seja por "Cartão de Crédito", deverá pedir o número de cartão de crédito e o código de segurança.
- Deverá ser capaz de validar se a compra pode ser realizada com o stock disponível.
- Deverá ser capaz de receber múltiplas operações de finalizar compra dos vários clientes.

2.2.12 Fazer registo numa feira

Utilizador

- Os vendedores podem inscrever-se numa dada feira, alugando um espaço virtual que lhes permite colocar à venda os seus produtos.

Sistema

- Quando é feito um registo, o sistema deve adicionar esse vendedor à lista de vendedores presentes nessa feira.

2.2.13 Registrar e colocar produto à venda

Utilizador

- O vendedor regista um produto para venda definindo o seu nome, descrição, preço, quantidade disponível, categoria do mesmo e parâmetros de negociação.
- O vendedor pode escolher se esse produto pode ser negociado ou não.
- Os vendedores podem colocar à venda os seus produtos nas feiras em que estão envolvidos.
- O vendedor tem a funcionalidade de alterar a informação de um produto.

Sistema

- Quando é colocado um produto à venda, o sistema adiciona esse produto à lista de produtos vendidos pelo mesmo, na feira em questão.
- O sistema permite alterar as informações dos produtos, alterando o seu registo.
- O sistema permite remover um produto da lista de produtos vendidos por ele.
- Não devem existir produtos com o mesmo nome na mesma feira.

2.2.14 Consultar produtos à venda

Utilizador

- O vendedor deverá ter acesso aos produtos que coloca à venda.
- Pode consultar todas as feiras onde tem produtos.
- Para cada feira pode consultar todos os produtos que registou.
- O vendedor para cada produto tem a opção de removê-lo da respetiva feira.

Sistema

- Deve validar se o utilizador está autenticado como vendedor.
- Deve ter acesso às feiras que cada vendedor participa.

- Deve ter acesso aos produtos registrados pelo vendedor nas feiras.

2.2.15 Definir parâmetros de negociação

Utilizador

- O vendedor tem a possibilidade de indicar ou não os parâmetros de negociação de um produto, conforme ele queira que este seja ou não negociável.
- Se o vendedor pretender que o produto seja negociável, tem de indicar os fatores de tolerância, aceitação e resposta.

Sistema

- O sistema adiciona esses parâmetros de negociação ao registo do produto.

2.2.16 Consultar histórico de vendas

Utilizador

- O vendedor pode consultar o seu histórico de vendas.
- O vendedor pode consultar as vendas dos últimos 6 meses.
- O vendedor tem a possibilidade de definir um intervalo de tempo customizado.
- O vendedor deve ter acesso ao valor monetário total recebido durante esse período.

Sistema

- Para cada compra, deve armazenar o timestamp associado.
- Deve saber calcular a soma dos valores das compras associadas a um vendedor, durante um período de tempo.
- Deve conseguir encontrar as vendas realizadas por um vendedor durante um intervalo de tempo.

2.3 Validação dos requisitos estabelecidos

Após alguns esclarecimentos e após algumas conversas entre a equipa de desenvolvimento e os *stakeholders* da nossa aplicação, ambas as partes concordaram que esta devia ser a base de partida para o desenvolvimento da nossa aplicação, pelo que seguiu-se o processo de especificação e modelação do software que nos foi proposto desenvolver.

3 Especificação e Modelação do Software

Após o levantamento dos requisitos funcionais e não funcionais e a validação dos mesmos, a equipa de desenvolvimento procedeu à especificação da aplicação.

3.1 Apresentação geral da especificação

Aqui seguimos uma abordagem orientada aos requisitos, de modo a conseguir identificar, para cada modelo apresentado, quais o(s) requisito(s) base que o originaram. Decidimos também seguir com uma abordagem a alto nível, não entrando em grandes detalhes de implementação. Assim, nos aspetos estruturais demos preferência, nesta fase, a modelos mais abstratos como modelo de domínio e diagrama de componentes. Nos aspetos comportamentais, para cada requisito elaboramos o(s) use case(s) correspondente(s), acrescentando alguns diagramas de atividade e máquinas de estados para modelar comportamentos específicos do sistema.

De uma forma abstrata, representada pela figura 3.1, a nossa solução será dividida em 3 camadas: interface com o utilizador, camada da lógica de negócio e camada de dados.

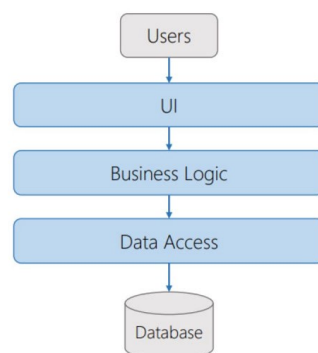


Figura 3.1: Arquitetura em camadas

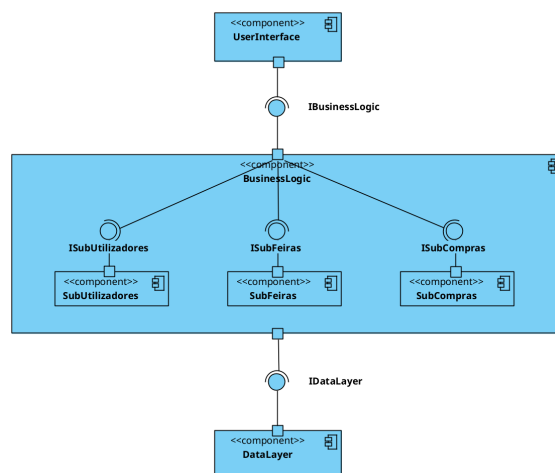


Figura 3.3: Diagrama de componentes

3.3 Aspetos comportamentais

3.3.1 Casos de uso

De modo a explicitar o comportamento do sistema face a cenários criados através dos requisitos indicados anteriormente, recorreremos ao desenvolvimento de casos de uso e ao respetivo diagrama. Um *use case* descreve as sequências de interações entre atores externos e um sistema, para que um dado cenário de utilização seja suportado. Assim, nesta secção apresentaremos os *use cases* que levantamos através dos requisitos definidos e a especificação dos que consideramos relevantes para a modelação das interações entre o cliente e o sistema. As restantes especificações efetuadas encontram-se presentes em anexo.

O primeiro passo passou pela identificação dos atores do sistema. Nesta aplicação, existem interações entre clientes e vendedores, ambos utilizadores. Existe também um gestor do sistema que, por sua vez, é também um utilizador. Para representar as funcionalidades das quais estes utilizadores podem usufruir, realizamos o seguinte diagrama da figura 3.4.

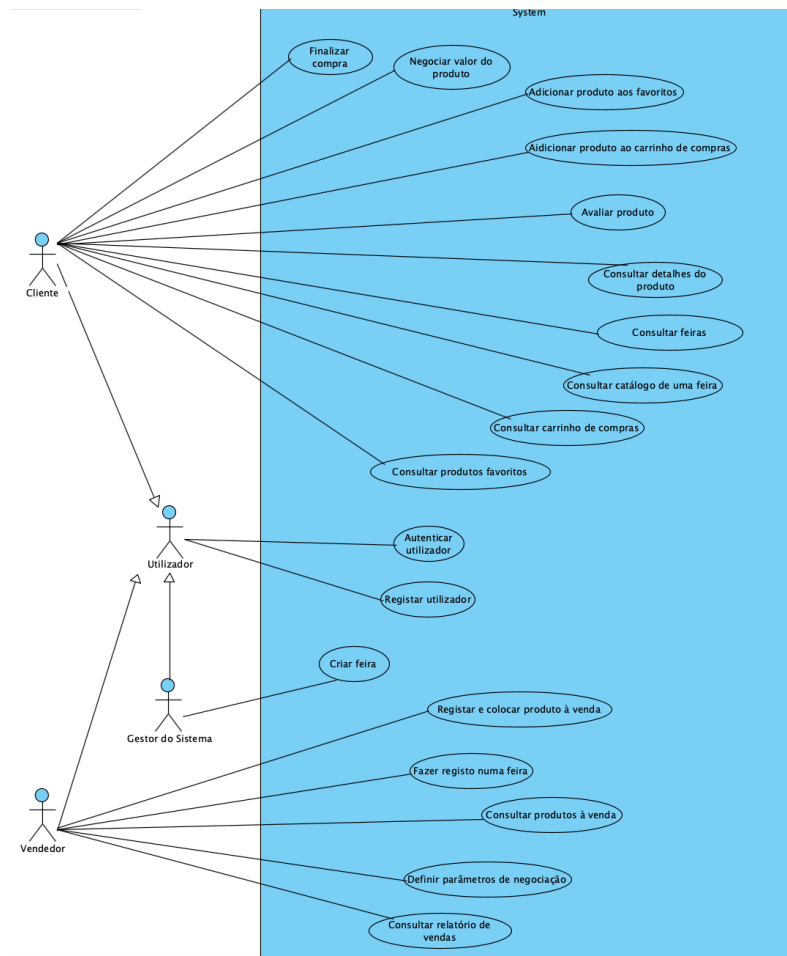


Figura 3.4: Diagrama de *use cases*

Em seguida, apresentamos os exemplos de *use cases* que consideramos relevantes em função das funcionalidades levantadas: “Consultar detalhes do produto”, “Adicionar produto ao carrinho de compras” e “Negociar valor do produto”.

Consultar Produto

O *use case* “Consultar produto” exprime todas as funcionalidades que um cliente tem disponível ao consultar um dado produto. Assim, sempre que seleciona um produto, o cliente tem disponível dados sobre o mesmo como nome, preço, descrição, categoria, vendedor, classificação global, quantidade estimada em *stock* e, possivelmente, imagem, se estiver disponível. Estando autenticado, recebe ainda informações sobre a sua própria avaliação do produto e é apresentado se o mesmo está ou não nos favoritos. Após ter consultado as informações necessárias, o cliente pode tomar várias decisões: avaliar o produto, adicionar/remover produto dos favoritos ou adicionar produto ao carrinho de compras.

Use Case	Consultar detalhes do produto	
Ator	Cliente	
Cenários		
Descrição	O cliente consulta um determinado produto.	
Pré-Condição	Cliente selecionou um produto.	
Pós-Condição	O produto é visualizado.	
	Ator	Sistema
Fluxo Normal		1. Apresenta imagem, nome, preço, descrição, categoria, vendedor, avaliação global e quantidade em stock.
	2. Seleciona adicionar produto ao carrinho de compras.	
		3. <<include>> Adicionar produto ao carrinho de compras.
Fluxo Alternativo [Cliente está autenticado] (passo 1)		1.1 Apresenta imagem, nome, preço, descrição, categoria, vendedor, avaliação global e quantidade em stock.
		1.2 Verifica que possui uma avaliação para o produto.
		1.3 Apresenta avaliação do cliente.
		1.4 Verifica que o produto está nos seus favoritos.
		1.5 Apresenta produto como favorito.
		1.6 Regressa a 2.
Fluxo Alternativo [Não possui avaliação] (passo 1.2)		1.2.1 Indica que a avaliação não existe.
		1.2.2 Regressa a 1.4.
Fluxo Alternativo [Não está nos favoritos] (passo 1.4)		1.4.1 Indica que o produto não está nos favoritos.
		1.4.2 Regressa a 1.6.
Fluxo Alternativo [Cliente quer avaliar produto] (passo 2)	2.1 Seleciona a opção de avaliar produto.	
		2.2 Valida que o cliente está autenticado.
		2.3 <<include>> Avaliar produto.
Fluxo Alternativo [Cliente quer adicionar/remover produto aos favoritos] (passo 2)	2.1 Seleciona a opção de adicionar/remover produto aos favoritos.	
		2.2 Valida que o cliente está autenticado.
		2.3 Altera estado de favorito desse produto para o cliente.
Fluxo Alternativo [Cliente não está autenticado] (passo 2.2)		2.2.1 <<include>> Autenticação
		2.2.2 Regressa a 2.3

Figura 3.5: Use case “Consultar Produto”

Adicionar Produto ao Carrinho de Compras

Caso opte pela opção “Adicionar produto ao carrinho de compras” na página relativa à funcionalidade anterior, o cliente terá acesso a um pequeno menu, onde poderá selecionar se pretende ou não realizar negociação. Pretendendo negociar o valor do produto, o sistema terá, ainda assim, de validar se o mesmo é negociável (ou seja, se o vendedor definiu os parâmetros de negociação para aquele produto). Não sendo negociável, o cliente poderá decidir cancelar a compra desse produto, constituindo esse um fluxo de exceção.

Use Case	Adicionar produto ao carrinho de compras	
Ator	Cliente	
Descrição	O cliente adiciona um produto ao carrinho	
Pré-Condição	O cliente selecionou um produto.	
Pós-Condição	Um novo produto é adicionado ao carrinho de compras	
	Ator	Sistema
Fluxo Normal		1. Apresenta opções de negociar preço ou adicionar diretamente ao carrinho.
	2. Indica que não quer negociar o valor estipulado.	
		3. Adiciona o produto ao carrinho.
Fluxo Alternativo [Cliente realiza negociação] (passo 2)	2.1. <<include>> Realizar negociação	
		2.2. Volta a 3.
Fluxo Alternativo [Produto não é negociável] (passo 2)		2.1. Indica que o preço do produto é fixo.
		2.2. Pergunta se pretende mesmo assim adicionar o produto.
	2.3. Indica que pretende adicionar produto ao carrinho.	
		2.4 Adiciona produto ao carrinho do cliente.
Fluxo de Exceção [Cliente não pretende adicionar ao carrinho sem negociar] (passo 2.3.)	2.3.1. Indica que não pretende adicionar ao carrinho.	
		2.3.1 Cancela operação.

Figura 3.6: Use case “Adicionar Produto ao Carrinho de Compras”

Negociar Produto

O use case “Negociar valor do produto” exprime as várias interações possíveis entre o cliente e o sistema (a representar o vendedor) quando é decidido realizar uma negociação. Deste modo, o sistema começa por apresentar o valor original do produto, valor que o cliente pretende negociar, pelo que este atribui um novo valor ao produto. Em seguida, o sistema poderá: aceitar o valor do cliente - se este for próximo do valor inicial; cancelar a negociação - caso o valor proposto pelo cliente for demasiado baixo; continuar a negociação - indicando uma resposta. Os critérios que definem o comportamento do sistema vão ser mais à frente explicados. Caso o sistema opte por continuar a negociação, o cliente pode aceitar ou não o valor do produto. No primeiro caso, a negociação deverá continuar com a resposta do sistema, até se chegar a um acordo ou falha.

Use Case	Negociar valor do produto	
Ator	Cliente	
Descrição	O cliente realiza a negociação do preço de um determinado produto	
Pré-Condição	O cliente selecionou um produto e a opção "Negociar preço"	
Pós-Condição	O cliente adiciona o produto ao seu carrinho de compras com o novo preço negociado.	
	Ator	Sistema
Fluxo Normal		1. Apresenta o valor original do produto.
	2. Indica o preço que atribui ao produto.	
		3. Calcula a resposta consoante o valor do cliente.
	4. Aceita o novo valor.	
		5. Regista a compra com o preço acordado.
Fluxo Alternativo [Valor proposto pelo cliente é aceite] (passo 3)		3.1. Volta a 5.
Fluxo de Exceção [Valor proposto é demasiado baixo e negociação é terminada] (passo 3)		3.1. Informa que as negociações estão terminadas.
		3.2. Cancela a compra do cliente.
Fluxo Alternativo [O cliente não aceita o valor proposto] (Passo 4)	4.1. Propõe novo valor.	
	4.2. Volta a 3.	

Figura 3.7: Use case “Negociar Produto”

3.3.2 Diagramas de atividades

Para especificar o comportamento do sistema no momento da compra de um ou mais produtos, elaboramos 3 diagramas de atividades. O primeiro diagrama representa o processo de compra superficialmente, o segundo aprofunda o processo de negociação que pode ou não ocorrer durante a compra e o terceiro aborda a finalização da compra.

O primeiro diagrama, da figura 3.8, representa todo o processo de fazer uma compra, desde a consulta dos produtos até à finalização da mesma. Para efetuar uma compra, o cliente deve começar por adicionar um produto ao carrinho de compras, podendo tentar efetuar uma negociação sobre este produto, isto se o vendedor disponibilizar negociação sobre o mesmo. Em seguida, poderá proceder à finalização da compra, onde vai inserir os dados pessoais e de pagamento, ou atrasar a mesma, adicionando novos produtos ao carrinho.

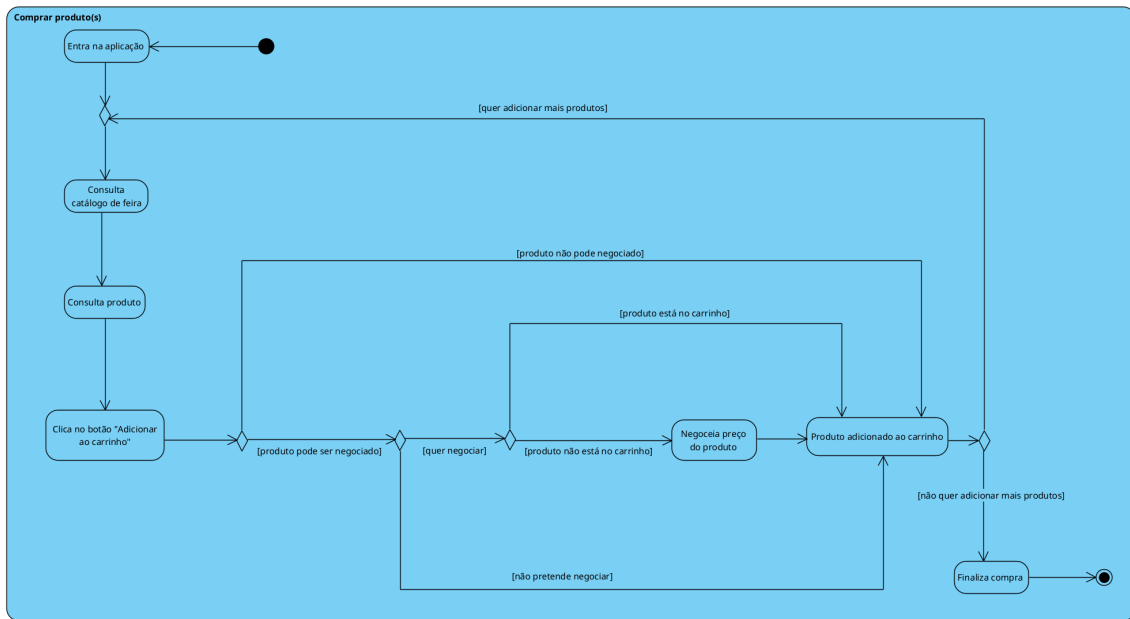


Figura 3.8: Diagrama de atividades de comprar produto(s)

Durante a compra de um produto, surge uma das funcionalidades mais importantes da nossa aplicação que é a de permitir negociações do valor do produto entre um cliente e o sistema (representando o vendedor). Deste modo, para permitir um maior conhecimento do algoritmo de negociação de um dado produto, procedemos à realização do diagrama de atividades da figura 3.9.

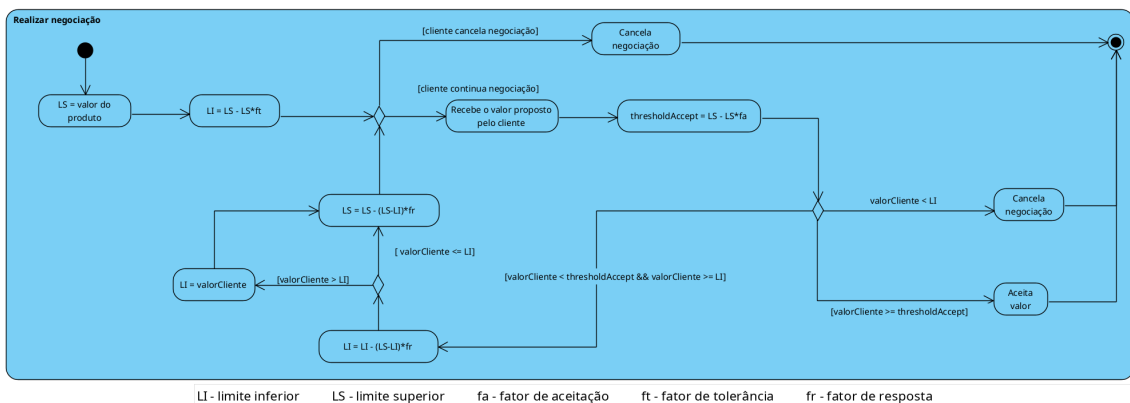


Figura 3.9: Diagrama de atividades de realizar negociação

Após o utilizador decidir finalizar a sua compra, o sistema deve verificar se há *stock* disponível de todos os produtos requeridos e, em caso afirmativo, atualizar a informação, decrementando

essas mesmas quantidades. Para isto, é necessário controle de concorrência para evitar estados de incoerência. Em seguida, o sistema valida os dados de pagamento e deve registrar a compra na base de dados. Caso haja uma falha na validação ou o *stock* de algum produto não seja suficiente, a compra deve ser imediatamente cancelada.

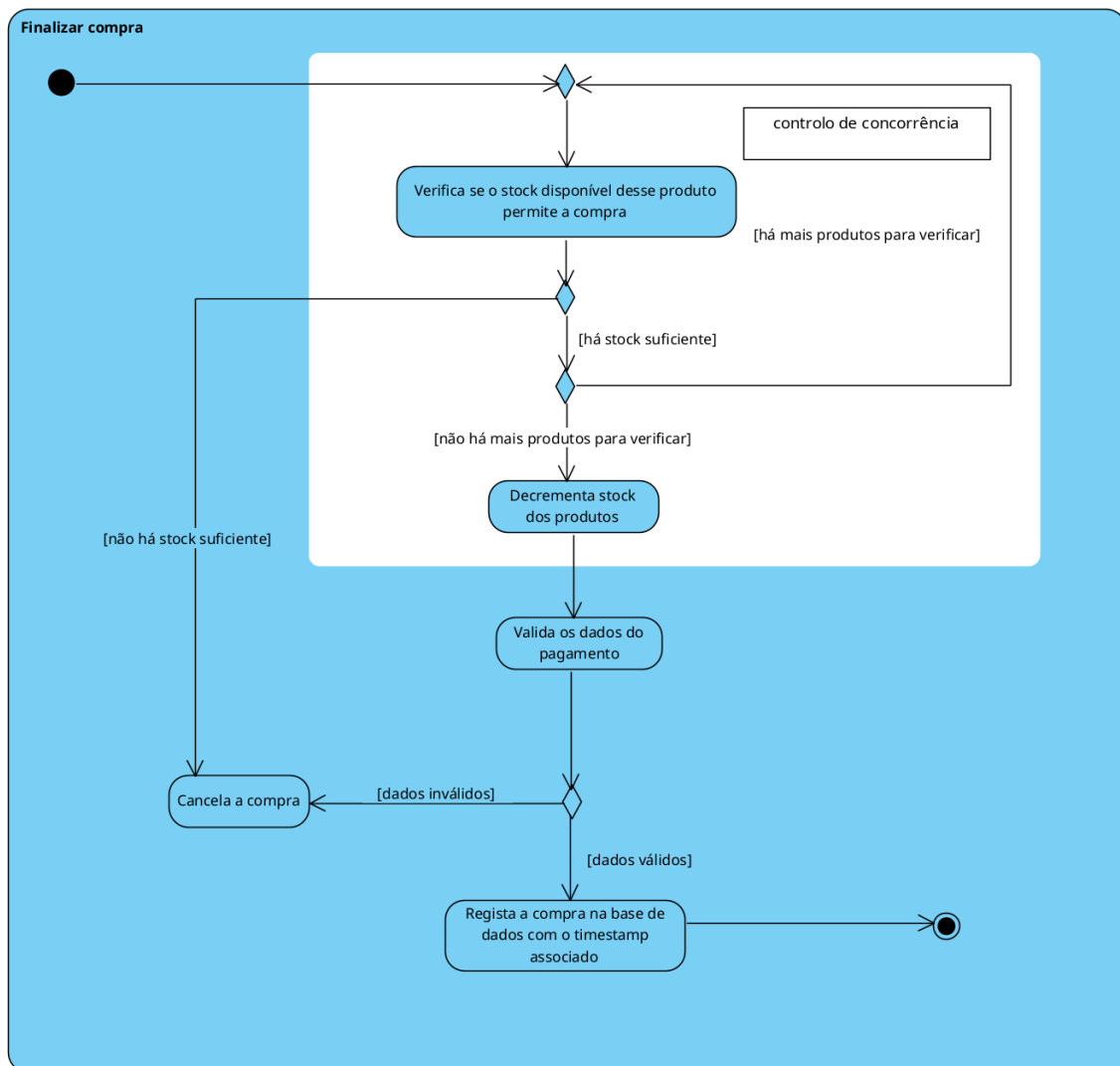


Figura 3.10: Diagrama de atividades de finalizar compra

4 Conceção do Sistema de Dados

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

Para a conceção do sistema de dados, decidiu-se que a base de dados a desenvolver seria relacional, tendo em conta que este tipo permite armazenar e aceder a dados de uma forma eficiente e fléxivel. Com uma base de dados relacional, os dados são organizados em tabelas que contêm informações sobre cada entidade e representam categorias pré-definidas através de linhas e colunas. Esta forma de organização de dados é o motivo pelo qual as bases de dados relacionais são as mais comuns.

Por conseguinte, apresentamos o modelo lógico implementado na figura 4.1.

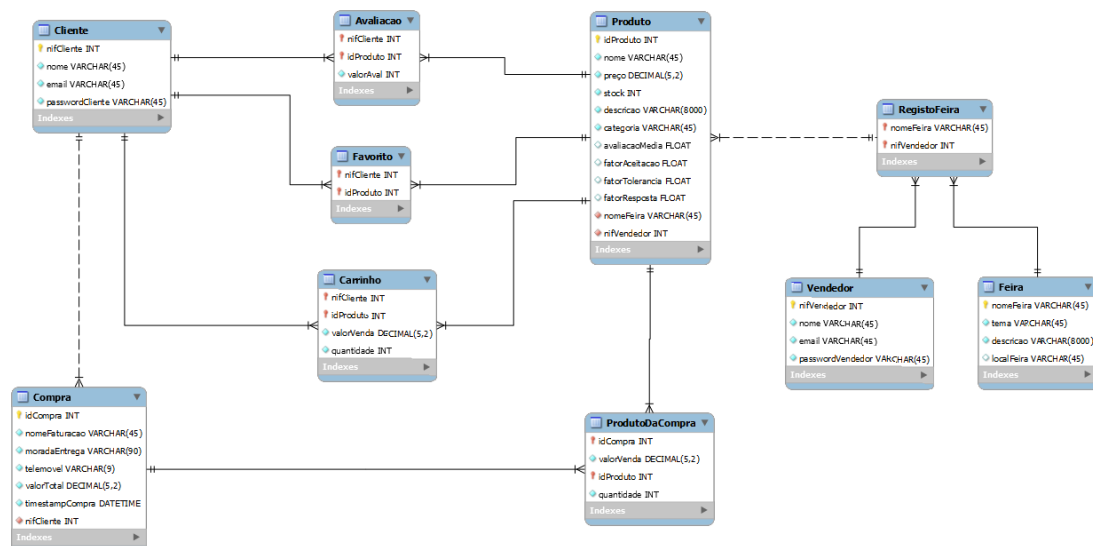


Figura 4.1: Esquema lógico do sistema de dados.

4.2 Descrição detalhada dos vários elementos de dados e seus relacionamentos

Neste esquema lógico, podemos distinguir 5 tabelas principais (Cliente, Compra, Produto, Feira e Vendedor) que resultam diretamente do mapeamento das entidades principais levantadas no modelo de domínio já apresentado. Destas entidades surgem também associadas 5 tabelas secundárias que exprimem os relacionamentos entre estas - Avaliacao, Carrinho, Favorito, ProdutoDaCompra e RegistoFeira.

Relativamente aos utilizadores diretos do sistema (clientes e vendedores), podemos reparar que estes foram divididos em duas tabelas diferentes, apesar de terem atributos semelhantes. No entanto, isto aumenta a legibilidade do esquema e facilita as *queries* de consulta à base de dados.

Um dos requisitos funcionais levantados foi a existência de um separador de favoritos para os clientes, algo que exige que sejam guardados de forma persistente em base de dados. Para tal, existirá um relacionamento $N : M$ entre a tabela “Cliente” e “Produto”, uma vez que um cliente pode ter N produtos na sua lista de favoritos e um produto pode estar presente na lista de vários clientes. Este relacionamento originará uma tabela à qual chamamos “Favorito” com duas chaves estrangeiras.

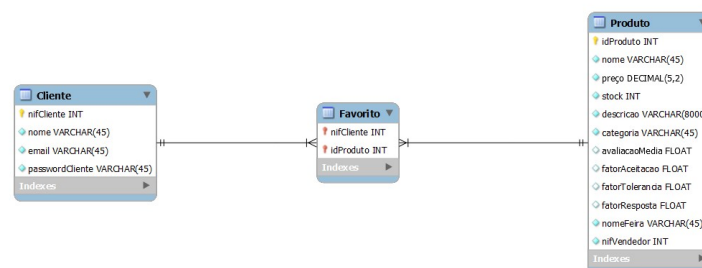


Figura 4.2: Relacionamento favoritos

O cliente poderá avaliar um produto e, desta forma, também visualizar a sua avaliação na página do respetivo produto. De forma a apresentar a avaliação de um produto, é necessário persistir essa informação. Com vista a resolver este problema, criamos novamente um relacionamento $N : M$ entre as tabelas “Cliente” e “Produto”, que origina uma tabela com duas chaves estrangeiras. Posto isto, adicionamos um atributo que representa o valor da avaliação.

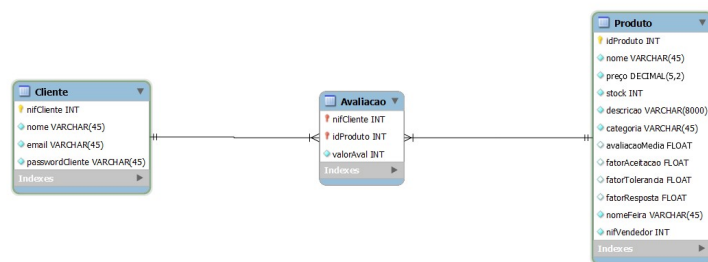


Figura 4.3: Relacionamento avaliações

O carrinho de compras é uma das funcionalidades presentes nos requisitos funcionais. Optamos por guardar o estado de um carrinho de compras em base de dados, com o objetivo de um certo cliente não perder os produtos que pretendia comprar caso feche o *site*. Deste modo, temos um relacionamento $N : M$ de “Cliente” para “Produto”, uma vez que um cliente pode ter vários produtos no seu carrinho, mas ao mesmo tempo um produto pode estar presente em diversos carrinhos. Isto permite-nos guardar os produtos com os respetivos preços e quantidades de um dado utilizador.

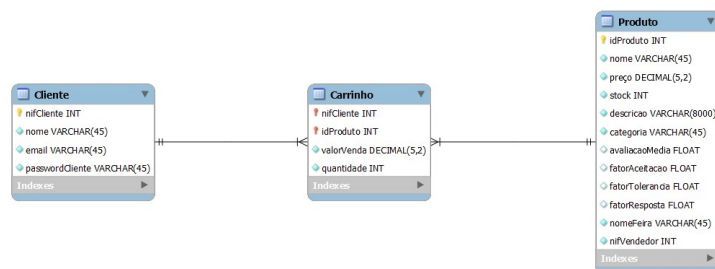


Figura 4.4: Relacionamento carrinho de compras

Um cliente pode realizar várias compras, pelo que existirá um relacionamento $1 : N$ entre a tabela “Cliente” e “Compra”, originando uma chave estrangeira no lado da compra com a informação do número de identificação fiscal cliente.

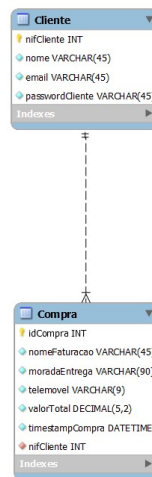


Figura 4.5: Relacionamento entre cliente e compra

Outro dos relacionamentos $N : M$ presentes no nosso modelo lógico é o relacionamento entre a tabela “Compra” e “Produto”. Uma compra pode estar associada a vários produtos e um produto pode estar presente em várias compras. A tabela originada é nos útil, na medida em que temos acesso a todas as compras realizadas e, por isso, podemos responder facilmente ao requisito do histórico de vendas inicialmente definido.

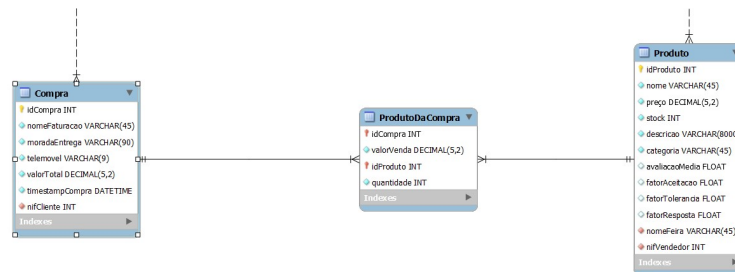


Figura 4.6: Relacionamento entre compra e produto

Relativamente às entidades vendedor e feira, um vendedor poderá registar-se em múltiplas feiras, assim como uma feira pode albergar vários vendedores, pelo que irá existir um relacionamento $N : M$. A tabela que resulta deste relacionamento irá indicar todos os registos dos vendedores efetuados nas feiras.

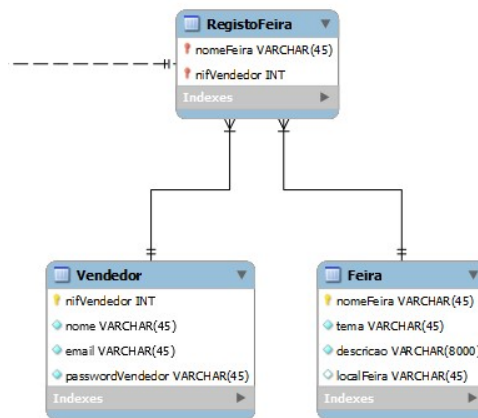


Figura 4.7: Relacionamento entre vendedor e feira

Finalmente, um registo efetuado por um vendedor numa feira poderá ter vários produtos. Com isto, existirá um relacionamento 1 : N da tabela que foi criada, anteriormente referida, e a tabela “Produto”. O produto passa assim a ser identificado por um vendedor e uma feira, algo que será indispensável para a realização de várias operações.

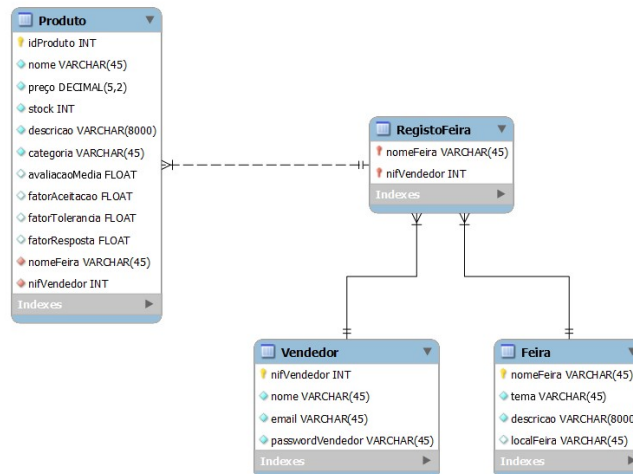


Figura 4.8: Relacionamento entre produto e registo de feira

5 Esboço das Interfaces do Sistema

De forma a representar as várias interfaces que o sistema promove, realizamos vários esboços que se encontram maioritariamente em anexo. Para criar estes esboços, recorremos à plataforma draw.io, uma vez que o objetivo passou por realizar um desenho simples. Nesta secção, iremos apresentar a sequência lógica entre estes *mockups* e fazer a respetiva associação com os *use cases* definidos.

5.1 Estrutura geral das interfaces do sistema

Através do diagrama de máquina de estados da figura 5.1 reproduzimos a sequência lógica da paginação da nossa aplicação, isto é, as transições existentes entre os *mockups*. Este tipo de diagrama UML é útil, uma vez que a ênfase é colocada no estado do sistema, modelando-se todos os estados possíveis que o sistema atravessa em resposta a diversos eventos que, neste caso, se tratam da mudança de página.

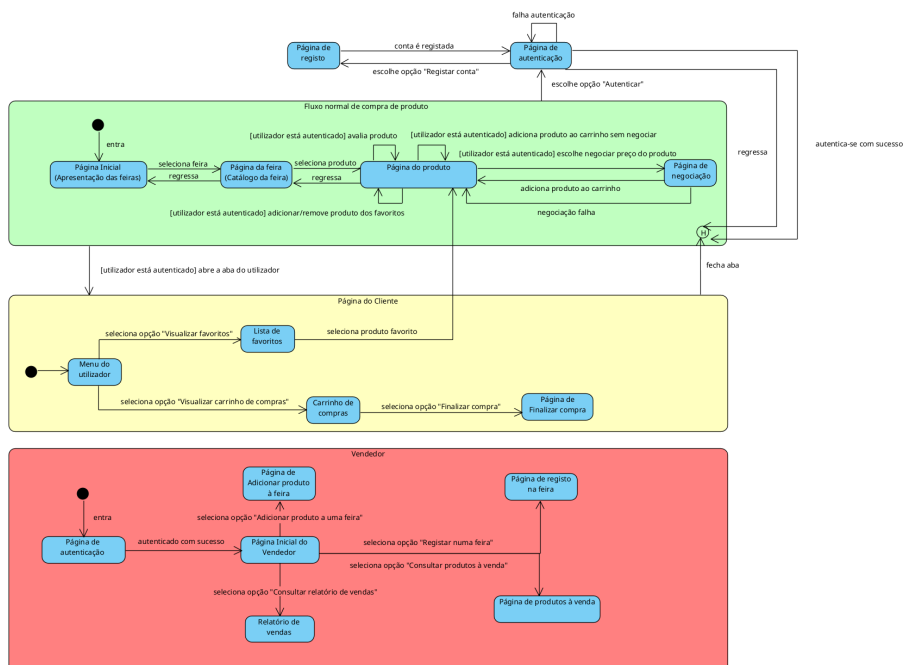


Figura 5.1: Transições entre *mockups*

5.2 Caracterização das interfaces

Ao realizar estes esboços das interfaces do sistema, procuramos ter por base os *use cases* definidos anteriormente. Assim, estes esboços acabam por complementar graficamente os mesmos.

Relembrando as especificações de *use cases* abordadas anteriormente neste relatório - “Consultar produto”, “Adicionar produto ao carrinho de compras” e “Negociar produto” -, passamos à associação entre os *mockups* e estas especificações. A representação gráfica dos *use cases* “Consultar produto” e “Negociar produto” encontram-se nas figuras 5.2 e 5.2, respetivamente.

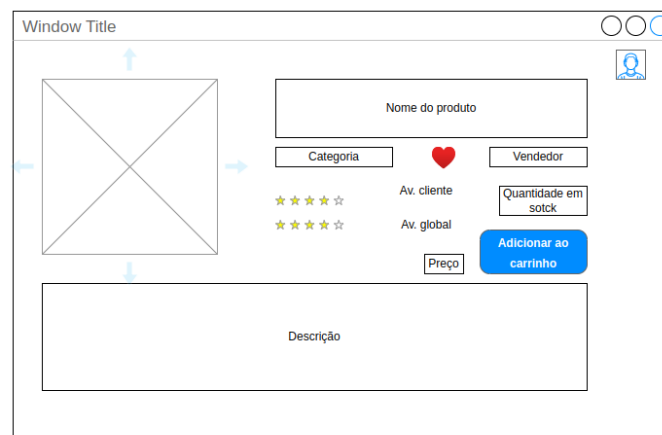


Figura 5.2: *Mockup* de apresentação de um produto

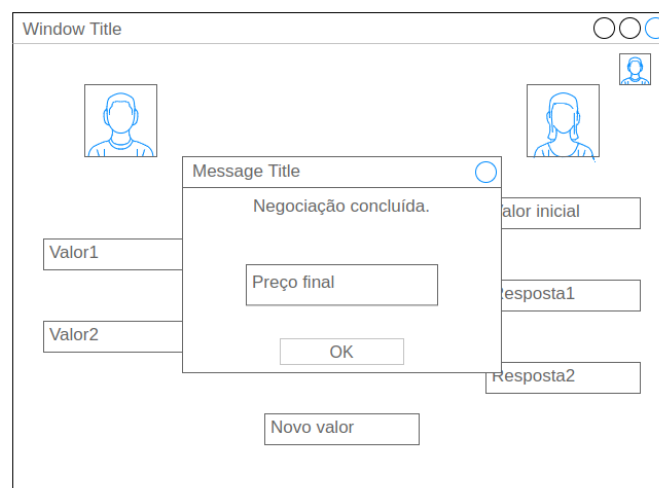


Figura 5.3: *Mockup* de negociação de um produto

6 Implementação da Aplicação

6.1 Processo de implementação

Depois de levantados os requisitos do sistema e de se ter especificado quer aspetos estruturais quer aspetos comportamentais da nossa aplicação, partimos para a implementação de uma solução *web*, tendo como base a tecnologia “.NET Core“.

Assim, de modo a reduzir conflitos de trabalho e a aumentar a produtividade geral do processo de desenvolvimento, os elementos da equipa foram divididos pelas seguintes componentes:

Elemento	Área de Trabalho
Gabriela Santos Ferreira da Cunha	Desenvolvimento da Interface Web
João António Redondo Martins	Criação, acesso e manipulação da Base de Dados
João Pedro Antunes Gonçalves	Analista de Testes
Miguel de Sousa Braga	Desenvolvimento da Interface Web
Nuno Guilherme Cruz Varela	Implementação do servidor

Tabela 6.1: Distribuição do grupo pelos componentes da implementação.

Isto foi possível devido a todo o planeamento prévio realizado que nos permitiu encarar esta fase do trabalho com menos dúvidas e mais certezas sobre a forma como cada um dos componentes e serviços iriam ser implementados.

Relativamente à implementação do servidor, começou-se por realizar os DAOs de acesso à base de dados, utilizando a ferramenta Dapper. Depois, foram criadas, uma por por uma, as operações fundamentais de cada um dos serviços.

Quanto ao processo de implementação das interfaces, começou-se por fazer o planeamento da estrutura e definir o esquema de cores. Foi também adicionada a funcionalidade de interação com o servidor, à medida que a ligação à base de dados se tornava possível. Para além disso,

foi adicionado algum código funcional C# de apoio à interface gráfica, recorrendo à linguagem Razor.

6.2 Apresentação dos serviços implementados e estrutura final da aplicação

Uma das fases mais importantes da implementação da aplicação é a divisão do sistema em serviços. Como tal, podemos identificar 3 subsistemas na nossa aplicação que vão corresponder a serviços: utilizadores, compras e feiras.

Uns dos requisitos que foram alvo de mais foco por parte do grupo foram a parte do processo de realizar uma compra e a negociação de um produto. Relativamente à finalização de uma compra, vimos a necessidade de ter controlo de concorrência, uma vez que é estritamente necessário descartar todas as situações de erro e incoerência de estados. A não utilização de controlo de concorrência poderia levar a estados em que um cliente poderia ter efetuado uma compra, mas na realidade não haver *stock* dos produtos, algo que tem de ser evitado. Para tal, recorreremos a *SQL Transactions* que nos permite realizar transações com todas as condições reunidas para garantir que não há estados de incoerências. É importante referir que o controlo de concorrência é garantido pela base de dados nos seus acessos e nas transações efetuadas. A negociação de um certo produto é algo que nos distingue de outras aplicações da concorrência, pelo que foram objeto de maior foco da equipa de desenvolvimento. Todo este processo consiste na “conversa” entre o cliente e o algoritmo implementado.

Ainda, relativamente ao *back-end* da aplicação, todas as operações correspondentes aos requisitos inicialmente definidos foram implementados. No entanto, não nos foi possível apresentar no *front-end* a totalidade destas funcionalidades.

As funcionalidades implementadas no *front-end* da aplicação foram as seguintes:

- Criação de feiras por parte do administrador do sistema;
- Entrar numa feira por parte do cliente;
- Consultar catálogo de uma feira;
- Consultar detalhes de produto;
- Adicionar produto ao carrinho de compras;
- Realizar negociação;

- Realizar compra;

Não foram implementadas funcionalidades relativas à gestão de cliente, como o *login* e o registo, pelo que, por consequência, não conseguimos implementar funcionalidades como “adicionar produto aos favoritos” ou “classificar produto”. Da parte do Vendedor, podemos registar numa feira, adicionar produto a uma feira e ver os produtos associados a esse vendedor.

6.3 Análise e avaliação da aplicação desenvolvida

Uma vez que não implementamos o *login* e o registo de um utilizador, foi criada uma página temporária, da figura 6.1, de modo a poder aceder às funcionalidades de cada tipo de utilizador que requerem autenticação, através de botões.

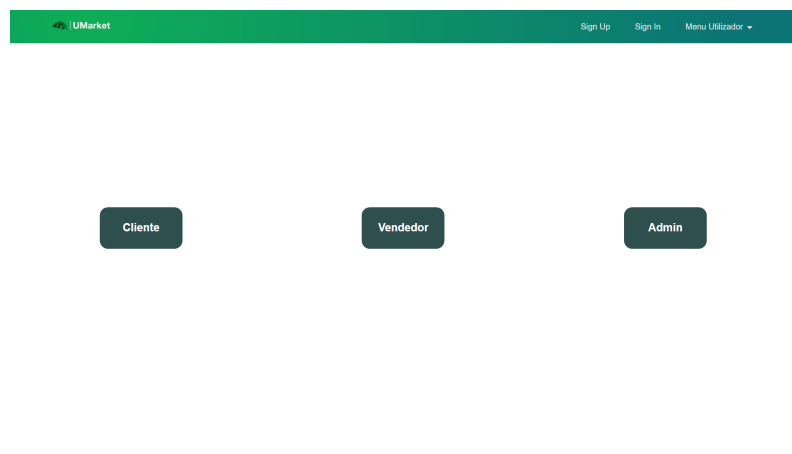


Figura 6.1: Página inicial

O cliente terá acesso à página principal das feiras onde serão apresentadas as feiras existentes e onde pode pesquisar por uma específica.



Figura 6.2: Página de apresentação de feiras

Assim que escolhe uma feira, terá acesso ao catálogo dessa feira, onde poderá também pesquisar por um produto específico.



Figura 6.3: Página de produtos de uma feira

Em seguida, o utilizador tem acesso à informação de um dado produto, podendo adicioná-lo ao carrinho, aos favoritos ou atribuir-lhe uma classificação.



Figura 6.4: Página de um produto

No lado vendedor, a página representada na figura 6.5 permite a este registar-se numa determinada feira.

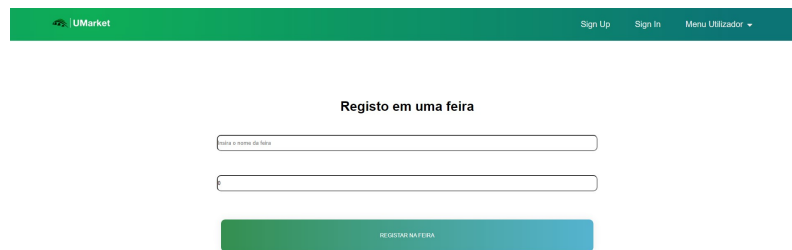


Figura 6.5: Página para registo em uma feira

Para adicionar produtos a uma feira, o vendedor deve indicar o nome, preço, *stock*, categoria, descrição e definir os parâmetros de negociação para este produto, se assim pretender.



Figura 6.6: Página para adicionar produto a uma feira

Assim que um utilizador finalize a sua compra, deve indicar os seus dados e escolher o modo de pagamento.

The screenshot displays the UMarket checkout interface. At the top, a green header bar contains the UMarket logo on the left and 'Sign Up', 'Sign In', and 'Menu Utilizador' links on the right. Below the header, the title 'Insira Dados de Faturação' is centered. A large, light blue rectangular form contains four input fields stacked vertically, labeled 'Nome:', 'Apelido:', 'Número de telemóvel:', and 'Morada:'. Below this form, three payment method logos are displayed horizontally: MB WAY, MULTIBANCO (MB logo), and mastercard. A thin horizontal line is positioned below the payment options.

Figura 6.7: Página para efetuar o pagamento de uma compra

Posto isto, grande parte dos requisitos que foram estabelecidos no início do projeto foram realizados. No entanto, há funcionalidades definidas no *back-end* que não têm suporte gráfico na aplicação. Uma dessas funcionalidades é a apresentação do relatório de vendas, que constitui algo importante para os vendedores terem uma perceção da dimensão do seu negócio. Outra dessas funcionalidades é a classificação dos produtos.

6.4 Ferramentas Utilizadas

O processo de desenvolvimento foi realizado tendo por base tecnologias Microsoft.

Uma vez que a implementação da aplicação é feita em ambiente *web*, foi utilizada a *framework* Blazor, baseada em HTML, CSS, C# e Razor e executada via *WebAssembly*, um formato portátil, leve e com tempo de carregamento eficiente, adequado para compilação em *web*.

Para o desenvolvimento da base de dados foi utilizado Microsoft SQL Server, devido à facilidade de utilização e integração no programa associadas, sendo a utilização deste sistema de gestão de base de dados bastante comum no mercado em contexto real. Ainda associado à base de dados, utilizamos a ferramenta *SQL Server Management Studio*, de modo a podermos gerir de uma maneira mais fácil todos os componentes da base de dados.

Com vista a facilitar o processo de desenvolvimento, procuramos utilizar um IDE que fornecesse um diverso conjunto de funcionalidades . A nossa escolha traduziu-se, portanto, no Visual Studio 2022, uma das ferramentas mais utilizadas no mercado atual, que suporta as várias linguagens utilizadas no projeto.

No que toca a ferramentas utilizadas mais ligadas à implementação do *software*, utilizamos um ORM para podermos utilizar a base de dados de forma abstrata. A escolha do ORM caiu sobre a ferramenta *Dapper*. Poderíamos também ter optado pela *Entity Framework Core*, no entanto o *Dapper* foi fortemente aconselhado nos seminários da unidade curricular e decidimos usá-lo.

7 Conclusões e Trabalho Futuro

Após a realização deste projeto, entendemos o impacto que um planeamento prévio correto causa na implementação do mesmo. Efetivamente, a planificação que havíamos definido durante os meses anteriores condicionou o desenvolvimento da aplicação. Esta última fase revelou-se, particularmente, a mais difícil e complexa, uma vez que fomos expostos a um ambiente sobre o qual não tínhamos experiência. Com isto, o tempo de adaptação a este ambiente acabou por demorar mais do que o esperado, levando a que os resultados que tínhamos em expectativa não fossem concretizados na totalidade.

Um primeiro aspeto a melhorar seria a totalidade da implementação do sistema, algo que, como referido anteriormente, não foi conseguido. De um ponto de vista estético, acreditamos que a aplicação poderia também vir a ser melhorada, uma vez que não se encontra propriamente atrativa a nível visual.

Em jeito de conclusão, e apesar de termos consciência dos vários aspetos que podiam vir a ser melhorados, fazemos um balanço positivo a nível de aprendizagem e acreditamos que a elaboração deste projeto tenha sido útil tendo em vista a entrada para o mercado de trabalho.

Referências

- [1] Sommerville, Ian. (2015). *software Engineering* (10ª edição). Pearson

Lista de Siglas e Acrónimos

UML *Unified Modelling Language*

IDE *Integrated Development Environment*

ORM *Object Relational Mapping*

Anexos

I. Especificação de *use cases*

Use Case	Criar feira	
Ator	Gestor do Sistema	
Cenários		
Descrição	O gestor do sistema cria uma feira.	
Pré-Condição	True	
Pós-Condição	A feira é adicionada ao sistema.	
	Ator	Sistema
Fluxo Normal	1. Indica o nome, descrição e tema da feira.	
		2. Valida o nome da feira.
	3. Indica o local da feira presencial.	
		4. Regista a feira.
Fluxo de exceção [Nome da feira já utilizado] (Passo 2)		2.1 Termina o processo.

Figura 7.1: Use case “Criar feira”

Use Case	Autenticar utilizador	
Ator	Utilizador	
Cenários		
Descrição	O utilizador autentica-se como cliente/vendedor.	
Pré-Condição	O utilizador está registado no sistema.	
Pós-Condição	O utilizador é autenticado.	
	Ator	Sistema
Fluxo Normal	1. Indica username e password.	
		2. Valida os dados introduzidos.
Fluxo de exceção [O utilizador introduziu dados inválidos] (Passo 2)		2.1 Termina o processo.

Figura 7.2: Use case “Autenticar utilizador”

Use Case	Realizar Compra	
Ator	Cliente	
Cenários		
Descrição	O cliente compra os produtos do carrinho de compras.	
Pré-Condição	O cliente está devidamente autenticado e tem que ter pelo menos um produto no carrinho.	
Pós-Condição	A compra é efetuada e o carrinho esvaziado.	
	Ator	Sistema
Fluxo normal		1. Pede o nome, a morada, o número de telemóvel e NIF de faturação.
	2. Insere o nome e a morada.	
	3. Insere o contacto (número de telemóvel).	
	4. Insere o NIF de faturação.	
		5. Apresenta as diversas formas de pagamento.
	6. Seleciona o método de transferência bancária.	
		7. Pede o nome e IBAN da conta de destino.
	8. Insere o nome e IBAN da conta de destino.	
		9. Regista a compra do cliente.
Fluxo alternativo [Não insere contacto] (passo 3)		3.1. Volta a 4.
Fluxo alternativo [Não insere NIF] (passo 4)		4.1. Volta a 5.
Fluxo alternativo [Seleciona o método MBWay] (passo 6)		6.1 Pede o número do telemóvel.
	6.2. Insere o número de telemóvel.	
		6.3. Volta a 9.
Fluxo alternativo [Seleciona o método Cartão de Crédito] (passo 6)		6.1. Pede o número do cartão.
	6.2. Insere o número do cartão de crédito.	
		6.3. Volta a 9.

Figura 7.3: Use case “Realizar compra”

Use Case	Registar Utilizador	
Ator	Utilizador	
Cenários		
Descrição	Um novo utilizador faz o registo	
Pré-Condição	True	
Pós-Condição	Um novo utilizador é adicionado à base de dados do sistema	
	Ator	Sistema
Fluxo Normal	1. Indica o seu nome de utilizador, palavra-passe e o email para se registar como Cliente.	
		2. Valida os dados inseridos.
		3. Adiciona novo utilizador à base de dados.
Fluxo Alternativo [Utilizador regista-se como Vendedor] (passo 1)	1.1 Indica o seu nome de utilizador, palavra-passe, email e o NIF para se registar como Vendedor.	
		1.2. Valida os dados inseridos.
		1.3. Adiciona novo utilizador à base de dados.
Fluxo de exceção [Nome de utilizador já existente ou NIF inválido] (passo 2)		2.1. Termina o processo.
Fluxo de exceção [Nome de utilizador já existente ou NIF inválido] (Passo 1.2)		1.2.1 Termina o processo

Figura 7.4: Use case “Registar utilizador”

Use Case	Consultar detalhes de feira	
Ator	Cliente	
Cenários		
Descrição	Um cliente consulta os detalhes de uma feira	
Pré-Condição	True	
Pós-Condição	Os detalhes da feira são apresentados ao cliente.	
	Ator	Sistema
Fluxo Normal	1. O cliente indica o nome da feira que pretende consultar.	
		2. O sistema verifica se a feira existe e apresenta os detalhes da feira ao cliente, incluindo vendedores associados, descrição, tema, etc.
Fluxo de Exceção [A feira procurada não existe] (passo 2)		3.1. Termina o processo

Figura 7.5: Use case “Consultar detalhes de feira”

Use Case	Atualizar informação do produto	
Ator	Vendedor	
Cenários		
Descrição	Um vendedor pretende atualizar a informação de um determinado produto	
Pré-Condição	O produto está registado no sistema	
Pós-Condição	A informação do produto é atualizada	
	Ator	Sistema
Fluxo Normal	1. O vendedor insere o nome do produto	
		2. O sistema valida o nome e pergunta se pretende atualizar o preço, a quantidade ou os dois
	3. O vendedor insere a opção de alterar o preço	
		4. O sistema pede o novo preço do produto
	5. O vendedor insere o novo preço do produto	
		6. A informação do produto é atualizada
Fluxo Alternativo [O vendedor escolhe a opção de alterar a quantidade do produto] (passo 3)		4.1. O sistema pede o novo valor para a quantidade
	4.2. O vendedor insere o novo valor para o produto	
		4.3. Regressa a 7
Fluxo Alternativo [O vendedor escolhe a opção de alterar a quantidade e o preço do produto] (passo 3)		4.1. O sistema pede o novo preço e quantidade do produto
	4.2. O vendedor insere o novo preço, e a nova quantidade do produto	
		4.3. Regressa a 7
Fluxo de Exceção [O nome do produto inserido não existe] (passo 2)		2.1. Termina o processo

Figura 7.6: Use case “Atualizar informação do produto”

Use Case	Registrar produto	
Ator	Vendedor	
Cenários		
Descrição	O Vendedor regista um novo produto para venda	
Pré-Condição	O Vendedor está devidamente autenticado	
Pós-Condição	Um novo produto é adicionado à lista de produtos para venda	
	Ator	Sistema
Fluxo Normal		1. Pede o nome do produto.
	2. Insere o nome do produto que quer registar.	
		3. Verifica que não existe um produto com o nome inserido.
		4. Apresenta as categorias disponíveis.
	5. Seleciona a categoria do produto que quer registar.	
		6. Pede a descrição, preço e quantidade em stock.
	7. Insere a descrição, o preço e quantidade em stock do produto que quer registar.	
		8. Pergunta se o preço do produto pode ser negociado.
	9. Indica que o preço do produto pode ser negociado.	
		10. Confirma que o vendedor tem sistema de negociação.
		11. Informa que o produto foi adicionado para venda.
		12. Adiciona o registo do produto ao sistema.
Fluxo de exceção [Produto com o mesmo nome já registado] (passo 3)		3.1. Informa que já há um produto registado com o mesmo nome
		3.2. Cancela o registo do produto.
Fluxo alternativo [Indica que o preço do produto não pode ser negociado] (passo 10)	10.1. Regressa a 11.	
Fluxo de exceção [Vendedor não tem sistema de negociação] (passo 11)		11.1. Indica que o vendedor não tem sistema de negociação.
		11.2 Cancela o registo do produto.

Figura 7.7: Use case “Registrar produto”

Use Case	Definir parâmetros de negociação	
Ator	Vendedor	
Cenários		
Descrição	O vendedor define os parâmetros para a negociação de um dado produto.	
Pré-Condição	O vendedor está autenticado no sistema e está a registar um produto	
Pós-Condição	Os parâmetros de negociação para esse produto estão definidos.	
	Ator	Sistema
Fluxo normal		1. Pede o fator de tolerância.
	2. Indica o fator de tolerância.	
		3. Valida o factor de tolerância.
		4. Pede o factor de aceitação.
	5. Indica o fator de aceitação.	
		6. Valida o factor de aceitação.
		7. Pede o factor de resposta.
	8. Indica o fator de resposta.	
		9. Valida o fator de resposta.
		10. Adiciona os parâmetros de negociação ao produto
Fluxo de exceção [Fator de tolerância ≥ 100 ou ≤ 0] (Passo 3)		4.1 Termina o processo.
Fluxo de exceção [Fator de aceitação \geq fator de tolerância ou ≤ 0] (Passo 6)		6.1. Termina o processo.
Fluxo de exceção [Fator de resposta ≥ 100 ou ≤ 0] (Passo 9)		9.1. Termina o processo.

Figura 7.8: Use case “Definir parâmetros de negociação”

Use Case	Avaliar vendedor	
Ator	Cliente	
Cenários		
Descrição	O cliente atribui uma classificação a um vendedor após realizar uma dada compra.	
Pré-Condição	O cliente está autenticado no sistema.	
Pós-Condição	A classificação dada pelo cliente é adicionada à pontuação geral do vendedor.	
	Ator	Sistema
Fluxo Normal	1. Indica o nome do vendedor que quer classificar.	
		2. Valida que esse vendedor existe e que o cliente tem compras associadas a esse vendedor.
	3. Indica a classificação que pretende atribuir ao vendedor (0 a 5).	
		4. Atualiza a classificação global do vendedor.
Fluxo de exceção [O vendedor não existe] (passo 2)		2.1. Indica que o vendedor indicado não existe.
		2.2. Cancela a classificação.
Fluxo de exceção [O cliente não tem compras associadas a esse vendedor] (passo 2)		2.1. Indica que não há registo de compras efetuadas pelo cliente a esse vendedor.
		2.2. Cancela a classificação.

Figura 7.9: Use case “Avaliar vendedor”

Use Case	Fazer registo numa feira	
Ator	Vendedor	
Cenários		
Descrição	O vendedor seleciona uma feira e faz o registo nessa feira.	
Pré-Condição	O vendedor está autenticado no sistema.	
Pós-Condição	O vendedor é adicionado à lista de vendedores de uma feira e pode começar a vender produtos.	
	Ator	Sistema
Fluxo Normal	1. Seleciona a feira que deseja entrar.	
		2. Valida que a feira existe e que existe pelo menos um stand de venda livre.
	3. Seleciona um dos stands disponíveis.	
		4. Regista o vendedor como novo feirante na feira selecionada.
Fluxo de exceção [Feira não existe] (passo 2)		2.1. Indica que a feira indicada não existe.
		2.2. Cancela o registo do vendedor.
Fluxo de exceção [Feira não tem stands disponíveis] (passo 2)		2.1. Indica que a feira indicada atingiu a lotação máxima.
		2.2. Cancela o registo do vendedor.

Figura 7.10: Use case “Fazer registo numa feira”

Use Case	Consultar catálogo de uma feira	
Ator	Cliente	
Cenários		
Descrição	O cliente consulta os produtos disponíveis na feira	
Pré-Condição	Cliente selecionou a feira pretendida	
Pós-Condição	O cliente visualiza os produtos disponíveis na feira	
	Ator	Sistema
Fluxo Normal	1. Seleciona a opção consultar catálogo.	
	2. Aplica consulta sem filtros.	
		3. Apresenta a lista dos produtos disponíveis.
Fluxo Alternativo [Pesquisar por vendedor] (passo 2)	2.1 Aplica pesquisa com filtro por vendedor.	
		2.2 Apresenta a lista dos produtos disponíveis.
Fluxo Alternativo [Pesquisar por preço] (passo 2)	2.1 Aplica pesquisa com filtro por preço ascendente/descendente.	
		2.2 Apresenta a lista dos produtos disponíveis.
Fluxo Alternativo [Pesquisar por categoria] (passo 2)	2.1 Aplica pesquisa com filtro por categoria.	
		2.2 Apresenta a lista dos produtos disponíveis.

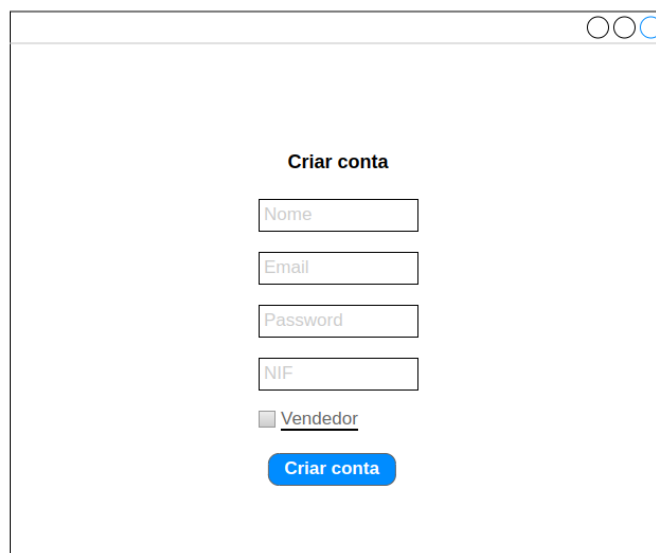
Figura 7.11: Use case “Consultar catálogo de uma feira”

Use Case	Adicionar produto aos favoritos	
Ator	Cliente	
Cenários		
Descrição	O cliente adiciona um produto aos favoritos.	
Pré-Condição	O produto existe.	
Pós-Condição	O produto é adicionado aos favoritos.	
	Ator	Sistema
Fluxo Normal	1. Indica produto que pretende adicionar aos favoritos.	
		2. Regista adição.

Figura 7.12: Use case “Adicionar produto aos favoritos”

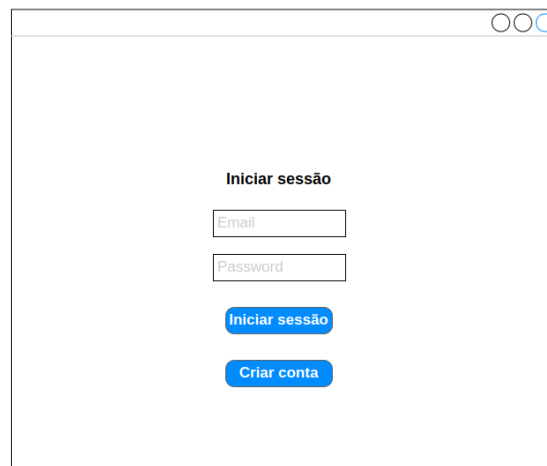
II. Mockups adicionais

Cliente



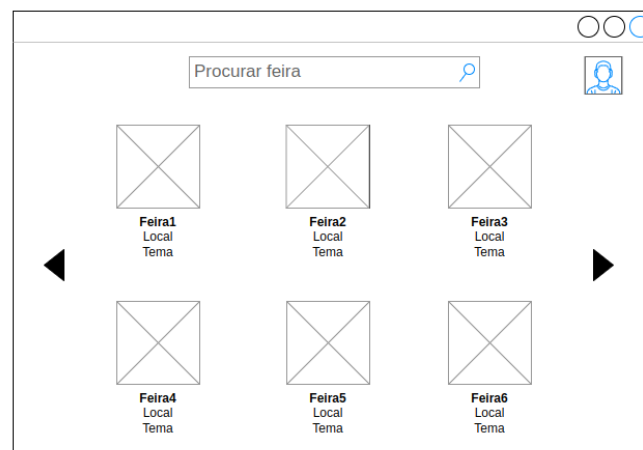
Mockup de uma interface de registo (criar conta) para um cliente. O formulário contém campos para Nome, Email, Password, e NIF. Há também uma opção para "Vendedor" com um checkbox desativado. Um botão azul "Criar conta" está no final.

Figura 7.13: Mockup de registo



A mockup of a login form within a window frame. The form is centered and contains the following elements: a title "Iniciar sessão", an "Email" input field, a "Password" input field, a blue "Iniciar sessão" button, and a blue "Criar conta" button.

Figura 7.14: *Mockup* de autenticação



A mockup of a fair presentation screen within a window frame. The screen features a search bar at the top with the placeholder text "Procurar feira" and a magnifying glass icon. To the right of the search bar is a user profile icon. Below the search bar, there are six placeholder boxes arranged in a 2x3 grid. Each box contains a square with an 'X' and the text "Feira1 Local Tema" (for the first row) and "Feira4 Local Tema" (for the second row). Navigation arrows (left and right) are positioned on the sides of the grid.

Figura 7.15: *Mockup* de apresentação das feiras



Figura 7.16: *Mockup* de apresentação do catálogo de uma feira

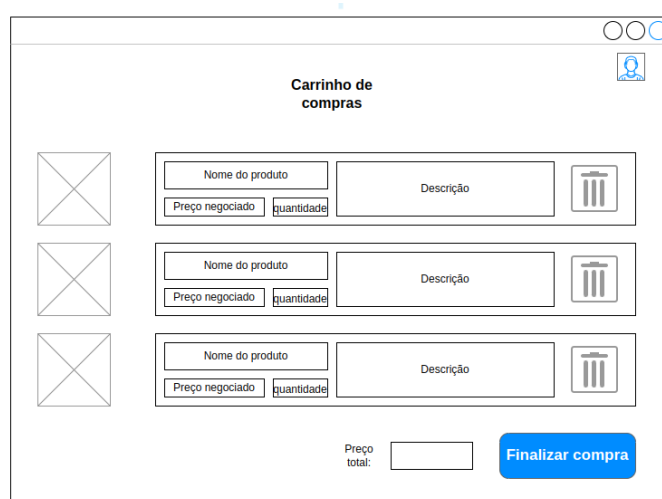


Figura 7.17: *Mockup* de apresentação do carrinho de compras

Window Title

Nome de faturação

Morada

Nº de telemóvel

MBWAY

Transferência

Cartão de crédito

Figura 7.18: *Mockup* de finalização de compra

Window Title

Favoritos

Nome do produto

Preço

Descrição

Nome do produto

Preço

Descrição

Nome do produto

Preço

Descrição

Figura 7.19: *Mockup* de apresentação dos favoritos

Vendedor

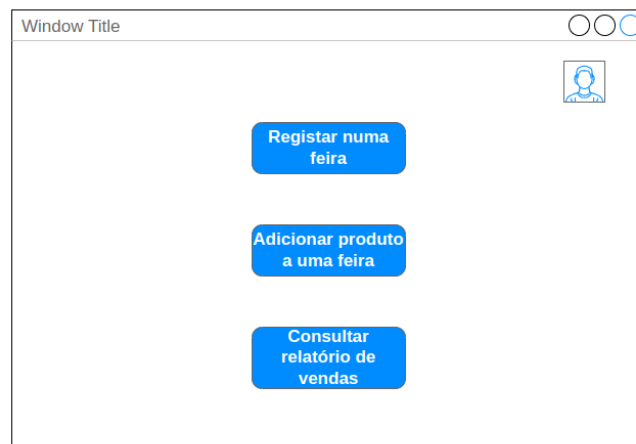


Figura 7.20: *Mockup* de seleção vendedor

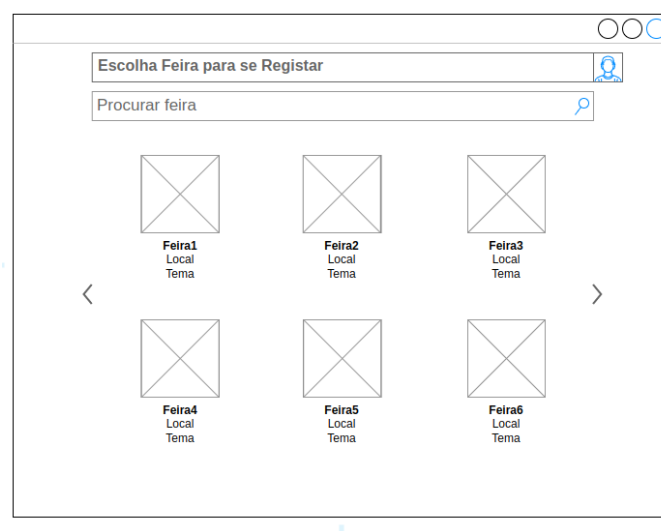


Figura 7.21: *Mockup* de seleção de feira para o vendedor

Adicionar Produto

Parâmetros de negociação

Adicionar Produto

Figura 7.22: *Mockup* de adição de produto para venda para o vendedor

Produtos à Venda

Feira 1

	Nome do Produto	Categoria	Preço	Quantidade		
	Nome do Produto	Categoria	Preço	Quantidade		
	Nome do Produto	Categoria	Preço	Quantidade		
	Nome do Produto	Categoria	Preço	Quantidade		

Feira 2

	Nome do Produto	Categoria	Preço	Quantidade		
	Nome do Produto	Categoria	Preço	Quantidade		
	Nome do Produto	Categoria	Preço	Quantidade		
	Nome do Produto	Categoria	Preço	Quantidade		

Figura 7.23: *Mockup* de consulta de produto à venda para o vendedor

Editar Produto

Nome do Produto

Quantidade

Categoria

Preço

Descrição

Fator de tolerância

Fator de aceitação

Fator de resposta

Guardar Alterações

Figura 7.24: *Mockup* de edição de produto à venda para o vendedor

Histórico de vendas

Produto	Data	Valor	Cliente
Nome 1	Item 1	Item 1	Item 1
Nome 2	Item 2	Item 2	Item 2
Nome 3	Item 3	Item 3	Item 3
Nome 4	Item 4	Item 4	Item 4

Filtrar por datas:

October 2014

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

Valor total:

Figura 7.25: *Mockup* de consulta do relatório de vendas para o vendedor