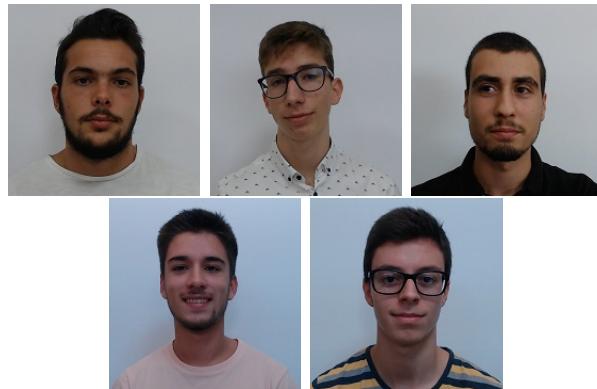


Universidade do Minho
Mestrado Integrado em Engenharia Informática
3º Ano - 2º Semestre

Laboratórios de Informática IV

Relatório do Trabalho Prático

Grupo Prático 5 - PL5



a83732 – Gonçalo Rodrigues Pinto
a84197 – João Pedro Araújo Parente
a84829 – José Nuno Martins da Costa
a85059 – Diogo Paulo Lopes de Vasconcelos
a85851 – Ricardo Filipe Dantas Costa

5 de Julho de 2020

Conteúdo

| | | |
|----------|--|-----------|
| 1 | Resumo | 6 |
| 2 | Introdução | 7 |
| 2.1 | Contextualização | 7 |
| 2.2 | Apresentação do Caso de Estudo | 8 |
| 2.3 | Motivação e Objectivos | 9 |
| 2.4 | Estrutura do Relatório | 10 |
| 3 | Fundamentação do Sistema | 11 |
| 3.1 | Justificação do Sistema | 11 |
| 3.2 | Utilidade do Sistema | 11 |
| 3.3 | Análise de Requisitos e Funcionalidades do Sistema | 12 |
| 4 | Planeamento do Projecto | 13 |
| 4.1 | Identificação dos Recursos Necessários | 13 |
| 4.2 | Plano de Desenvolvimento | 13 |
| 4.3 | Viabilidade de Crescimento Futuro | 14 |
| 5 | Modelação De Domínio | 15 |
| 6 | Modelo de Use Case | 16 |
| 6.1 | Diagrama | 16 |
| 6.2 | Actores | 16 |
| 6.2.1 | Utilizador | 16 |
| 6.2.2 | Cliente | 16 |
| 6.2.3 | Funcionário | 16 |
| 6.3 | Use Cases | 17 |
| 6.3.1 | Autenticação | 17 |
| 6.3.2 | Desautenticação | 17 |
| 6.3.3 | Visualizar produtos disponíveis | 18 |
| 6.3.4 | Acrescentar um produto à lista de favoritos | 18 |
| 6.3.5 | Visualizar pedidos anteriores | 18 |
| 6.3.6 | Efectuar pedido | 19 |
| 6.3.7 | Visualizar estado do pedidos | 19 |
| 6.3.8 | Consultar o número do último pedido e do próximo pedido a ser atendido | 19 |
| 6.3.9 | Reclamações | 20 |
| 6.3.10 | Ver informações da Empresa | 20 |
| 6.3.11 | Mudar estado do sistema | 21 |
| 6.3.12 | Visualizar Pedidos por preparar | 21 |
| 6.3.13 | Mudar o estado do pedido. | 21 |

| | | |
|-----------|---|-----------|
| 7 | Diagrama de Classes | 22 |
| 7.1 | Cliente | 22 |
| 7.2 | Funcionário | 23 |
| 7.3 | Servidor | 24 |
| 8 | Base de Dados | 25 |
| 8.1 | Modelo Conceptual de Dados | 25 |
| 8.1.1 | Identificação das Identidades | 25 |
| 8.1.2 | Identificação dos Relacionamentos | 25 |
| 8.1.3 | Diagrama do Modelo Conceptual | 26 |
| 8.2 | Modelo Lógico de Dados | 27 |
| 8.2.1 | Conversão do esquema conceptual para o esquema lógico | 27 |
| 8.2.2 | Diagrama do Modelo Lógico | 28 |
| 8.3 | Modelo Físico de Dados | 28 |
| 8.3.1 | Conversão do esquema lógico para o esquema físico . . | 28 |
| 9 | Interface | 29 |
| 9.1 | Menu Inicial | 29 |
| 9.2 | Autenticar | 30 |
| 9.3 | Aplicação do Cliente | 30 |
| 9.3.1 | Criar Conta | 30 |
| 9.3.2 | Menu | 31 |
| 9.3.3 | Pedido Cliente | 31 |
| 9.3.4 | Ver Produtos | 32 |
| 9.3.5 | Lista Favoritos | 32 |
| 9.3.6 | Histórico | 33 |
| 9.3.7 | Reclamações | 33 |
| 9.3.8 | Sobre | 34 |
| 9.4 | Aplicação do Funcionário | 34 |
| 9.4.1 | Menu | 34 |
| 10 | Implementação das funcionalidades/tecnologias | 35 |
| 10.1 | Modelo MVC | 35 |
| 10.2 | Servidor | 36 |
| 10.2.1 | Funcionamento | 36 |
| 10.2.2 | Multiplexagem e segurança | 36 |
| 10.2.3 | Conexões e forma como comunicam | 37 |
| 10.2.4 | Gestão da informação | 37 |
| 10.3 | Inclusão da Base de Dados | 38 |
| 10.4 | Incorporação da API externa OneSignal | 38 |

| | |
|--|-----------|
| 11 Produto Final | 39 |
| 11.1 Aplicação do Cliente | 39 |
| 11.1.1 Carregar a aplicação | 39 |
| 11.1.2 Criar Conta | 40 |
| 11.1.3 Autenticar | 40 |
| 11.1.4 Menu Cliente | 41 |
| 11.1.5 Ver Produtos (Bebidas) | 41 |
| 11.1.6 Ver Produtos (Compal Pera) | 42 |
| 11.1.7 Adicionar ao carrinho | 42 |
| 11.1.8 Ver carrinho | 43 |
| 11.1.9 Fazer Pedido | 43 |
| 11.1.10 Estado Pedido (por preparar) | 44 |
| 11.1.11 Notificação | 44 |
| 11.1.12 Histórico | 45 |
| 11.1.13 Sobre | 45 |
| 11.2 Aplicação do Funcionário | 46 |
| 11.2.1 Autenticar | 46 |
| 11.2.2 Menu Inicial | 46 |
| 11.2.3 Menu Inicial (com pedidos) | 47 |
| 11.2.4 Informação de um pedido | 47 |
| 12 Discussão | 48 |
| 13 Conclusões | 49 |

Lista de Figuras

| | | |
|----|---|----|
| 1 | Diagrama de Gantt. | 14 |
| 2 | Diagrama do Modelo de Domínio. | 15 |
| 3 | Diagrama de Use Cases. | 16 |
| 4 | Diagrama de Classes do Cliente. | 22 |
| 5 | Diagrama de Classes do Funcionário. | 23 |
| 6 | Diagrama de Classes do Servidor. | 24 |
| 7 | Diagrama do Modelo Conceptual. | 26 |
| 8 | Diagrama do Modelo Lógico. | 28 |
| 9 | Mockup - Menu Inicial. | 29 |
| 10 | Mockup - Autenticar. | 30 |
| 11 | Mockup - Criar Conta. | 30 |
| 12 | Mockup - Menu Cliente. | 31 |
| 13 | Mockup - Pedido Cliente. | 31 |
| 14 | Mockup - Ver Produtos. | 32 |
| 15 | Mockup - Lista Favoritos. | 32 |
| 16 | Mockup - Histórico. | 33 |
| 17 | Mockup - Reclamações. | 33 |
| 18 | Mockup - Sobre. | 34 |
| 19 | Mockup - Menu Funcionário. | 34 |
| 20 | Interface - Carregar aplicação. | 39 |
| 21 | Interface - Criar Conta. | 40 |
| 22 | Interface - Autenticar. | 40 |
| 23 | Interface - Menu Cliente. | 41 |
| 24 | Interface - Ver Produtos (bebidas). | 41 |
| 25 | Interface - Ver produto (compal pera). | 42 |
| 26 | Interface - Adicionar ao carrinho. | 42 |
| 27 | Interface - Ver carrinho. | 43 |
| 28 | Interface - Fazer Pedido. | 43 |
| 29 | Interface - Estado Pedido (por preparar). | 44 |
| 30 | Interface - Notificação. | 44 |
| 31 | Interface - Histórico. | 45 |
| 32 | Interface - Sobre. | 45 |
| 33 | Interface - Autenticar. | 46 |
| 34 | Interface - Menu Inicial. | 46 |
| 35 | Interface - Menu Inicial (com pedidos). | 47 |
| 36 | Interface - Informação de um pedido. | 47 |

Lista de Tabelas

| | | |
|---|--|----|
| 1 | Identificação e caracterização das potenciais entidades envolvidas | 25 |
| 2 | Caracterização dos diversos relacionamentos que possam existir entre as entidades estabelecidas. | 25 |

1 Resumo

Este relatório é o resultado do trabalho elaborado no âmbito da Unidade Curricular de Laboratórios de Informática IV na qual nos possibilitou a criação de um software com o tema "Click & Collect". Basicamente é um sistema para encontrar preços e encomendar em qualquer lugar, tornando ainda mais rápido o processo de encomenda dos favoritos sempre que se visita um estabelecimento.

Desta forma, o grupo, depois de uma profunda reflexão, chegou à decisão de criar a aplicação *MyBar*, uma aplicação móvel que oferece todos estes requisitos e mais alguns que serão adiante devidamente referidos e explicados. Assim, ao longo deste relatório do projecto serão expostos detalhadamente todos os passos que tivemos de efectuar desde o surgimento da ideia e motivação para a mesma até à sua justificação e viabilidade, culminando desta forma com o estabelecimento da identidade do projecto. Definimos ainda um conjunto de medidas que achamos importantes para o sucesso da nossa aplicação, esperando que estas sejam atingidas na sua totalidade.

Elaborou-se um plano de desenvolvimento a longo prazo, no qual, com a ajuda de um Diagrama de Gantt, listamos todas as tarefas que terão de ser executadas para finalmente atingir a conclusão do projecto, juntamente com o período de tempo esperado para cada uma delas.

Foi ainda detalhadamente especificado todo o sistema com os devidos diagramas, de modo a criar uma correta percepção do funcionamento do mesmo. Adjacente à especificação surge a toda a complexa modelação da base de dados para a aplicação.

Por fim, foi idealizada uma interface intuitiva e a pensar unicamente no utilizador e nas interacções deste com o sistema.

Área de Aplicação: Especificação e desenvolvimento de aplicações móveis do mundo real, utilizando metodologias no âmbito da Engenharia de Software.

Palavras-Chave: Engenharia de Software, Aplicações móveis, Microsoft.

2 Introdução

2.1 Contextualização

No 2º semestre do 3º ano do Curso de Engenharia Informática da Universidade do Minho, existe uma unidade curricular denominada por Laboratórios de Informática IV, que tem como objectivo fundamentar, projectar e gerir o desenvolvimento de um sistema de software como também analisar e especificar de forma completa todos os requisitos operacionais e funcionais de um sistema de software e ainda desenvolver, testar, documentar e instalar sistemas de software.

Por exemplo, a quantos de nós já calhou na hora de ponta ir ao bar da universidade e ficar retido ou atrasado para uma aula ou para um compromisso.

Após algum debate sobre o assunto a opinião da equipa foi uniforme, a resposta a esta questão tratava-se de facto de uma necessidade da sociedade académica. Deste modo, deparamos-nos perante uma ideia inovadora que tinha o necessário para se tornar um sistema de software com potencialidade. Assim, através de uma necessidade surgiu a *MyBar*, uma aplicação móvel com um sistema para encontrar preços e encomendar em qualquer lugar.

Podendo ser adaptado para cafés, restaurantes ou para qualquer estabelecimento que ofereça serviços alimentares aos clientes.

O principal objectivo da aplicação é fornecer uma forma mais rápida e cómoda de obter um determinado produto que pretendemos do estabelecimento a que dirigimos efectuando o pedido pela nossa aplicação.

2.2 Apresentação do Caso de Estudo

A *MyBar* estará disponível ao público que possuem smartphones do tipo Android, sendo livre de quaisquer custos para o utilizador.

Uma vez que se tratará de uma aplicação móvel, o utilizador necessitará apenas de descargar a nossa aplicação. Neste momento, o utilizador estará a um pequeno passo de poder desfrutar de todas as suas vantagens. Para isso necessitará previamente de efectuar o registo na aplicação no qual terá de fornecer alguns dados relevantes ao seu reconhecimento, isto é, o seu nome, email e uma password, os quais terá de utilizar sempre que se pretender autenticar na aplicação. Todas estas informações serão directamente inseridas na base de dados da aplicação, após concluído o registo.

Sendo o login efectuado o utilizador depara-se com uma interface bastante simples e intuitiva que onde permite navegar. Assim, o utilizador pode facilmente ver produtos no qual a plataforma encarrega-se de procurar os produtos registados. Assim, desta lista, o utilizador terá de seleccionar aqueles que pretende acrescentar ao pedido logo de seguida encontra-se totalmente apto para efectuar um pedido. Após recolher o pedido e já depois de ter desfrutado do seu pedido, o utilizador pode e deve inserir na plataforma uma pequeno comentário da sua experiência no local em questão. Um utilizador poderá adicionar um determinado produto aos seus favoritos de modo a que possa, numa outra situação consultar tanto este como os seus pedidos mais recentes. Poderá ainda obter informações do estabelecimento para saber quando pode efectuar um pedido.

Para os utilizadores poderem usufruir é necessário existir funcionários para atenderem os pedidos, um funcionário necessita de efectuar o registo na aplicação no qual já foi inserido previamente. Um funcionário pode visualizar os pedidos que são feitos como mudar o estado dos mesmos. Após preparar o pedido notifica os clientes para recolherem o pedido. Caso sejam encarregues de abrirem ou fecharem o estabelecimento podem e devem alterar o estado do sistema permitindo ou impossibilitando de haver pedidos, respectivamente.

2.3 Motivação e Objectivos

O começo da implementação do projecto e até mesmo a própria decisão em avançar com o mesmo foi alvo de uma grande reflexão por parte de toda a equipa no sentido de melhorar a nossa visão da credibilidade do mesmo. Um dos principais motivos que levou a equipa a desenvolver o projecto foi o facto de realmente não haver disponível no mercado português nenhuma aplicação com funcionalidades iguais ou semelhantes, e desta forma acabamos por ter uma arma bastante poderosa do nosso lado que é a originalidade.

Para além disso, olhando do ponto de vista do cidadão comum, chegamos à conclusão que esta plataforma pode vir a tornar-se bastante útil para a comunidade, uma vez que é capaz encomendar em qualquer lugar de forma aproveitar sempre e só está pronto quando o utilizador estiver.

A elaboração desta aplicação mobile tem como principais objectivos a navegação pelo menu no próprio tempo do utilizador, personalizando o pedido e guardando os pedidos favoritos para poder pedir repetidamente com o toque de um botão. Não importa onde se peça, a comida será preparada quando apenas clica-se para efectuar o pedido. Encomendando e pagando pela comida na nossa aplicação e começa-se a prepará-la para o utilizador assim que ele quiser. Recolhendo a refeição no balcão.

De um modo mais geral e com base nos aspectos que referimos em cima, podemos chegar à conclusão que o desenvolvimento desta aplicação trará novas e deliciosas experiências e um conceito inovador para o ambiente mobile, facilitando assim a escolha e a procura da perfeito pedido aos futuros utilizadores da aplicação.

2.4 Estrutura do Relatório

Após apresentada a contextualização com que surge toda esta idealização da aplicação *MyBar*, do caso de estudo em que se insere e a motivação e objectivos que levaram a todo este enredo, as secções seguintes deste relatório fazem uma descrição mais detalhada de como o nosso grupo pretendeu conseguir que a aplicação tenha sucesso no mercado mobile.

Encontraremos, de seguida, a secção Justificação do Sistema, onde surge uma ampla explicação para a existência e principais objectivos da aplicação, evidenciando a unicidade da ideia por trás da mesma. Posteriormente encontraremos a Utilidade do Sistema, onde, muito detalhadamente, é explicada a sua utilidade, sendo completada pela análise de requisitos e funcionalidades do sistema, na qual são enumeradas as expectáveis funcionalidades do produto.

De seguida, entrando numa parte mais técnica, são identificados os recursos que a equipa achou necessários ao desenvolvimento do produto final. Terminando a parte da identificação, é apresentado o plano de desenvolvimento que a equipa pretendeu seguir nos meses de desenvolvimento como também uma análise sobre a viabilidade de crescimento futuro.

Logo após, surge a modelação de domínio com a ilustração e explicação do modelo de domínio elaborado pela equipa para uma mais concreta idealização do sistema.

Consequentemente, aparece a secção do modelo de use case, no qual são apresentados e especificados todos os use cases do sistema, assim como os actores intervenientes no mesmo. Logo depois, surge o diagrama de classes do nosso sistema. Ainda num registo de especificação do sistema, surge toda a elaboração e análise da base de dados associada à aplicação, desde a implementação do diagrama do modelo conceptual até à transição deste para o modelo físico correspondente.

Terminando a secção de especificação, são apresentados os vários mockups da interface da aplicação.

Na secção seguinte, começaremos com uma breve descrição da implementação de cada uma das ferramentas e técnicas utilizadas para obtermos o produto final. Posteriormente, apresentaremos a arquitectura final do software desenvolvido.

Por último apresentamos uma conclusão que refere os pontos que conseguimos implementar e os que não conseguimos, faremos também uma referência aos pontos deste projecto que nos permitiram aumentar o nosso conhecimento.

3 Fundamentação do Sistema

3.1 Justificação do Sistema

Num mundo cada vez mais evoluído, tanto em termos tecnológicos como intelectuais, o ser humano procura a todo o custo tornar o seu dia-a-dia menos cansativo e o mais confortável possível. Nesta linha de pensamento surge o *MyBar*, a aplicação que torna o acto de fazer um pedido rápido e agradável.

Para além de ser uma ideia bastante original, a escassez no que diz respeito a este tipo de aplicações à disposição do público foi o ponto que mais nos motivou e encorajou a avançar com o projecto em questão.

Deste modo, como dito anteriormente, a principal razão que motivaria uma pessoa a utilizar esta aplicação centra-se na comodidade e conveniência que esta oferece aos seus utilizadores. Todas as complicações e inconvenientes que surgem no momento de fazer um pedido no bar da universidade, são deste modo eliminados de uma forma bastante fácil e cómoda.

3.2 Utilidade do Sistema

O sistema assenta numa plataforma móvel que tem por finalidade ajudar os utilizadores a efectuar pedidos que melhor se enquadram com os seus interesses num determinado instante. Um utilizador a qualquer momento e em qualquer lugar pode recorrer a esta plataforma/aplicação sem quaisquer restrições ou custos associados.

Imaginemos que deslocamos a um estabelecimento e deparamos com uma enorme fila. De forma a melhorar consideravelmente a nossa experiência decidimos utilizar esta aplicação com o objectivo de fazer o pedido sem que para isso seja necessário gastar tempo na fila. Para isso basta iniciar a aplicação e logo de seguida especificar o que nos apetece naquele momento e o sistema imediatamente se encarregaria de efectuar o pedido. Depois do utilizador efectuar o pedido apenas tinha que o recolher.

3.3 Análise de Requisitos e Funcionalidades do Sistema

O *MyBar* é uma aplicação inserida na categoria alimentar que pode em algumas situações servir de apoio à gestão de um estabelecimento. Cada indivíduo desde que registado poderá consultar e encomendar produtos conforme as suas preferências. O ideal seria se, após cada pedido, o cliente desse a sua opinião do serviço prestado, melhorando desta maneira a qualidade da aplicação.

Segue uma breve lista dos requisitos funcionais que são esperados na aplicação:

- Autenticação;
- Registar-se na aplicação;
- Ver informações da empresa;
- Visualizar pedidos anteriores;
- Efectuar uma reclamação;
- Visualizar estado do pedidos;
- Acrescentar um produto à lista de favoritos;
- Visualizar produtos disponíveis;
- Efectuar Pedido;
- Consulta o nº do último pedido e do pedido a ser atendido;
- Mudar estado do sistema;
- Visualizar pedidos por preparar;
- Mudar o estado do pedido;
- Desautenticação;

Segue uma breve lista dos requisitos não funcionais que são esperados na aplicação:

- Interface amigável;
- Segurança dos dados dos utilizadores;
- Ordem dos pedidos tem uma abordagem FIFO;

Segue uma breve lista dos requisitos de negócio que são esperados na aplicação:

- Cobrar um montante mensal fixo à empresa que utilizar os nossos serviços;

4 Planeamento do Projecto

4.1 Identificação dos Recursos Necessários

No processo de formulação de qualquer aplicação é fundamental ter em conta os recursos necessários, sendo componentes cruciais ao bom funcionamento da aplicação, uma vez que sem eles seria impossível implementar o produto final. O recurso mais importante é a forma como iremos obter e inserir os dados propriamente ditos na nossa aplicação, isto é, as informações sobre os vários produtos e os pedidos que se encontram por atender.

Relativamente ao desenvolvimento da aplicação, tivemos obrigatoriamente fazer uso de várias ferramentas auxiliares da Microsoft e ainda externas. São exemplos *Visual Studio Community 2019*, mais concretamente a função de desenvolvimento de apps cross-platform “*Xamarin.forms*”, para desenvolver a interface gráfica da aplicação; *Overleaf*, para elaboração de todo o presente relatório; *Visual Paradigm*, para modelação de todo o sistema através de UML; *MySQL Workbench* para criação e gestão de toda a base de dados; incorporação da plataforma *OneSignal* de forma a notificar os clientes; e ainda *Pencil* para desenvolver os mockups da interface gráfica.

A linguagem de programação usada para implementar o código do servidor será o C#.

Concluindo, achamos que estes, no seu conjunto, foram todos os recursos necessários tendo em vista a colocação em prática daquilo que inicialmente projectamos para a nossa aplicação, e que achamos que foi suficiente para obtermos um resultado final bastante satisfatório.

4.2 Plano de Desenvolvimento

O desenvolvimento deste projecto foi realizado em três fases distintas: a fundamentação, a especificação e por fim, a construção de todo o software por trás do que será o *MyBar*.

A primeira fase passou pela idealização e fundamentação da aplicação, definindo um conjunto de pontos importantes ao seu desenvolvimento desde o contexto em que surge às diferentes medidas de sucesso expectadas para o seu futuro. Posteriormente foi realizado uma série de diagramas, constituindo os modelos de sistemas de software presentes na especificação que a equipa utilizará adiante na construção da aplicação. Todo este projecto terminou com a construção da própria aplicação seguindo à risca todos os pontos já definidos pela fundamentação e especificação anteriormente realizadas.

Podemos verificar que todo este processo exigiu da equipa os conhecimentos no âmbito da Engenharia de Software combinados com a prática frequente de programação de mobile software. A realização de um planeamento prévio de todas as tarefas foi imprescindível, e como tal, uma das primeiras preocupações pós-idealização foi a construção de um Diagrama de Gantt.

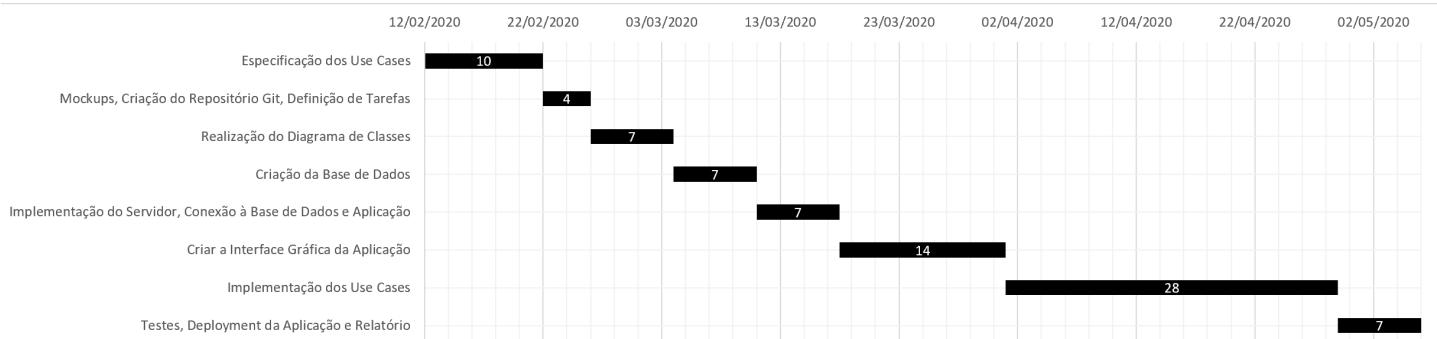


Figura 1: Diagrama de Gantt.

4.3 Viabilidade de Crescimento Futuro

A longevidade de uma aplicação depende, sobretudo, da sua capacidade em se adaptar a novos requisitos funcionais que surjam. A plataforma apresentada neste trabalho está limitada ao âmbito da universidade. Contudo, é possível imaginar novos requisitos, dentro do contexto alimentar em estudo que seriam facilmente implementáveis e tornariam a aplicação ainda mais útil para outros estabelecimentos.

5 Modelação De Domínio

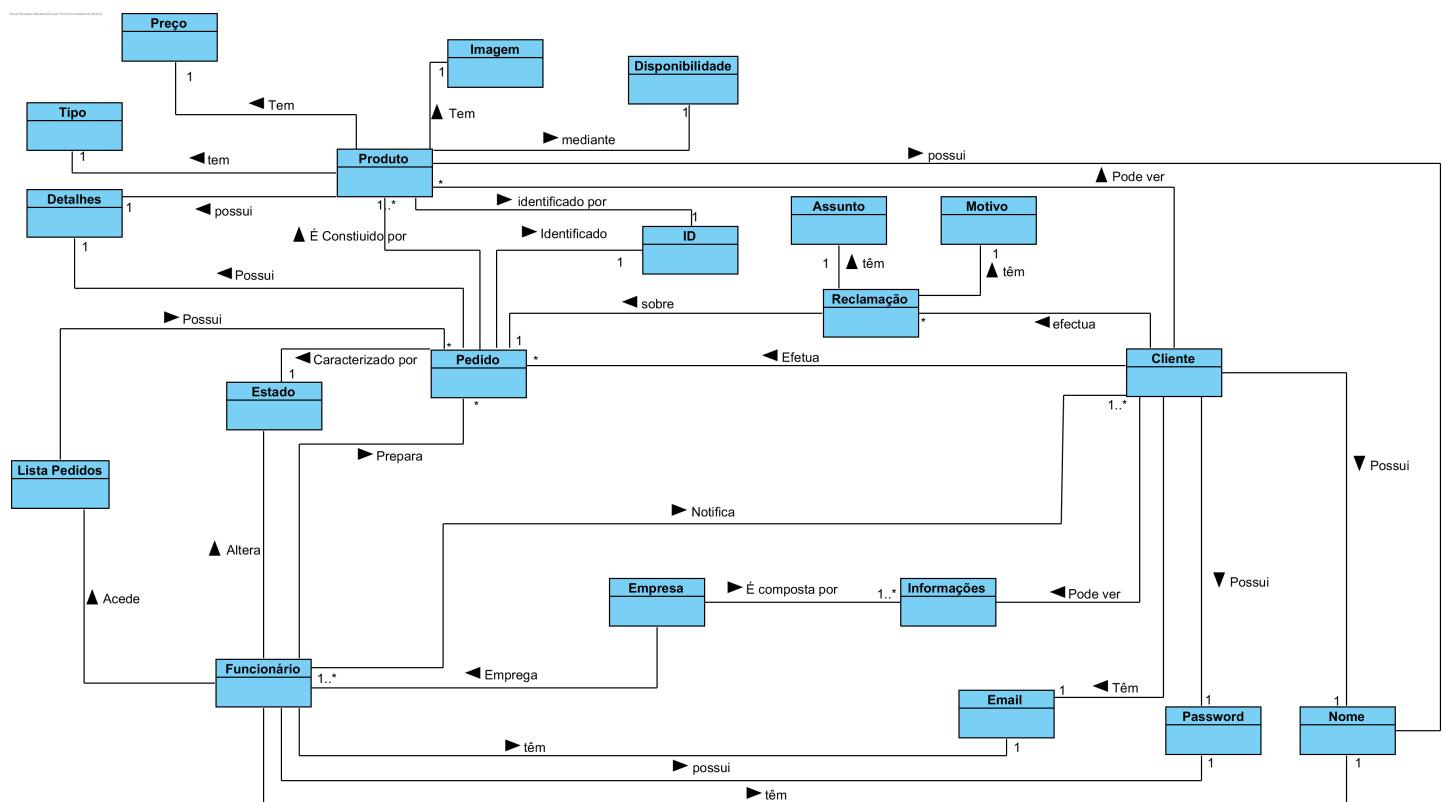


Figura 2: Diagrama do Modelo de Domínio.

Os Clientes, assim que registados no sistema, possuem três entidades relevantes: nome, email e uma password. O sistema deve, portanto, saber identificar correctamente cada Cliente para o seu correto funcionamento. É ainda de referir que, qualquer Cliente da aplicação pode efectuar pedidos e reclamações como também pode ver produtos e informações associadas à Empresa. Produtos esses que tem um nome, preço, imagem, disponibilidade, detalhes, um tipo e são identificados uns dos outros através de um ID. Estes Produtos fazem parte do Pedido caracterizado por um estado, detalhes e um identificador. Estes Pedidos são preparados por uma entidade do sistema denominada Funcionário que acede através de uma lista de pedidos por preparar. Um Funcionário tal como as outras identidades referida possui credenciais que permitem aceder à aplicação.

6 Modelo de Use Case

6.1 Diagrama

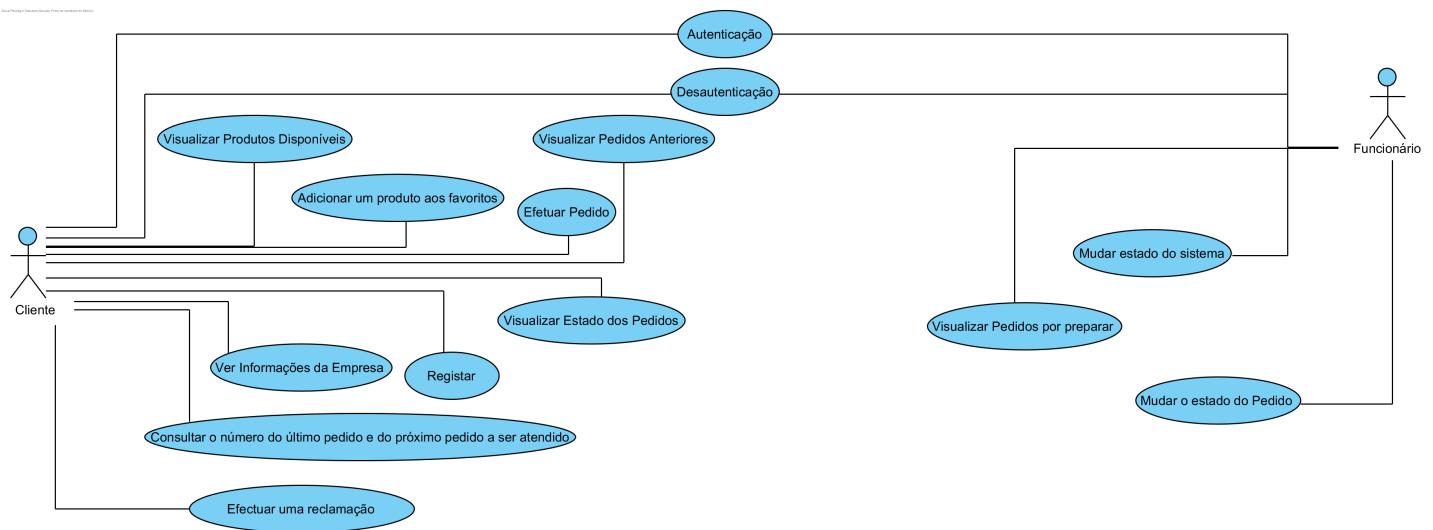


Figura 3: Diagrama de Use Cases.

6.2 Actores

6.2.1 Utilizador

Representa o utilizador que ainda não pode efectivamente desfrutar da aplicação, uma vez que ainda não tem o registo na mesma efectuado.

6.2.2 Cliente

Representa um utilizador pós-registo, isto é, com o registo devidamente concluído este pode agora usufruir de praticamente todas as funcionalidades da aplicação lhe tem para oferecer.

6.2.3 Funcionário

Representa o utilizador pós-registo, isto é, uma pessoa que exerce uma função remunerada no estabelecimento cabendo a responsabilidade por atender pedidos e alterar o estado dos mesmos como também alterar o estado sistema.

6.3 Use Cases

6.3.1 Autenticação

Use Case: Autenticação.

Descrição: O utilizador autentica-se no sistema.

Pré-condição: O utilizador não se encontra autenticado.

Pós-condição: O utilizador fica autenticado.

Fluxo Normal:

1. O utilizador pretende introduzir as suas credenciais;
2. O utilizador insere o seu email e password;
3. O sistema valida a combinação inserida;
4. O sistema apresenta o menu principal;

Fluxo Alternativo: [O utilizador, que é um cliente, ainda não se encontra registado] (Passo 1)

- 1.1. O cliente introduz o seu nome, email e a password;
- 1.2. O sistema valida o email introduzido;
- 1.3. O sistema cria uma nova conta com os dados fornecidos;

Fluxo de Excepção 1: [Combinação inválida] (Passo 3)

- 3.1. O sistema avisa sobre combinação inválida;

Fluxo de Excepção 2: [O email introduzido é inválido] (Passo 1.2)

- 1.2.1. O sistema informa que o email introduzido é inválido;

Fluxo de Excepção 3: [O email introduzido já está associado a outro utilizador] (Passo 1.2)

- 1.2.1. O sistema informa que o email introduzido já está associado a uma conta;

6.3.2 Desautenticação

Use Case: Desautenticação.

Descrição: O utilizador sai do sistema.

Pré-condição: O utilizador está autenticado.

Pós-condição: O utilizador deixa de estar autenticado.

Fluxo Normal:

1. O utilizador indica que pretende sair da aplicação;
2. Aplicação é encerrada;

6.3.3 Visualizar produtos disponíveis

Use Case: Visualizar produtos disponíveis.

Descrição: O cliente visualiza os produtos disponíveis.

Pré-condição: O cliente está autenticado na aplicação.

Pós-condição: O cliente conseguiu visualizar os produtos que queria.

Fluxo Normal:

1. O cliente selecciona a categoria dos produtos que pretende;
2. O sistema apresenta todos os produtos registados dessa categoria;

6.3.4 Acrescentar um produto à lista de favoritos

Use Case: Acrescentar um produto à lista de favoritos;.

Descrição: O cliente acrescenta um produto à lista de favoritos;.

Pré-condição: O cliente está autenticado.

Pós-condição: O cliente efectuou a adição de um novo produto à lista de favoritos

Fluxo normal:

1. O cliente selecciona o produto que quer adicionar;
2. O sistema apresenta as informações do mesmo
3. O cliente efectua adição;
4. O sistema regista o produto na lista de favoritos desse cliente;

Fluxo de Excepção: [O cliente cancela a adição] (Passo 2)

- 2.1. O cliente não adiciona o produto;
- 2.2. O sistema não efectua a adição;

6.3.5 Visualizar pedidos anteriores

Use Case: Visualizar pedidos anteriores.

Descrição: O cliente visualiza os pedidos que realizou anteriormente.

Pré-condição: O cliente está autenticado.

Pós-condição: O cliente consegue visualizar pedidos anteriores.

Fluxo Normal:

1. O cliente indica que pretende visualizar pedidos realizados;
2. O sistema apresenta a informação pretendida;

6.3.6 Efectuar pedido

Use Case: Efectuar pedido.

Descrição: O cliente efectua um pedido.

Pré-condição: O cliente está autenticado.

Pós-condição: O pedido é efectuado.

Fluxo normal:

1. O cliente selecciona os produtos que quer;
2. O cliente procede com o pagamento;
3. O sistema informa o cliente do número do pedido dele;

Fluxo Alternativo: [O cliente adiciona uma observação ao seu pedido] (Passo 2)

- 2.1. O cliente indica uma observação ao seu pedido;
- 2.2. O cliente procede com o pagamento;
- 2.3. Volta a 3;

6.3.7 Visualizar estado do pedidos

Use Case: Visualizar estado do pedidos.

Descrição: O cliente visualiza o estado do pedidos que ainda não foram entregues.

Pré-condição: O cliente está autenticado.

Pós-condição: O cliente consegue visualizar o estado.

Fluxo Normal:

1. O cliente indica que pretende visualizar estado dos pedidos;
2. O sistema apresenta o pedido com o actual estado em que se encontra;

6.3.8 Consultar o número do último pedido e do próximo pedido a ser atendido

Use Case: Consultar o número do último pedido e do próximo pedido a ser atendido.

Descrição: O cliente consulta o número do último pedido atendido e do próximo pedido a atender.

Pré-condição: O cliente está autenticado.

Pós-condição: O cliente consegue consultar o número dos últimos pedidos.

Fluxo normal:

1. O sistema mostra o número do último pedido atendido e do próximo pedido a atender;

6.3.9 Reclamações

Use Case: Reclamações.

Descrição: O cliente pretende efectuar uma reclamação referente a um pedido de um produto que já lhe foi entregue.

Pré-condição: O cliente está autenticado.

Pós-condição: O cliente efectuou a reclamação referente a um pedido de um produto que já lhe foi entregue.

Fluxo Normal:

1. O sistema mostra os pedidos que o cliente efectuou que já foram realizados e entregues ao cliente;
2. O cliente indica o pedido do qual quer reclamar;
3. O sistema pede motivo da reclamação e a reclamação em si;
4. O cliente insere o motivo e a reclamação;
5. O sistema regista a reclamação;

Fluxo de Excepção: [O cliente deseja cancelar a reclamação] (Passo 3)

- 3.1. O cliente indica que quer cancelar a reclamação;
- 3.2. O sistema cancela a reclamação;

6.3.10 Ver informações da Empresa

Use Case: Ver informações da Empresa.

Descrição: O cliente visualiza as informações sobre a empresa.

Pré-Condição: O cliente está autenticado.

Pós-Condição: O cliente visualiza as informações sobre a empresa.

Fluxo Normal:

1. O sistema mostra as informações da empresa;

6.3.11 Mudar estado do sistema

Use Case: Mudar estado do sistema.

Descrição: Um determinado funcionário altera o estado do sistema.

Pré-condição: Um funcionário está autenticado.

Pós-condição: O sistema ficou com um estado diferente do que se encontrava.

Fluxo normal:

1. O funcionário inicializa o sistema;
2. O sistema permite que os clientes façam pedidos;

Fluxo Alternativo: [O funcionário desliga o sistema] (Passo 1)

- 1.1. O cliente desliga o sistema;
- 1.2. O sistema não permite que os clientes façam mais pedidos;

6.3.12 Visualizar Pedidos por preparar

Use Case: Visualizar Pedidos por preparar.

Descrição: Um funcionário quer ver quantos pedidos faltam atender.

Pré-condição: O funcionário está autenticado.

Pós-condição: O funcionário conseguiu visualizar o que queria.

Fluxo normal:

1. O sistema mostra ao funcionário os pedidos e as respectivas informações dos mesmos que estão por atender;

6.3.13 Mudar o estado do pedido.

Use Case: Mudar o estado do pedido.

Descrição: Um funcionário pretende mudar um estado de um pedido.

Pré-condição: O funcionário está autenticado.

Pós-condição: O funcionário alterou o estado de um pedido.

Fluxo normal:

1. O sistema mostra os pedidos que estão por preparar ou em preparação;
2. O funcionário selecciona o pedido do qual deseja alterar o estado;
3. O sistema altera o estado do produto para o estado seguinte (estado por preparar->estado em preparação ->estado pronto);
4. O sistema notifica o cliente desta mudança;

7 Diagrama de Classes

A partir do modelo de domínio e do diagrama/especificação dos use case já esquematizados, conseguimos identificar as principais entidades e, consequentemente, potenciais candidatas a serem classes no desenvolvimento da nossa aplicação. Como tal, surge agora o momento de dar um passo em frente na modelação do sistema de software da *MyBar*, implementando para isso os designados diagramas de classe.

7.1 Cliente

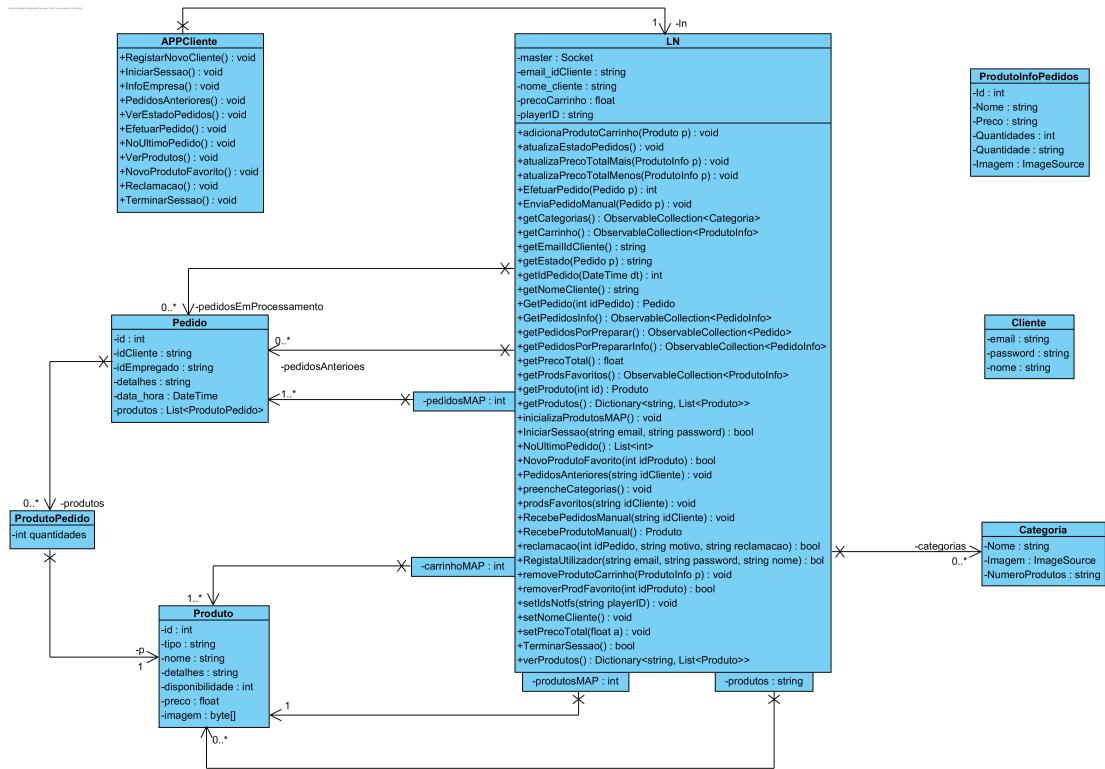


Figura 4: Diagrama de Classes do Cliente.

Começando pela classe principal, APPCliente, esta é a classe que vai gerir toda a comunicação com o Cliente comunicando com a classe LN que é encarregue de estabelecer contacto com o servidor, esta última possui informação sobre o Cliente no qual está registado como também os pedidos anteriores e em processamento e ainda o actual carrinho do mesmo (como o actual preço dele). Possui ainda conhecimento dos vários produtos como categorias existentes.

7.2 Funcionário

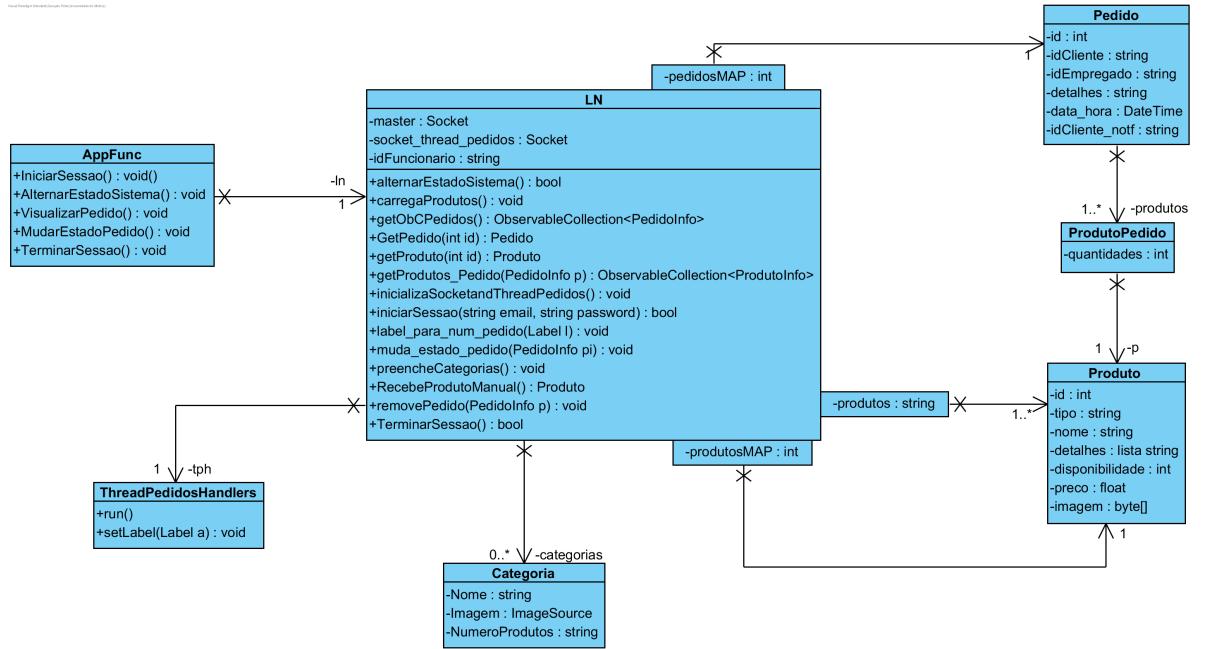


Figura 5: Diagrama de Classes do Funcionário.

Começando pela classe principal, AppFunc, esta é a classe que vai gerir toda a comunicação com o Funcionário comunicando com a classe LN que é encarregue de estabelecer contacto com o servidor, esta última possui informação sobre o funcionário, sobre os pedidos que não se encontram entregues, sobre os vários produtos registados e ainda as categorias. Possui também uma thread dedicada para atender pedidos.

7.3 Servidor

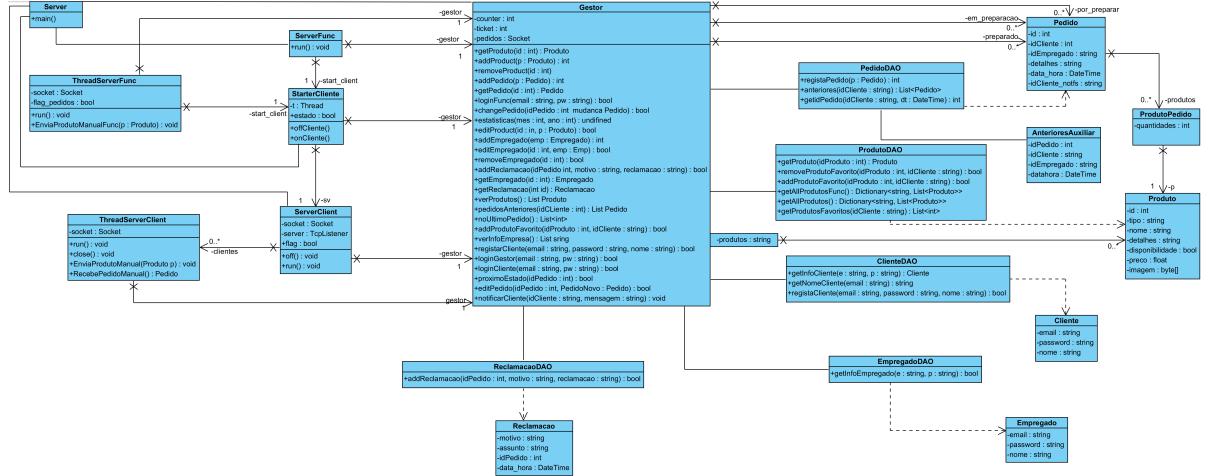


Figura 6: Diagrama de Classes do Servidor.

Começando pela classe principal, **Server**, esta é a classe que vai instanciar os diferentes servidores e criar a classe **Gestor**, esta funciona de intermediário entre os diferentes utilizadores conectados ao sistema através das classes **ThreadServer** (Func ou Client) e a informação armazenada. Para isso isolou-se a ligação à Base de Dados na Camada de Dados utilizando o padrão DAO(DataAccess Object) que implementa a camada de persistência permitindo aceder às diferentes classes criadas na aplicação que resolvem as funcionalidades propostas.

8 Base de Dados

8.1 Modelo Conceptual de Dados

A finalidade do modelo conceptual foi capturar os requisitos de informação e regras de negócio sob o ponto de vista do negócio. Para isto, tornou-se necessário o entendimento e a correta aplicação dos mecanismos de abstracção.

8.1.1 Identificação das Identidades

Entidades formam um conjunto de “coisas” com conceitos comuns às quais desejou-se armazenar os dados.

| Entidade | Descrição | Descrição |
|------------|--|--|
| Cliente | Termo geral que descreve todos clientes do nosso sistema. | Um cliente possui nome, email, password. |
| Produto | Termo geral que descreve todos os produtos do nosso sistema. | Um produto possui um identificador, um tipo, um nome, detalhes, disponibilidade, preço e uma imagem. |
| Pedido | Termo geral que descreve todos os pedidos do nosso sistema. | Um pedido possui um identificador e uma data. |
| Empregado | Termo geral que descreve todos os empregados do nosso sistema. | Um empregado possui um email, password, um nome e um identificador a indicar se é gestor ou não. |
| Reclamação | Termo geral que descreve todas reclamações do nosso sistema. | Uma reclamação possui um motivo, um assunto e uma data. |

Tabela 1: Identificação e caracterização das potenciais entidades envolvidas

8.1.2 Identificação dos Relacionamentos

Relacionamentos são associações entre entidades com um significado específico dentro do mundo real. Os objectos do mundo real não ocorrem de forma isolada, eles associam-se ou vinculam-se. Desta forma criou-se os possíveis relacionamentos entre as entidades acima apresentadas.

| Entidade 1 | Multiplicidade | Relação (1 para 2) | Multiplicidade | Entidade 2 |
|------------|----------------|--------------------|----------------|------------|
| Cliente | N | adquire | N | Produto |
| Cliente | N | gosta de | N | Produto |
| Cliente | 1 | efectua | N | Pedido |
| Produto | N | faz parte | N | Pedido |
| Empregado | 1 | encarrega-se | N | Pedido |
| Reclamação | 1 | refere-se | 1 | Pedido |

Tabela 2: Caracterização dos diversos relacionamentos que possam existir entre as entidades estabelecidas.

8.1.3 Diagrama do Modelo Conceptual

Com base nos requisitos apresentados, nas potenciais entidades envolvidas e os diversos relacionamentos concebeu-se o seguinte diagrama para a base de dados requerida:

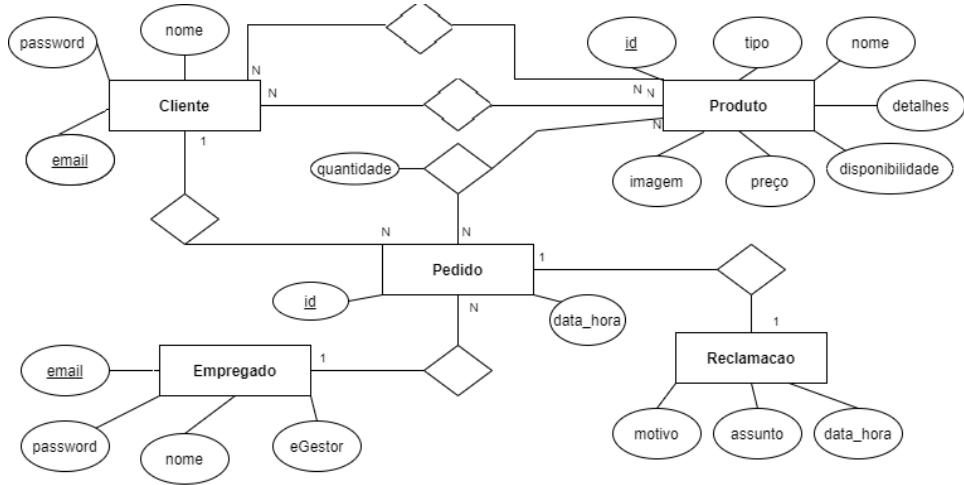


Figura 7: Diagrama do Modelo Conceptual.

As entidades foram representadas através de um rectângulo com o nome da entidade escrito em seu centro. Os relacionamentos são apresentados através de losangos. Todos os nossos relacionamentos são binários, isto é, apenas duas entidades participam no relacionamento. Os atributos são informações que caracterizam as entidades e os relacionamentos, desta forma eles encontram-se representados por círculos ligados as entidades. O atributo identificador é o que permite distinguir as diferentes identidades uma das outras, encontra-se sublinhado.

8.2 Modelo Lógico de Dados

8.2.1 Conversão do esquema conceptual para o esquema lógico

Depois de criado o modelo conceptual, o próximo passo foi a criação do modelo lógico. Utilizando a ferramenta visual de design de base de dados que integra desenvolvimento, administração, design, criação e manutenção de SQL em um único ambiente de desenvolvimento integrado para o sistema de base de dados MySQL, “MySQLWorkbench”, convertemos o esquema conceptual produzido na alínea anterior para o seu correspondente esquema lógico.

Seguindo as seguintes regras, todas as entidades viram uma tabela; um atributo identificador torna-se chave primária na tabela; atributos compostos tornam-se atributos simples, mapeados em colunas, uma coluna para cada atributo; atributos derivados não são mapeados; atributos multi-valorados dão origem a uma nova tabela.

Em relação aos relacionamentos, num relacionamento 1:1, escolhe-se uma das relações para receber a chave estrangeira, chave essa primária da relação que ocorre o relacionamento. Em um relacionamento 1:N, escolhe-se a relação “N” e inclui como chave estrangeira a chave primária da outra relação.. Em um relacionamento N:N, cria-se uma tabela que inclui como chave estrangeira, as chaves primárias das relações em que ocorre o relacionamento.

8.2.2 Diagrama do Modelo Lógico

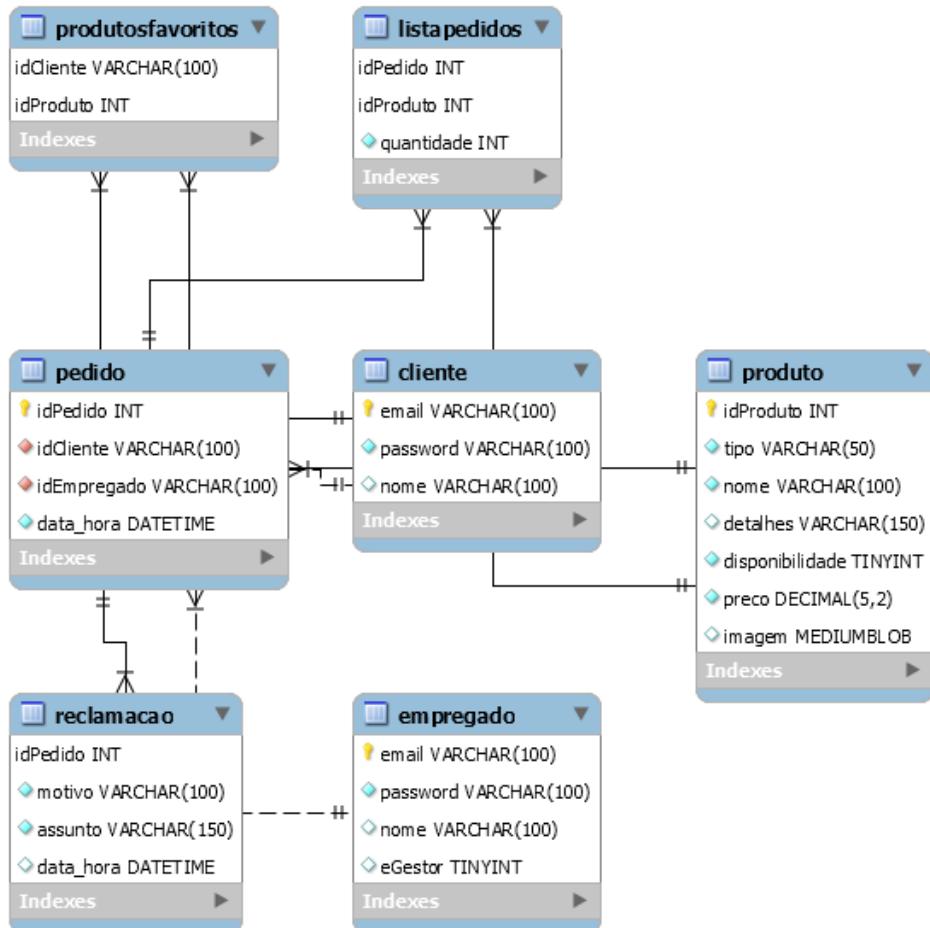


Figura 8: Diagrama do Modelo Lógico.

8.3 Modelo Físico de Dados

8.3.1 Conversão do esquema lógico para o esquema físico

O sistema de gestão de base de dados escolhido foi o “MySQL”, como foi dito anteriormente, porque é um sistema que apresenta um elevado desempenho, é robusto e permite vários utilizadores. Partindo do modelo criado no “MySQL Workbench” como foi dito anteriormente, para traduzir o esquema lógico para o esquema físico automaticamente utilizou-se a opção “forward engineer”.

9 Interface

Nesta secção segue-se a interface idealizada pela equipa do *MyBar*. Toda a interface da aplicação foi pensada e construída de modo a ser o mais user-friendly possível, isto é, uma aplicação com um uso muito intuitivo proporcionando ao utilizador uma adaptação fácil e rápida.

Todos os botões, caixas de formulário e diversos tipos de apresentação de informações foram conseguidos com um alto nível de consistência, sendo partilhados por múltiplos menus, mantendo assim a sua utilidade e criando um ambiente apelativo ao utilizador.

A interface foi também desenvolvida a pensar em futuras alterações, desde novas funcionalidades a alterações na estrutura do sistema, sendo, portanto, uma interface flexível às futuras adversidades pelas quais a equipa terá de ultrapassar.

9.1 Menu Inicial



Figura 9: Mockup - Menu Inicial.

9.2 Autenticar



Figura 10: Mockup - Autenticar.

9.3 Aplicação do Cliente

9.3.1 Criar Conta



Figura 11: Mockup - Criar Conta.

9.3.2 Menu



Figura 12: Mockup - Menu Cliente.

9.3.3 Pedido Cliente



Figura 13: Mockup - Pedido Cliente.

9.3.4 Ver Produtos



Figura 14: Mockup - Ver Produtos.

9.3.5 Lista Favoritos



Figura 15: Mockup - Lista Favoritos.

9.3.6 Histórico



Figura 16: Mockup - Histórico.

9.3.7 Reclamações



Figura 17: Mockup - Reclamações.

9.3.8 Sobre



Figura 18: Mockup - Sobre.

9.4 Aplicação do Funcionário

9.4.1 Menu

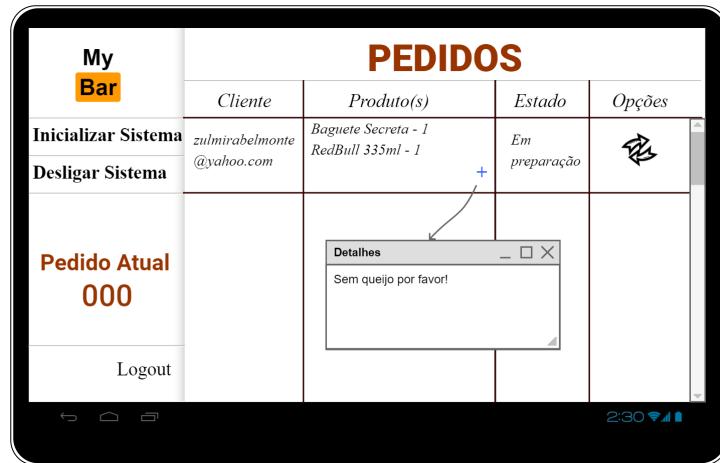


Figura 19: Mockup - Menu Funcionário.

10 Implementação das funcionalidades/tecnologias

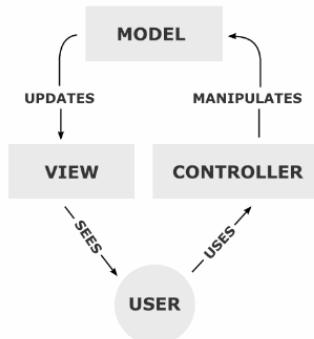
De modo a que a aplicação cumprisse os requisitos inicialmente definidos tivemos de recorrer a diversas ferramentas e técnicas que nos possibilitaram tornar o seu processo de criação bastante mais fluído e possível.

Recordando novamente que a linguagem de programação usada para implementar para todo o sistema foi o C#. E em termos da interface gráfica da aplicação usou-se a função de desenvolvimento de apps cross-platform “*Xamarin.forms*”.

Nesta secção iremos apresentar todas estas técnicas e ferramentas utilizadas para a criação do sistema de software final.

10.1 Modelo MVC

Para a elaboração deste projecto baseamo-nos no padrão Model-View-Controller (MVC), que é um padrão de arquitectura de software que separa as funcionalidades do sistema em três camadas distintas:



O modelo (*Model*) representa a camada de negócio, que consiste nos dados da aplicação, lógica e funções.

A visão (*View*), que faz parte da camada da interface, é aquela que possui todas as funcionalidades para a apresentação da informação ao utilizador da aplicação, sendo esta informação fornecida à *view* pelo controlador.

O controlador (*Controller*), que faz parte da camada de input, é responsável por controlar todo o fluxo de informação que flui no sistema. Ou seja, é o controlador que controla e comanda a interação entre os modelos e as várias views, funcionando como um intermediário.

Com este padrão foi possível obter uma separação de conceitos, sendo mais fácil para nós fazer qualquer tipo de alterações no código que estava a ser desenvolvido.

10.2 Servidor

Efectuar e atender pedidos foi algo extremamente delicado, principalmente porque pretendeu-se armazenar estes para um consulta posterior. E a situação é ainda mais delicada quando existem dois ou mais clientes a tentarem efectuar pedidos. Desta forma, foi fundamental e óbvio para o grupo que devia existir uma sistema interno, evitando, assim, a chance dos pedidos perderem-se. Sistema interno esse passou pela existência de um servidor de forma a gerir, armazenar, enviar informações e processar dados o tempo todo.

Actualmente existem muitas alternativas que podemos encontrar em *cloud* desde a priorização ao armazenamento dos dados até à recuperação de desastres existe um servidor pronto a usar.

Contudo o nosso grupo optou por criar um **servidor local de raiz** pois apesar de sabermos que temos que ter mais responsabilidade temos muito mais controlo na troca de informação, além do facto que não gastou-se fundos nem se esteve limitado nas capacidades fornecidas. Realçando o factor de desta forma garantimos maior segurança aos utilizadores.

10.2.1 Funcionamento

Desta forma, só podemos ter partido das aplicações, tanto a do cliente e a do funcionário, se o Servidor estiver em execução. Após colocarmos em execução o Servidor, este vai lançar dois servidores mais pequenos, um para cada uma das aplicações responsável por atender e responder aos pedidos da respectiva aplicação.

10.2.2 Multiplexagem e segurança

De forma a receberem pedidos de um cliente/funcionário (depende da aplicação pelo qual estão responsáveis), os servidores criam uma thread responsável por tratar de todos os pedidos da respectiva entidade a conectar-se, por exemplo: autenticação, ver produtos, etc.

Este comportamento do servidor permite que sejam atendidos várias entidades do sistema ao mesmo tempo, ou seja, que o servidor criado tenha **multiplexagem**, e como cada entidade possui uma conexão ao servidor também garantimos a segurança e privacidade porque a informação direcionada a uma entidade só chega a essa mesma entidade.

10.2.3 Conexões e forma como comunicam

Toda a comunicação entre as aplicações e o servidor ocorre por **sockets TCP**, o que nos garante que todas as comunicações beneficiam das suas vantagens, ou seja, garantia que a informação chega ordenada e que não existe perdas de informação.

A comunicação neste sistema começa sempre pela entidade que pretende obter informação, enviando um identificador da operação pretendida para a outra aplicação e a aplicação que recebe este identificador responde-lhe com o respectivo comportamento a esse comando.

10.2.4 Gestão da informação

O servidor para responder a todos os pedidos das aplicações tem de guardar a informação, neste seguimento podemos distinguir esta informação em dois tipos sendo cada uma tratada de maneira diferente:

- **informação temporária** : considerando que o servidor é reiniciado todos os dias no final de um dia de trabalho, esta informação é descartada, por exemplo o caso do ticket actual;
- **informação permanente** : que não queremos que se perca de forma alguma, é o caso das informações do utilizador, registos dos pedidos, etc;

No caso do primeiro tipo de informação esta foi guardada em variáveis numa classe do servidor (nomeadamente na classe gestor), se for o outro tipo de informação esta é colocada na base de dados apresentada anteriormente.

Um caso especial da informação permanente é a gestão dos pedidos, quando um pedido é feito, num pequeno intervalo de tempo são feitas várias alterações ao estado dele, por isso mesmo, para não estarmos a fazer constantemente consultas à base de dados, este é guardado temporariamente na classe gestor do servidor até chegar ao estado final (pronto a levantar), sendo que quando pedido for entregue é armazenado na base de dados.

10.3 Inclusão da Base de Dados

Para a implementação da base de dados do projecto recorremos, como já indicado, a uma base de dados relacional, suportada pelo *MySQLWorkbench* que nos permitiu armazenar os dados, e ao mesmo tempo garantir eficiência nas consultas e transacções aplicadas à BD.

Para isso tivemos que instalar uma extensão no *Visual Studio Community 2019* que permitiu comunicar com a conexão criada no *MySQLWorkbench* com o hostname e root correto.

Tal como foi referido previamente o que se propôs a fazer foi isolar a ligação à Base de Dados na Camada de Dados. Para isso utilizou-se o padrão DAO(DataAccess Object) que implementa a camada de persistência. DAOs são classes que persistem objectos em Bases de Dados, criando objectos a partir da informação na Base de Dados e encapsulando queries SQL. Podendo ainda persistir uma classe ou várias classes. Resumindo, são as fachadas da Camada de Dados.

Desta forma o servidor acede a estas classes com esta denominação para armazenar ou obter informação de forma a cumprir as diferentes funcionalidades propostas.

10.4 Incorporação da API externa OneSignal

De modo a conseguir enviar uma notificação para o cliente de que o seu pedido está pronto a ser levantado, utilizou-se a API externa OneSignal, pois é uma das plataformas de notificações push mais populares do mercado, e além de ser fácil de utilizar é também gratuita.

Para a notificação chegar ao cliente, é gerado um identificador do dispositivo a ser utilizado, este é enviado junto de cada pedido ao servidor, o servidor reencaminha esse pedido para a aplicação do funcionário, esta por sua vez altera o estado do pedido como foi referido previamente. Quando o pedido atingir um estado de conclusão faltando apenas que seja levantado, a aplicação do funcionário trata de enviar uma mensagem para o servidor com o tal identificador que foi associado ao cliente, o servidor com este identificador faz um "HTTP POST" com um ficheiro JSON com toda a informação necessária para que a plataforma OneSignal consiga efectuar a notificação.

11 Produto Final

11.1 Aplicação do Cliente

11.1.1 Carregar a aplicação

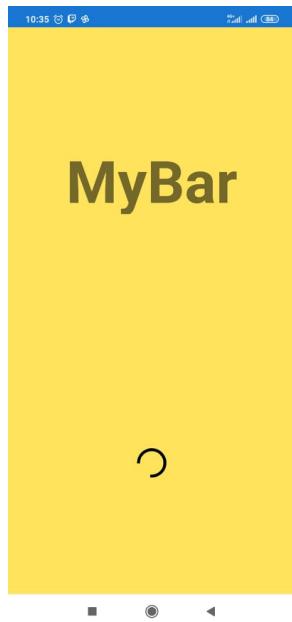


Figura 20: Interface - Carregar aplicação.

11.1.2 Criar Conta



Figura 21: Interface - Criar Conta.

11.1.3 Autenticar



Figura 22: Interface - Autenticar.

11.1.4 Menu Cliente



Figura 23: Interface - Menu Cliente.

11.1.5 Ver Produtos (Bebidas)

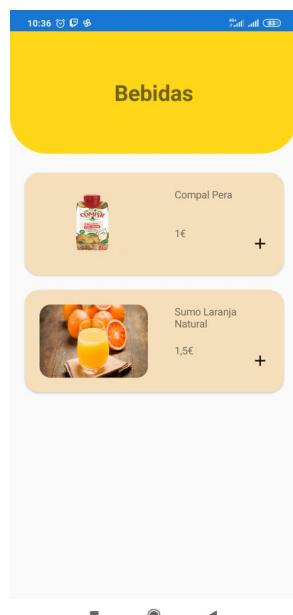


Figura 24: Interface - Ver Produtos (bebidas).

11.1.6 Ver Produtos (Compal Pera)

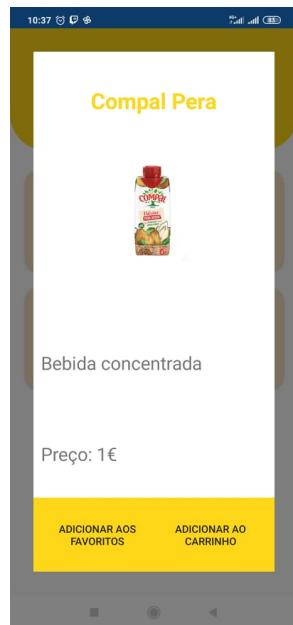


Figura 25: Interface - Ver produto (compal pera).

11.1.7 Adicionar ao carrinho

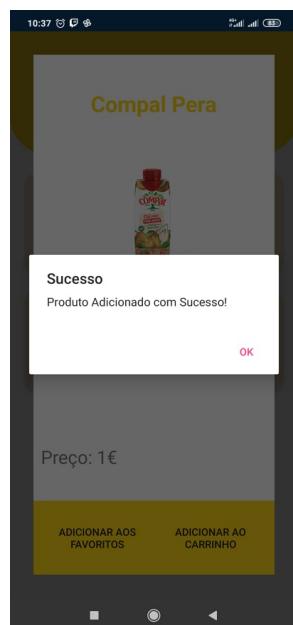


Figura 26: Interface - Adicionar ao carrinho.

11.1.8 Ver carrinho

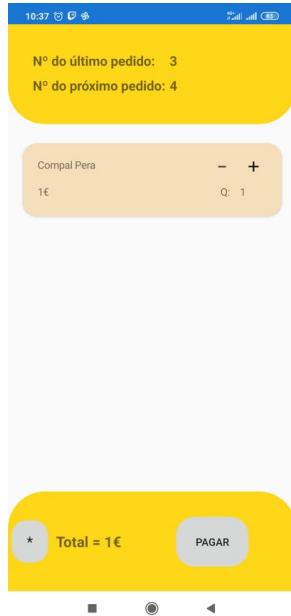


Figura 27: Interface - Ver carrinho.

11.1.9 Fazer Pedido

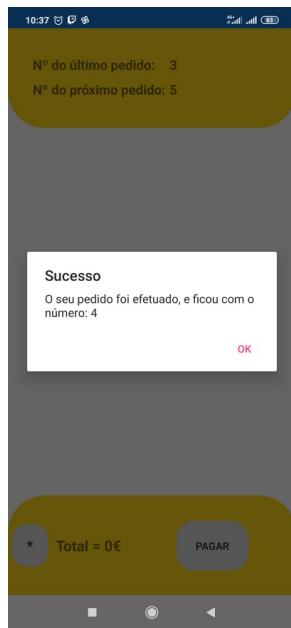


Figura 28: Interface - Fazer Pedido.

11.1.10 Estado Pedido (por preparar)



Figura 29: Interface - Estado Pedido (por preparar).

11.1.11 Notificação

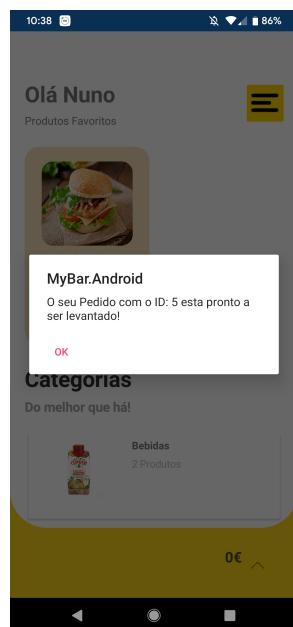


Figura 30: Interface - Notificação.

11.1.12 Histórico



Figura 31: Interface - Histórico.

11.1.13 Sobre



Figura 32: Interface - Sobre.

11.2 Aplicação do Funcionário

11.2.1 Autenticar

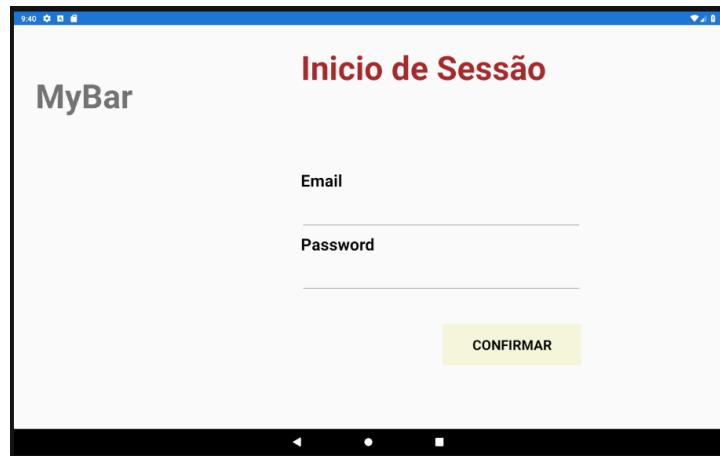


Figura 33: Interface - Autenticar.

11.2.2 Menu Inicial

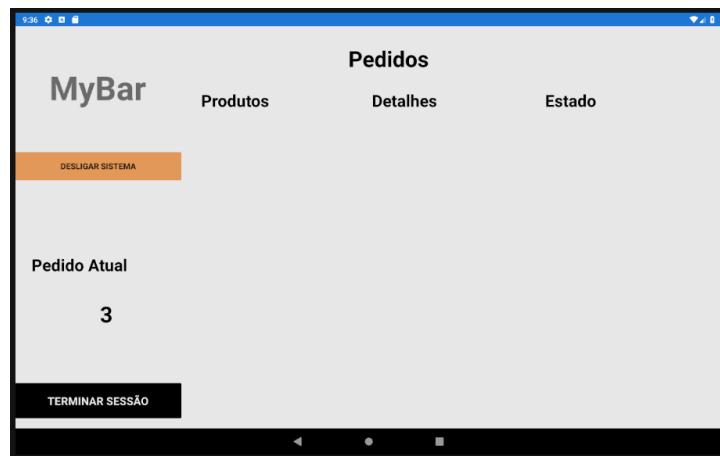


Figura 34: Interface - Menu Inicial.

11.2.3 Menu Inicial (com pedidos)

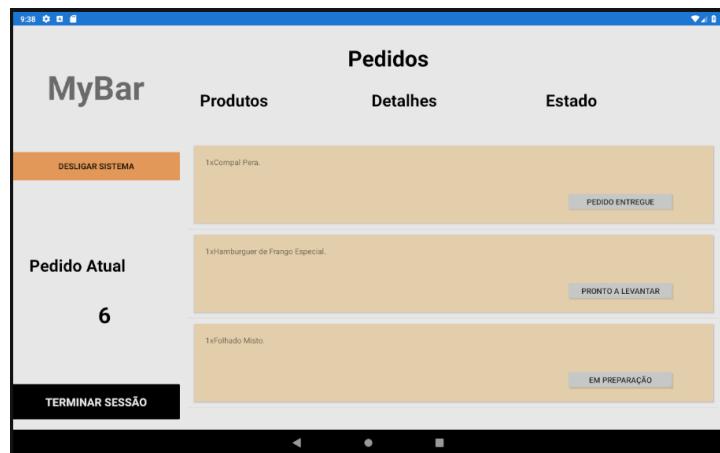


Figura 35: Interface - Menu Inicial (com pedidos).

11.2.4 Informação de um pedido

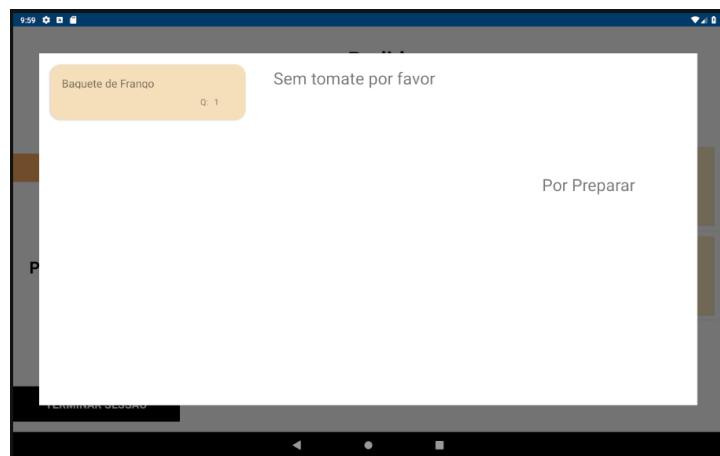


Figura 36: Interface - Informação de um pedido.

12 Discussão

Considerando que nunca antes tínhamos tido contacto com muitas das ferramentas e recursos que foram imprescindíveis para a correta implementação da aplicação achamos que o nosso produto final foi aceitável.

Com o escasso tempo que tivemos ao nosso dispor, fomos forçados a aprender uma nova linguagem de programação que o foi C# juntamente com a ferramenta Xamarin para desenvolver a parte gráfica. O mesmo se passou com a utilização de um servidor local nunca tínhamos criado algo complexo no processo de elaboração de uma aplicação e que, desta forma, nos consumiu bastante tempo de modo a encontrarmos a melhor forma de utilizar. Apesar de sabermos que era possível utilizar um servidor remoto o qual facilitava nos bastante o processo de troca de informação mas devido à pouca experiência na área aplicou-se os conhecimentos previamente adquiridos nas outras unidades curriculares.

Em suma, o produto obtido está pronto a ir para o mercado contudo poderia-se ter adicionado algumas funcionalidades que trariam ainda mais destaque, por exemplo, na aplicação do cliente poderia-se implementar a dinâmica do pagamento, quer este fosse em loja ou numa plataforma de pagamento online, através da incorporação da API do PayPal, por exemplo.

Em relação à aplicação do funcionário não ficou com uma interface gráfica muito desenvolvida, mas achamos boa o suficiente para o uso diário que esta poderá ter.

Em relação à aplicação do cliente tentou-se que a interface gráfica fosse o mais agradável possível e intuitiva de se olhar e utilizar, respectivamente. Temos consciência que mais funcionalidades poderiam ter sido implementadas tais como a possibilidade de alterar um pedido (se este ainda não tivesse sido atendido ainda) e a habilidade de avaliar um ou mais produtos e respectivamente conseguir visualizar os produtos com melhor votação.

Numa perspectiva futura e realista poderia-se criar uma nova aplicação destinada aos gestores dos estabelecimentos, sendo que nesta poderia ser possível adicionar novos produtos, removê-los ou editá-los, assim como adicionar ou remover empregados e efectuar consultas estatísticas.

13 Conclusões

Nunca antes em nenhuma outra Unidade Curricular tivemos uma abordagem a um produto de software como agora pois nunca tivemos de realizar uma fundamentação/idealização tão elaborada. Entendemos, no entanto, a sua necessidade uma vez que se trata de um produto inovador e começado de raiz. Também essencial termos em consideração diversos outros factores, entre dos quais a utilização de diversas ferramentas (APIs, por exemplo) com que a equipa jamais tivera contacto e a adaptação a uma nova linguagem/método de programação de modo a conseguirmos satisfazer os nossos próprios objectivos.

Durante a elaboração da aplicação propriamente dita, vários foram os obstáculos que nos foram surgindo, aos quais tivemos de dar a volta e seguir em frente em direcção ao nosso objectivo. Uma das grandes dificuldades com que nos debatemos esteve relacionada com a Pandemia de COVID-19 pois deparamos com uma nova situação para todos os elementos o que impossibilitou a reunião da equipa de trabalho dificultando a comunicação, consenso de certas ideias/problemas que foram surgindo como o próprio plano de desenvolvimento foi afectado. Outra das dificuldades prendeu-se com o facto de a plataforma escolhida para desenvolver a interface gráfica não ser muito intuitiva e prática de utilizar. Apesar da escassez de tempo e das várias dificuldades com que nos fomos debatendo, conseguimos chegar a um produto final que obedece aos requisitos originalmente previstos. A aplicação do cliente permite a qualquer utilizador ver produtos, efectuar um pedido, acrescentar um produto aos favoritos, adicionar comentários aos pedidos, ver o estado dos pedidos e ver pedidos anteriores. E a aplicação do funcionário permite alterar o estado do sistema e principalmente atender pedidos.

Sendo este o primeiro trabalho em que realmente tivemos de ser nós a aprender o funcionamento de grande parte das ferramentas utilizadas e não aplicar os conhecimentos adquiridos durante as aulas, por um lado foi bom pois possibilitou-nos desenvolver capacidades de auto-aprendizagem e autocrítica, mas por outro lado tornou o processo muito mais demorado e o produto final não tão completo como o desejado. Apesar disso, estamos orgulhosos com o produto final aqui desenvolvido e pensamos que todo este trabalho valeu realmente a pena, na medida em que enriqueceu todos os elementos do grupo.

Concluindo, agora que finalmente atingimos o patamar final da elaboração deste projecto, recordando o inicio do mesmo apercebemos-nos o quanto importante este trabalho prático foi para todos os elementos do grupo, na medida em que, para além de nos ter possibilitado adquirir um inúmero conhecimento no que toca à implementação de sistemas de software e permitiu-nos ficar um pouco a conhecer como é que funciona o mundo de trabalho no que diz respeito a esta área tecnológica.