

Universidade do Minho

Escola de Engenharia

Departamento de Informática

Rui Pedro Neto Reis - A84930

Diogo Pinto Ribeiro - A84442

Rui Filipe Moreira Mendes - A83712

João Miguel Fernandes Cerqueira Soares - A83581

Rui Nuno Borges Cruz Oliveira - A83610

**Sistema de Gestão de Restaurantes
SGR**

Dezembro 2022

RESUMO

No contexto da Unidade Curricular de Laboratórios de Informática IV foi desenvolvido um Sistema de Gestão de Restaurantes com o intuito de potencialmente ser utilizado em cenários reais. Um restaurante precisa de conseguir gerir muitas áreas: mercadoria, funcionários, menus, entre outros. É nesses pontos que o nosso sistema se torna útil, permitindo automatizar e facilitar diversos processos do dia-a-dia. Neste relatório iremos apresentar todas as fases de desenvolvimento deste projeto, bem como as dificuldades com as quais nos deparamos, bem como as decisões e implementações que realizamos. No entanto, a documentação de código, optámos por não incluir neste relatório, e assim, ter uma [Wiki](#) completa no repositório do projeto.

CONTEÚDO

1	INTRODUÇÃO	1
1.1	Contexto	1
1.2	Motivação	1
1.3	Objetivos	1
1.4	Relatório	2
2	PLANEAMENTO	3
2.1	Sistema de Gestão de Restaurantes	3
2.2	Funcionalidades	3
2.3	Estrutura	4
2.4	Tecnologias	4
2.5	Inovação	5
3	DESENVOLVIMENTO	6
3.1	Etapa 1 — Definição do Projeto	6
3.2	Etapa 2 — Modelação e Requisitos	7
3.2.1	Requisitos BackOffice	8
3.2.2	Requisitos Mobile	9
3.3	Etapa 3 — Modelação e Design	9
3.4	Etapa 4 — Implementações Iniciais	13
3.5	Etapa 5 — Login e Protótipos Funcionais	13
3.6	Etapa 6 — Gestão e Melhorias	14
3.7	Etapa 7 — Detalhes e Correção de Erros	15
3.8	Etapa 8 — Documentação e Ajustes Finais	15
3.9	Considerações sobre o Desenvolvimento do Projeto	16
4	RESULTADOS	17
4.1	BackOffice	17
4.2	Aplicação	20
5	CONCLUSÃO	23
5.1	Objetivos e Resultados Obtidos	23
5.2	Trabalho Futuro	23

LISTA DE FIGURAS

Figura 1	Diagrama de Gantt Final	7
Figura 2	Modelo de Domínio	8
Figura 3	Diagrama de Classes sem DAO	10
Figura 4	Diagrama de Classes com DAO	10
Figura 5	Modelo Final da Base de Dados	11
Figura 6	BackOffice: Adicionar Funcionário	12
Figura 7	BackOffice: Eliminar Funcionário	12
Figura 8	Mobile: Adicionar Artigo e Login	12
Figura 9	BackOffice: Login	14
Figura 10	BackOffice: Página Inicial I	17
Figura 11	BackOffice: Página Inicial II	18
Figura 12	BackOffice: Listar Funcionários	18
Figura 13	BackOffice: Adicionar Artigo	19
Figura 14	BackOffice: Ver Artigo	19
Figura 15	BackOffice: Editar Artigo	19
Figura 16	BackOffice: Apagar Artigo	20
Figura 17	Aplicação: Login	20
Figura 18	Aplicação: Serviço Mesa e Balcão	21
Figura 19	Aplicação: Ver Pedidos e Conteúdo do Pedido	21
Figura 20	Aplicação: Confirmação de Pedido	22
Figura 21	Aplicação: Categorias de Artigos	22

INTRODUÇÃO

1.1 CONTEXTO

No contexto da Unidade Curricular de Laboratórios de Informática IV, foi-nos proposto escolher um tema para um projeto que pudesse vir a ser utilizado num cenário real. Nessa ótica propusemo-nos a criar um sistema que servisse de apoio a um restaurante.

O processo de gerir um restaurante é complexo e demorado sendo que existem diversas ineficiências. Essa foi a ideia que nos fez optar por este tema.

Sendo que uma boa parte dos restaurantes não possuem sistemas informáticos ou possuindo algum, estes estão maioritariamente desatualizados, achámos que a nossa ideia se enquadra no mercado que existe, sendo que não encontrámos evidências de nenhum software que representasse a nossa ideia.

1.2 MOTIVAÇÃO

A nossa motivação neste projeto passa por diversos pontos, nomeadamente dois grandes grupos: produzir um sistema de qualidade e adquirir novas competências, bem como melhorar as que temos.

A nossa motivação em produzir um sistema de qualidade consiste em querermos desenvolver um sistema que cumpra os requisitos definidos e possa efetivamente vir a ser usado, e, não seja mais um software qualquer que acaba por "morrer" pelo facto de não fazer o que seria esperado.

A nossa motivação em adquirir novas competências deriva da nossa ambição em aprender novas e melhores metodologias de desenvolvimento de software, trabalho em equipa e aprimorar o nosso leque de conhecimentos técnicos.

1.3 OBJETIVOS

Com a elaboração deste projeto, logo nas fases iniciais, definimos alguns objetivos que gostaríamos de alcançar ao longo das fases de desenvolvimento. Esses objetivos não são

requisitos do sistema em si, mas sim objetivos do projeto em si. Assim sendo os nossos objetivos passam por:

- Cumprir todos os prazos
- Desenvolver todos os requisitos
- Ter um produto funcional e apelativo
- Ter um produto pronto a implementar num cenário real
- Adquirir novas competências em ferramentas novas

1.4 RELATÓRIO

O presente relatório é composto por 5 capítulos sendo que não inclui a documentação do código, dado que esta se encontra na wiki do projeto. Assim sendo, os capítulos existentes são:

- Introdução
- Planeamento
- Desenvolvimento
- Sistema
- Conclusão

Na **Introdução** iremos abordar a temática do projeto bem com alguns aspetos essenciais. No **Planeamento** iremos detalhar todo o processo de definição da ideia, escolhas que fizemos e outras questões importantes. No capítulo do **Desenvolvimento** iremos detalhar o nosso progresso a cada reunião feita com o docente da UC. No capítulo **Sistema** iremos apresentar o produto final e os resultados obtidos. Na **Conclusão** iremos tecer alguns comentários sobre o projeto e fazer algumas reflexões gerais.

PLANEAMENTO

2.1 SISTEMA DE GESTÃO DE RESTAURANTES

De maneira a conseguir tornar o projeto o mais prático possível, eficiente e útil, mantendo a sua simplicidade de uso, achámos que este sistema idelamente deveria ser composto por duas partes:

- Backoffice: esta será a base do sistema. Sendo necessária autenticação de um responsável, permite gerir todos os aspetos do restaurante, bem como aceder a toda a informação do sistema.
- Mobile: traduzindo-se numa aplicação, o objetivo desta componente é possibilitar o uso dinâmico por parte de funcionários do restaurante, permitindo assim, adicionar informações como pedidos, a partir de um dispositivo móvel, integrando assim o nosso sistema em operações do dia-a-dia.

2.2 FUNCIONALIDADES

Devido a sua elevada interligação de diferentes componentes, o SGR induz o seu usuário numa vasta teia de funcionalidades, desenhadas com o intuito prévio de maximizar os ganhos e eficiência do seu utilizador. A título de exemplo, podemos mencionar algumas das funcionalidades.

- Registar Trabalhadores: Nada melhor para a gestão duma empresa do que ter todos os seus constituintes devidamente catálogos.
- Criação de Horários: O gestor possui a capacidade de criar os horários dos funcionários, sendo que pode decidir notificar estes via e-mail. Permitindo assim um

workflow mais subtil.

- Registrar Mercadorias: Fornece à entidade patronal um mecanismo para controlo de mercadorias, permitindo assim uma maior organização empresarial.

Evidentemente, o universo de diferentes funcionalidades é mais extenso, sendo que o utilizador pode a todo o momento verificar a página de funcionalidades e verificar as diferentes opções ao seu dispor.

2.3 ESTRUTURA

De maneira a conseguir tornar o projeto o mais prático possível, eficiente e útil, mantendo a sua simplicidade de uso, achámos que este sistema idealmente deveria ser composto por duas partes:

- Backoffice: esta será a base do sistema. Sendo necessária autenticação de um responsável, permite gerir todos os aspetos do restaurante, bem como aceder a toda a informação do sistema.
- Mobile: traduzindo-se numa aplicação, o objetivo desta componente é possibilitar o uso dinâmico por parte de funcionários do restaurante, permitindo assim, adicionar informações como pedidos, a partir de um dispositivo móvel, integrando assim o nosso sistema em operações do dia-a-dia.

2.4 TECNOLOGIAS

A principal ferramenta utilizada foi a linguagem C#, em ambas componentes principais. Sendo que um dos requisitos seria utilizar uma ferramenta Microsoft. Por um motivo de aproveitar o facto de estarmos a aprender C#, optámos por utilizar o Xamarin para desenvolver a parte mobile, podendo assim utilizar a mesma linguagem e desenvolver uma aplicação cross-platform. Em concreto, no Backoffice utilizamos a ferramenta ASP.NET Core MVC, pois achámos que este padrão seria o mais adequado para o serviço web. Para o armazenamento de dados, optámos por criar uma base de dados em MySQL.

2.5 INOVAÇÃO

A escolha do nosso projeto deveu-se em grande parte à nossa vontade de criar um sistema inovador no mercado. Um dos grande problemas nos restaurantes centra-se na grande ineficiência em certas atividades do quotidiano, que custam tempo e recursos a uma empresa. Apesar de existirem certos produtos no mercado voltados para a gestão de restaurantes, nenhum conjugou as nossas ideias para melhorar o funcionamento do restaurante.

Alguns dos pontos inovadores do nosso projeto, apesar de simples, permitem melhorar e automatizar o *workflow* de um restaurante, sendo que alguns são:

- **Gestão Automática de Inventários** - tendo a informação da quantidade de um produto em stock e de qual a quantidade mínima que se deseja ter, o nosso sistema propõe-se a fazer um levantamento dos produtos com falha de stock, não sendo necessário dispendir recursos extra nessa tarefa.
- **Gestão de Horários dos Funcionários** - tendo os horários e turnos dos funcionários no sistema, é fácil automatizar o envio dos horários para os funcionários, sendo que é possível poupar tempo numa tarefa essencial.
- **Integração Mobile** - nos tempos atuais, a integração da componente móvel num sistema é fulcral para o seu sucesso, sendo que o nosso sistema se propõe a efetuar, nomeadamente, pedidos diretamente numa aplicação, refletindo-se essa alteração no resto do sistema.
- **Tratamento de Dados** - cada vez mais os dados são essenciais para o sucesso de um negócio. Analisando as vendas do restaurante é fácil produzir estatísticas que nos digam que artigos devem ser removidos da carta e quais devem ser mantidos. Apesar de simples, esta é uma tarefa essencial para muitos negócios poderem otimizar lucros e satisfação de clientes.

DESENVOLVIMENTO

Nesta secção, iremos abordar o desenvolvimento de todo o projeto, desde a sua definição, passando pela implementação. Para isso iremos dividir esta linha temporal em etapas, sendo que uma etapa corresponde ao período de tempo entre reuniões feitas com o docente da unidade curricular.

3.1 ETAPA 1 — DEFINIÇÃO DO PROJETO

A primeira etapa consistiu em desenvolver uma ideia para a realização do projeto. Refletimos sobre vários possíveis produtos e acabamos por optar pelo Sistema de Gestão de Restaurantes. Foi escolhido este tema devido à concordância entre os elementos do grupo e pela inexistência de um produto semelhante no mercado.

Para sustentar o nosso projeto, definimos alguns requisitos iniciais de modo a tornar a ideia mais concreta. Alguns requisitos inicialmente definidos foram:

- O sistema deverá possuir uma página de estatísticas.
- O sistema deverá possibilitar que um utilizador defina os horários de trabalho dos funcionários.
- O sistema possuirá um backoffice em interface web, será desenvolvido em C#.
- O sistema deverá possuir uma aplicação mobile para faturação e registo de pedidos.
- Apenas pode existir um pedido aberto em simultâneo por mesa.

Além de alguns requisitos iniciais, também definimos algumas das tecnologias que irão ser utilizadas no projeto. Para a definição da evolução temporal do projeto, em conjunto com o nosso docente, elaboramos um cronograma contendo as fases de desenvolvimento do projeto. Este cronograma sofreu algumas correções como iremos ver mais à frente. Para auxiliar o desenvolvimento do projeto, recorreremos a uma funcionalidade do nosso

repositório do GitHub, o GitHub Projects, sendo que inserimos as fases e tarefas do projeto nessa plataforma para melhor gerir o projeto. Também foi atribuída uma duração a cada fase do projeto. Assim sendo, o diagrama de Gantt que foi realizado (incluindo já todas as correções) tem a seguinte forma:

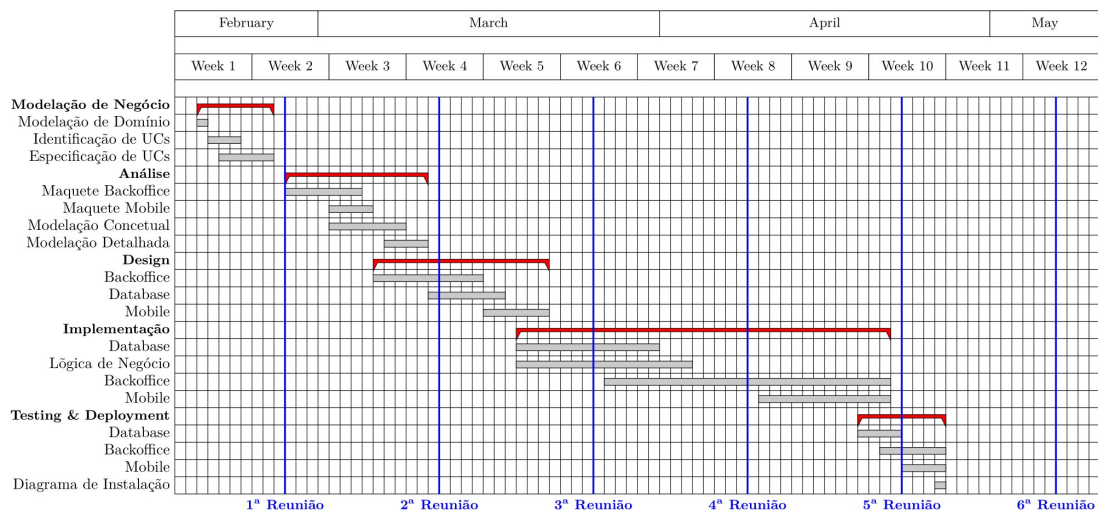


Figura 1: Diagrama de Gantt Final

O Cronograma desenvolvido ficou dividido nestas secções:

1. Modelação de Negócio (aprox. 2 semanas)
2. Análise (aprox. 2 semanas)
3. Design (aprox. 3 semanas)
4. Implementação (aprox. 4 semanas)
5. Testing & Deployment (aprox. 1 semana)

3.2 ETAPA 2 — MODELAÇÃO E REQUISITOS

Nesta etapa foi feito um ajuste ao nosso Diagram de Gantt sendo que o docente propôs que efetuássemos alterações nas durações de certas etapas. O resultado consistiu em adiantar a fase de implementação. A partir desta etapa, mudamos ligeiramente o nosso diagrama de Gantt para retroceder certos prazos de entrega, como o design da base de dados. Assim iniciando a implementação do programa ainda mais cedo.

- | | |
|-----------------------------|-------------------------------------|
| 9. Adicionar Mercadorias. | 18. Adicionar Reservas. |
| 10. Remover Mercadorias. | 19. Remover Reservas. |
| 11. Editar Mercadorias. | 20. Editar Reservas. |
| 12. Adicionar Fornecedores. | 21. Adicionar Pedidos. |
| 13. Remover Fornecedores. | 22. Remover Pedidos. |
| 14. Editar Fornecedores. | 23. Editar Pedidos. |
| 15. Adicionar Horários. | 24. Gerar Estatísticas. |
| 16. Remover Horários. | 25. Verificar mercadorias em Falta. |
| 17. Editar Horários. | |

3.2.2 *Requisitos Mobile*

1. Login.
2. Logout.
3. Registrar Pedido.
4. Faturar Pedido.
5. Editar Pedido.
6. Inserir artigo em Pedido.
7. Remover artigo de Pedido.

3.3 ETAPA 3 — MODELAÇÃO E DESIGN

Nesta etapa do projeto procedemos à modelação conceptual da nossa base de dados, criámos o nosso Diagrama de Classes, fizemos os *Mockups* das aplicações mobile e web, e, começamos com a ativação de tecnologias. Os Diagramas de Classes foram desenvolvidos em 2 versões, sendo que uma conta com DAO (Data Access Objects) e outra não. Os diagramas têm a seguinte forma:

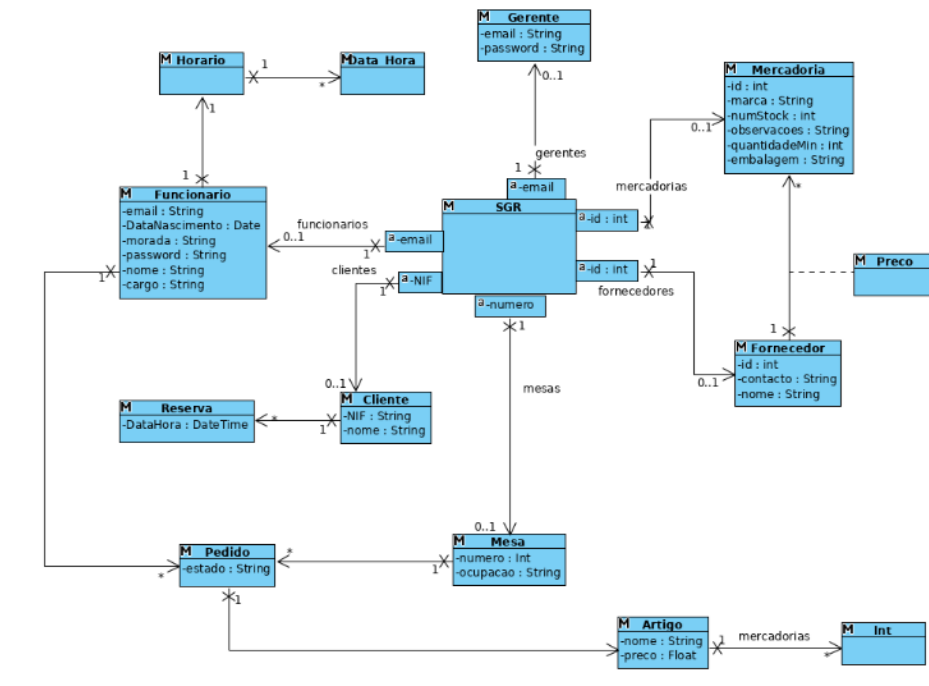


Figura 3: Diagrama de Classes sem DAO

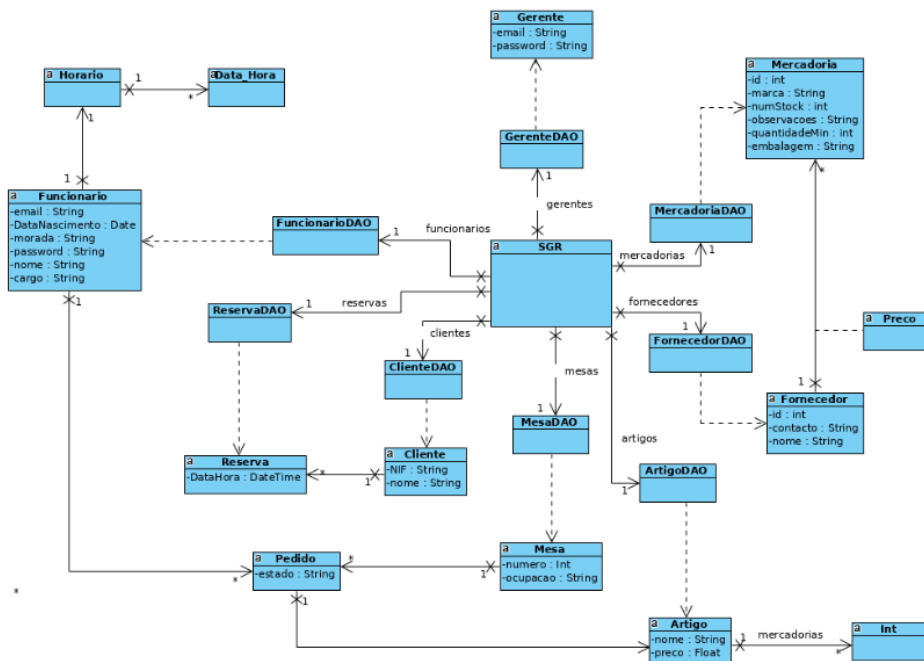


Figura 4: Diagrama de Classes com DAO

Relativamente à nossa base de dados, recorreremos aos nossos conhecimentos da Unidade Curricular de **Bases de Dados** para efetuar a sua modelação, sendo que a ferramenta que utilizamos para o efeito foi o **MySQLWorkbench**. A nossa base de dados sofreu várias iterações ao longo do projeto para acomodar necessidades que surgiram em outras etapas, no entanto, como as alterações são ligeiras, o modelo a seguir apresentado é já a iteração mais recente:

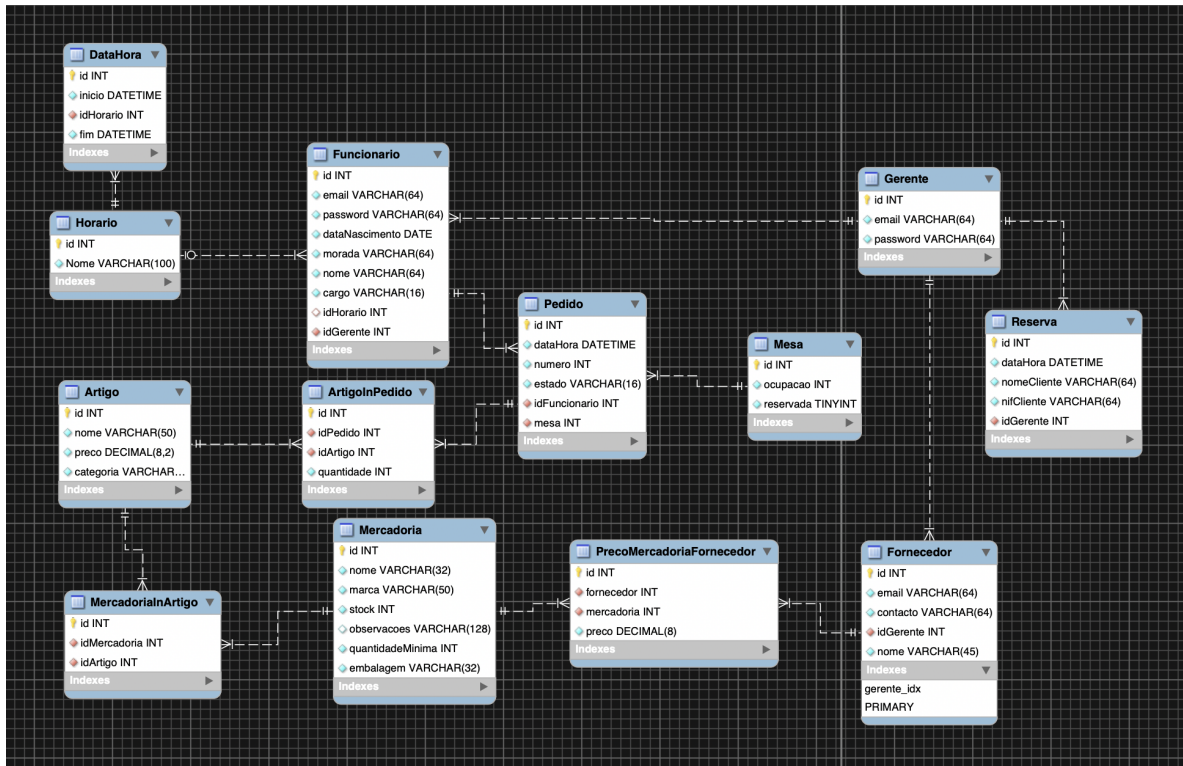
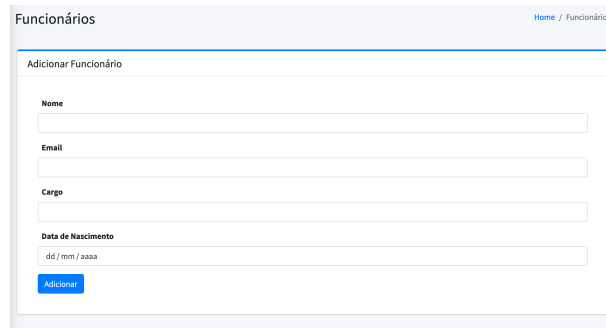


Figura 5: Modelo Final da Base de Dados

Nesta etapa realizamos também a Ativação de Tecnologia e criámos alguns *Mockups*. A ativação consistiu em colocar num servidor local uma instância inicial do Backoffice, sendo que logo nesta fase existiu a preocupação em criar um produto eficiente e estruturado. Para isso implementamos uma instância inicial utilizando a **framework ASP.NET Core** e seguindo o padrão **MVC - Model View Controller**. Para esta instância inicial não efetuamos ligação com a base de dados e apenas criámos um sistema **CRUD** (Create Read Update Delete) para uma entidade Funcionário simplificada. Para a aplicação, foi também feito um protótipo simples utilizando o **Xamarin**.

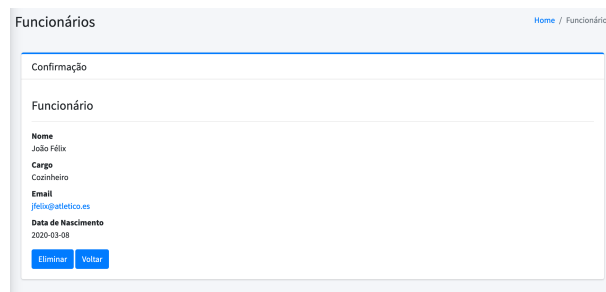
Relativamente aos *Mockups*, de modo a evitar perder tempo com ferramentas de design, criámos já as páginas necessárias usando **HTML**, **CSS** e **JavaScript** de modo a ter interfaces responsivas. De seguida, utilizamos alguns *screenshots* da interface desenvolvida para

servirem de *Mockups*, sendo que até ao final do projeto adotámos este design. De seguida apresentam-se alguns dos *Mockups* produzidos:



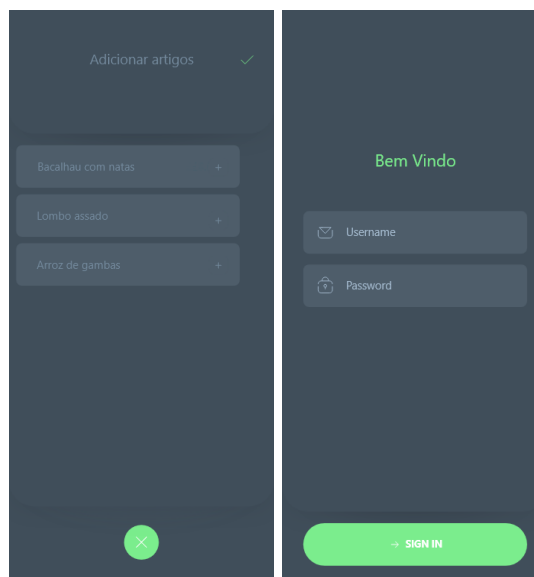
The mockup shows a web interface for adding a new employee. It features a header with the title 'Funcionários' and a breadcrumb 'Home / Funcionários'. Below this is a form titled 'Adicionar Funcionário' with fields for 'Nome', 'Email', 'Cargo', and 'Data de Nascimento' (with a date picker). A blue 'Adicionar' button is at the bottom.

Figura 6: BackOffice: Adicionar Funcionário



The mockup shows a confirmation screen for deleting an employee. It has a header with 'Funcionários' and a breadcrumb 'Home / Funcionários'. The main content is titled 'Confirmação' and 'Funcionário', displaying the employee's details: 'Nome: João Félix', 'Cargo: Cozinheiro', 'Email: jfelix@atletico.es', and 'Data de Nascimento: 2020-03-08'. At the bottom are 'Eliminar' and 'Voltar' buttons.

Figura 7: BackOffice: Eliminar Funcionário



The image shows two mobile app mockups side-by-side. The left screen, titled 'Adicionar artigos', has a green checkmark and a list of items: 'Bacalhau com natas', 'Lombo assado', and 'Arroz de gambas', each with a plus icon. A green close button is at the bottom. The right screen, titled 'Bem Vindo', has input fields for 'Username' and 'Password', and a green 'SIGN IN' button at the bottom.

Figura 8: Mobile: Adicionar Artigo e Login

3.4 ETAPA 4 — IMPLEMENTAÇÕES INICIAIS

Nesta etapa do projeto, após estarmos familiarizados com as ferramentas a utilizar, procedemos à implementação de componentes base quer do BackOffice, quer da aplicação. Nesta etapa, efectuámos também o deploy da nossa base de dados bem como das conexões necessárias com o backoffice e a aplicação.

Relativamente ao BackOffice, foi feito o core do sistema, sendo que consistiu em criar um sistema **CRUD** (Create Read Update Delete) para cada tabela da nossa base de dados. Dado o facto de estarmos a usar o modelo **MVC**, este processo tornou-se mais prático e organizado, sendo que o resultado foi bastante produtivo. Para o acesso à nossa base de dados usamos 2 pacotes disponíveis: 1 para efetuar a ligação a uma base de dados, que é essencial e padrão, e, usámos o **Entity Framework Core** (EF Core). O EF Core é uma framework disponível para .NET que nos permite abstrair os DAO e usar essa mesma abstração para comunicar com a base de dados. Para isso, basta-nos criar um modelo para cada uma das nossas tabelas, especificando alguns tipos para cada campo da tabela, otimizando assim o seu tratamento nos formulários e processamento de dados. Para obter todos os modelos e conseguir o máximo de coerência entre a base de dados e o que é exibido no BackOffice, usamos a CLI dotnet para fazer o **reverse engineering** do modelo da base de dados.

Para o servidor remoto decidimos utilizar **Microsoft Azure**, na sua versão gratuita, sendo que após o período experimental, será feita a migração para outra plataforma gratuita.

Relativamente à aplicação, para efetuar a ligação com a base de dados, utilizámos o package *MySQL.Data*, sendo que a comunicação com a mesma passa a ser feita através de instâncias *SqlConnection*.

3.5 ETAPA 5 — LOGIN E PROTÓTIPOS FUNCIONAIS

Nesta fase foram implementados protótipos funcionais do nosso sistema, isto é, versões não finais do sistema, mas cujas funcionalidades implementadas estejam funcionais. No nosso caso, os protótipos funcionais consistiram na implementação de todos os requisitos relativos a entidades no sistema. Dados que esses requisitos consistem numa grande parte de todos os requisitos, consideramos os nossos protótipos razoavelmente completos.

A maioria do trabalho nesta fase consistiu em resolver dependências e correção de erros, sendo que continuámos a utilizar várias ferramentas e bibliotecas intrínsecas ao .NET.

No BackOffice foi adicionada uma funcionalidade bastante importante na forma de um sistema de autenticação, **Login**. Este sistema foi feito de modo a um gerente poder efetuar o login no sistema e manter a sua sessão iniciada até que opte por fazer logout. Embora possa parecer uma falha de segurança manter a sessão aberta, num cenário real, o gerente não pretende estar constantemente a efetuar login no sistema devido ao tempo de inatividade. Para esse efeito recorremos a **Cookies** para guardar os dados de sessão.



Figura 9: BackOffice: Login

Na aplicação mobile afastamo-nos dos *Mockups* anteriormente definidos, tendo redesenhado a interface de modo a ter um resultado visualmente mais apelativo.

3.6 ETAPA 6 — GESTÃO E MELHORIAS

Os objetivos desta etapa passaram pela implementação dos fatores mais interessantes e inovadores do nosso sistema, bem como a consotante implementação de melhorias quer no BackOffice, quer na aplicação.

No BackOffice, foi implementada uma página de estatísticas com dados pertinentes à gestão do restaurante. Outra funcionalidade adicionada foi a possibilidade de ver quais as Mercadorias em falta no restaurante. Relativamente a tecnologias, o ponto forte destas implementações centrou-se em **JavaScript**, dados que permitimos que os componentes sejam dinâmicos e devido à utilização de gráficos, mais concretamente, **ChartJS**. A página de estatísticas permite-nos visualizar informações como:

- Total faturado nos últimos dias
- Total número de pedidos nos últimos dias

- Artigos mais vendidos
- Mercadorias em falta
- Número de Pedidos do dia

Relativamente à aplicação, foram trabalhados diversos aspetos. Implementámos a funcionalidade de concretizar pagamento de um pedido por parte de um empregado de balcão, acrescentamos informação visível na área de visualização dos pedidos e resolvemos alguns problemas presentes na ligação da aplicação com a base de dados. Foram feitas também melhorias a nível da interface da aplicação, sendo que este é um processo iterativo.

3.7 ETAPA 7 — DETALHES E CORREÇÃO DE ERROS

Nesta etapa, apesar de já termos o projeto na sua fase final, foram necessárias algumas implementações de detalhes que o docente sugeriu, bem como correções de erros (**bugs**) que ainda persistem no sistema.

Alguns detalhes que adicionámos passam por efetuar pedidos na aplicação com os artigos separados por categorias, visualizar conteúdo de um pedido depois de este ser feito, entre outras.

Uma parte fulcral em qualquer projeto é o controlo de qualidade. Sem este processo não podemos garantir que iremos entregar a um potencial cliente um produto que o possa servir no dia-a-dia, muito menos, servir o seu negócio. Para esse efeito, realizamos diversos testes, quer com conjuntos de dados de modo a testar a página de estatísticas, quer na aplicação ao efetuar pedidos monitorizando a base de dados, quer nos formulários do backoffice ao inserir registos. Os casos de teste não se limitaram aos inumerados, no entanto permitiram-nos detetar e corrigir um número considerável de erros, sendo que neste momento considerámos que existe um número reduzido de erros no nosso sistema.

De modo a evitar colocar excertos de código e ocupar o relatório deste projeto com componentes técnicas, sendo estas necessárias, optámos por criar em paralelo uma *Wiki*, sendo que nesta etapa iniciámos a sua construção.

3.8 ETAPA 8 — DOCUMENTAÇÃO E AJUSTES FINAIS

Nesta última etapa, o nosso principal foco foi tornar o nosso produto legível e compreensível, isto é, focamo-nos na componente de documentação do projeto. Dado que estamos a criar um sistema para terceiros, considerámos que é uma boa prática ter uma documentação técnica a acompanhar, dado que no futuro nos irá ajudar a efetuar revisões e melhorias sem ter que rever todo o código, e, porque ajuda a desmistificar o código para alguém que não

tenha desenvolvido o projeto. Após analisarmos algumas opções, nomeadamente inserir a documentação com o código, achámos que essa não seria a melhor opção, sendo que optámos por criar uma *Wiki*, suportando-a com excertos de código sempre que necessário.

Relativamente ao sistema em si, apenas foram efetuadas ligeiras afinações na forma de correção de erros que ainda tenham persistido de etapas anteriores, bem como a implementação de mais algumas funcionalidades extra a pedido do docente. Um desses ajustes passou pela utilização de uma **API** externa, sendo que no nosso caso decidimos criar um requisito extra para utilizar a dita API. O nosso sistema passa agora a permitir que um Gerente, ao visualizar a lista de Funcionários, tenha a opção **Enviar Horário**. Esta opção envia ao Funcionário em questão um email com a informação do seu Horário de trabalho no sistema. Para esse efeito recorremos à API **mailjet**, dado que esta nos pareceu fiável e muito bem documentada.

3.9 CONSIDERAÇÕES SOBRE O DESENVOLVIMENTO DO PROJETO

Após a conclusão da fase de desenvolvimento do projeto, é também necessário fazer uma avaliação em da relação entre objetivos definidos e objetivos alcançados, bem como dos resultados obtidos.

Relativamente à relação entre os objetivos que propusemos ao longo deste capítulo e aquilo que fomos capazes de produzir e cumprir, podemos fazer um balanço bastante positivo. No que toca ao seguimento do projeto, mesmo com algumas dificuldades iniciais todas as etapas foram concluídas com sucesso e todos os requisitos foram implementados na sua totalidade, sendo que ainda houve espaço para incluir alguns extras. No que toca à Unidade Curricular, cremos que fizemos uma gestão do tempo bastante boa, na medida em que mesmo com as circunstâncias atuais e apesar de os prazos serem adiados, o nosso grupo conseguiu concluir o projeto para a data inicialmente prevista.

RESULTADOS

Nesta secção iremos apresentar alguns elementos do sistema desenvolvido, sendo que iremos também proceder à explicação sobre o funcionamento dos mesmos. Assim sendo iremos mostrar alguns componentes mais essenciais de cada componente:

4.1 BACKOFFICE

A primeira página com a qual nos deparámos quando tentámos aceder ao BackOffice é a página de **Login**. Após efetuar o login com sucesso, é-nos apresentada a página inicial, ou, caso tenha sido feito o acesso a uma página específica antes do Login, somos redirecionados para essa página. Por exemplo, se tentarmos aceder à página de listar os Funcionários, após efetuar o login, somos redirecionados. A página inicial consiste numa página de estatísticas, seguindo sempre a mesma filosofia de design em todo o BackOffice, mantendo a barra lateral e superior em todas as páginas. Deste modo evitámos a repetição de código e tornámos tudo mais eficiente.

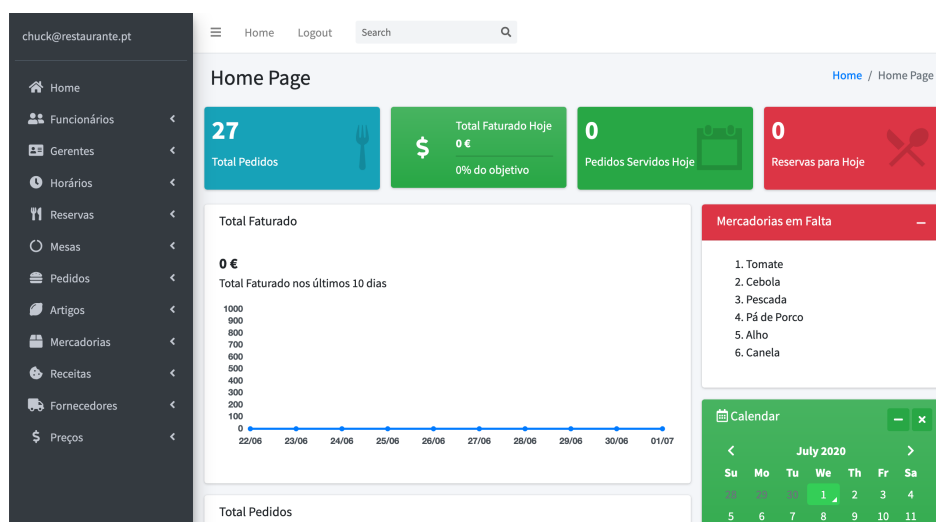


Figura 10: BackOffice: Página Inicial I

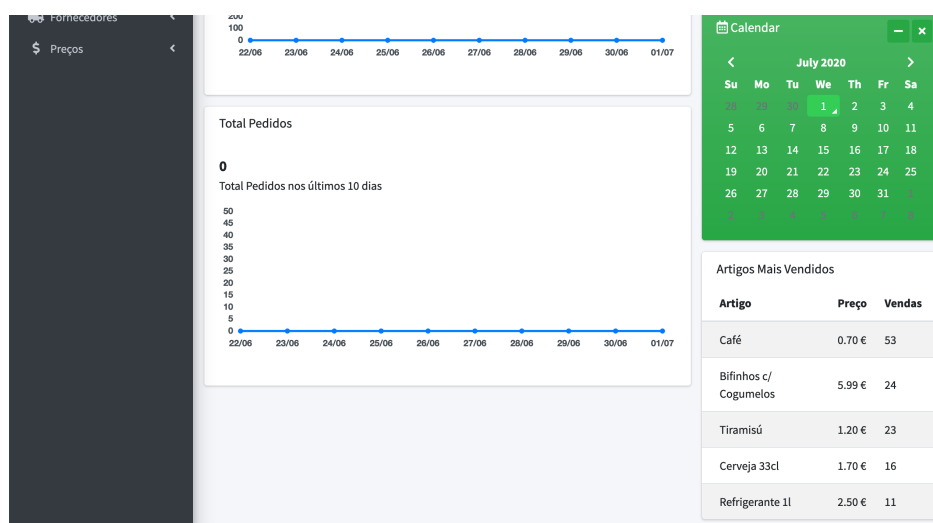


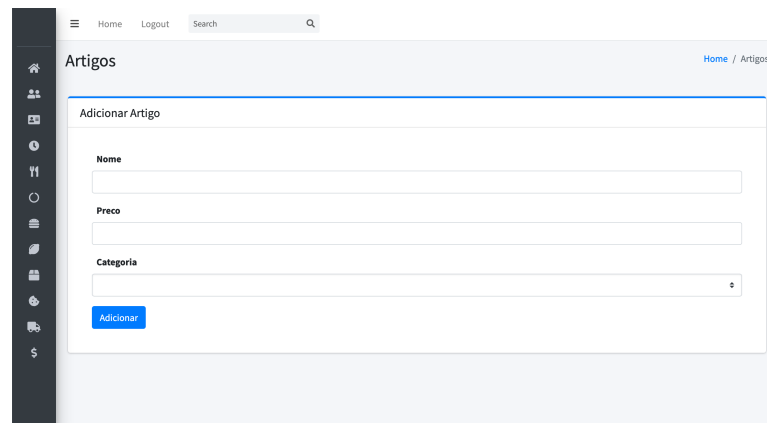
Figura 11: BackOffice: Página Inicial II

Nesta página inicial, através de **JavaScript** conseguimos arrastar alguns dos elementos para permitir um certo nível de customização, sendo que cada gerente poderá querer exibir em destaque uma estatística diferente. Outra funcionalidade adicionada consiste no facto de o BackOffice ser **responsivo**, isto é, suportar dispositivos móveis. Deste modo ampliámos a gama de dispositivos que podem aceder eficazmente ao nosso sistema. Outra preocupação nossa passou por efetuar a recolha e redirecionamento dos erros **HTTP 404**. Outro ponto essencial do BackOffice consiste na gestão de diversos elementos do restaurante como Funcionários, Artigos, Mercadorias, Pedidos, entre muitos outros. Para cada um destes elementos foram criadas as operações **CRUD**, sendo que iremos mostrar alguns excertos dessas operações para os Funcionários e Artigos:

The screenshot shows the 'Funcionários' (Employees) page in the BackOffice. It includes a search bar, a table of employees, and a sidebar with navigation icons.

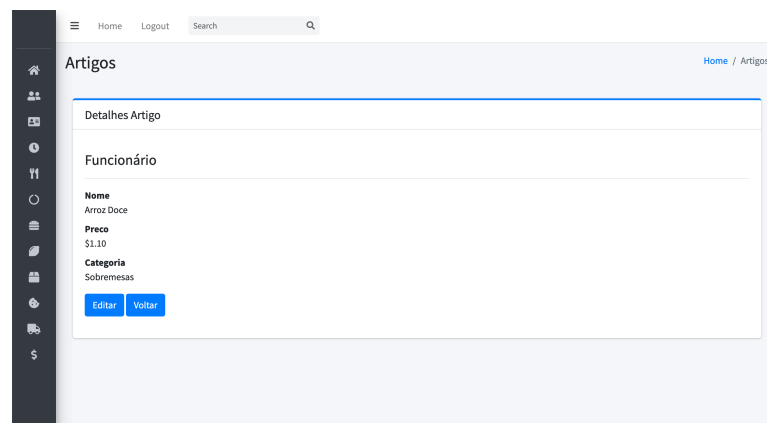
Email	Nome	Cargo	Horário	Id Gerente	Opções
alice@restaurante.pt	Alice Gama	Emp. Mesa	Turno Almoço	2	Editar Detalhes Eliminar Enviar Horário
clemente@restaurante.pt	Clemente Azevedo	Emp. Mesa	Turno Jantar	2	Editar Detalhes Eliminar Enviar Horário
emilio@restaurante.pt	Emilio Torres	Cozinheiro	Turno Almoço	1	Editar Detalhes Eliminar Enviar Horário
enzo@restaurante.pt	Enzo Nunes	Cozinheiro	Turno Jantar	1	Editar Detalhes Eliminar Enviar Horário
gervasio@restaurante.pt	Gervásio Fidalgo	Emp. Balcão	Turno Almoço	3	Editar Detalhes Eliminar Enviar Horário
hugo@restaurante.pt	Hugo Águas	Emp. Balcão	Turno Jantar	3	Editar Detalhes Eliminar Enviar Horário
juliana@restaurante.pt	Juliana Gama	Emp. Mesa	Turno Almoço	2	Editar Detalhes Eliminar Enviar Horário
rafaela@restaurante.pt	Rafaela Coelho	Emp. Mesa	Turno Jantar	2	Editar Detalhes Eliminar Enviar Horário
roberto@restaurante.pt	Roberto Agostinho	Cozinheiro	Turno Jantar	1	Editar Detalhes Eliminar Enviar Horário

Figura 12: BackOffice: Listar Funcionários



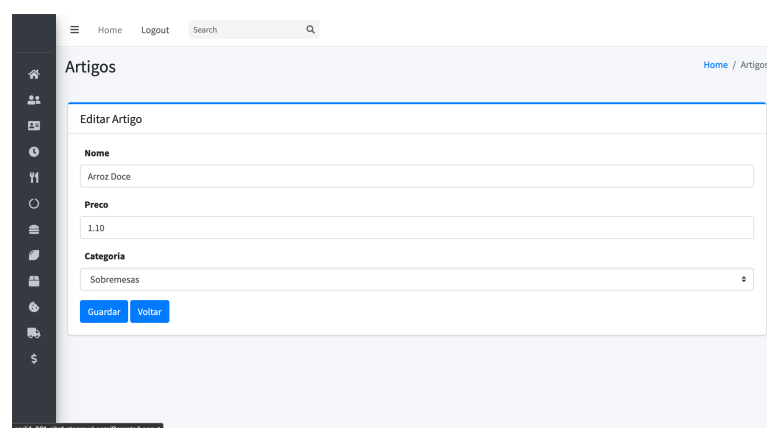
The screenshot shows the 'Adicionar Artigo' (Add Article) form in the BackOffice. The form is titled 'Artigos' and has a breadcrumb 'Home / Artigos'. It contains three input fields: 'Nome' (Name), 'Preço' (Price), and 'Categoria' (Category). Below the fields is a blue 'Adicionar' (Add) button. The left sidebar contains a vertical menu with icons for Home, Users, Articles, Categories, and a search icon. The top navigation bar includes 'Home', 'Logout', and a search bar.

Figura 13: BackOffice: Adicionar Artigo



The screenshot shows the 'Ver Artigo' (View Article) page in the BackOffice. The page is titled 'Artigos' and has a breadcrumb 'Home / Artigos'. It displays the details of an article, including the 'Funcionário' (Employee) section. The article details are: 'Nome' (Name) is 'Arroz Doce', 'Preço' (Price) is '\$1.10', and 'Categoria' (Category) is 'Sobremesas'. Below the details are two buttons: 'Editar' (Edit) and 'Voltar' (Back). The left sidebar and top navigation bar are the same as in the previous screenshot.

Figura 14: BackOffice: Ver Artigo



The screenshot shows the 'Editar Artigo' (Edit Article) page in the BackOffice. The page is titled 'Artigos' and has a breadcrumb 'Home / Artigos'. It displays the details of an article, including the 'Funcionário' (Employee) section. The article details are: 'Nome' (Name) is 'Arroz Doce', 'Preço' (Price) is '1.10', and 'Categoria' (Category) is 'Sobremesas'. Below the details are two buttons: 'Guardar' (Save) and 'Voltar' (Back). The left sidebar and top navigation bar are the same as in the previous screenshots.

Figura 15: BackOffice: Editar Artigo

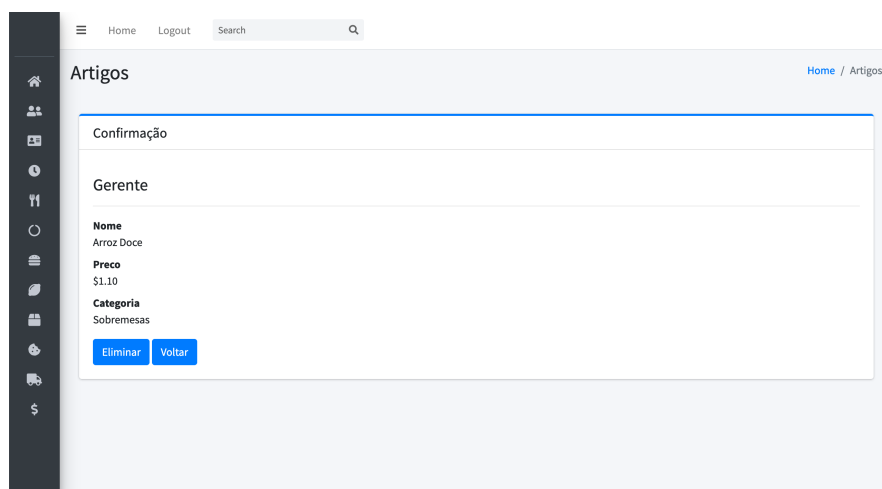


Figura 16: BackOffice: Apagar Artigo

Todas as entidades do sistema possuem uma estrutura semelhante, variando apenas nos campos que possuem. Nestes formulários entram em uso as especificações do tipo de dados que fizemos nos modelos da EF Core, sendo que existe uma ligeira diferença no suporte a alguns dos campos entre navegadores. O Google Chrome suporta uma maior variedade de campos do que o Firefox, no entanto, a diferença não é muito acentuada.

4.2 APLICAÇÃO

Para o desenvolvimento da aplicação, utilizámos o **Xamarin**, que nos permitiu desenvolver uma componente móvel **cross-platform**. Nesta secção iremos mostrar excertos da versão **Android**. A aplicação por sua vez, também possui um sistema de autenticação, que dados o email e password, efetua a verificação comparando os valores com o que está na base de dados. Após verificar que os dados de acesso são válidos, é feita uma verificação ao perfil do Funcionário de modo a saber se se trata de de um Empregado de Mesa ou de um Empregado de Balcão. Consoante o cargo do Funcionário, é apresentado o ecrã correspondente. Consoante o ecrã apresentado, variam as opções possíveis, sendo que o Empregado de Mesa além de retroceder e ver os pedidos existentes, pode criar novos pedidos, enquanto que o Empregado de Balcao em vez de criar pedidos, pode faturá-los.

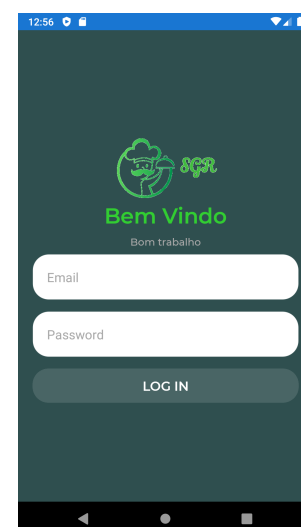


Figura 17: Aplicação: Login

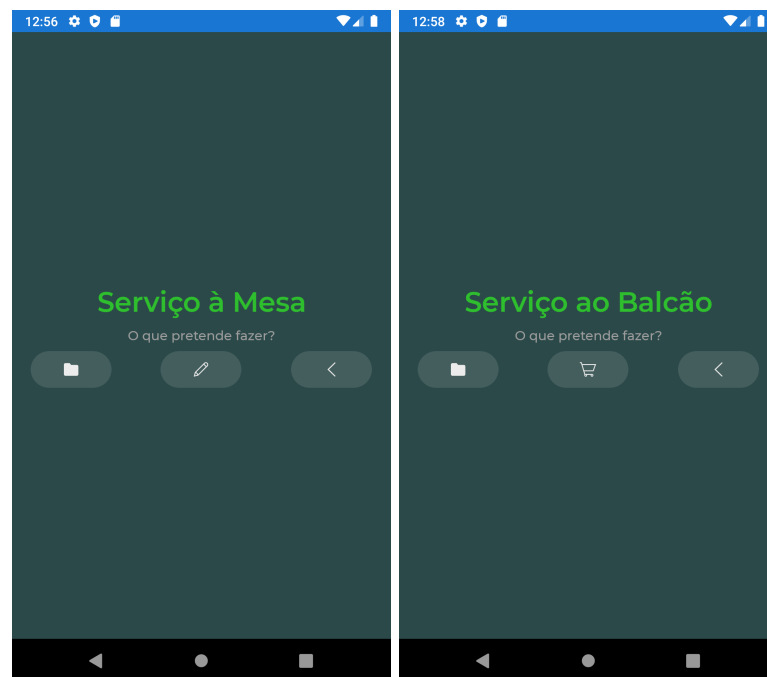


Figura 18: Aplicação: Serviço Mesa e Balcão

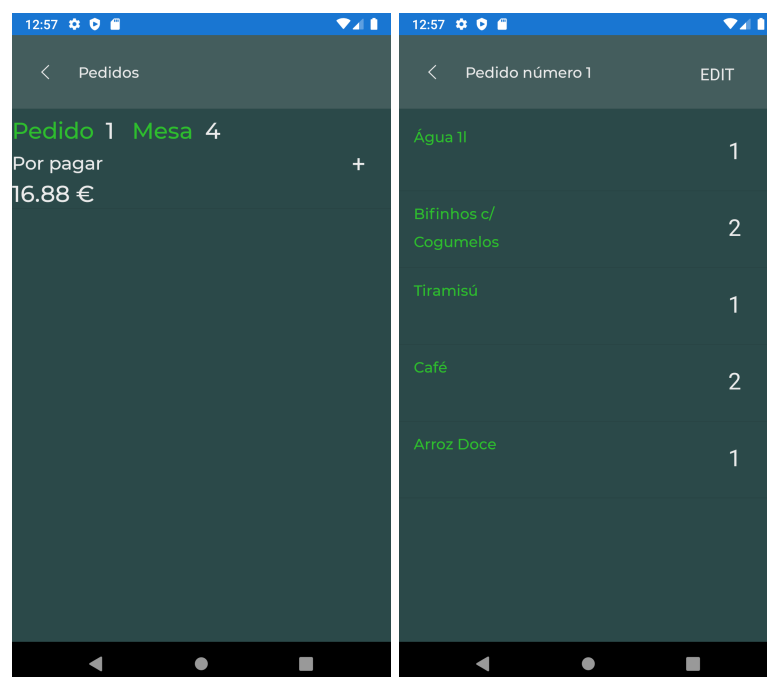


Figura 19: Aplicação: Ver Pedidos e Conteúdo do Pedido

Dado que a aplicação tem como objetivo ser usada num cenário real, sob condições de pressão, torna-se essencial conjugar a rapidez com a certeza na hora de efetuar e faturar os pedidos. Isso significa que temos que conjugar o número de confirmações necessárias para uma ação e o número de opções que se tem de selecionar para chegar a uma determinada ação. No nosso caso, as confirmações são feitas usando os elementos nativos do sistema móvel em questão (Android ou iOS).

Um dos ecrãs por nós desenvolvidos, apesar de adicionado durante a fase de desenvolvimento, demonstra a natureza iterativa do projeto. Referimo-nos ao ecrã das **Categorias** de um Artigo. Apesar de não nos ter ocorrido fazer essa separação, é uma ideia que permite efetuar uma procura de Artigos na aplicação de maneira muito mais eficiente e ordenada. Esta foi uma das melhorias presentes durante as etapas de desenvolvimento da nossa aplicação.

Após a finalização da aplicação, podemos dizer que ficámos e entender melhor alguns conceitos relacionados com o desenvolvimento de aplicações mobile. Esses conceitos variam entre conceitos relacionados com o design de uma interface intuitiva e visualmente apelativa, ou, relacionados com a experiência que o utilizador final irá ter quando estiver a utilizar a nossa aplicação (UI & UX).

Apesar de existir sempre espaço para melhorias, acreditámos que construímos uma aplicação que segue os padrões desejados de qualidade mantendo-se relevante para o problema em causa.

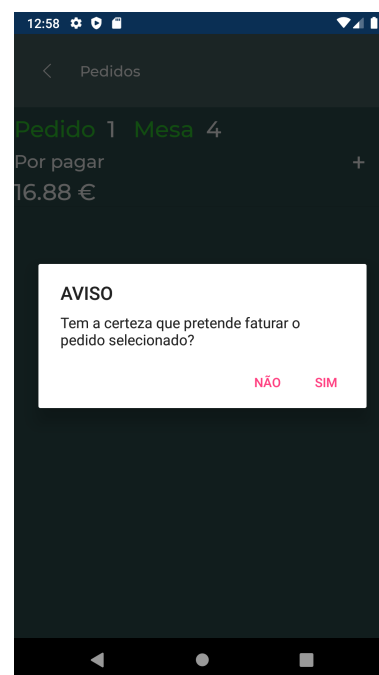


Figura 20: Aplicação: Confirmação de Pedido



Figura 21: Aplicação: Categorias de Artigos

CONCLUSÃO

5.1 OBJETIVOS E RESULTADOS OBTIDOS

No capítulo inicial, propusemos uma série de objetivos que pretendíamos alcançar até à finalização do projeto. Tendo dado o projeto por terminado, chegou a hora de efetuar o balanço entre os objetivos propostos e os objetivos alcançados e cumpridos. Podemos afirmar que fomos capazes de alcançar todos os objetivos na sua íntegra bem como ir além do que nos propusémos. Fomos capazes de efetuar uma boa gestão do tempo e cumprir todos os prazos estabelecidos, tendo implementado todos os requisitos propostos, bem como alguns requisitos adicionais. Podemos afirmar que o nosso produto final é funcional e considerámos o mesmo apelativo. O nosso produto encontra-se pronto para implementar num cenário real, sendo que apenas são necessárias as respetivas configurações intrínsecas à publicação de uma plataforma do género.

Com a realização deste projeto conseguimos também atingir o nosso objetivo de adquirir competências em ferramentas novas. Fomos capazes de entender melhor o conceito MVC, e aprendemos os fundamentos de aplicações feitas com .NET e com Xamarin.

Fazendo uma reflexão generalizada sobre este projeto, chegámos à conclusão unânime de que conseguimos fazer tudo aquilo ao qual nos propusemos, tendo conseguido trabalhar alguns detalhes adicionais que considerámos relevantes. Assim sendo, fizemos um balanço positivo do trabalho realizado, sendo que acima do projeto realizado, considerámos que estamos melhor preparados para enfrentar desafios futuros em cenários reais.

5.2 TRABALHO FUTURO

Apesar de considerarmos o nosso projeto bem sucedido, existe sempre um espaço para serem feitas melhorias. Após reflectir sobre o que poderíamos ter feito de modo diferente ou incluir, chegámos à conclusão que a nossa página de estatísticas poderia sofrer iterações

futuras de modo a se tornar mais eficiente e menos pesada em termos de acessos à base de dados. Um aspeto que não foi tido em conta devido à sua complexidade foi o aspeto da segurança do sistema. Esta seria uma implementação fundamental a acrescentar no futuro.