

## **Carta de Restaurante Digital**

35392, João Paulo Silva Rodrigues

39592, José Pedro Sampaio Vicente Gandarez

Orientadores      Eng<sup>a</sup>. Paula Graça

Relatório final realizado no âmbito de Projeto e Seminário,  
do curso de licenciatura em Engenharia Informática e de Computadores  
Semestre de Verão 2016/2017

Julho de 2017



# **Instituto Superior de Engenharia de Lisboa**

Licenciatura em Engenharia Informática e de Computadores

## **Carta de Restaurante Digital**

35392, João Paulo Silva Rodrigues

39592, José Pedro Sampaio Vicente Gandarez

Orientadores      Eng<sup>a</sup>. Paula Graça

Relatório final realizado no âmbito de Projeto e Seminário,  
do curso de licenciatura em Engenharia Informática e de Computadores  
Semestre de Verão 2016/2017

Julho de 2017



# Resumo

No projeto Carta de Restaurante Digital, foi desenvolvida uma solução que poderá melhorar a capacidade de resposta aos pedidos dos estabelecimentos de restauração, e, consequentemente, a gestão dos mesmos. A verdade é que, devido a uma gestão ineficiente ou por falta de formação dos funcionários, verificam-se frequentemente erros na execução dos pedidos ou tempos de espera demasiado longos. Assim sendo, surge a necessidade de desenvolver soluções ágeis e inovadoras que proporcionem um serviço eficiente e de qualidade ao cliente. Considere-se, por exemplo, a solução existente no McDonald's para a realização do pedido pelo próprio cliente: não só permitiu reduzir o tempo de espera, como também modernizar e simplificar a experiência do cliente, apresentando um menu interativo que engloba todos os produtos.

Tendo por base o conceito existente no McDonald's [1], foi desenvolvida a presente solução, através da plataforma OutSystems, aplicável a qualquer estabelecimento de restauração. O projeto *Carta de Restaurante Digital* visa otimizar o processamento dos pedidos em unidades de restauração, agilizando a capacidade de resposta no atendimento ao cliente, podendo assim reduzir o tempo de espera. Além disso, pretende-se modernizar a experiência do cliente, não só apresentando os menus de uma forma mais apelativa e interativa através de uma aplicação móvel, como possibilitando também o envio da fatura para o *email*.

Foi, então, implementada uma aplicação de *front-end* que permite consultar a oferta da unidade de restauração e efetuar o pedido através de um *tablet* sem intervenção do funcionário. A configuração do sistema é executada a nível do *back-office*. Para o envio da fatura por *email*, foi implementada a *framework InvoiceXpress* [2] que constitui um programa de faturação online, certificado pela Autoridade Tributária(AT) e Aduaneira de Portugal e que se caracteriza precisamente por permitir o envio de faturas por correio eletrónico, sem ser necessário recorrer a outros serviços externos. Por outro lado, esta *framework* possibilita também a partilha automática dos dados com a AT. Todos os desenvolvimentos técnicos foram realizados de acordo com as melhores práticas aconselhadas pela OutSystems [3].

Estamos convictos que foi desenvolvido um produto com aplicação à vida real que poderá ter um impacto significativo no sector da restauração, não só na melhoria da relação entre funcionários e clientes, como também na eficiência do serviço prestado.

Palavras-chave: aplicação móvel, aplicação web, menu digital, outsystems, restauração, framework.



# Agradecimentos

Agradecemos, em primeiro lugar, à Prof. Eng.<sup>a</sup> Paula Graça que com o seu apoio, dedicação, orientação e confiança, ajudou a construir o caminho que percorremos desde a idealização até à concretização deste projeto. Os desafios que colocou, as palavras que corrigiu e até os desabafos que ouviu foram fundamentais para concretização deste projeto.

Queremos agradecer a todos os professores do ISEL pelos conhecimentos e técnicas que transmitiram ao longo do curso no qual foram importantes no desenrolar deste trabalho.

Agradecemos também ao Márcio e ao Prof. Dr. Fernando Sousa por terem participado neste projeto, permitindo que este se tivesse concretizado.

À Inês Estrelinha, pela paciência, amizade, incentivo e apoio incondicional que me deu nos últimos anos. Agradeço principalmente pela presença e pela força que transmitiu nos meus últimos três anos de trabalhador-estudante.

Tendo consciência que sozinho nada disto teria sido possível, dirijo um agradecimento especial aos meus pais, por serem modelos de coragem, pelo seu apoio incondicional, incentivo, amizade e paciência demonstrados e total ajuda na superação dos obstáculos que ao longo desta caminhada foram surgindo. A eles dedico este trabalho!

À minha família, em especial atenção ao meu irmão, por estarem sempre presentes e por acreditarem em mim.

Por fim, agradeço a todos os meus amigos que acompanharam o meu percurso académico, em especial atenção à Débora Caeiro pela sua orientação em *design* e aos meus amigos da Infosistema pela compaixão, ajuda e pela força que transmitiram ao longo do tempo para terminar o curso.



# Índice

Resumo .....	5
Agradecimentos .....	7
Lista de Figuras.....	11
1. Introdução .....	15
1.1. Enquadramento, Objetivos e Motivação.....	15
1.2. Organização do documento .....	16
2. Formulação do Problema .....	17
3. Solução Carta de Restaurante Digital .....	21
3.1. Segurança.....	21
3.2. Casos de uso .....	21
3.3. Perfis.....	22
3.4. Modelo de dados.....	23
3.5. Ações genéricas .....	25
3.6. Tema .....	26
3.7. Aplicação de <i>back-office</i> .....	27
3.7.1. Registo da empresa .....	27
3.7.2. <i>Dashboard</i> .....	28
3.7.3. Configuração da empresa.....	29
3.7.4. Gestão de taxas .....	30
3.7.5. Gestão de categorias de produtos .....	31
3.7.6. Gestão de tipos de produtos .....	33
3.7.7. Gestão do menu .....	35
3.7.8. Gestão de mesas .....	37
3.7.9. Gestão de utilizadores.....	38
3.8. Aplicação de <i>front-office</i> .....	40
3.8.1. Gestão de mesas .....	40
3.8.2. Gestão de pedidos de mesas.....	41
3.8.3. Gestão de pedido de mesas .....	43
3.8.4. Pagamento .....	44
3.9. Aplicação móvel ( <i>tablet</i> ).....	45
3.9.1     Fluxo da aplicação .....	45
3.9.2     On boarding.....	46

3.9.3	Listagem de mesas .....	46
3.9.4	Menu .....	47
3.9.5	Detalhe do produto .....	48
3.9.6	Detalhe do pedido .....	49
3.9.7	Sincronização da base de dados local .....	50
3.10.	Faturação.....	57
3.10.1	<i>InvoiceXpress API</i> .....	57
3.10.2.	Processos.....	66
3.10.3.	Temporizadores.....	67
4.	Avaliação Experimental.....	69
4.1.	Aplicação de <i>back-office</i> .....	69
4.2.	Aplicação de <i>front-office</i> .....	77
4.3.	Aplicação móvel ( <i>tablet</i> ).....	83
	Conclusões .....	87
	Referências.....	89

# **Lista de Figuras**

Figura 1 - Arquitetura OutSystems .....	18
Figura 2 - Arquitetura da aplicação .....	19
Figura 3 - Casos de uso .....	21
Figura 4 - Modelo de dados .....	23
Figura 5 - Modelo de dados local .....	24
Figura 6 - Ações genéricas.....	25
Figura 7 - Ações do <i>eSpace Core</i> .....	25
Figura 8 - Tema do OutSystems .....	26
Figura 9 - Ecrã de registo da empresa.....	27
Figura 10 - Ecrã de <i>dashboard</i> .....	28
Figura 11 - Ecrã de configuração da empresa.....	29
Figura 12 - Ecrã de gestão de taxas .....	30
Figura 13 - Ecrã de edição de taxas .....	30
Figura 14 - Ecrã de gestão de categorias de produtos.....	31
Figura 15 - Ecrã de edição de categorias de produtos .....	32
Figura 16 - Ecrã de gestão de tipos de produtos .....	33
Figura 17 - Ecrã de edição de tipos de produto .....	34
Figura 18 - Ecrã de gestão do menu.....	35
Figura 19 - Ecrã de edição do produto.....	36
Figura 20 - Ecrã de gestão de mesas .....	37
Figura 21 - Ecrã de edição da mesa .....	37
Figura 22 - Ecrã de gestão de utilizadores .....	38
Figura 23 - Ecrã de edição do utilizador .....	39
Figura 24 - Casos de uso do <i>front-office</i> .....	40
Figura 25 - <i>User interface</i> da gestão de mesas e lógica da gestão de mesas .....	40
Figura 26 - Ecrãs de gestão de pedidos do balcão e da cozinha .....	41
Figura 27 - <i>User interface</i> e lógica do <i>web block Order</i> .....	42
Figura 28 - <i>User interface</i> e lógica do <i>web block OrderItems</i> .....	42
Figura 29 - Lógica da gestão do pedido.....	43
Figura 30 - <i>User interface</i> e lógica do <i>popup</i> de pagamento .....	44
Figura 31 - Casos de uso (aplicação mobile) .....	45
Figura 32 - Fluxo da aplicação móvel.....	45
Figura 33 – <i>user interface</i> do ecrã de <i>On boarding</i> .....	46
Figura 34 - Lógica da listagem de mesas.....	46
Figura 35 - Lógica do menu.....	47

Figura 36 - <i>Lazy load</i> de imagens .....	47
Figura 37 - Lógica do detalhe do produto.....	48
Figura 38 - Lógica do detalhe do pedido .....	49
Figura 39 – Arquitetura da sincronização .....	50
Figura 40 – Modelo de dados da sincronização de <i>read only</i> .....	51
Figura 41 - Lógica de sincronização ( <i>read-only</i> ) do cliente .....	52
Figura 42 - Lógica de sincronização ( <i>read-only</i> ) do cliente .....	52
Figura 43 - Modelo de dados da sincronização <i>read/write</i> .....	53
Figura 44 - Lógica de sincronização ( <i>read/write</i> ) do cliente .....	54
Figura 45 - Lógica de sincronização ( <i>read/write</i> ) do servidor .....	55
Figura 46 - lógica de sincronização .....	56
Figura 47 - Importação dos métodos .....	57
Figura 48 - Método de criação fatura em OutSystems .....	58
Figura 49 - Função de transformação de nomes e <i>assign</i> dos nomes .....	60
Figura 50 - Estrutura invoice .....	60
Figura 51 - Método de edição do estado da fatura .....	61
Figura 52 - Estrutura de alteração do estado da fatura.....	61
Figura 53 - Método de envio.....	62
Figura 54 – Exemplo do email enviado .....	62
Figura 55 - Estrutura de envio de email.....	63
Figura 58 - Processo assíncrono da geração da fatura .....	66
Figura 59 – <i>Business process</i> .....	66
Figura 60 - <i>Automatic activity</i> CreateInvoice .....	66
Figura 61 - Timer para sincronização de taxas de iva .....	67
Figura 62 - <i>Dropdown</i> com a informação das taxas de iva.....	67





# **1. Introdução**

## **1.1. Enquadramento, Objetivos e Motivação**

Atualmente, num mundo em que cada vez são usadas plataformas tecnológicas para facilitar e gerir os serviços aos quais recorremos no nosso dia a dia, somos também cada vez mais exigentes com a eficiência e qualidade dos mesmos. Os estabelecimentos de restauração são um exemplo nos quais é frequente a falta de resposta por ineficiência de gestão, ou mesmo por falta de formação dos seus funcionários, sendo alta a probabilidade de ocorrerem erros na execução dos pedidos ou a sujeição dos clientes a largos tempos de espera. Além disso, as formas de pagamento aceites por estes estabelecimentos são por norma pouco inovadoras, limitando-se a pagamentos em numerário ou por multibanco. Endereçando estas preocupações, o projeto Carta de Restaurante Digital visa otimizar o processamento dos pedidos em unidades de restauração, ajudando na capacidade de resposta no atendimento às mesas, podendo assim reduzir o tempo de espera. Pretende-se também modernizar a experiência do cliente, não só apresentando os menus de uma forma mais apelativa e interativa através de uma aplicação móvel, como possibilitando também a utilização de formas de pagamento mais inovadoras e a facilidade de envio da fatura para o email do cliente.

Motivados pelas considerações e objetivos acima expostos, implementámos uma App móvel designada por Carta de Restaurante Digital, podendo esta ser instalada em tablets pertencentes aos estabelecimentos de restauração, substituindo assim os cardápios tradicionais. Deste modo, os pedidos de cada mesa podem ser efetuados através da aplicação, agilizando assim o envio das solicitações para a cozinha, dado que não é necessário aguardar pela chegada de um empregado de mesa disponível. Por outro lado, a informação apresentada ao cliente para cada uma das ofertas da ementa é mais detalhada, podendo incluir imagens, informação nutricional e sugestões de acompanhamento. Por fim, o pagamento pode ser efetuado também através da aplicação, utilizando formas de pagamento alternativas, ou através de numerário ou multibanco.

## **1.2. Organização do documento**

O restante relatório encontra-se organizado da seguinte forma:

- Formulação do problema: o estado de arte, ou seja, a problemática que motivou o desenvolvimento desta solução;
- Solução: parte técnica do relatório, dividido pelos temas mais relevantes como a segurança, arquitetura, modelo de dados e as aplicações desenvolvidas;
- Avaliação experimental: diferentes cenários, de modo a apresentar as funcionalidades da aplicação;
- Conclusões: apresentação dos resultados obtidos e se estes são favoráveis em relação ao problema em questão.

## 2. Formulação do Problema

Atualmente são cada vez mais o numero de queixas contra os estabelecimentos de restauração por ineficiência de gestão ou por falta de formação dos seus funcionários. Os clientes podem até não gostar de certos pratos ou ficarem descontentes com as condições de higiene, contudo o problema maior incide-se na falta de formação e disciplina dos funcionários. A falta de atenção e a desinformação estão também entre as principais queixas dos clientes. É cada vez mais frequente os funcionários esquecerem-se de pedidos, ou porque não apontam ou porque por vezes são demasiados clientes para um único funcionário. Outro problema grave dos funcionários é não recolherem informação da cozinha de quais os pratos que terminaram ou porque simplesmente se esquecem. É comum em certos estabelecimentos de restauração o funcionário recolher vários pedidos das mesas e no final é que entrega tudo na cozinha, contudo isso é um erro porque a cozinha não consegue ter controlo da ordem de saída dos pedidos.

Nos últimos anos começaram a surgir novas ideias e novos produtos para combater a falta de gestão dos estabelecimentos, como foi o caso do McDonald's. O McDonald's disponibiliza máquinas para realizar pedidos automáticos, isto é, sem auxilio de um funcionário. Estas máquinas disponibilizam um menu interativo com a possibilidade de alterar ingredientes dos produtos e ainda a possibilidade de efetuar o pagamento no balcão ou na própria máquina de vendas. Isto é um conceito bastante inovador porque permite que os clientes façam os seus pedidos sem pressas e sem preocupações, evitando assim de passar imenso tempo em filas. O McDonald's disponibiliza um ecrã gigante com a indicação dos pedidos que estão em processamento e os pedidos que estão prontos a levantar.

Para combater os problemas dos estabelecimentos de restauração pretendemos aplicar o conceito dos pedidos automáticos do McDonald's nos restaurantes tradicionais, e para tal foi decidido usar a tecnologia OutSystems.

O OutSystems é uma *framework* de desenvolvimento rápido e *low code* que tira partido da metodologia *Agile* e que a partir de um único *IDE* é capaz de gerar código .Net e .Java.

De seguida apresenta-se na Figura 1 a arquitetura geral da aplicação, bem como todas as ferramentas disponibilizadas pela OutSystems de modo a agilizar o tempo de desenvolvimento. O **Service Studio** é utilizado para desenvolver código OutSystems, isto é, é onde está toda a lógica da aplicação, desde a geração da aplicação *mobile*, controlo de versões e publicações no servidor. Contudo, podem existir requisitos em que é preciso algo mais do que já foi pensado ou desenvolvido em OutSystems, existindo para isso o **Integration Studio**, em que é possível criar extensões em C# ou Java.

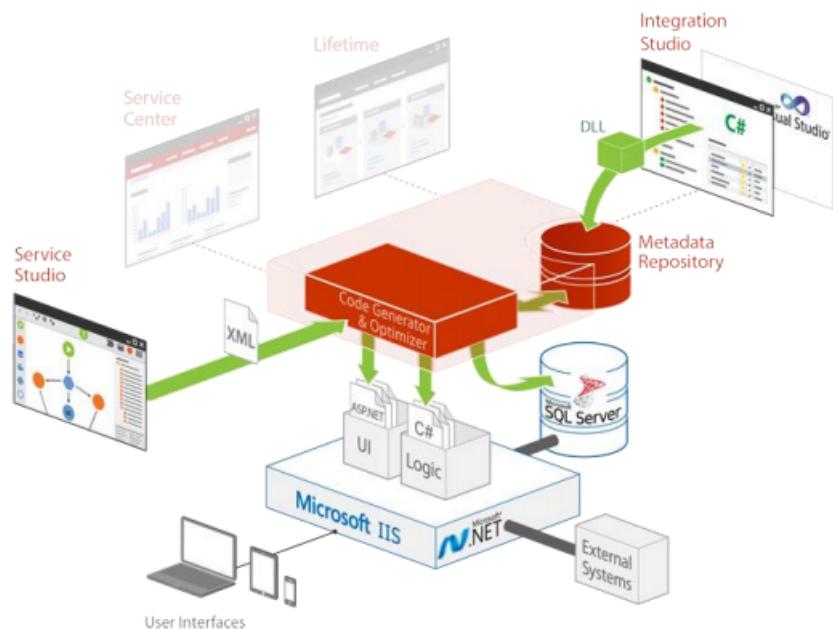


Figura 1 - Arquitetura OutSystems

O **Service Center** e o **Lifetime** são ferramentas onde é feito o registo *log* de erros, as configurações de cada módulo, a verificação dos estados dos servidores, passagens entre ambientes e criação de utilizadores.

Para a construção da arquitetura da aplicação é utilizado o *4 Layer Canvas* que consiste em 4 camadas ilustradas na Figura 2:

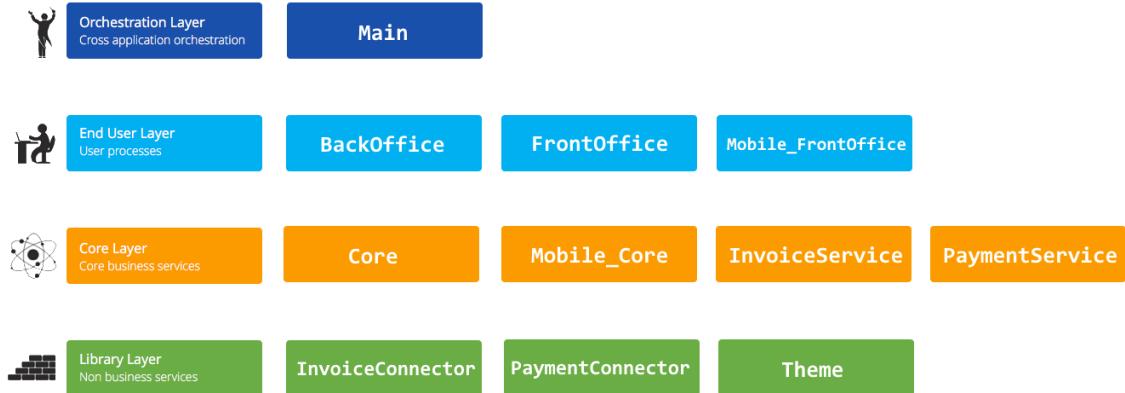


Figura 2 - Arquitetura da aplicação

**Orchestration Layer:** os componentes nesta camada são criados quando é necessário apresentar ao utilizador informação de várias aplicações, permitindo-lhe uma experiência transversal (exemplo: *dashboards*);

**End User Layer:** é a camada onde se encontram todos os ecrãs e processos da aplicação, que de modo a possibilitar a sua construção, reutiliza os componentes das camadas abaixo;

**Core Layer:** é a camada onde se encontram os serviços relacionados com os conceitos do negócio, exposição de entidades e regras de negócios;

**Library Layer:** servem de componentes de execução de serviços, como por exemplo conexões a serviços externos ou padrões de UI.

A plataforma OutSystems permite a criação de perfis, bem como a associação desses perfis a utilizadores registados na aplicação *web*. Os perfis são por sua vez associados a ecrãs e funcionalidades que requerem privilégios de visualização e/ou alteração, controlado assim o acesso aos conteúdos.

**Perfis de Sistema:** os perfis de sistema estão pré-configurados e disponíveis na plataforma OutSystems, consistindo nos seguintes:

- *Anonymous*: a atribuição deste perfil a uma página *web* permite que qualquer utilizador aceda ao seu conteúdo, mesmo que seja um utilizador não autenticado;
- *Registered*: a atribuição deste perfil a uma página *web* garante que esta só é acessível para utilizadores autenticados.



### 3. Solução Carta de Restaurante Digital

A Carta de Restaurante Digital consiste na criação de uma aplicação móvel utilizando a plataforma OutSystems com o objetivo de substituir as tradicionais cartas de menu nos estabelecimentos de restauração. Assim, os pratos disponíveis podem ser consultados e os pedidos efetuados através de um *tablet*, para o qual foi implementada uma aplicação de *front-end*. Toda a configuração do sistema é feita a nível do *back-office*.

#### 3.1. Segurança

A presente secção define o modelo de segurança a ser implementado na nossa aplicação *web* e explica a configuração das restrições de acesso.

#### 3.2. Casos de uso

A presente seção fornece uma visão dos casos de uso implementados através da aplicação *web*. A Figura 3  
**Erro! A origem da referência não foi encontrada.** apresenta os quatro principais perfis e os casos de uso associados:

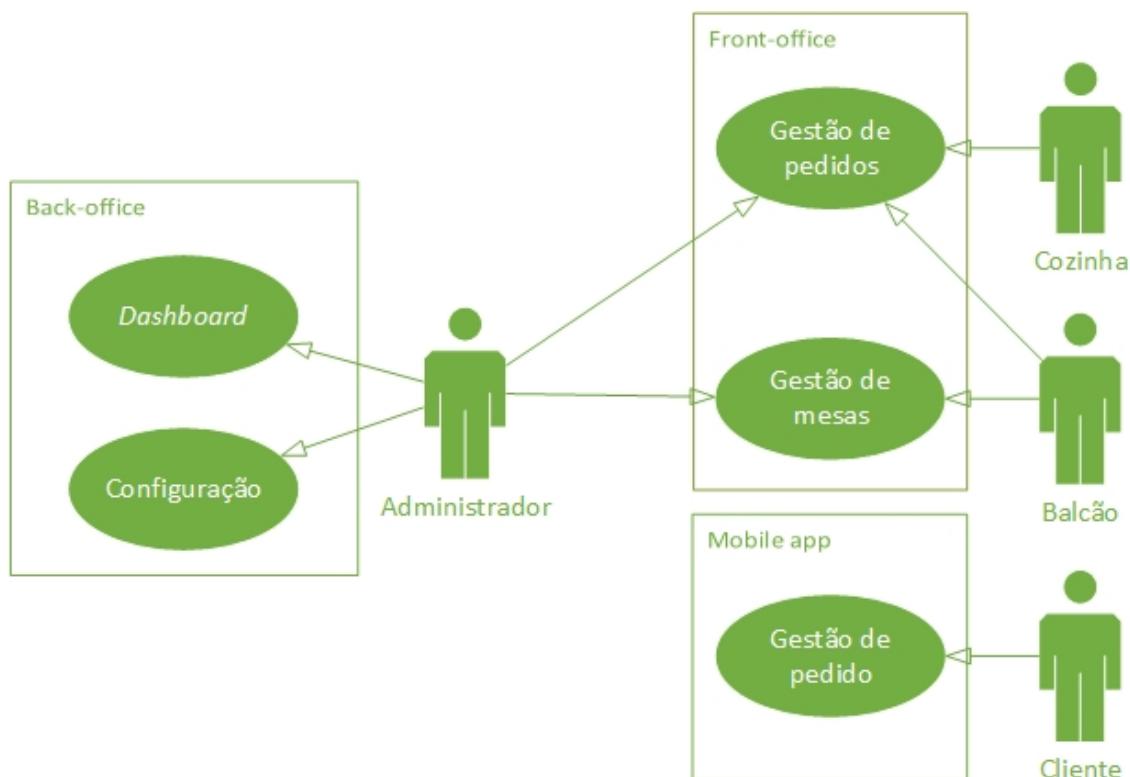


Figura 3 - Casos de uso

O *back-office* é apenas acessível pelo administrador da aplicação, dispondo de um ecrã de *dashboard* onde são apresentados gráficos para análise complementar e ecrãs de configuração de produtos, mesas, categorias, tipos, empresa e utilizadores.

O *front-office* é acessível pelo administrador, cozinha e empregado de mesa/balcão. No *front-office* é possível gerir mesas e pedidos. A gestão de mesas permite ao utilizador verificar que mesas têm um pedido a decorrer e o tempo decorrido. A gestão de pedidos disponibiliza também ferramentas de edição dos produtos associados ao pedido.

A *mobile app* é acessível pelo cliente e empregado de mesa. O empregado é responsável por indicar a mesa do dispositivo móvel, podendo o cliente gerir o seu pedido a partir desse momento.

### 3.3. Perfis

A plataforma OutSystems permite a criação de perfis, bem como a associação desses perfis a utilizadores registados na aplicação *web*. Os perfis são por sua vez associados a ecrãs e funcionalidades que requerem privilégios de visualização e/ou alteração, controlado assim o acesso aos conteúdos.

**Perfis da Aplicação:** a plataforma OutSystems permite a criação de perfis personalizados. A melhor prática é a promoção da separação entre *view mode* e *edit mode*, permitindo uma abordagem mais fina na configuração de segurança das páginas. Para gerir os conteúdos da nossa aplicação foram identificados quatro perfis diferentes, em que cada um tem acesso a funcionalidades distintas:

- Atendimento ao balcão: *dashboard* de gestão de pedidos do balcão;
- Atendimento cozinha: *dashboard* de gestão de pedidos para a cozinha;
- Atendimento cliente: lista de mesas disponíveis da empresa de modo a permitir ao cliente associar o seu pedido;
- Administrador: *dashboard* com informações acerca dos pedidos, produtos e gráficos complementares. O *dashboard* disponibiliza também ligações rápidas às páginas de gestão da empresa, nomeadamente às páginas de gestão de produtos, mesas e utilizadores.

### 3.4. Modelo de dados

A presente secção ilustra na Figura 4 o modelo de dados no qual a Carta de Restaurante Digital se baseia.

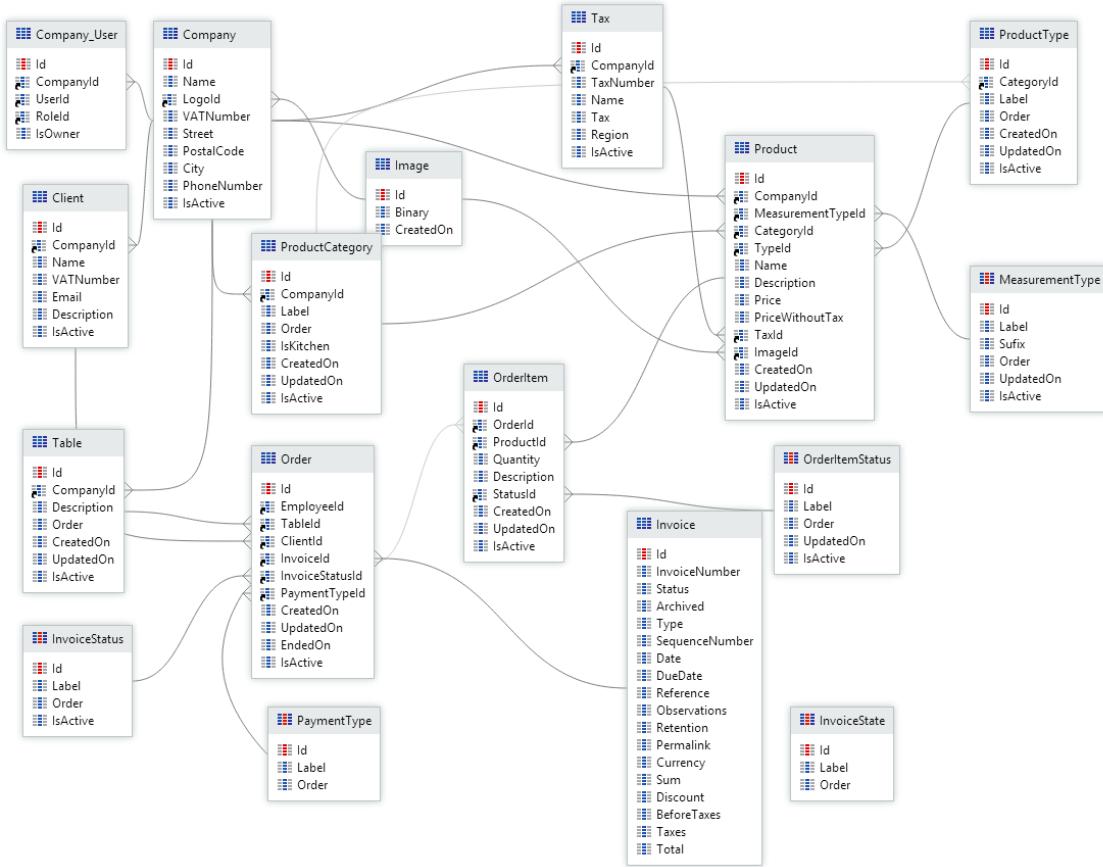


Figura 4 - Modelo de dados

De seguida apresenta-se o modelo de dados com as entidades descritas:

1. **Company:** Entidade responsável por manter os dados da empresa;
2. **Product:** Entidade responsável por manter os produtos da empresa, nomeadamente pratos, bebidas, sobremesas, etc.;
3. **Table:** Entidade responsável por manter as mesas disponíveis associadas à empresa;
4. **Client:** Entidade responsável por manter o registo dos clientes, nomeadamente o nome e contribuinte;
5. **Order:** Entidade responsável por manter os pedidos dos clientes. Quando um cliente submete um pedido, é criado um registo nesta tabela com a informação da mesa;
6. **OrderItem:** Entidade responsável por manter o registo do produto escolhido, associado ao pedido do cliente;
7. **OrderItemStatus:** Entidade estática com os vários estados dos pedidos (novo, em progresso ou processado) associado ao item do pedido;

8. OrderItemLog: Entidade responsável por guardar o registo *log* do processamento de cada item do pedido;
9. Image: Entidade responsável por guardar o binário das imagens utilizadas no registo de produtos e logo da empresa;
10. User: Entidade de sistema (fornecida pelo OutSystems) com o registo dos utilizadores da aplicação;
11. Company\_User: Entidade responsável por guardar os utilizadores permitidos na aplicação (por exemplo o *staff*).
12. Invoice: Entidade responsável por guardar a resposta do serviço de geração de faturas.
13. InvoiceState: Entidade estática com os estados das faturas.
14. PaymentType: Entidade estática com os tipos de pagamentos disponíveis na aplicação.
15. Tax: Entidade responsável por guardar as taxas disponíveis na aplicação.
16. MeasurementType: Entidade estática com os tipos de medidas (doses ou kg).
17. InvoiceStatus: Entidade estática com os vários estados possíveis da geração do pagamento até ao envio por email do cliente.

Devido à existência de uma aplicação móvel, foi criada uma base de dados local que contém as entidades e atributos necessários para o funcionamento dos requisitos da mesma.

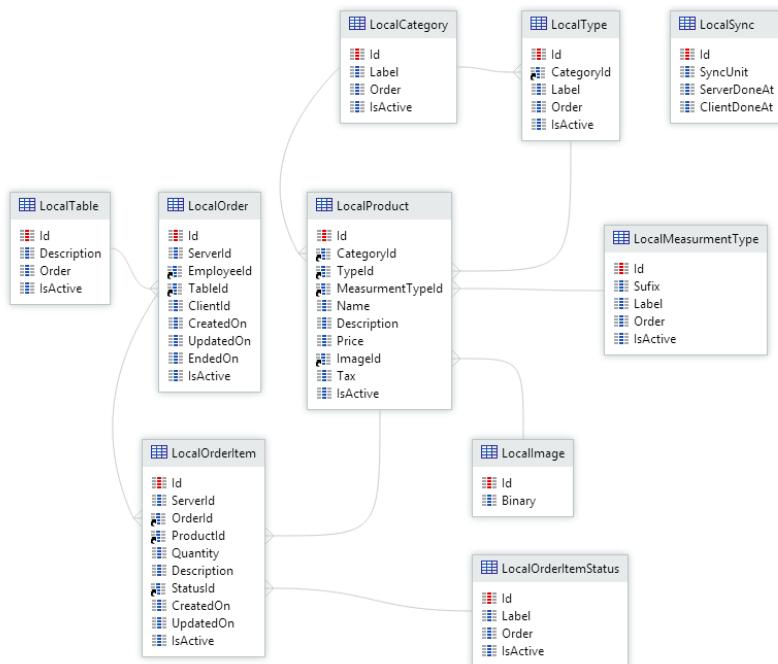


Figura 5 - Modelo de dados local

### 3.5. Ações genéricas

O *eSpace Core* disponibiliza ações genéricas que são utilizadas por toda a aplicação. A sua estrutura divide-se na forma de ações privadas e públicas, como é apresentado na Figura 6. As ações privadas são colocadas em diferentes pastas dependendo da entidade principal utilizada. Isto é, se é uma ação de edição da entidade Company então está deve ser inserida na pasta Company. As ações públicas encontram-se na pasta PublicAPI.

Todas as ações deve, começar com um verbo (ação) seguido por um *underscore* (\_) e a entidade a que esta se destina. Na Figura 7 estão representadas todas as ações deste *eSpace*.

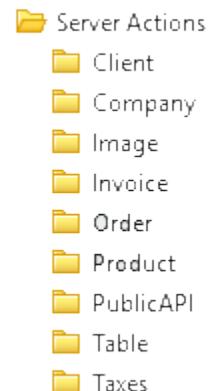


Figura 6 - Ações genéricas

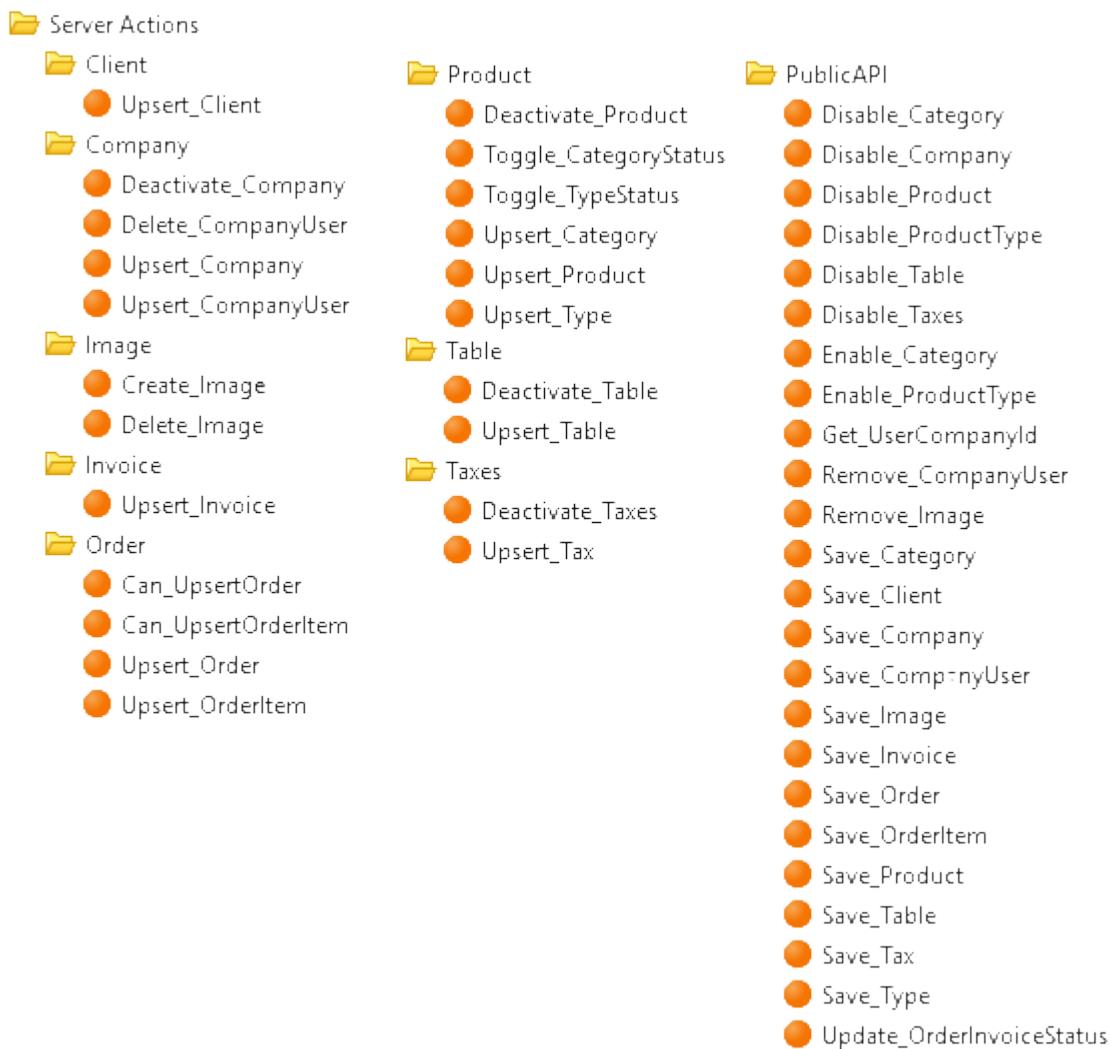


Figura 7 - Ações do *eSpace Core*

### 3.6. Tema

A plataforma OutSystems disponibiliza um mecanismo de criação de temas. Esta funcionalidade permite a criação de *layouts*, adição de css e configurações internas.

Theme	
Name	Theme
Description	
Public	Yes
Mobile	No
Base Theme	LisbonTheme
Style Sheet	html,
OnException Handler	
Grid	
Grid Type	Fluid (from LisbonTheme)
Web Blocks	
Layout	Layouts\Layout_Normal
Info Balloon	
Pop-up Editor	LayoutPopup (from LisbonTheme)
Email	
Header	Common\Header
Menu	Common\Menu
Footer	Common\Footer
Images	
True Image	Icon_Check_Green
False Image	
Info Balloon Image	InfoBalloon_Info

Figura 8 - Tema do OutSystems

### 3.7. Aplicação de *back-office*

A presente secção fornece a visão do papel do *back-office* e mostra as respetivas páginas *web*. A aplicação de *back-office* é a responsável pela administração da empresa, nomeadamente para a configuração de todos os seus dados, bem como a configuração dos produtos disponíveis.

#### 3.7.1. Registo da empresa

O ecrã de registo da empresa ilustrado na Figura 9 é o primeiro ecrã disponível na aplicação. O administrador terá acesso a este ecrã no momento que faz a sua primeira autenticação na aplicação para registar a sua empresa. A página disponibiliza três *breadcrumbs* para guiar o administrador nas etapas de registo necessárias.

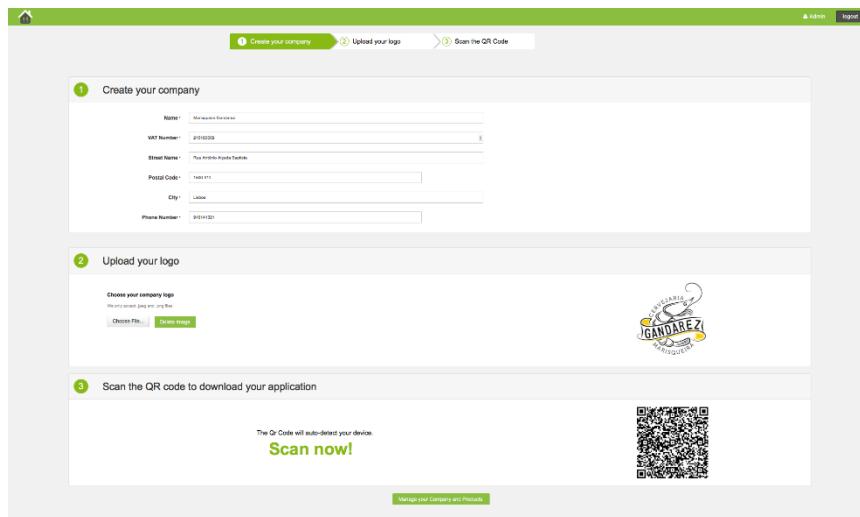


Figura 9 - Ecrã de registo da empresa

A primeira etapa é preencher os dados fiscais da empresa, nomeadamente o nome, número de contribuinte, morada e telefone. Estes dados serão posteriormente usados na geração das faturas. Todos os campos são de preenchimento obrigatório.

A segunda etapa é fazer um *upload* do logotipo da empresa. O logotipo será usado na geração da fatura para o cliente.

A terceira etapa disponibiliza um *QR Code* ao administrador após finalizar a etapa um e dois. O conceito do *QR Code* tem como objetivo facilitar o administrador na aquisição da aplicação para os seus *tablets*. Uma vez que a aplicação só estará disponível internamente e não em nenhuma '*store*', vamos disponibilizar ao administrador a possibilidade de, através de qualquer *tablet* da empresa, fazer a leitura do *QR Code* e o *download* da sua aplicação. O *QR Code* estará disponível posteriormente no ecrã de configuração da empresa.

### 3.7.2. Dashboard

O ecrã de *dashboard* ilustrado na Figura 10 do administrador tem como objetivo fornecer informações acerca das vendas da empresa, nomeadamente as categorias com o maior número de pedidos dos clientes e os produtos mais vendidos no estabelecimento, exibido num formato de gráfico de colunas (*column chart*) e num gráfico circular (*pie chart*), respetivamente. É também disponibilizado ao administrador um gráfico de áreas (*area chart*) com o valor total de vendas mensalmente, durante todo o ano.

Após o fecho de conta, este ecrã (componente opcional) terá um *overview* dos comentários dos clientes.

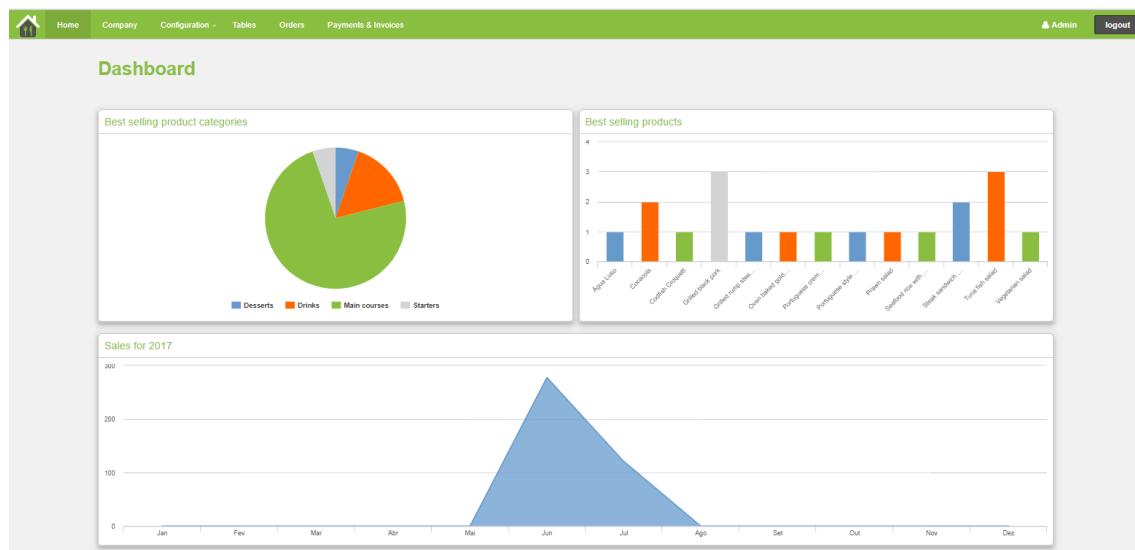


Figura 10 - Ecrã de *dashboard*

### 3.7.3. Configuração da empresa

O ecrã de configuração da empresa ilustrado na Figura 11, permite modificar a qualquer momento os dados fiscais da empresa. É também possível neste ecrã alterar o seu logotipo e realizar o *download* da aplicação através do *QR Code* disponibilizado pela a aplicação.

Name \*: Marisqueira Gандarez  
VAT Number \*: 215190009  
Street Name \*: Rua António Alçada Baptista  
Postal Code \*: 1500-911  
City \*: Lisboa  
Phone Number \*: 916141321

Choose your company logo  
We only accept .jpg and .png files  
Choose File... Delete image

The Qr Code will auto-detect your device.  
**Scan now!**

Cancel Update Company

Figura 11 - Ecrã de configuração da empresa

### 3.7.4. Gestão de taxas

O ecrã de gestão de taxas ilustrado na Figura 12, é o ecrã responsável por listar, criar e editar as taxas que serão aplicadas ao valor dos produtos.

The screenshot shows a web-based application interface for managing taxes. At the top, there is a navigation bar with links for Home, Company, Configuration, Tables, Orders, Payments & Invoices, Admin, and logout. Below the navigation bar, there is a search section titled "Tax Search" with a search input field and "Search" and "Clear" buttons. The main area is titled "List of Taxes" and contains a table with three rows of data. The columns in the table are Name, Tax, Is Active, and Actions. The data in the table is as follows:

Name	Tax	Is Active	Actions
6%	6.00	✓	
13%	13.00	✓	
23%	23.00	✓	

At the bottom left of the table area, it says "3 records".

Figura 12 - Ecrã de gestão de taxas

No ecrã de *search* está disponível um filtro para taxas, o que permite procurar valores de taxas específicas. É também disponibilizado ao administrador a possibilidade de ordenar as colunas da tabela, selecionando e clicando no nome de cada linha do *header*.

Como o nome indica, o botão de *search* permite procurar na entidade *Tax* as taxas que satisfazem o filtro preenchido pelo utilizador, listando o resultado na tabela de taxas. O botão *Clear* faz um *reset* ao filtro, à *query* e à navegação da tabela.

Cada taxa disponível na lista de taxas pode ser editada, para isso está disponível um ícone de edição com um *link* que redireciona para o ecrã de edição de taxas ilustrado na Figura 13, passando o identificador da taxa como argumento. As taxas podem também ser desativados clicando no ícone de desativar correspondente a cada taxa na tabela, ficando assim indisponíveis no momento da criação ou edição de produtos.

The screenshot shows a web-based application interface for editing a tax. At the top, there is a navigation bar with links for Home, Company, Configuration, Tables, Orders, Payments & Invoices, Admin, and logout. Below the navigation bar, the title "Edit Tax" is displayed. The main area is titled "Tax Information" and contains a form with two fields: "Name \*" and "Tax \*". The "Name \*" field has "6%" entered, and the "Tax \*" field has "6" entered. At the bottom right of the form, there are "Cancel" and "Save" buttons.

Figura 13 - Ecrã de edição de taxas

O botão de *Save* da página de criação ou edição é responsável por criar ou atualizar taxas na entidade *Tax*. Dado que a *framework* de faturação *InvoiceXpress* necessita de conhecer estas taxas para poder emitir as respetivas faturas, foi adicionado no botão Save ações que invocam os serviços para criar ou atualizar o valor das taxas no lado da *framework*. O serviço de criação de registo de taxas devolve um identificador, no qual será mantido na entidade *Tax* e usado para futuras atualizações ao registo. Os detalhes dos serviços serão mencionados no capítulo 3.10.

### 3.7.5. Gestão de categorias de produtos

O ecrã de gestão de categorias de produtos ilustrado na Figura 14, é o ecrã responsável por listar, criar e editar as categorias que posteriormente serão associados aos produtos.

Product Category	Order	Is Kitchen	Is Active	Actions
Starters	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<a href="#"></a> <a href="#"></a>
Main courses	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<a href="#"></a> <a href="#"></a>
Desserts	3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<a href="#"></a> <a href="#"></a>
Drinks	4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<a href="#"></a> <a href="#"></a>
Desserts 2	5	<input type="checkbox"/>	<input type="checkbox"/>	<a href="#"></a> <a href="#"></a>

Figura 14 - Ecrã de gestão de categorias de produtos

No ecrã de *search* está disponível um filtro para a categoria, o que permite procurar categorias específicas. É também disponibilizado ao administrador a possibilidade de ordenar as colunas da tabela, selecionando e clicando no nome de cada linha do *header*.

O botão de *search* permite procurar na entidade *ProductCategory* as categorias que satisfazem o filtro preenchido pelo utilizador, listando o resultado na tabela de categorias. O botão *Clear* faz um *reset* ao filtro, à *query* e à navegação da tabela.

Cada categoria disponível na lista de categorias pode ser editada, para isso está disponível um ícone de edição com um *link* que redireciona para o ecrã de edição de categorias de produtos ilustrado na Figura 15, passando o identificador da categoria como argumento. As categorias podem também ser desativados clicando no ícone de desativar correspondente a cada categoria na tabela, desativando consequentemente todos os tipos de produtos e os respetivos produtos

associados. As categorias desativadas ficam também indisponíveis no momento da criação ou edição de produtos.

The screenshot shows a web-based application interface for managing product categories. At the top, there is a navigation bar with links for Home, Company, Configuration (selected), Tables, Orders, Payments & Invoices, Admin (logged in), and logout. The main content area has a title 'Edit Product Category' and a sub-section titled 'Product Category Information'. Within this section, there are three input fields: 'Product Category' with the value 'Starters', 'Order' with the value '1', and a checkbox labeled 'Is Kitchen' which is unchecked. At the bottom right of the form are two buttons: 'Cancel' and 'Save'.

Figura 15 - Ecrã de edição de categorias de produtos

O botão de *Save* da página de criação ou edição é responsável por criar ou atualizar os dados do formulário na entidade *ProductCategory*. A *checkbox* *Is Kitchen* permite indicar que todos os produtos daquela categoria são produtos confeccionados na cozinha do estabelecimento.

### 3.7.6. Gestão de tipos de produtos

O ecrã de gestão de tipos de produtos ilustrado na Figura 16, é o ecrã responsável por listar, criar e editar os tipos de produtos que posteriormente serão associados às categorias e aos produtos.

The screenshot shows a web-based application interface for managing product types. At the top, there is a navigation bar with links for Home, Company, Configuration (with a dropdown menu for Tables, Orders, Payments & Invoices), Admin, and logout. Below the navigation bar is a search section titled "Product Type Search" with a search input field containing placeholder text "Search by product category or product type" and two buttons: "Search" and "Clear". The main content area is titled "List of Product Types" and contains a table with the following data:

Product Category	Product Type	Order	Is Active	Actions
Main courses	Meat	1	✓	
Main courses	Fish	2	✓	
Main courses	Vegan	2	✓	
Main courses	Salads	3	✓	
Drinks	Waters	4	✓	
Drinks	Wine	5	✓	
Drinks	Non Alcholic	6	✓	
Starters	Fish	7	✓	

Below the table, it says "8 records".

Figura 16 - Ecrã de gestão de tipos de produtos

No ecrã de *search* está disponível um filtro para os tipos, o que permite procurar categorias e tipos específicos. É também disponibilizado ao administrador a possibilidade de ordenar as colunas da tabela, selecionando e clicando no nome de cada linha do *header*.

O botão de *search* permite procurar na entidade *ProductType* os tipos que satisfazem o filtro preenchido pelo utilizador, listando o resultado na tabela de tipos de produtos. O botão *Clear* faz um *reset* ao filtro, à *query* e à navegação da tabela.

Cada tipo de produto disponível na lista de tipos de produtos pode ser editado, para isso está disponível um ícone de edição com um *link* que redireciona para o ecrã de edição de tipos de produtos ilustrado na Figura 17, passando o identificador do tipo como argumento. Os tipos podem também ser desativados clicando no *icon* de desativar correspondente a cada tipo na tabela, desativando consequentemente todos os tipos de produtos e os respetivos produtos associados. Os tipos de produtos desativados ficam também indisponíveis no momento da criação ou edição de produtos.

Product Type Information

Product Category \*

Product Type \*

Order \*

Cancel Save

Figura 17 - Ecrã de edição de tipos de produto

O botão de *Save* da página de criação ou edição é responsável por criar ou atualizar os dados do formulário na entidade *ProductType*.

### 3.7.7. Gestão do menu

O ecrã de gestão do menu ilustrado na Figura 18, é o ecrã responsável por listar, criar e editar os produtos que ficam disponíveis na aplicação móvel.

The screenshot shows a mobile application interface for managing products. At the top, there is a navigation bar with links for Home, Company, Configuration, Tables, Orders, Payments & Invoices, Admin, and Logout. Below the navigation bar is a search section titled "Product Search" with fields for Name, Price, and Tax, each with dropdown menus. There are also dropdown menus for Category and Type, and buttons for "Search" and "Clear". Below the search section is a table titled "List of Products" with the following columns: Name, Category, Type, Price, Tax, Measurement Type, Is Active?, and Actions. The table lists various food items such as Água Luso, CocaCola, Codfish "a bras", Codfish Croquet, etc., along with their details like category (Drinks, Main courses, Starters, Desserts, Salads), type (Waters, Fish, Meat), price, tax rate (13% or 23%), measurement type (Doses or Mass (in kg)), and whether they are active (indicated by a checked checkbox). A "New Product" button is located at the top right of the product list. At the bottom left of the list, it says "18 records".

Figura 18 - Ecrã de gestão do menu

No ecrã de *search* estão disponíveis filtros para os produtos, o que permite procurar produtos específicos. É também disponibilizado ao administrador a possibilidade de ordenar as colunas da tabela, selecionando e clicando no nome de cada linha do *header*.

Como o nome indica, o botão de *search* permite procurar na entidade *Product* os produtos que satisfazem os filtros preenchidos pelo utilizador, listando o resultado na tabela de produtos. O botão *Clear* faz um *reset* aos filtros, à *query* e à navegação da tabela.

Cada produto disponível na lista de produtos pode ser editado, para isso está disponível um ícone de edição com um *link* que redireciona para o ecrã de edição de produtos ilustrado na Figura 19, passando o identificador do produto como argumento. Os produtos podem também ser desativados clicando no *icon* de desativar correspondente a cada produto na tabela, ficando assim indisponíveis na aplicação móvel. O administrador pode voltar a ativar o produto a qualquer momento, no entanto é feito a validação se o produto tem uma categoria associada.

O botão *New Product* redireciona o administrador para o ecrã de criação de produtos ilustrado na Figura 19, no qual só estará disponível ao administrador.

Figura 19 - Ecrã de edição do produto

A forma como o ecrã distingue se está em modo criação ou em modo edição, é pelo identificador recebido por argumento. Se esse valor for diferente de zero, indica que está em modo edição e então irá buscar a informação daquele produto à base de dados, senão disponibiliza apenas os campos no ecrã para o administrador preencher com os dados do novo produto. Todos os campos são de preenchimento obrigatório.

Uma vez que o administrador salve os dados para a base de dados, é criado primeiro um registo na entidade *Image* com o binário da imagem e só depois é que é criado um registo na entidade *Product*, incluindo o identificador do registo da imagem. O botão de *Cancel* redireciona o administrador para o ecrã da listagem de produtos.

### 3.7.8. Gestão de mesas

O ecrã de gestão de mesas ilustrado na Figura 20, é o ecrã responsável por listar, criar e editar as mesas disponíveis da empresa. As mesas configuradas no *back-office* ficam disponíveis na aplicação móvel, de forma a que os pedidos estejam associados às mesas respetivas.

The screenshot shows a mobile application interface for managing tables. At the top, there is a navigation bar with icons for Home, Company, Configuration (with a dropdown arrow), Tables, Orders, Payments & Invoices, Admin, and logout. Below the navigation bar is a search section titled 'Table Search' with a search input field and 'Search' and 'Clear' buttons. The main area is titled 'List of Tables' and contains a table with the following data:

Name	Order	Is Active	Actions
Table one	1	✓	
Table two	2	✓	
Table three	3	✓	
Table four	4	✓	

Below the table, it says '4 records'.

Figura 20 - Ecrã de gestão de mesas

No ecrã de *search* está disponível um filtro para as mesas, o que permite procurar uma mesa específica. É também disponibilizado ao administrador, a possibilidade de ordenar as colunas da tabela, selecionando e clicando no nome de cada linha do *header*.

O botão de *search*, permite procurar na entidade *Table* as mesas que satisfazem o filtro preenchido pelo utilizador, listando o resultado na tabela de mesas. O botão *Clear* faz um *reset* aos filtros, à *query* e à navegação da tabela.

Cada mesa disponível na lista de mesas pode ser editada, para isso está disponível um *icon* de edição com um *link* que redireciona para o ecrã de edição de mesas ilustrado na Figura 21, passando o identificador da mesa como argumento.

The screenshot shows the 'Edit Table' screen. At the top, there is a navigation bar with icons for Home, Company, Configuration (with a dropdown arrow), Tables, Orders, Payments & Invoices, Admin, and logout. Below the navigation bar is a title 'Edit Table'. The main area is titled 'Table Information' and contains a form with the following fields:

Name *	<input type="text" value="Table one"/>
Order *	<input type="text" value="1"/>

At the bottom right are 'Cancel' and 'Save' buttons.

Figura 21 - Ecrã de edição da mesa

O ecrã de criação e edição de mesas está disponível exclusivamente ao administrador. A forma como o ecrã distingue se está em modo criação ou em modo edição, é pelo identificador recebido por parâmetro. Se este for diferente de zero, indica que está em modo edição indo buscar a informação daquela mesa à base de dados, senão disponibiliza apenas os campos no ecrã para o administrador preencher com os dados da nova mesa. Todos os campos são de preenchimento obrigatório.

Uma vez que o administrador salve os dados para a base de dados, é criado um registo na entidade Table. O botão de *Cancel* redireciona o administrador para o ecrã de listagem de mesas.

### 3.7.9. Gestão de utilizadores

O ecrã de gestão de utilizadores ilustrado na Figura 22, é o ecrã responsável por listar, criar e editar utilizadores na aplicação. Todos os utilizadores criados na aplicação são automaticamente associados ao perfil de sistema *registered*.

The screenshot shows a web application interface for managing users. At the top, there is a navigation bar with links for Home, Company, Configuration (with a dropdown arrow), Tables, Orders, Payments & Invoices, Admin (logged in as Admin), and logout. Below the navigation bar is a search section titled "Users Search" with a "Name" input field, a "Search" button, and a "Clear" button. Underneath is a table titled "List of Users" with a "New User" button. The table has columns for Name, Username, Role, and Actions. It lists six records:

Name	Username	Role	Actions
Admin	admin	Administrator	
Employee	employee	Employee	
Counter	counter	Employee	
Kitchen	kitchen	Kitchen	
Marcio	marcio	Administrator	
José Gandarez	josepgandarez	Administrator	

At the bottom left of the table area, it says "6 records".

Figura 22 - Ecrã de gestão de utilizadores

No ecrã de *search* está disponível um filtro para o nome ou *username*, o que permite procurar por um utilizador associado à nossa aplicação. É também disponibilizado ao administrador, a possibilidade de ordenar as colunas da tabela, selecionando e clicando no nome de cada linha do *header*.

O botão de *search*, permite procurar na entidade *Company\_User* os utilizadores que satisfazem o filtro preenchido pelo utilizador, listando o resultado na tabela de utilizadores. O botão *Clear* faz um *reset* aos filtros, à *query* e à navegação da tabela.

Dado que a aplicação dispõe de quatro perfis, é possível na página de gestão de utilizadores ilustrado na Figura 23 associar um dos perfis ao utilizador através da *dropdown Role*. O administrador pode a qualquer momento remover ou alterar os perfis dos utilizadores.

The screenshot shows a user interface for editing user information. At the top, there is a navigation bar with links for Home, Company, Configuration, Tables, Orders, Payments & Invoices, and a user icon labeled 'Admin' with a 'logout' button. Below the navigation bar is a section titled 'User Information'. This section contains five input fields: 'Name \*' with the value 'Admin', 'Username \*' with the value 'admin', 'Role \*' with the value 'Administrator' (selected from a dropdown menu), 'Password' (empty field), and 'Confirm Password' (empty field). At the bottom right of this section are 'Cancel' and 'Save' buttons. The overall background is light grey, and the main content area has a white background with a thin border.

Figura 23 - Ecrã de edição do utilizador

### 3.8. Aplicação de *front-office*

A presente secção ilustrada na Figura 24 fornece a visão do papel da aplicação de *front-office* e os respetivos ecrãs. A aplicação é a responsável pela execução de pedidos por parte do empregado de balcão.

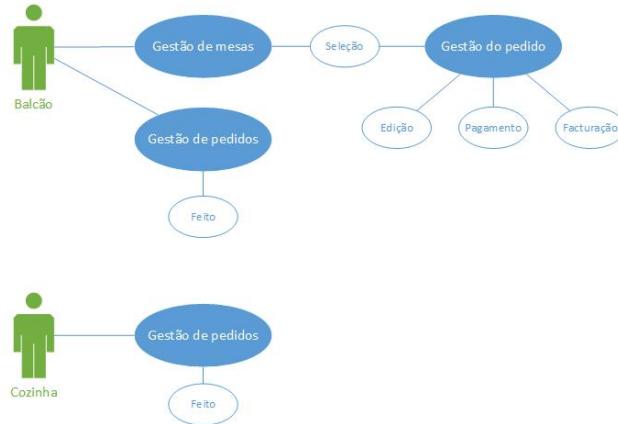


Figura 24 - Casos de uso do *front-office*

#### 3.8.1. Gestão de mesas

O utilizador com o perfil de atendimento ao balcão consegue ver todas as mesas existentes no estabelecimento onde está inserido, tendo cada mesa a indicação se possui um pedido em curso. Esta informação é obtida pela query GetTables.

Ao selecionar uma mesa, o utilizador é redirecionado para o ecrã de gestão do pedido ilustrado na Figura 25.

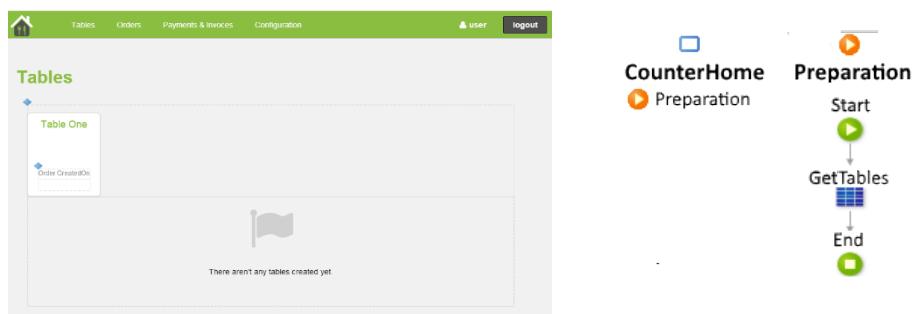


Figura 25 - *User interface* da gestão de mesas e lógica da gestão de mesas

### 3.8.2. Gestão de pedidos de mesas

O ecrã de gestão de pedidos das mesas ilustrado na Figura 26 apresenta uma listagem com os pedidos ativos, ou seja, dos pedidos que ainda estão a decorrer. Por cada pedido é apresentado todos os produtos selecionados, a hora de inicio, um símbolo com a indicação do estado do pedido e a opção de marcar um produto com feito. Isto é, os produtos quando são adicionados ao pedido estão no estado “por realizar” e é necessário que um utilizador com o perfil de atendimento de balcão ou cozinha registe que o produto está pronto.

A diferença entre estes dois perfis, são os produtos apresentados. O perfil de atendimento de balcão consegue ver todos os pedidos, mas o perfil de cozinha apenas consegue ver os pedidos que são para a cozinha. O atributo IsKitchen da entidade ProductCategory realiza esta distinção.

De modo a ter um acesso diferente para cada perfil foram criados dois ecrãs: KitchenHome e Orders. Ambos têm o mesmo *web block* denominado Orders, o que difere é o parâmetro de entrada IsKitchen.

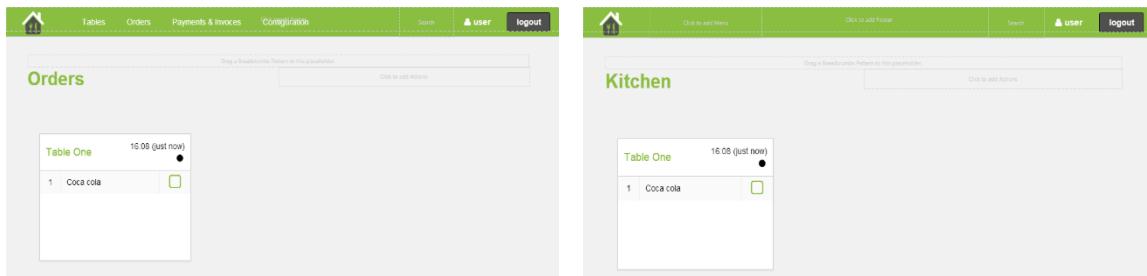


Figura 26 - Ecrãs de gestão de pedidos do balcão e da cozinha

O *web block Orders* ilustrado na Figura 27 obtém no *preparation* os pedidos ativos e a data e hora do produto que está à mais tempo por realizar. Em cada pedido, é utilizado o *web block OrderItems* de modo a apresentar os produtos.

Este *web block* também é responsável por atualizar a sua lista a cada 10 segundos. Como tal, foi criado um botão escondido com a classe *RefreshButton* e foi adicionado o javascript necessário na página. Quando este botão é selecionado, a ação *Refresh\_Table* é executada.

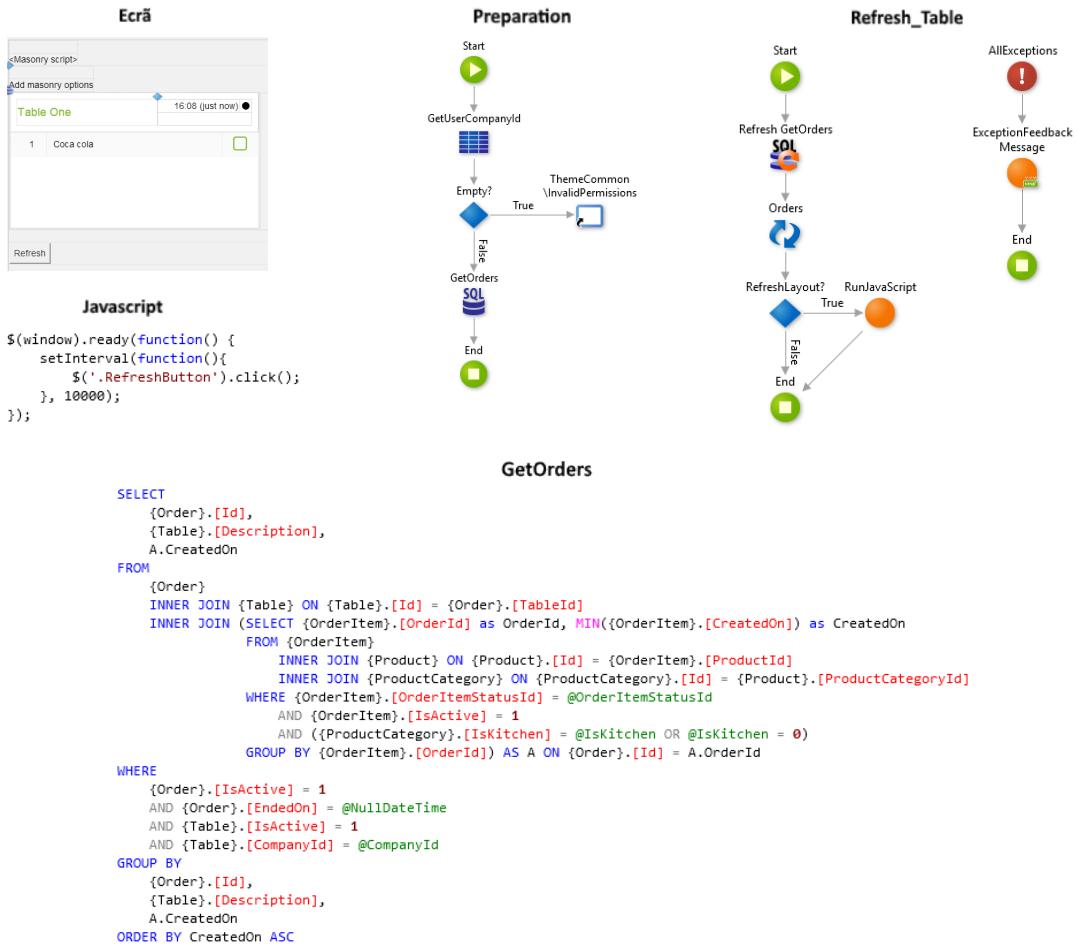


Figura 27 - User interface e lógica do web block Order

O web block *OrderItems* ilustrado na Figura 28 apresenta uma listagem dos produtos selecionados daquele pedido e tem a opção de registrar o produto como feito.

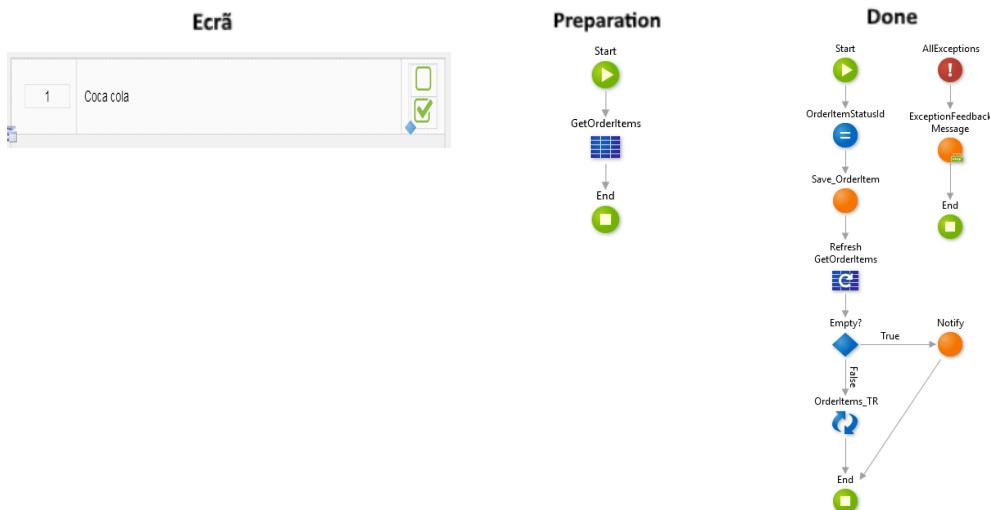


Figura 28 - User interface e lógica do web block OrderItems

### 3.8.3. Gestão de pedido de mesas

O ecrã de gestão de pedido de mesa, apresenta o estado atual do pedido e uma listagem de produtos. Esta contém todos os produtos que estão ativos para aquela empresa e o utilizador pode adicioná-los ao pedido através do botão *Add* de cada produto. Esta listagem tem a hipótese de ser filtrada através nome ou descrição do produto, categoria e tipo do mesmo.

Na Figura 29 é representada a lógica deste ecrã em que se divide nas seguintes ações:

- *Preparation*: obtenção dos dados necessários para apresentar a listagem de produtos e o pedido;
- *Add*: adição de um produto ao pedido e criação do pedido se este ainda não foi criado;
- *OnChange\_Category*: atualização dos tipos de produtos apresentados devido à alteração da categoria na pesquisa;
- *OnNotify\_OrderChanged*: atualização do preço total do pedido devido a ter ocorrido uma alteração nos produtos associados ao pedido;
- *Refresh\_Table*: atualização dos produtos apresentados devido a uma alteração dos filtros ou da paginação.

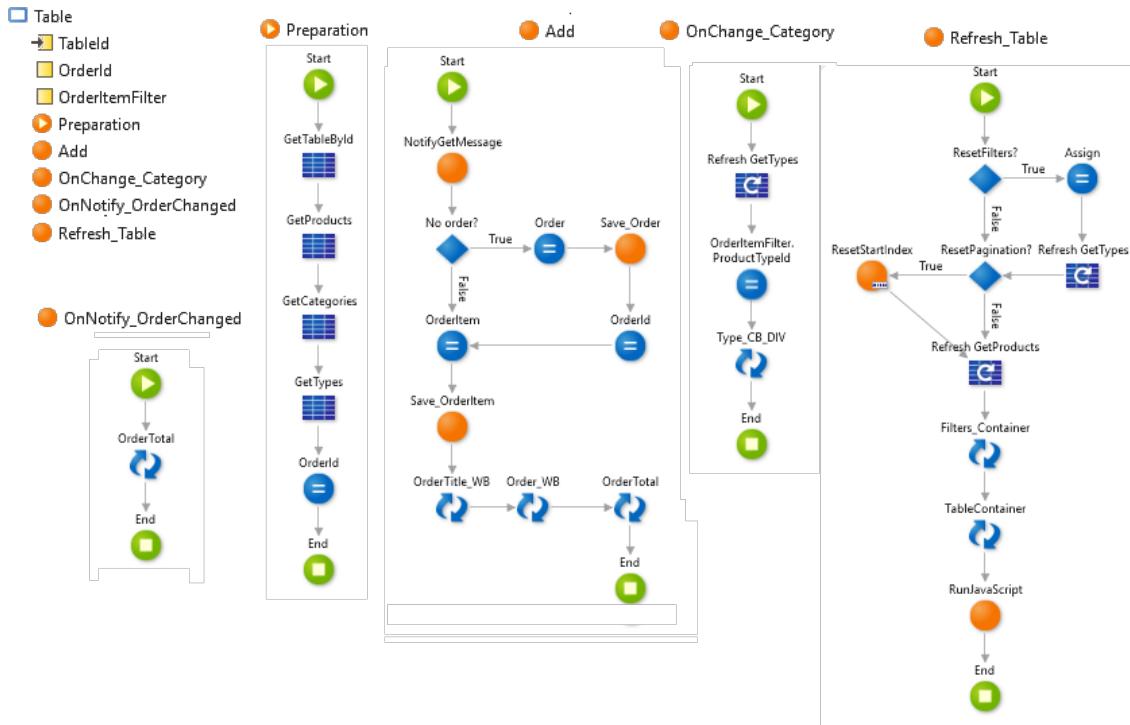


Figura 29 - Lógica da gestão do pedido

### 3.8.4. Pagamento

O *popup* de pagamento é responsável por efetuar o pagamento e o fecho de conta. De modo a efetuar o pagamento é necessário decidir se o tipo de pagamento é monetário ou multibanco. Conforme ilustra a Figura 30, existem mais três campos para preenchimento:

O campo de email é de preenchimento obrigatório e utiliza o padrão denominado *auto complete*. Isto é, à medida que o utilizador escreve o seu email, são apresentados os emails dos clientes que já efetuaram um pagamento e que comecem pelo mesmo conjunto de letras. Se um desses emails for selecionado, os campos de nome e número de faturação são preenchidos automaticamente.

Os campos de nome e número de faturação não são obrigatórios dado que o cliente pode optar por não introduzir esses dados. Como o nome indica, o campo de número de faturação apenas permite a introdução de números. Esta funcionalidade foi realizada com o auxílio de uma mascarada associada ao *input* desse mesmo campo.

Para efetuar o pagamento, o utilizador deve selecionar o botão de pagamento (Pay). Posteriormente são efetuadas as validações necessários e é apresentada uma mensagem de erro, se este existe. Após o pagamento, os dados são guardados na base de dados e é iniciado o processo de geração de fatura. Este processo será mencionado no capítulo 3.10.

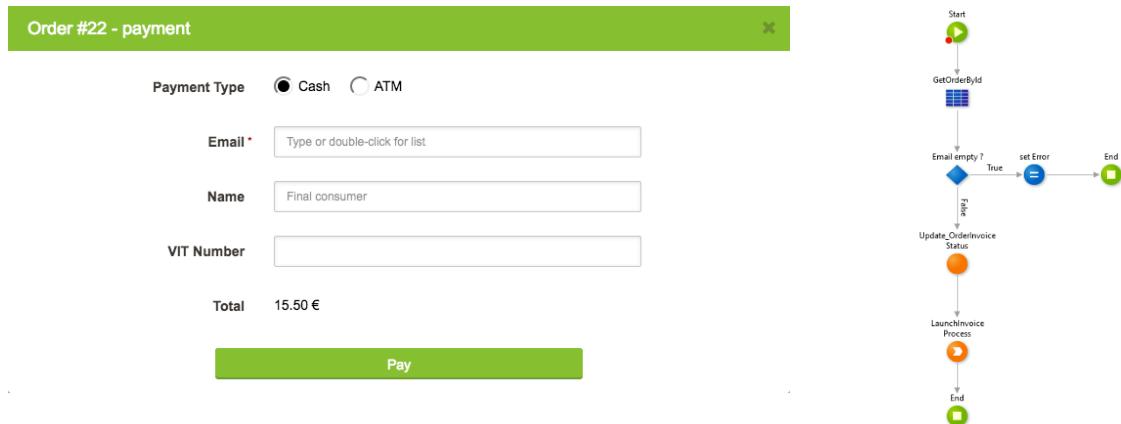


Figura 30 - *User interface* e lógica do *popup* de pagamento

### 3.9. Aplicação móvel (tablet)

A presente secção fornece a visão do papel da aplicação móvel e os respetivos ecrãs. A aplicação será responsável pela execução de pedidos por parte do cliente. A sua estrutura divide-se em dois *eSpaces*: CMD\_Core onde se encontram as ações genéricas e CMD\_FrontOffice que contém a user interface e a lógica associada à mesma.



Figura 31 - Casos de uso (aplicação mobile)

O fluxo da aplicação móvel inicia-se com a autenticação por parte do empregado da empresa no dispositivo, após é necessário realizar a escolha da mesa onde se encontra o *tablet*. A partir deste momento a interacção com a aplicação é realizada pelo cliente, e este começa no ecrã de *on boarding*. O *on boarding* passa por uma sucinta explicação da aplicação.

Após o ecrã acima referido, é apresentado o menu e o cliente pode realizar os seus pedidos.

#### 3.9.1 Fluxo da aplicação

Na Figura 32 apresenta-se o fluxo dos ecrãs da aplicação.

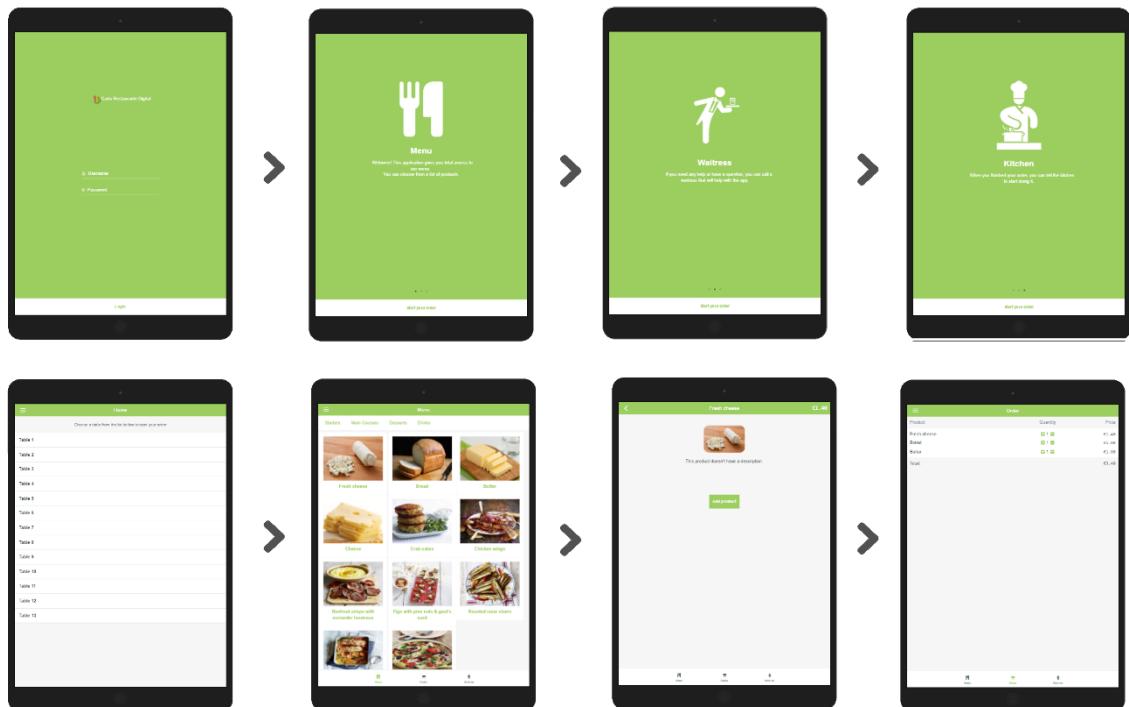


Figura 32 - Fluxo da aplicação móvel

### 3.9.2 On boarding

O utilizador começa o fluxo das suas ações neste ecrã. Apresenta um *carousel* com uma sucinta explicação sobre o propósito e as ações desta aplicação.

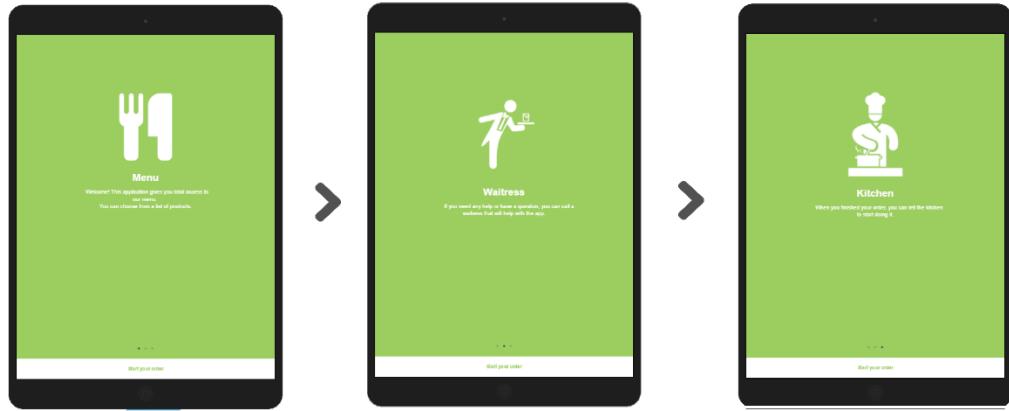


Figura 33 – user interface do ecrã de On boarding

### 3.9.3 Listagem de mesas

A página de listagem de mesas ilustrado na Figura 34 é necessária para indicar qual é a mesa onde se encontra o cliente, esta seleção é realizada pelo empregado. Por cada mesa é apresentada a sua descrição e, se está a decorrer um pedido na mesma, a data e hora do inicio do pedido e um ícone de uma caixa registadora. Ao selecionar uma mesa é criado um registo na entidade *LocalOrder* e o utilizador é redirecionado para o ecrã que contém o menu.

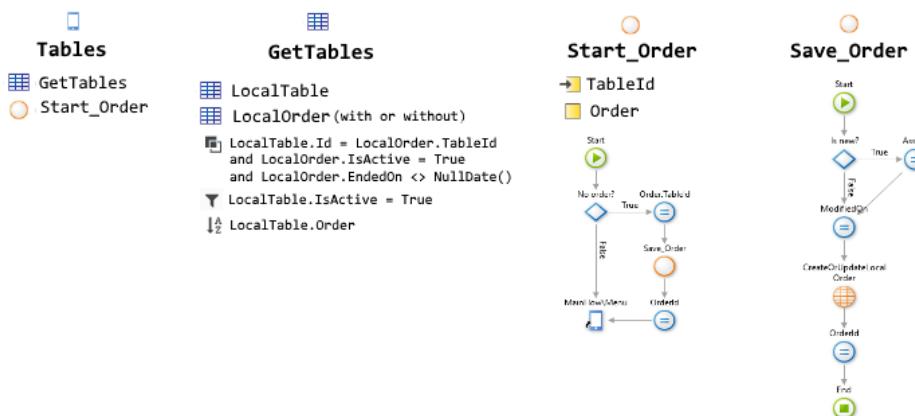


Figura 34 - Lógica da listagem de mesas

### 3.9.4 Menu

No ecrã do menu apresenta-se a listagem de produtos disponíveis da empresa. Existe um menu onde se pode escolher a categoria e o tipo do produto de modo a filtrar os produtos apresentados. Ao escolher uma categoria ou tipo de produto, é realizada a ação *OnCategoryClick* ou *OnTypeClick* que consistem numa atualização (*refresh*) da query *GetProducts*. Quando utilizador seleciona um produto, este é redirecionado à página com o detalhe desse mesmo produto.

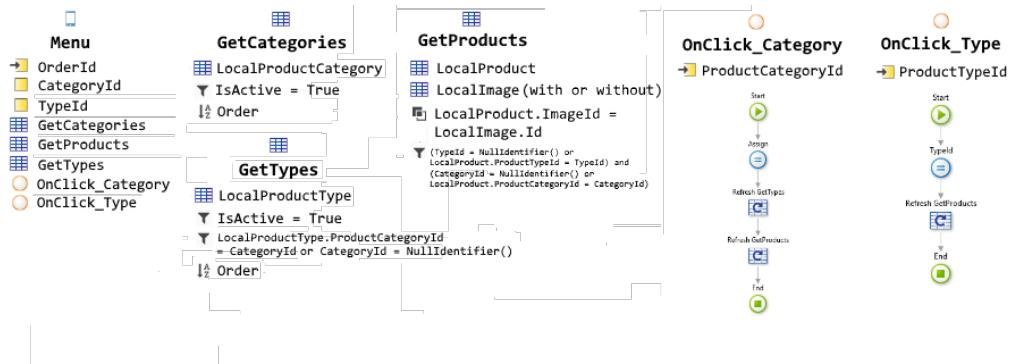


Figura 35 - Lógica do menu

Devido a um dispositivo móvel não ter a mesma capacidade de processamento que um *desktop*, foi decidido que o carregamento de imagens deve utilizar o padrão de *lazy load*. Isto é, em vez de obter todas as imagens de uma vez através de uma *query*, foi criado um *web block* que apenas obtém a imagem na ação de *on render*. Esta ação apenas é executada quando o elemento foi apresentado (“renderizado”) no dispositivo. Na Figura 36 está representada a lógica deste elemento. O parâmetro *IsLoading* identifica se a imagem está a ser obtida. A ação Refresh é executada nas condições mencionadas anteriormente e obtém a imagem da base de dados.

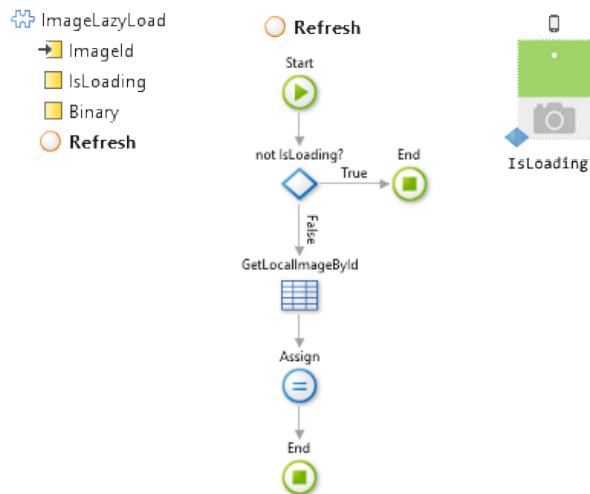


Figura 36 - *Lazy load* de imagens

### 3.9.5 Detalhe do produto

O detalhe do produto é composto pelos seguintes atributos: título, descrição, imagem e preço. A query *GetProduct* ilustrado na Figura 37 é responsável por obter esta informação.

Quando o utilizador seleciona o botão para adicionar este produto aos seus pedidos, a ação *Order* é executada. Nesta ação é criado um registo na entidade *OrderItem* que por sua vez está associada através de *foreign keys* com as entidades *Product* e *Order*.

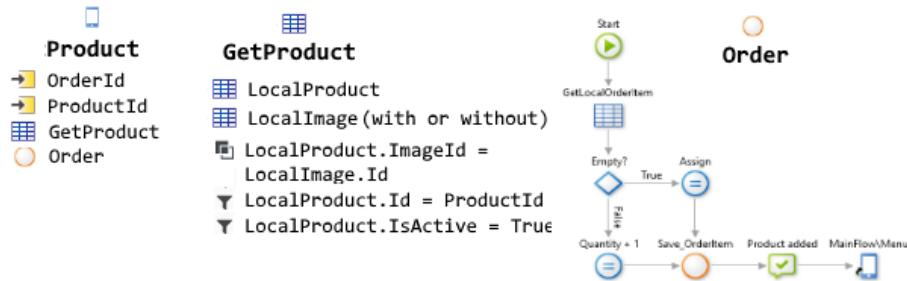


Figura 37 - Lógica do detalhe do produto

### 3.9.6 Detalhe do pedido

Até o pedido ser terminado (pago ou cancelado pelo administrador), é disponibilizada uma página com os produtos (pratos) que o utilizador selecionou. Esta listagem é obtida pela *query GetOrder* ilustrado na Figura 38 e por cada produto é apresentado o seu nome e o valor. No final da listagem encontra-se a soma total do valor de todos os produtos pedidos, que é suportado pela *query GetOrderTotal*.

O utilizador tem ainda a opção de aumentar ou reduzir a quantidade do produto. Ao realizar essa interação, a ação *Update\_OrderItem* é executada e o registo na entidade *OrderItem* é atualizado.

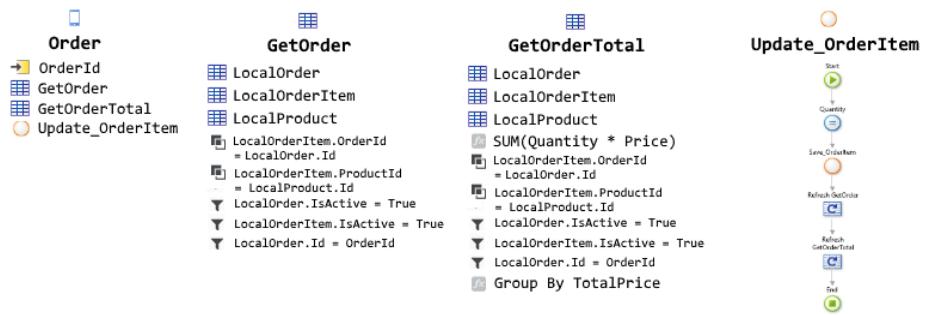


Figura 38 - Lógica do detalhe do pedido

### 3.9.7 Sincronização da base de dados local

Apesar da ideia de estarmos permanentemente conectados, os dispositivos móveis estão em constante movimento e como tal, a sua ligação à internet pode ser interrompida. De modo a oferecer uma experiência completa da aplicação móvel, as funcionalidades em modo *offline* têm de se manter. A plataforma OutSystems tem mecanismos preparados para suportar o modo *offline* (lógica *core* e *user interface*). Apenas é necessário implementar a base de dados local e a lógica da sincronização como ilustrado na Figura 39.

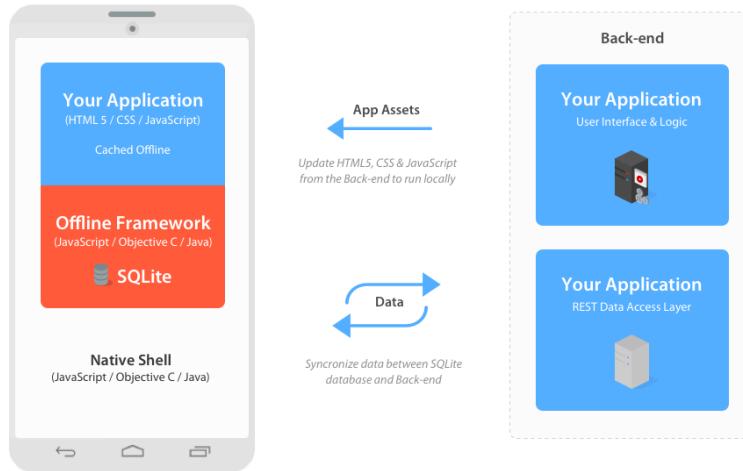


Figura 39 – Arquitetura da sincronização

A plataforma OutSystems recomenda os seguintes padrões de desenho:

- **Read-only data:** utilizado em entidades que são apenas de leitura e para pequenas quantidades de dados;
- **Read-only data optimized:** utilizado em entidades que são apenas de leitura e para grandes quantidades de dados;
- **Read/write data last write wins:** utilizado em entidades que podem ser alterados por vários utilizadores e é recomendado quando existe pouca probabilidade dos dados serem alterados ao mesmo tempo;
- **Read/write data with conflict detection:** utilizado em entidades que podem ser alteradas por vários utilizadores e é recomendado para cenários avançados quando os dados são alterados ao mesmo tempo;
- **Read/write data one-to-many:** utilizado em entidades com relações de associação de 1 para N, é recomendado quando existe pouca probabilidade dos dados serem alterados ao mesmo tempo.

Tendo em conta todos os padrões foram escolhidos os seguintes:

**Read-Only Data Optimized:** este padrão foi escolhido devido a ter um acréscimo da *performance* em relação aos restantes. As entidades que apenas têm dados de leitura e utilizam este padrão são as seguintes: ProductCategory, Image, MeasurmentType, OrderItemStatus, Product, Table, ProductType, LocalCategory, LocalImage, LocalMeasurmentType, LocalOrderItemStatus, LocalProduct, LocalTable e LocalType.

- A base de dados do servidor mantém os dados atualizados;
- A sincronização é otimizada de modo a transferir apenas os dados que foram alterados desde a última sincronização;
- As alterações efetuadas na aplicação móvel não são sincronizadas para o servidor.

A lógica deste padrão é descrita nos seguintes passos:

1.  Obtenção dos dados do servidor;
2.  Envio dos registos alterados desde a última sincronização;
3.  Atualização dos dados da aplicação móvel.

O modelo de dados foi desenhado com este padrão em consideração. Todas as entidades do servidor têm um atributo denominado UpdatedOn (1 da Figura 40) do tipo *date time*. E como o nome indica, é atualizado com o dia e hora atual sempre que é feita uma alteração.

Na aplicação móvel foi criada a entidade LocalSync que indica o tipo de sincronização e a data e hora que esta foi realizada (2 da Figura 40).

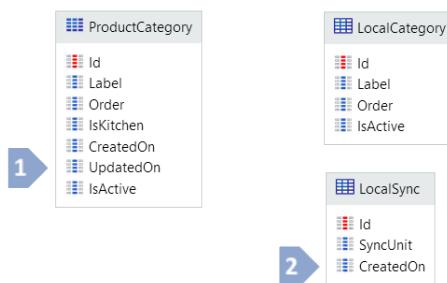


Figura 40 – Modelo de dados da sincronização de *read only*

A lógica de sincronização do cliente é apresentada na Figura 41.

1. Obtenção da data e hora da última sincronização;
2. Obtenção dos dados do servidor que foram alterados desde a última sincronização. A ação ServerSync retorna uma lista dos dados alterados e a data e hora desta sincronização;
3. Atualização dos dados do cliente;
4. Alteração da data e hora da sincronização;
5. Atualização desta sincronização.



Figura 41 - Lógica de sincronização (*read-only*) do cliente

A lógica de sincronização do cliente é apresentada na Figura 42.

1. Verificação se o utilizador tem as permissões necessárias;
2. *Log* da sincronização;
3. Obtenção dos dados atualizados desde a última sincronização;
4. Retorno da lista com os dados atualizados e data e hora da sincronização.

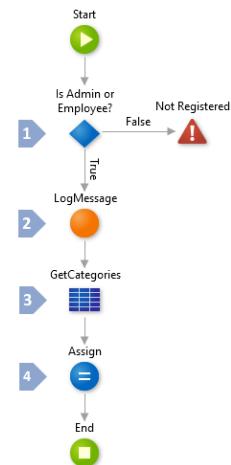


Figura 42 - Lógica de sincronização (*read-only*) do cliente

**Read/Write Data:** neste padrão é possível alterar valores no cliente e estes têm de ser sincronizados para o servidor e vice-versa. Não foi escolhido um dos padrões disponibilizados pela OutSystems mas sim uma aproximação ao que é proposto pelos mesmos. As entidades com dados de leitura e escrita que utilizam este padrão são as seguintes: Order e OrderItem.

- A base de dados do servidor mantém os dados atualizados;
- As alterações efetuadas no cliente (aplicação móvel) são sincronizadas para o servidor;
- A sincronização é otimizada de modo a transferir apenas os dados que foram alterados desde a última sincronização.

A lógica deste padrão é descrita nos seguintes passos:

1.  Envio dos registos alterados no cliente, desde a última sincronização;
2.  Atualização dos dados do servidor;
3.  Envio dos dados atualizados pelo cliente (identificador do registo no servidor);
4.  Envio dos dados alterados no servidor;
5.  Atualização dos dados do cliente;

O modelo de dados foi desenhado com este padrão em consideração. Todas as entidades do servidor têm um atributo denominado UpdatedOn (1 da Figura 43) do tipo *date time*. E como o nome indica, é atualizado com o dia e hora atual sempre que é feita uma alteração. O mesmo acontece na base de dados local, todas as entidades da mesma têm um atributo denominado UpdatedOn (3 da Figura 43). Na sincronização, do lado do cliente, é utilizado o atributo ServerId que indica o Id da entidade que se encontra no servidor.

Na aplicação móvel foi criada a entidade LocalSync com o tipo de sincronização e outros dois atributos: ServerDoneAt (2 da Figura 43) que indica a data e hora do servidor em que foi feita a última sincronização e ClientDoneAt que indica a data e hora do cliente em que foi feita a última sincronização (4 da Figura 43).

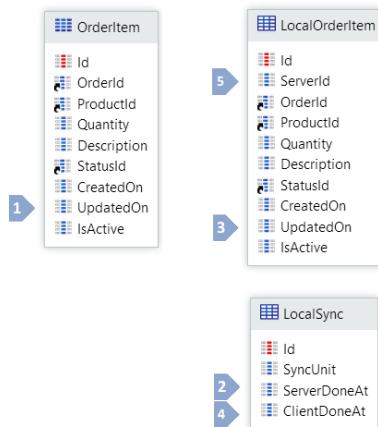


Figura 43 - Modelo de dados da sincronização *read/write*

A lógica de sincronização do cliente é apresentada na Figura 44.

1. Obtenção do pedido a decorrer no cliente (*GetLocalOrder query*);
2. Obtenção do pedido mas apenas se este foi alterado desde a última sincronização (*GetOrder2 query*);
3. Obtenção dos produtos do pedido que foram alteradores desde a última sincronização (*GetOrderItems query*);
4. Atualização dos dados do servidor (ação *ServerSync\_Order*);
5. Obtenção dos dados do servidor que foram alterados desde a última sincronização (ação *ServerSync\_Order*);
6. Se o cliente enviou o pedido, é necessário atualizar o mesmo com o identificador do servidor no atributo *ServerId* da entidade *Order* (*1º if*);
7. Se o cliente enviou produtos do pedido, é necessário atualizar com o identificador do servidor no atributo *ServerId* da entidade *OrderItem* (*1º for each*);
8. Se o servidor enviou produtos do pedido atualizados, é necessário atualizar os dados no cliente. Se existe um registo de *OrderItem* com o *ServerId* igual ao identificador enviado, o registo é atualizado. Se não, é criado um novo registo.
9. Alteração da data e hora da sincronização;
10. Atualização desta sincronização.

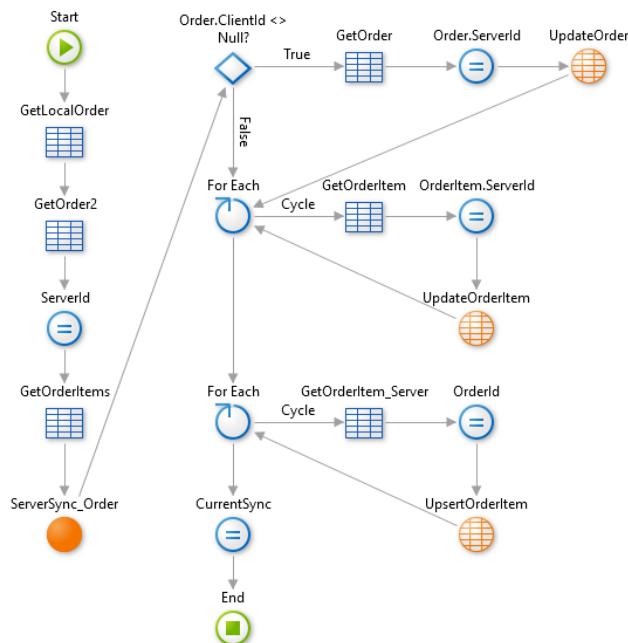


Figura 44 - Lógica de sincronização (*read/write*) do cliente

A lógica de sincronização do servidor é apresentada na Figura 45.

1. Verificação se o utilizador tem as permissões necessárias;
2. Cálculo da diferença de horas entre a hora do cliente e do servidor;
3. Log da sincronização;
4. Se o cliente enviou um pedido, atualizar os dados do mesmo pelo atributo ServerId;
5. Atualização dos produtos do pedido enviados pelo cliente;
6. Obtenção dos dados (produtos do pedido) atualizados desde a última sincronização;
7. Retorno da lista com os dados atualizados e data e hora da sincronização.

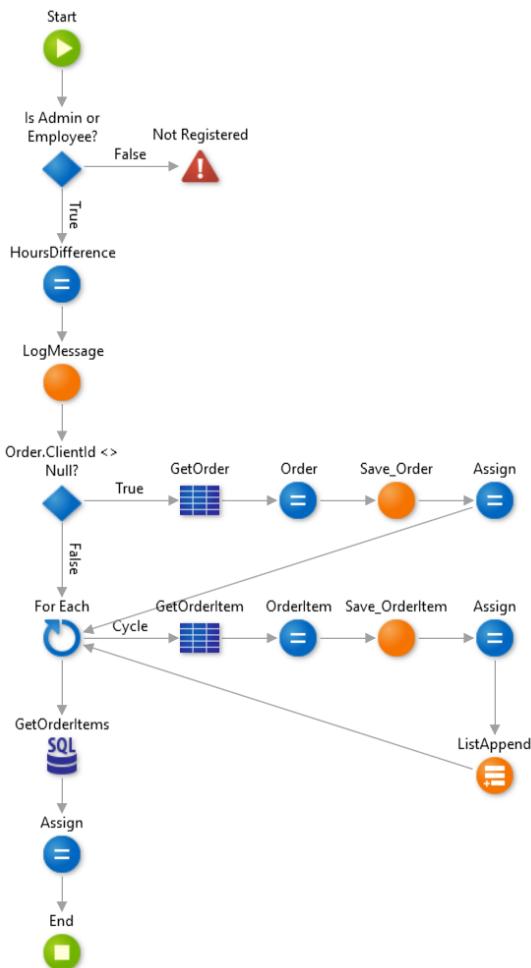


Figura 45 - Lógica de sincronização (read/write) do servidor

Após a criação da lógica de sincronização, é necessário decidir quando é que os dados devem ser sincronizados e quais as entidades a sincronizar em cada um desses momentos. Como tal, foram criadas as seguintes ações:

- DataSync\_1stLogin: utilizada no ecrã de login quando é a primeira sincronização. As seguintes entidades são atualizadas: OrderItemStatus, Category, Type, Table, Image, Product e MeasurmentType;
- DataSync\_StaticEntities: utilizada no ecrã de login quando já foram realizadas outras sincronizações. As seguintes entidades são atualizadas: OrderItemStatus e MeasurmentType.
- DataSync\_Table: utilizada no ecrã de listagem de mesas. As seguintes entidades são atualizadas: Table.
- DataSync\_Menu: utilizada no ecrã de menu. As seguintes entidades são atualizadas: Image, Product, Category e Type.
- DataSync\_Order: utilizada no ecrã de detalhe de pedido. As seguintes entidades são atualizadas: Order e OrderItem.

De modo a realizar a sincronização, a ação Trigger\_DataSync disponibilizada pela OutSystem é acionada. O parâmetro de entrada SyncUnit indica quais das sincronizações realizar. Esta ação vai realizar a ação Sync\_Data, a lógica é executada assíncronamente e está representada na Figura 46.

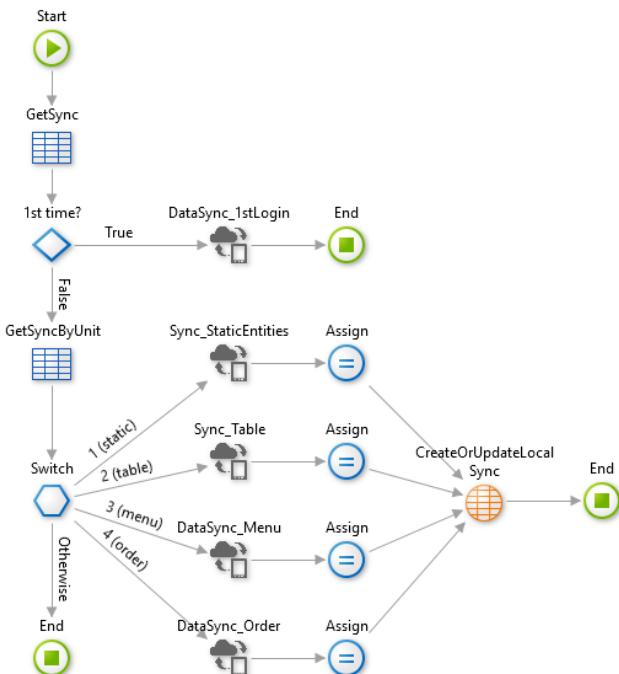


Figura 46 - lógica de sincronização

## 3.10. Faturação

Neste capítulo iremos abordar a *framework* de faturação escolhida para a nossa aplicação e todo o processo à volta da mesma. A Framework escolhida para a aplicação foi a *InvoiceXpress* porque é uma *framework* conhecida no mercado português e porque disponibiliza todos os serviços necessários que necessitamos.

### 3.10.1 *InvoiceXpress* API

O *InvoiceXpress* é um programa de faturação online, certificado pela Autoridade Tributária(AT) e Aduaneira de Portugal. Uma fatura eletrónica tem o mesmo valor legal do que uma fatura em papel, a diferença está na substituição da sua assinatura manual por uma assinatura digital e na apresentação do documento em ficheiro PDF.

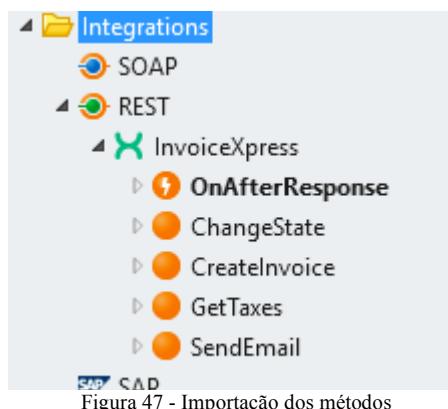
A principal vantagem do *InvoiceXpress* é permitir enviar faturas por correio eletrónico, sem ter de recorrer a serviços externos e evitando custos de correio. Outra vantagem é que é possível partilhar automaticamente os dados com a AT.

A *InvoiceXpress* oferece uma API bastante complexa e extensa, no qual alguns desses métodos foram incorporados na nossa aplicação. Os métodos implementados na nossa aplicação foram:

- Criar fatura;
- Alterar o estado da fatura;
- Enviar fatura para o email do cliente;
- Adicionar taxas de iva;
- Editar taxas de iva.

A API está disponibilizada em <https://invicexpress.com/api/overview> e o URL de acesso à API é o <https://isel-1.app.invoicexpress.com>

Foram importados os quatro métodos em OutSystems, ilustrados na Figura 47, através da integração de serviços REST disponibilizado pelo *Service Studio*. Os atributos e a estrutura de



dados de cada método vão ser detalhados posteriormente. Todos os métodos necessitam de uma *API\_KEY*, disponibilizada pela *InvoiceXpress*.

## Método de criação fatura

O método de criação de fatura ilustrado na Figura 48, permite criar uma fatura no estado *draft*, ou seja, uma consulta de mesa.

A chamada ao método é feita através do protocolo HTTP POST, enviando uma solicitação para o seguinte URL com os dados da fatura no corpo do pedido em XML:

[https://isel-1.app.invoicexpress.com/invoices.xml?api\\_key={ApiKey}](https://isel-1.app.invoicexpress.com/invoices.xml?api_key={ApiKey})

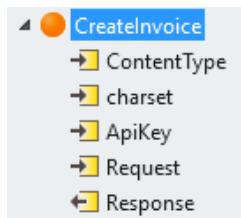


Figura 48 - Método de criação fatura em OutSystems

## Exemplo de um pedido:

XML enviado no corpo

```
<invoice>
  <date>7/6/2017</date>
  <due_date>7/6/2017</due_date>
  <client>
    <name>João</name>
    <fiscal_id>215190009</fiscal_id>
  </client>
  <items type="array">
    <item>
      <name>P4</name>
      <description>Cheese</description>
      <unit_price>2.50</unit_price>
      <quantity>1</quantity>
      <tax>
        <id>0</id>
        <name>23%</name>
        <value>23</value>
      </tax>
      <discount>0.0</discount>
      <subtotal>0.0</subtotal>
      <tax_amount>0.0</tax_amount>
      <discount_amount>0.0</discount_amount>
      <total>0.0</total>
    </item>
  </items>
</invoice>
```

XML recebido no corpo

```
<invoice>
  <id>12821058</id>
  <status>draft</status>
  <archived>false</archived>
  <type>Invoice</type>
  <sequence_number>draft</sequence_number>
  <date>12/06/2017</date>
  <due_date>12/06/2017</due_date>
  <permalink>www.ie.com/docs/1</permalink>
  <client><id>2937073</id><name>Joao</name></client>
  <currency>Euro</currency>
  <items type="array">
    <item>
      <name>P</name><description>Chicken</description>
      <unit_price>0.0</unit_price>
      <quantity>1.0</quantity>
      <tax>
        <id>13</id><name>23%</name><value>23.0</value>
      </tax>
      <discount>0.0</discount><subtotal>0.0</subtotal>
      <tax_amount>0.0</tax_amount>
      <discount_amount>0.0</discount_amount>
      <total>0.0</total>
    </item>
  </items>
  <sum>0.0</sum><discount>0.0</discount>
  <before_taxes>0.0</before_taxes>
  <taxes>0.0</taxes><total>0.0</total>
  <invoice_timeline type="array">
    <event>
      <type>create</type><user>35392</user>
      <date>12/06/2017</date><time>11:10</time>
    </event>
  </invoice_timeline>
</invoice>
```

Na aplicação foram criadas duas estruturas semelhantes às estruturas idênticas às estruturas apresentadas no exemplo da chamada ao método de criação de fatura. Inicialmente foi criada apenas uma estrutura com todos os atributos, mas a plataforma preenche automaticamente os atributos com o seu valor padrão. Como tal, o serviço retornava uma exceção dado que eram enviados mais atributos do que este estava à espera.

Foi usado o componente *XMLRecords* disponível na *Forge* que permite transformar uma estrutura OutSystems em XML e vice-versa.

Dado o facto de OutSystems não permitir haver duas estruturas com o mesmo nome, o componente é também responsável por alterar, no momento da transformação, as *tags invoiceIn* e *invoiceOut* na tag *invoice* ilustradas na Figura 50, passando uma lista com a transformação de nomes. Essa lista foi construída através de uma função em OutSystems que devolve uma lista da estrutura *XMLConfig*, ilustrado na Figura 49.

Os dados de resposta do serviço são guardados na tabela *Invoice*, no qual são usados posteriormente na chamada aos seguintes serviços. O serviço retorna o *id* da fatura no sistema interno do *InvoiceXpress* e o URL do documento.

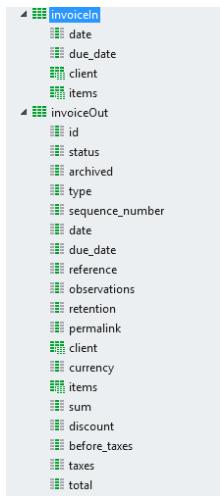


Figura 50 - Estrutura invoice

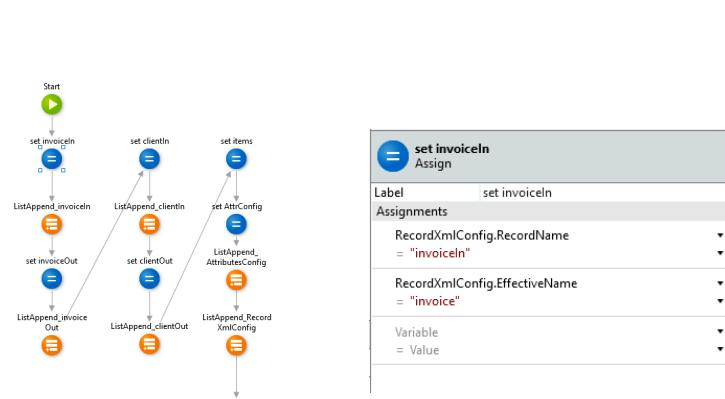


Figura 49 - Função de transformação de nomes e *assign* dos nomes

## Método de edição do estado da fatura

O método de edição do estado da fatura permite alterar o estado *draft* para o estado *final*, ou seja, será emitida uma fatura digital para o cliente.

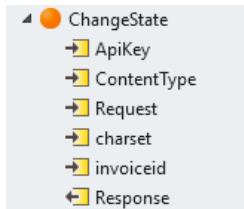


Figura 51 - Método de edição do estado da fatura

A chamada ao método é feita através do protocolo HTTP PUT, enviando uma solicitação para o seguinte URL com o novo estado da fatura no corpo do pedido em XML:

[https://isel-1.app.invoicexpress.com/invoices/{invoiceid}/change-state.xml?api\\_key={ApiKey}](https://isel-1.app.invoicexpress.com/invoices/{invoiceid}/change-state.xml?api_key={ApiKey})

### Exemplo de um pedido:

XML enviado no corpo

```
<invoice>
  <state>finalized</state>
</invoice>
```

Resposta

HTTP: 200 OK

Para este método foi criada uma única estrutura, uma vez que não tem uma resposta em XML.

Dado que o nome da estrutura em OutSystems é *invoiceState* e o serviço está à espera de receber a tag *invoice*, foi necessário passar novamente ao componente uma lista com a transformação de nomes.

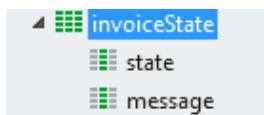


Figura 52 - Estrutura de alteração do estado da fatura

Quando a resposta é retornada com sucesso, é realizado um *update* à entidade *Invoice* com o estado final (valor da entidade *InvoiceState*).

## Método de envio da fatura

O método de envio da fatura para o email do cliente, ilustrado na Figura 53, tal como o nome indica, é responsável por enviar um email ao cliente com uma mensagem e um URL para fazer o download da fatura digital.

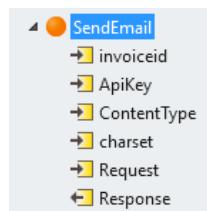
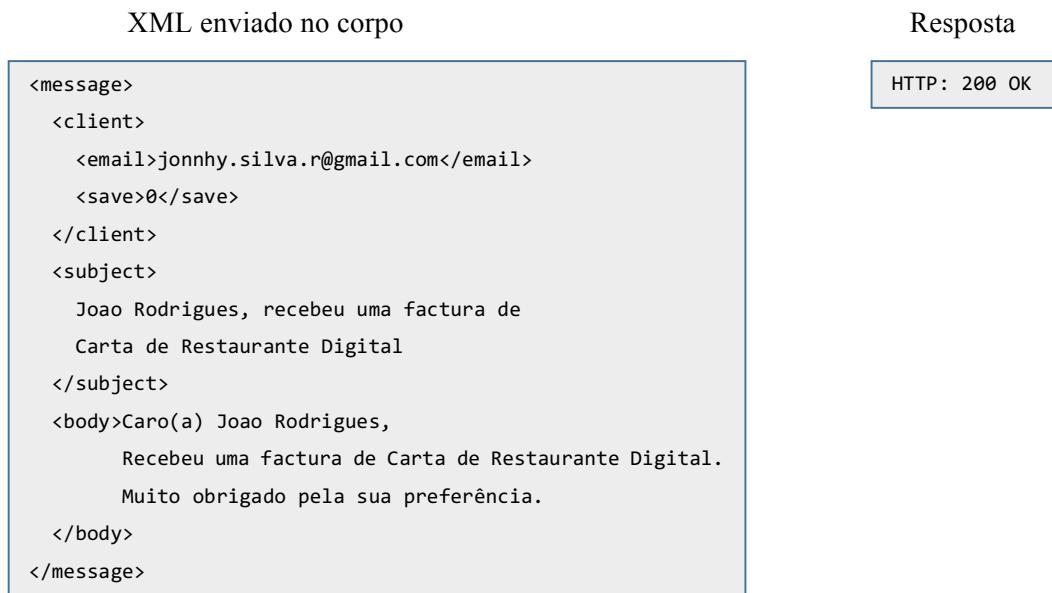


Figura 53 - Método de envio da fatura

A chamada ao método é feita através do protocolo HTTP PUT, enviando uma solicitação para o seguinte URL com a mensagem no corpo do pedido em XML:

[https://isel-1.app.invoicexpress.com/invoices/{invoiceid}/email-document.xml?api\\_key={ApiKey}](https://isel-1.app.invoicexpress.com/invoices/{invoiceid}/email-document.xml?api_key={ApiKey})

### Exemplo de um pedido:



Exemplo de um *email* ilustrado na Figura 54 enviado para o cliente:

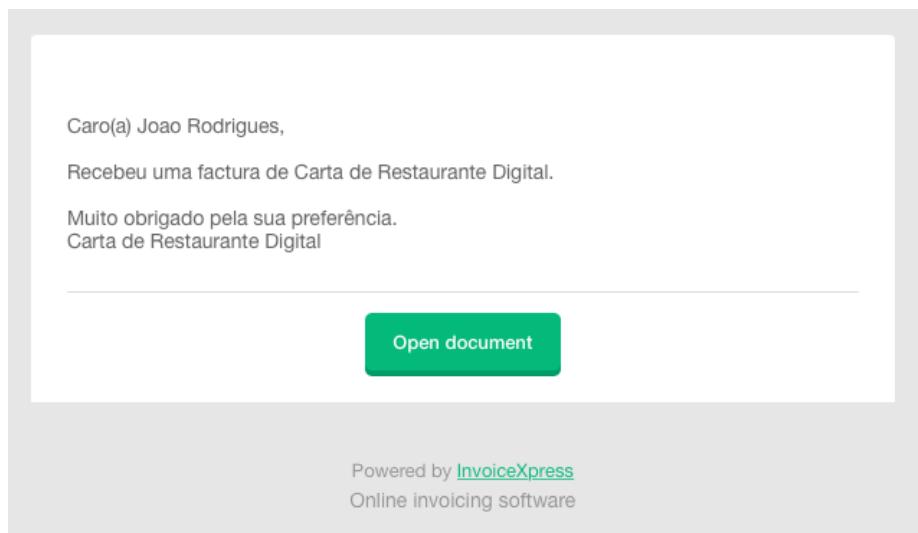


Figura 54 – Exemplo do email enviado

Para este método foi criada uma única estrutura ilustrada na Figura 55, uma vez que não tem uma resposta em XML. Dado que o nome da estrutura em OutSystems é *clientEmail* e o serviço está à espera de receber a tag *client*, foi necessário passar novamente ao componente uma lista com a transformação de nomes. Quando a resposta é retornada com sucesso, é realizado um *update* à tabela de *Order* a indicar que o processamento da fatura e o envio da mesma por email foram concluídos.

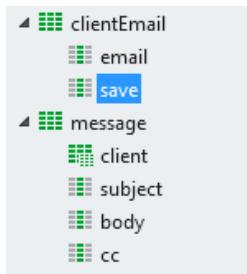


Figura 55 - Estrutura de envio de email

## Método de adicionar taxas de iva

O método de adicionar taxas de iva é responsável por adicionar no lado da Framework de faturação as taxas de iva configuradas no *back-office*.

A chamada ao método é feita através do protocolo HTTP POST, enviando uma solicitação para o seguinte URL:

[https://isel-1.app.invoicexpress.com/taxes.xml?api\\_key={ApiKey}](https://isel-1.app.invoicexpress.com/taxes.xml?api_key={ApiKey})

### Exemplo de um pedido:

XML enviado no corpo

```
<tax>
    <name>IVA23</name>
    <value>23.0</value>
    <region>PT</region>
    <default_tax>1</default_tax>
</tax>
```

Resposta em XML

```
<tax>
    <id>31540</id>
    <name>IVA23</name>
    <value>23.0</value>
    <region>PT</region>
    <default_tax>1</default_tax>
</tax>
```

A resposta do serviço retorna um novo elemento *id*, no qual é guardado na entidade *Taxes* associado à taxa submetida no serviço. Esse identificador vai ser usado posteriormente na chamada ao serviço de *update* de taxas.

Para este método foi apenas criado uma única estrutura com a informação das taxas. Neste método não houve a necessidade de alterar os nomes dos atributos.

## Método de modificar taxas de iva

O método de modificar taxas de iva é responsável por modificar no lado da Framework de faturação as taxas de iva configuradas no *back-office*.

A chamada ao método é feita através do protocolo HTTP PUT, enviando uma solicitação para o seguinte URL:

[https://isel-1.app.invoicexpress.com/taxes/{taxid}.xml?api\\_key={ApiKey}](https://isel-1.app.invoicexpress.com/taxes/{taxid}.xml?api_key={ApiKey})

### Exemplo de um pedido:

XML enviado no corpo

Resposta em XML

```
<tax>
    <id>31540</id>
    <name>IVA23</name>
    <value>23.0</value>
    <region>PT</region>
    <default_tax>1</default_tax>
</tax>
```

HTTP: 200 OK

Para este método foi usado a estrutura anteriormente criada na a criação das taxas.

### 3.10.2. Processos

De forma a que a aplicação tenha o máximo performance possível, foi criado um processo assíncrono em OutSystems que é responsável por chamar os métodos anteriormente mencionados.

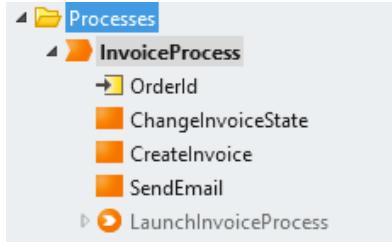


Figura 56 - Processo assíncrono da geração da fatura

O processo foi dividido em três partes (três *automatic activities*, ou seja, não necessita de intervenção do utilizador para executarem a função), nomeadamente na criação da fatura, na alteração do estado da fatura e no envio do email.

Para cada *automatic activity* é feito um *update* na tabela de Order com o estado do processo e é chamado a ação de encapsulamento da chamada ao serviço, ou seja, ação responsável por preencher a estrutura de dados necessária para enviar para o serviço, chamar o serviço, e tratar a sua resposta. As outras *Automatic Activities* são processadas uma de cada vez, esperando a conclusão da *Automatic Activity* anterior.

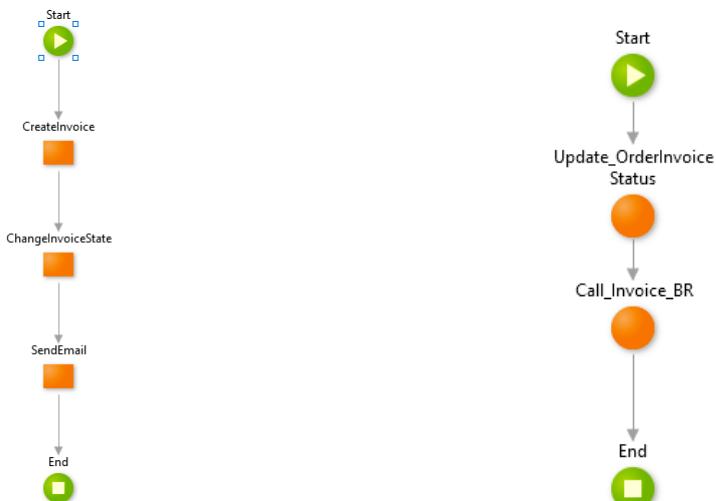


Figura 57 – Business process

Figura 58 - Automatic activity CreateInvoice

### 3.10.3. Temporizadores

No método de criar faturas mencionado anteriormente, é necessário associar taxas de iva a cada *item*, taxas essas que são pré-configuradas no *back-office*. Dado que existe a possibilidade de o serviço estar temporariamente indisponível e o administrador precisar de adicionar taxas de iva sem impossibilitar a criação ou os *updates* das taxas, foi criado um serviço para sincronizar todos os dias as taxas de iva existentes no *back-office* com a *framework*.

Deste modo, foi criado um *timer* em OutSystems ilustrado na Figura 61 que permite sincronizar as taxas de iva de ambos os sistemas. O *timer* é configurável e por *default* tem um *schedule* para sincronizar todos os dias durante a noite.

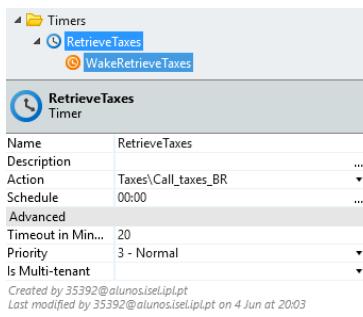


Figura 59 - Timer para sincronização de taxas de iva

As taxas de iva são atualizadas com o seu identificador na entidade *Tax* e usadas na *dropdown* de taxas no ecrã de configuração de produtos do *back-office*, ilustrado na Figura 62.



Figura 60 - Dropdown com a informação das taxas de iva



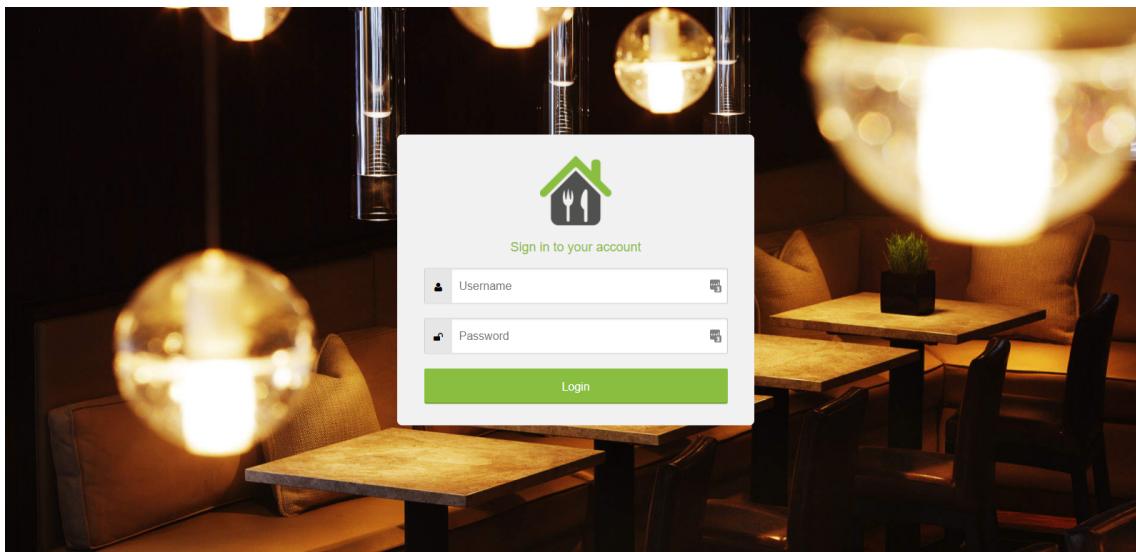
## 4. Avaliação Experimental

A avaliação da nossa solução é apresentada neste capítulo, onde é apresentado o resultado da solução. Na aplicação foram criados os seguintes utilizadores:

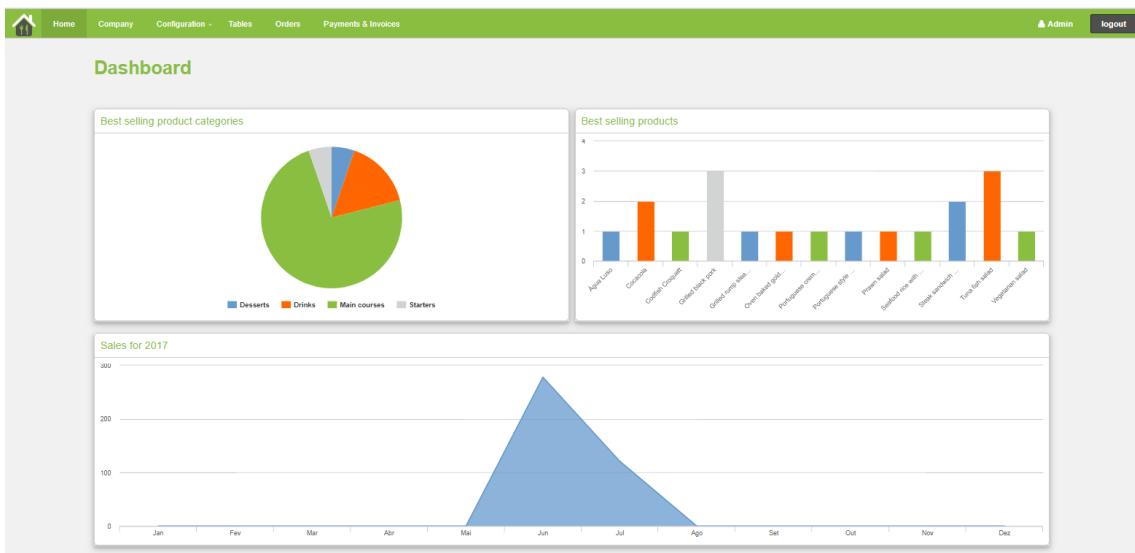
- Admin: utilizador com perfil de administrador.
  - Username: admin
  - Password: 123456
- Employee: utilizador com perfil de empregado.
  - Username: employee
  - Password: 123456
- Kitchen: utilizador com perfil de cozinha.
  - Username: kitchen
  - Password: 123456

### 4.1. Aplicação de *back-office*

1. Acesso à página de *login* através do link <https://cartarestaurantedigital.outsystemscloud.com/Main/>;



2. Autenticação com o utilizador Admin;



3. Acesso ao menu “Company”;

The company information form includes the following fields:

Name *	Marisqueira Gandarez	Postal Code *	1500-911
VAT Number *	215190009	City *	Lisboa
Street Name *	Rua António Alçada Baptista	Phone Number *	916141321

The logo section shows a placeholder for a company logo, with options to "Choose File..." or "Delete Image".

4. Acesso ao menu “Configuration” – “Tables”;

The table list includes the following data:

Name	Order	Is Active	Actions
Table one	1	✓	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Table two	2	✓	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Table three	3	✓	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>

3 records

5. Selecionar o botão “New Table”, preencher o formulário e carregar no botão “Save”;

**Create new Table**

Table Information

Name *	Table 4
Order *	4

**Cancel** **Save**

6. Verificar a mensagem de sucesso e a nova mesa na listagem;

**Table Search**

The table was created with success.

List of Tables				
Name	Order	Is Active	Actions	
Table one	1	✓	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Table two	2	✓	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Table three	3	✓	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Table 4	4	✓	<input type="checkbox"/>	<input checked="" type="checkbox"/>

4 records

**New Table**

7. Aceder ao menu “Configuration” – “Product Categories”;

**Product Category Search**

The category was created with success.

List of Product Categories				
Product Category	Order	Is Kitchen	Is Active	Actions
Starters	1		✓	<input type="checkbox"/> <input checked="" type="checkbox"/>
Main courses	2	✓	✓	<input type="checkbox"/> <input checked="" type="checkbox"/>
Desserts	3		✓	<input type="checkbox"/> <input checked="" type="checkbox"/>
Drinks	4		✓	<input type="checkbox"/> <input checked="" type="checkbox"/>

4 records

**New Category**

8. Selecionar o botão “New Category”, preencher o formulário e carregar no botão “Save”;

Create new Product Category

Product Category Information

Product Category \*

Order \*

Is Kitchen

Cancel Save

9. Verificar a mensagem de sucesso e a nova categoria na listagem;

Product Category Search

The category was created with success.

Search by product category  Search Clear

List of Product Categories

New Category

Product Category	Order	Is Kitchen	Is Active	Actions
Starters	1		✓	<input checked="" type="checkbox"/> <input type="checkbox"/>
Main courses	2	✓	✓	<input checked="" type="checkbox"/> <input type="checkbox"/>
Desserts	3		✓	<input checked="" type="checkbox"/> <input type="checkbox"/>
Drinks	4		✓	<input checked="" type="checkbox"/> <input type="checkbox"/>
Desserts 2	5		✓	<input checked="" type="checkbox"/> <input type="checkbox"/>

5 records

10. Aceder ao menu “Configuration” – “Product Types”;

Product Type Search

Search by product category or product type  Search Clear

List of Product Types

New Type

Product Category	Product Type	Order	Is Active	Actions
Main courses	Meat	1	✓	<input checked="" type="checkbox"/> <input type="checkbox"/>
Main courses	Fish	2	✓	<input checked="" type="checkbox"/> <input type="checkbox"/>
Main courses	Vegan	2	✓	<input checked="" type="checkbox"/> <input type="checkbox"/>
Main courses	Salads	3	✓	<input checked="" type="checkbox"/> <input type="checkbox"/>
Drinks	Waters	4	✓	<input checked="" type="checkbox"/> <input type="checkbox"/>
Drinks	Wine	5	✓	<input checked="" type="checkbox"/> <input type="checkbox"/>
Drinks	Non Alcholic	6	✓	<input checked="" type="checkbox"/> <input type="checkbox"/>

7 records

11. Selecionar o botão “New Type”, preencher o formulário e carregar no botão “Save”;

Product Type Information

Product Category \* Starters

Product Type \* Fish

Order \* 7

Cancel Save

12. Verificar a mensagem de sucesso e o novo tipo na listagem;

The type was created with success.

Product Category	Product Type	Order	Is Active	Actions
Main courses	Meat	1	✓	[Edit, Delete]
Main courses	Fish	2	✓	[Edit, Delete]
Main courses	Vegan	2	✓	[Edit, Delete]
Main courses	Salads	3	✓	[Edit, Delete]
Drinks	Waters	4	✓	[Edit, Delete]
Drinks	Wine	5	✓	[Edit, Delete]
Drinks	Non Alcholic	6	✓	[Edit, Delete]
Starters	Fish	7	✓	[Edit, Delete]

8 records

13. Aceder ao menu “Configuration” – “Products”;

Product Search

Name:  Category:

Price:  Type:

Tax:

List of Products

Name	Category	Type	Price	Tax	Measurement Type	Is Active?	Actions
Água Luso	Drinks	Waters	€1.50	13%	Doses	✓	[Edit, Delete]
Cocacola	Drinks	Waters	€1.50	13%	Doses	✓	[Edit, Delete]
Codfish "à brás"	Main courses	Fish	€9.50		Doses	✓	[Edit, Delete]
Codfish Croquett	Starters		€6.00	13%	Mass (in kg)	✓	[Edit, Delete]
Euromilhões			€10.00	6%		✓	[Edit, Delete]
Flan Custard	Desserts		€3.25	23%	Doses	✓	[Edit, Delete]
Grilled black pork	Main courses	Meat	€11.90	23%	Doses	✓	[Edit, Delete]
Grilled rump steak w/ fried chips	Main courses	Meat	€10.50	23%	Doses	✓	[Edit, Delete]

14. Selecionar o botão “New Product”, preencher o formulário e carregar no botão “Save”;

**Create new Product**

**Product Information**

Name*	Mouse	Measurement Type	Doses
Description			
Price*	12	Tax*	6%
Category*	Starters	Type	

**Edit image**

Choose your product image  
We only accept jpeg and png files

Choose File...

Cancel Save

15. Verificar a mensagem de sucesso e o novo produto na listagem;

**Product Search**

The product was created with success.

Name	Category	Type	Price	Tax	Measurement Type	Is Active?	Actions
Água Luso	Drinks	Waters	€1.50	13%	Doses	✓	
Cocacola	Drinks		€1.50	13%	Doses	✓	
Codfish "à brás"	Main courses	Fish	€9.50		Doses	✓	
Codfish Croquet	Starters		€5.00	13%	Mass (in kg)	✓	
Euromilhões			€10.00	6%		✓	
Flan Custard	Desserts		€3.25	23%	Doses	✓	
Grilled black pork	Main courses	Meat	€11.90	23%	Doses	✓	
Portuguese traditional rice	Main courses	Rice	€4.00	9%	Mass	✓	

**List of Products**

New Product

16. Aceder ao menu “Configuration” – “Taxes”;

**Tax Search**

Search by tax

Search Clear

**List of Taxes**

New Tax

Name	Tax	Is Active	Actions
6%	6.00	✓	
13%	13.00	✓	
23%	23.00	✓	

3 records

17. Selecionar uma taxa para editar, alterar o valor e selecionar o botão “Save”;

The screenshot shows a 'Edit Tax' form. At the top, there's a navigation bar with links for Home, Company, Configuration, Tables, Orders, Payments & Invoices, Admin, and logout. The main area is titled 'Edit Tax' and contains a 'Tax Information' section with two input fields: 'Name \*' (containing '6%') and 'Tax \*' (containing '6'). Below the form are 'Cancel' and 'Save' buttons.

18. Verificar a mensagem de sucesso e a edição da taxa na listagem;

The screenshot shows a 'List of Taxes' page. At the top, there's a navigation bar with links for Home, Company, Configuration, Tables, Orders, Payments & Invoices, Admin, and logout. A success message 'The table was updated with success.' is displayed above the search bar. The main area is titled 'List of Taxes' and contains a table with three records:

Name	Tax	Is Active	Actions
6%	6.00	✓	
13%	13.00	✓	
23%	23.00	✓	

3 records

19. Aceder ao menu “Configuration” – “Users”;

The screenshot shows a 'List of Users' page. At the top, there's a navigation bar with links for Home, Company, Configuration, Tables, Orders, Payments & Invoices, Admin, and logout. A search bar with the placeholder 'Name' is at the top. The main area is titled 'List of Users' and contains a table with five records:

Name	Username	Role	Actions
Admin	admin	Administrator	
Employee	employee	Employee	
Counter	counter	Employee	
Kitchen	kitchen	Kitchen	
Marcio	marcio	Administrator	

5 records

20. Selecionar o botão “New User”, preencher o formulário e carregar no botão “Save”;

User Information

Name \* José Gandarez

Username \* josepgandarez

Role \* Administrator

Password \* .....

Confirm Password \* .....

Cancel Save

21. Verificar a mensagem de sucesso e a edição da taxa na listagem;

The user was created with success.

Users Search

Name

Search Clear

List of Users

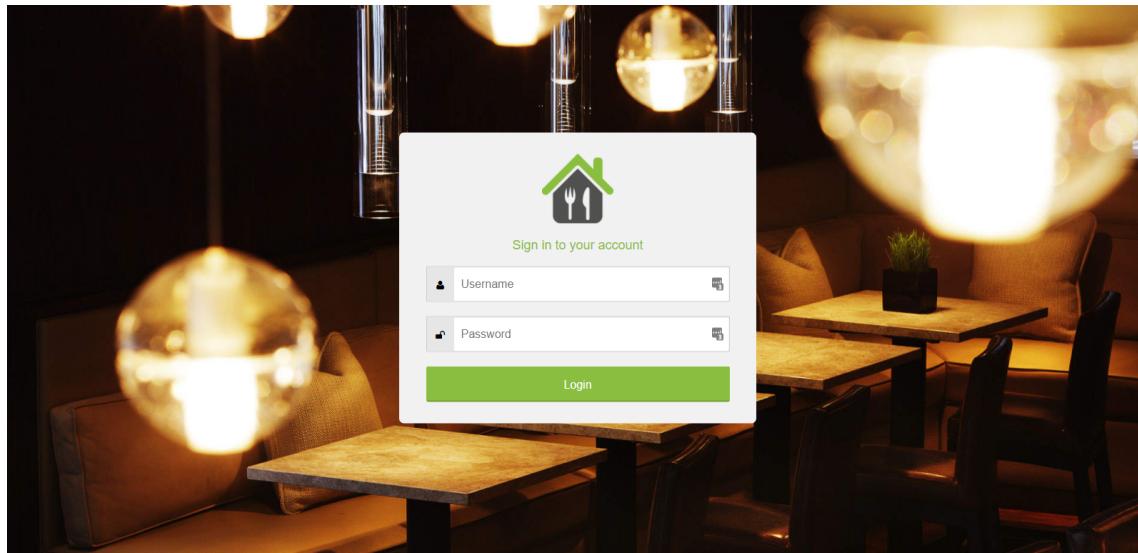
New User

Name	Username	Role	Actions
Admin	admin	Administrator	
Employee	employee	Employee	
Counter	counter	Employee	
Kitchen	kitchen	Kitchen	
Marcio	marcio	Administrator	
José Gandarez	josepgandarez	Administrator	

6 records

## 4.2. Aplicação de *front-office*

1. Acesso à página de *login* através do link <https://cartarestaurantedigital.outsystemscloud.com/Main/>;



2. Autenticação com o utilizador Employee;

A screenshot of the 'Tables' section of the application. It shows four tables labeled 'Table one', 'Table two', 'Table three', and 'Table four'. The top navigation bar includes links for 'Tables', 'Orders', 'Payments &amp; Invoices', and user authentication buttons ('Employee', 'logout').

3. Selecionar uma mesa;

A screenshot of the 'Order' management screen for 'Table one'. On the left, there's a grid of food items with 'Add' buttons. On the right, a summary table shows 'Product', 'Quantity', and 'Price' with a total of '0,00 €'. The top navigation bar includes links for 'Tables', 'Orders', 'Payments &amp; Invoices', and user authentication buttons ('Employee', 'logout').

4. Selecionar o botão “Add” de um produto e verificar a adição do mesmo à conta;

The screenshot shows the 'Tables' section of a restaurant management system. On the left, under 'Table one', there is a grid of food items with small images and 'Add' buttons. On the right, a detailed view of an order is shown with a header 'Order #24'. The order table has columns for Product, Quantity, and Price. One item, 'Água Luso', is listed with a quantity of 1 and a price of 1,50 €. A total of 1,50 € is displayed at the bottom.

Product	Quantity	Price
Água Luso	1	1,50 €

Total: 1,50 €

5. Selecionar o menu “Orders”;

The screenshot shows the 'Orders' section. It displays a list of orders for 'Table one' at 21:05. The first order is for 'Água Luso' with a quantity of 1. There is a checkbox next to the order.

Table one	21:05
1 Água Luso	<input type="checkbox"/>

6. Marcar o produto como realizado;

The screenshot shows the 'Orders' section again. The 'Água Luso' order from 'Table one' at 21:05 now has a checked checkbox next to it, indicating it has been marked as completed.

Table one	21:05
1 Água Luso	<input checked="" type="checkbox"/>

7. Selecionar o menu “Tables”;

The screenshot shows the 'Tables' section. It lists four tables: 'Table one' (status 21:05), 'Table two', 'Table three', and 'Table four'. 'Table one' is highlighted with a green background and a checkmark icon.

Tables
Table one 21:05
Table two
Table three
Table four

8. Selecionar a mesa com pedidos feitos;

The screenshot shows a software interface for managing a restaurant. On the left, there's a sidebar with icons for 'Tables', 'Orders', and 'Payments & Invoices'. The main area is titled 'Table one' and displays a grid of items. Each item has a thumbnail, its name, and a green 'Add' button. To the right of the grid is a summary for 'Order #24' showing a single item: 'Água Luso' at 1.50 €. At the bottom right of the screen is a small icon of a cash register.

Product	Quantity	Price
Água Luso	1	1,50 €

Total: 1,50 €

9. Selecionar a imagem da registador, de modo a efetuar o pagamento;

This screenshot shows the same software interface as the previous one, but with a modal dialog box overlaid. The dialog is titled 'Order #24 - payment' and contains fields for 'Payment Type' (radio buttons for 'Cash' and 'ATM'), 'Email' (text input with placeholder 'Type or double-click for list'), 'Name' (text input with placeholder 'Final consumer'), and 'VIT Number' (text input). Below these fields, it shows the total amount 'Total 1,50 €' and a large 'Pay' button. The background of the interface is dimmed to indicate the dialog is active.

10. Preencher o formulário;

Product	Quantity	Price
Água Luso	1	1,50 €

11. Selecionar o botão “Pay”;

The Invoice was sent with success.

12. Abrir a conta de email e verificar que a fatura foi recebida;

Caro(a) José Gandarez,  
Recebeu uma factura de Marisqueira Gandarez.  
Muito obrigado pela sua preferência.  
Marisqueira Gandarez

[Open document](#)

Powered by [InvoiceXpress](#)  
Online invoicing software

**Projecto e Seminario - Carta de Restaurante Digital**  
e-mail: a35392@alunos.isel.pt

Exmo.(s) Sr.(s)  
**José Gandarez**

**Original**

**Factura nº A/5**

Data	Contribuinte	Observações
2017/07/16	249635569	
Vencimento	W Ref.	

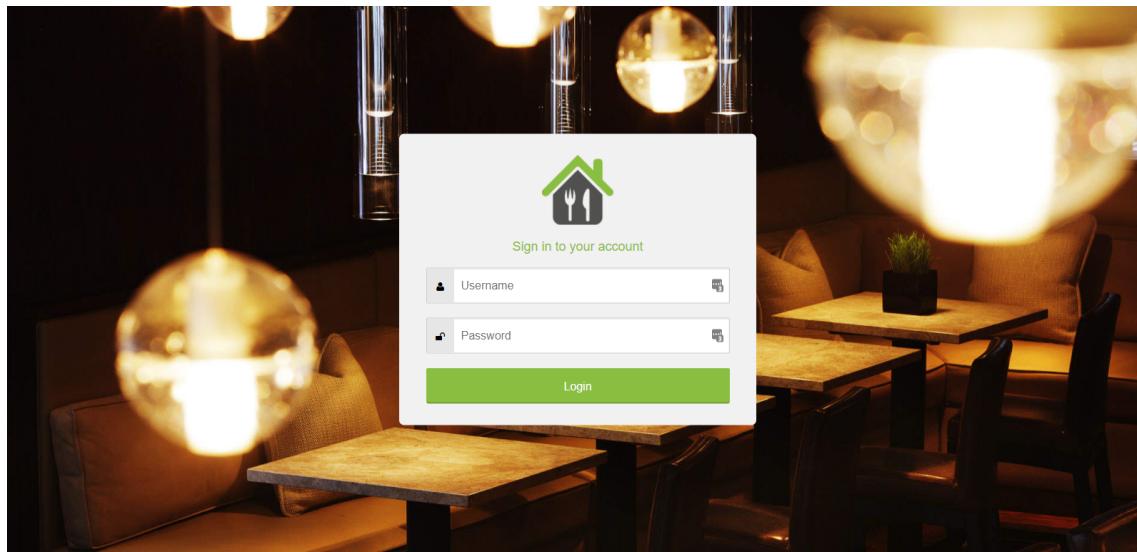
Código	Descrição	Preço Un.	Qty.	IVA %	Total Iliquido
P17	Água Luso	1,33 €	1,0	13,00	1,33 €

13. Aceder ao menu “Payment & Invoices”;

The screenshot shows a web-based application interface for managing payments and invoices. At the top, there is a navigation bar with links for Home, Company, Configuration, Tables, Orders, Payments & Invoices, Admin, and logout. Below the navigation bar is a search bar labeled "Search by name or description". Underneath the search bar are two dropdown menus: "Table" and "Payment Type", both currently set to "-". To the right of these dropdowns are "Search" and "Reset" buttons. The main content area displays a table with the following columns: Order #, Employee, Table, Check In, Check Out, Client, Order Status, Payment Type, Generate Invoice, Invoice Number, Send Email, and Actions. The table contains seven rows of data, each representing a different order entry.

Order #	Employee	Table	Check In	Check Out	Client	Order Status	Payment Type	Generate Invoice	Invoice Number	Send Email	Actions
25	Employee	Table one	16/07/2017 21:09:56	16/07/2017 21:10:16	José Gandarez	Completed	Money	✓	13220735	✓	
24	Employee	Table one	16/07/2017 21:05:54	16/07/2017 21:09:38	José Gandarez	Completed	Money	✓	13220729	✓	
23	Admin	Table one	12/07/2017 22:14:42	12/07/2017 22:15:53	João Rodrigues	Completed	Money	✓	13180932	✓	
22	Counter	Table one	10/07/2017 06:18:59	10/07/2017 06:42:39	João Rodrigues	Completed	Money	✓	13138445	✓	
17	Admin	Table two	27/06/2017 16:25:15	02/07/2017 21:47:19	João Rodrigues	Completed	Money	✓	13049273	✓	
6	Counter	Table three	17/06/2017 18:56:32	18/06/2017 15:11:35	João Rodrigues	Completed	Money	✓	12875947	...	

14. Selecionar o botão de “logout”;



15. Autenticação com o utilizador Kitchen;

The screenshot shows a dashboard titled "Kitchen". At the top, there is a navigation bar with links for Home, Kitchen, and logout. The main content area displays a table for "Table one" with the date "05.16". The table lists three items: "Codfish \"à brás\"", "Grilled black pork", and "Prawn salad", each with a small green checkmark icon next to it.

Table one		05.16
1	Codfish "à brás"	✓
1	Grilled black pork	✓
1	Prawn salad	✓

16. Marcar os pedidos como realizados;

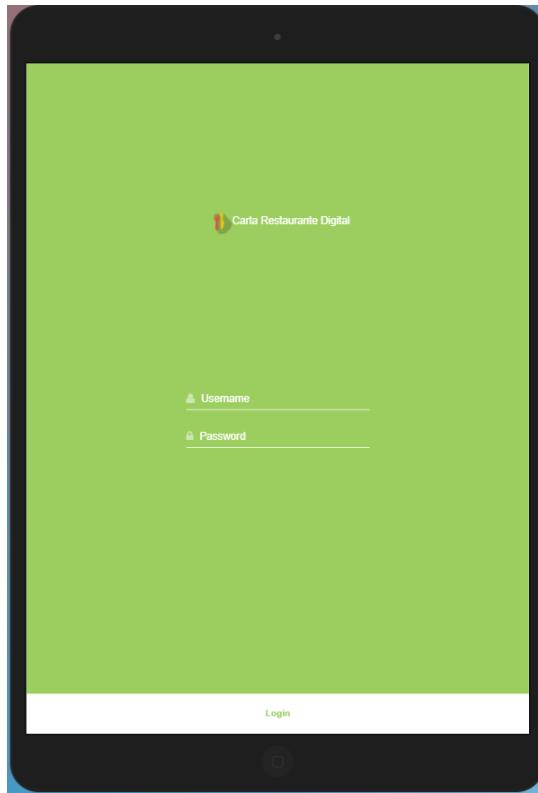
The screenshot shows a software interface titled "Kitchen". At the top, there's a green header bar with a house icon, the word "Kitchen", and a "logout" button. Below the header, the word "Kitchen" is displayed again in a larger font. A table titled "Table one" is shown, with a timestamp "05.16" and a green circular progress indicator. The table lists three items, each with a quantity of 4 and a checked status box:

Table one		05.16
4	Codfish "à brás"	<input checked="" type="checkbox"/>
4	Grilled-black-pork	<input checked="" type="checkbox"/>
4	Prawn-salad	<input checked="" type="checkbox"/>

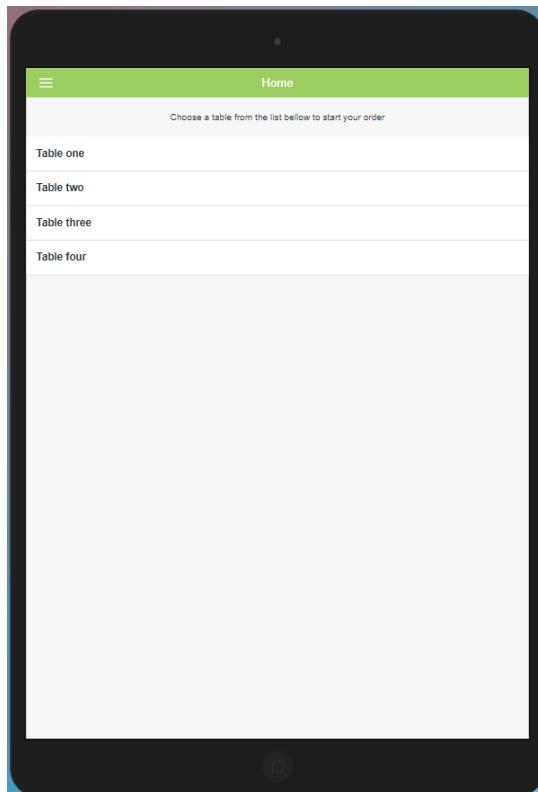
The screenshot shows the same "Kitchen" interface. The header and title are identical. However, the main content area is now empty, featuring a large gray placeholder box with a fork and knife icon and the text "There aren't orders to process."

#### **4.3. Aplicação móvel (*tablet*)**

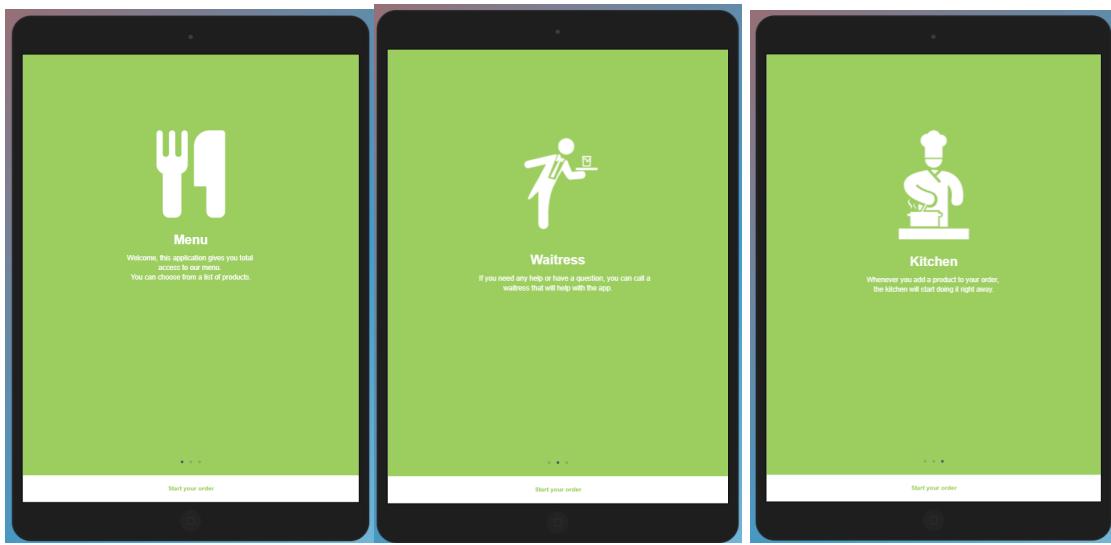
1. Acesso à aplicação através do tablet ([download](#));



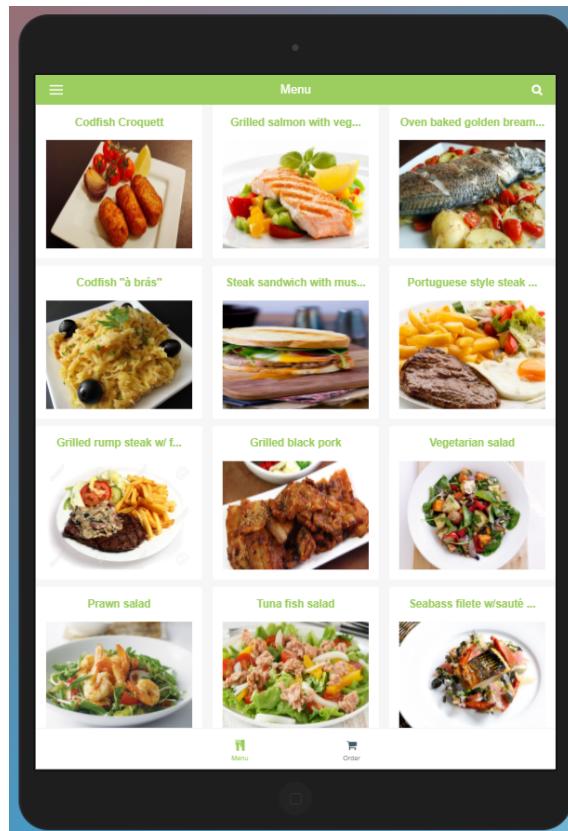
2. Autenticação com o utilizador Employee;



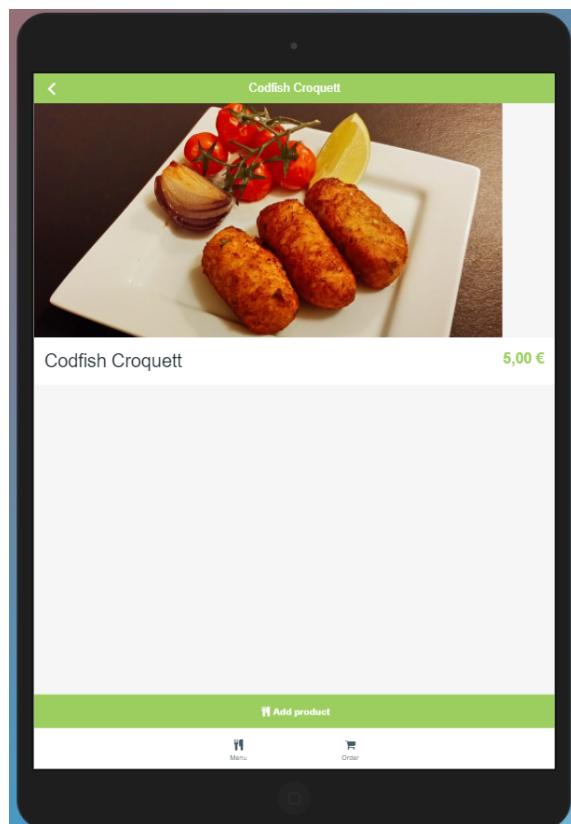
3. Selecionar uma mesa e navegar pelo ecrã de *on boarding*;



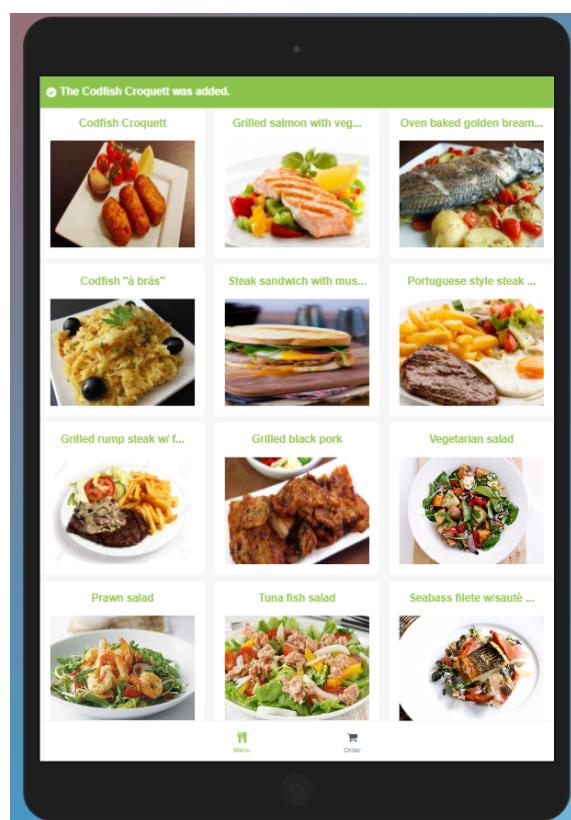
4. Iniciar o pedido;



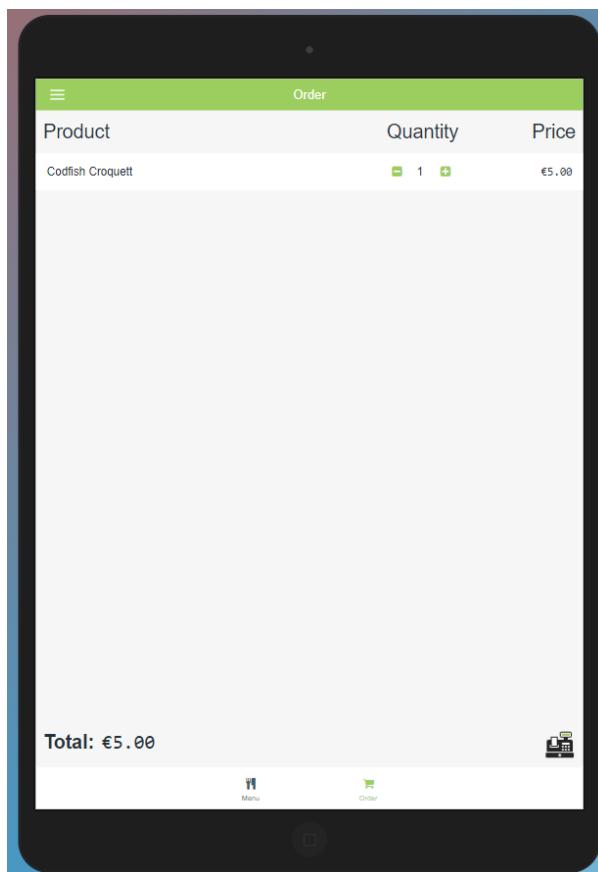
5. Selecionar um produto;



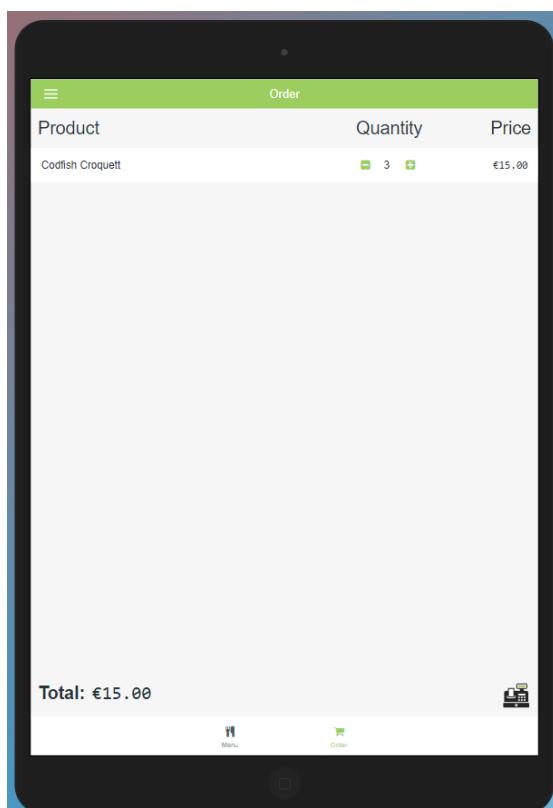
6. Adicionar produto ao pedido;



7. Aceder ao menu do pedido;



8. Adicionar nº de produtos;



## Conclusões

O projeto Carta de Restaurante Digital visa otimizar o processamento de pedidos em empresas de restauração, aumentando a sua capacidade de resposta no atendimento às mesas e assim reduzir o tempo de espera do cliente. Como tal, foram identificados dois grandes problemas: a demora do tempo de resposta do empregado perante o cliente e o tempo que leva a realizar os pedidos. Com o intuito de colmatar estas falhas, foram desenvolvidas duas aplicações através da plataforma OutSystems.

Uma aplicação *web* com foco nos empregados onde estes podem verificar pedidos, verificar o estado dos mesmos e configurar os dados do estabelecimento. Deste modo é possível minimizar erros por parte do empregado e manter um registo do que é necessário realizar de momento.

Uma aplicação *mobile* destinada aos clientes, sendo esta instalada em *tablets* do estabelecimento, disponibilizando funcionalidades para facilitar os pedidos do cliente. Assim, o segundo problema é mitigado, disponibilizando também ao cliente, uma forma digital onde este pode realizar os seus pedidos. Isto é, além do decréscimo do tempo de resposta, o estabelecimento ainda ganha valor por ter algo inovador. É de realçar também a funcionalidade de faturação. O cliente quando realiza o pagamento, tem a possibilidade de receber a fatura via *email* através da *framework InvoiceXpress*.

Todos os objetivos definidos foram alcançados, à exceção da implementação da *framework Easypay* que seria utilizada para a realização de pagamentos através da aplicação móvel, dado que surgiram alguns constrangimentos na obtenção de acessos à API para programadores que colocaram em causa a qualidade da implementação. Além disso, surgiram algumas dificuldades ao longo do projeto, de entre as quais se destacam a disponibilidade da *cloud* da OutSystems e a sincronização dos dados com a aplicação móvel. Apesar de a OutSystems disponibilizar aos seus programadores uma área pessoal para a criação e desenvolvimento de produtos, esta encontra-se muitas vezes sujeita a atualizações ou problemas de memória, atrasando os desenvolvimentos.

Como trabalho futuro e com o enfoque de continua melhoria, pretende-se implementar a *Framework Easypay* [4] para a realização dos pagamentos através da aplicação móvel, aumentar o sistema de *reporting* existente no ecrã de *Dashboard* do *back-office*, contribuindo para uma melhor gestão dos estabelecimentos de restauração, adicionar uma componente de sugestões dos clientes, de modo a melhorar a qualidade do serviço e para concluir, adicionar um sistema de controlo de *stock* dos produtos.

Para finalizar, tendo por base os conhecimentos técnicos obtidos durante o percurso académico e profissional, considera-se que os objetivos deste projeto foram alcançados. Foi desenvolvido um produto com aplicação à vida real que poderá ter um impacto significativo no sector da

restauração, tanto na melhoria da relação entre funcionários e clientes, como também na eficiência do serviço prestado.

## **Referências**

[1] McDonald's Order Machine. Consultado em Março 1, 2017 em:

<http://www.businessinsider.com/what-self-serve-kiosks-at-mcdonalds-mean-for-cashiers-2015-8>

[2] InvoiceXpress framework. Consultado em Março 1, 2017 em:

<https://invoicexpress.com>

[3] OutSystems Platform. Consultado em Março 1, 2017 em:

<https://www.outsystems.com/>

[4] Easypay framework. Consultado em Março 1, 2017 em:

<https://www.easypay.pt/pt>