

Aplicação de restauração

Projeto temático em Desenvolvimento de
Aplicações

1º semestre 2020/2021

Autores:

Sofia Rocha, nº 99991

Humberto Vaz, nº 97749

Miguel Marques, nº 97747

Leonardo Silva, nº 95381

Leonel, nº 85782

Aplicação de restauração

Projeto temático em Desenvolvimento de
Aplicações

1º semestre 2020/2021

Autores:

Sofia Rocha, nº 99991

Humberto Vaz, nº 97749

Miguel Marques, nº 97747

Leonardo Silva, nº 95381

Leonel, nº 85782

Orientador:

Rui Isidro

Índice

Índice	3
Nomenclatura	4
Introdução	5
Visão geral do sistema	5
Cliente do sistema	6
Objetivos do sistema	6
Planeamento	8
Modelo de requisitos	9
Requisitos funcionais	9
Restrições e requisitos não funcionais	13
Requisitos de interface e facilidade de uso	13
Requisitos de segurança e integridade dos dados	15
Requisitos de interface com sistema externos e ambientes de execução	17
Modelo de casos de utilização	19
Visão geral	19
Atores	19
Diagramas de casos de uso	20
Descrição dos casos de utilização	20
Modelo dinâmico	31
Configuração do sistema	31
Registo do pedido do cliente	32
Confecionar o pedido	33
Pagamento	34
Modelo estrutural	36

Nomenclatura

SGBD Sistema de Base de Dados

BD Base de Dados

SQL Structured Query Language

TPA Terminal de Pagamento Automático

IHC Interação Humano-Computador

API Application Programming Interface

1. Introdução

No âmbito da disciplina Projeto Temático de Desenvolvimento de Aplicações com as disciplinas de Engenharia de Software e Sistemas de Bases de Dado associadas foi-nos proposto o desenvolvimento de uma aplicação *stand-alone* utilizando uma linguagem orientada a objetos com acesso a uma base de dados.

No entanto o objetivo principal deste módulo é aprender a analisar os requisitos de um sistema de *software* e com isso construir um modelo conceptual do sistema incluindo o esquema relacional da base de dados e o modelo de desenho orientado a objetos da aplicação, implementar um SGBD cliente-servidor utilizando a linguagem SQL para a definição e manipulação de dados e por fim adotar uma estratégia de teste e especificar os casos de teste.

Com isto o grupo teve a opção de escolher o seu próprio tema que cumprisse o requisito acima. Assim escolheu-se uma aplicação que consiste em facilitar e aumentar a eficiência das tarefas na área da restauração. Esta app pretende ajudar o empregado da cozinha a não esquecer dos pedidos dos clientes do estabelecimento e também diminuir o tempo perdido na comunicação de cada pedido entre a cozinha e o empregado de mesa.

Por uma questão de todos os elementos conhecerem Java, ao contrário de outras linguagens, uma vez que foi lecionada no primeiro ano e agora no segundo ano, esta foi a linguagem escolhida para implementar a aplicação.

Assim, ao longo deste relatório é possível observar a evolução da solução proposta para resolver o problema mencionado. Para acompanhar foram feitos vários diagramas em UML e protótipos para que haja uma compreensão comum de toda a solução.

1.1. Visão geral do sistema

Como mencionado acima este sistema irá ser uma aplicação instalada em vários dispositivos para satisfazer as necessidades do cliente (ver secção [Cliente do sistema](#)). De uma forma geral esta aplicação consiste num programa em que serão registados no sistema os pedidos de cada consumidor de forma rápida e intuitiva. Este poderia adicionar ao pedido menus, criados pelo administrador que pode ser o chefe da empresa ou o gerente ou produtos individuais como um café ou uma sandes. Para além disso, cada produto poderá conter personalizações como extras, substituir algum ingrediente que o cliente não goste ou até mesmo retirar do produto.

Do lado da cozinha a aplicação receberia uma notificação cada vez que é feito um pedido novo para começar imediatamente a confeccioná-lo. Imediatamente a seguir começava um temporizador com um tempo definido pelo administrador para que evitasse esquecer do pedido por ter estado à espera de algo ou de alguém para concluir o pedido.

Assim que o pedido estivesse pronto todos os empregados de mesa receberiam um aviso de que há pedidos prontos.

Todas estas funcionalidades estão na mesma aplicação mas separadas por diversos logins (um para cada utilizador: o administrador, o empregado de cozinha e o empregado de mesa) para conseguir manter todas as funcionalidades separadas. E no caso de haver vários empregados de mesa há vários logins diferentes para que se possa distinguir que empregado fez o registo de determinado pedido. Também pode acontecer se houver mais de uma cozinha no mesmo estabelecimento.

Para além de tudo isto o administrador tem acesso a vários gráficos com o dinheiro ganho na empresa para conseguir ver quando vende mais determinado produto ou época.

1.2. Cliente do sistema

O cliente deste sistema são empresas da área da restauração que querem otimizar o seu estabelecimento, tal como um restaurante ou um café. Podendo ser usado para qualquer tamanho de empresas.

Nestes estabelecimentos acontece várias vezes que aparece uma enorme quantidade de clientes que aumentam a ineficácia do sistema tradicional. Neste sistema os empregados terão de decorar todos os pedidos e mesas associadas que se forem imensos, mais do que a memória de curto prazo consegue guardar, a tendência é esquecerem e terem de voltar a perguntar ao cliente. Isto aumenta a insatisfação do empregado e do cliente que terá de esperar mais tempo e o pedido poderá não ser o correto.

Para além disso toda a comunicação entre os empregados é feita “boca-a-boca” ou através de um papel e isso faz com que o empregado tenha de percorrer constantemente o estabelecimento para entregar toda a informação constantemente.

1.3. Objetivos do sistema

Assim, para resolver os problemas mencionados acima este sistema tem como objetivo facilitar o processo de registar o pedido do cliente e diminuir o tempo de espera para receber o mesmo. Com apenas alguns cliques toda a informação das ordens estará

registada no sistema permitindo que o empregado de mesa possa focar em atender outros clientes.

Enquanto isso, o empregado da cozinha começa imediatamente a preparar todos os pedidos que são registrados aumentando assim a autonomia de toda a equipa. E logo que esteja preparado todos os empregados de mesa recebem um aviso que podem ir buscar o pedido. Para ajudar o empregado de mesa ele pode ver na aplicação qual é a mesa em que é preciso ser entregue, qual o valor a pagar (para o estabelecimento cobrar se preferir que seja pago no ato de entrega, por exemplo) e quais os produtos solicitados e a sua quantidade para diminuir substancialmente o esquecimento que leva ao erro.

No entanto, para tudo isto acontecer o administrador tem de adicionar quantos produtos ele quiser que posteriormente sejam vendidos. Também pode adicionar menus e destacar produtos ou menus para serem mais facilmente vistos pelo empregado de mesa.

Para além disso, para saber como melhorar as suas estratégias de venda ou de marketing terá acesso a gráficos de quais indicam quando um produto é mais vendido, quando existem mais vendas e até mesmo quanto dinheiro ganhou num período de tempo escolhido pelo mesmo.

Em suma, a aplicação para além de aumentar a eficácia dos processos também ajudará a que o chefe consiga ter uma melhor gerência do estabelecimento.

2. Planeamento

Inicialmente o projeto pensado pelo grupo não seria *stand-alone* mas sim um *website*. Isto ocorreu por um erro de não verificar corretamente os requisitos do projeto, no entanto a ideia traria mais utilidades e comodidade para os utilizadores. Deste modo teria-se várias vantagens como poder atualizar o produto quando pudéssemos sem ter de ir a todos os dispositivos, poder aceder facilmente com qualquer telemóvel ou computador e também para ajudar na fase de desenvolvimento poderia-se utilizar *frameworks*.

Esta proposta conteria todas as funcionalidades já descritas anteriormente com a diferença que seria o cliente a interagir diretamente com a aplicação e que este poderia fazer encomendas. Assim o cliente escolheria os seus próprios produtos a consumir e poderia ver menus em destaque relacionados a uma época especial ou que o administrador pretendesse vender mais. Logo, toda a interface seria pensada para o cliente. Assim tanto no nosso telemóvel ou em dispositivos que seriam instalados no estabelecimento o cliente facilmente faria o seu pedido. No entanto, se o cliente não quisesse esperar para consumir poderia fazer uma encomenda em casa e levá-la no local.

Esta proposta para além de não ter sido mais discutida por reparar que teria de ser *stand-alone* levantava um pequeno problema. Muitas vezes em cafés ou restaurantes os clientes gostam de conversar com os empregados e isso aumenta as chances de o cliente voltar. Neste caso o website iria retirar esse facto importante tornando o estabelecimento menos sociável pois haveria menos interações entre cliente e empregado.

Assim sendo depois de toda a ideia ter sido mudada foi possível avançar na primeira fase do projeto, podendo assim definir os requisitos do projeto, como está referido na seguinte tabela, tanto funcionais como não funcionais. Esta primeira fase tem como objetivo fazer o planeamento do projeto, identificar e caracterizar as funcionalidades pretendidas para o sistema e por fim identificar e caracterizar os utilizadores do sistema. Na fase seguinte descreve-se todos os casos de utilizações e os diagramas associados e produzir o modelo de dados persistente. Por isso foram definidas várias tarefas como as representadas abaixo mas cumprir com as fases ditas.

Tabela 1. Tarefas

Tarefas
Escolher tema do projeto
Definir requisitos funcionais e não funcionais
Criar diagramas de casos de uso e atividade
Criar diagramas de sequência e classes

Definir que linguagem de programação utilizar
Fazer planeamento do projeto no MS Project

Com a penúltima tarefa concluída, o planeamento foi estruturado da seguinte forma. No entanto este planeamento contém algumas falhas em relação aos testes uma vez que ainda não foi lecionada nem escolhida qual a estratégia a adotar. Para além disso este planeamento não é o final, podendo por isso ser alterado. No entanto pode-se verificar, nas seguintes figuras, que as tarefas de planeamento estão quase todas concluídas exceto o modelo de dados persistente.

Assim após concluir toda a primeira fase e a segunda fase começa o momento de implementação. Esta implementação terá uma comunicação indireta com a base de dados uma vez que todos os dados serão validados a partir de uma API para que não seja escrito dados incorretos na base de dados. No entanto, inicialmente será usado uma base de dados local criada na aplicação Access da Microsoft para conseguir entender como funciona esta comunicação. As funcionalidades implementadas serão transferidas para a BDs da UA.

No fim serão criados um relatório, um manual do utilizador e um manual de instalação para que ajude o cliente a entender melhor como funciona o *software* e como usufruí-los sem a ajuda dos desenvolvedores tendo assim uma autonomia total.

3. Modelo de requisitos

Depois de o tema ter sido escolhido e as suas funcionalidades principais e utilizadores definidos faz-se o levantamento de requisitos para que seja aprimorada a estrutura do *software*. Assim sendo, começou-se por definir os requisitos funcionais que são mostrados já a seguir e logo a seguir todos os requisitos não funcionais e relacioná-los aos requisitos funcionais.

3.1. Requisitos funcionais

Assim, os requisitos funcionais permitem responder às necessidades descritas acima. Para além disso, cada requisito funcional tem uma prioridade associada uma vez que nem todas as funcionalidades serão cruciais para responder ao problema citado. Prioritariamente os requisitos funcionais relacionados com o registo do pedido são os mais prioritários seguido de funcionalidades associadas a notificações e personalizações

acabando em ver gráfico com o dinheiro ganho pela empresa e também temporizadores para cada produto.

Tabela 2. Requisitos funcionais

Ref ^a	Requisitos funcionais	Prioridade(1/3)
RF.1	Início de sessão para administrador, funcionário da cozinha/balcão e empregado de mesa	+++
RF.2	O pedido pode ter personalizações ou um ou mais menus	+++
RF.3	Cada pedido tem duas opções de consumo:consumir no local ou levar.	+++
RF.4	O empregado quando faz o pagamento tem a opção de escolher entre numerário e multibanco.	++
RF.5	Após finalizar o pedido, o empregado de mesa terá de confirmar o pedido.	+++
RF.6	Qualquer um dos funcionários pode alterar o pedido que ainda não foi pago.	++
RF.7	O administrador pode cadastrar a equipa.	+++
RF.8	Há sempre uma notificação quando há um pedido novo e quando o pedido está pronto.	+
RF.9	O administrador pode remover e adicionar produtos e consequentemente remover, adicionar e também alterar preços existentes.	+++
RF.10	O pedido após ter sido confirmado é vinculado a um número diferente de todos os outros pedidos e por ordem crescente.	+++
RF.11	Os pedidos na página de pedidos para serem confeccionados são organizados por número crescente de pedidos.	++
RF.12	Todos os produtos têm um nome, uma descrição, uma categoria e preço associados	+++

RF.13	O administrador no final do dia vê o dinheiro ganho num período de tempo escolhido pelo mesmo	+
RF.14	Todos os empregados de mesa têm acesso a todos os pedidos já confeccionados.	+++
RF.15	Ao fazer o pagamento o funcionário tem de colocar o dinheiro recebido pelo cliente para a aplicação poder fazer o troco.	+
RF.16	O administrador pode ver um gráfico de barras com toda a faturação do mês.	+
RF.17	O administrador pode remover vários produto	++
RF.18	O administrador só pode criar menus com produtos existentes.	+++
RF.19	Os utilizadores após terminarem a sessão não podem aceder às funcionalidades da aplicação.	+++
RF.20	Nenhum dos utilizadores após iniciar sessão pode aceder a outras funcionalidades a não ser as definidas para o mesmo.	+++
RF.21	O administrador pode configurar quantas mesas existem na sua empresa.	+++
RF.22	Sempre que o funcionário entrar na página dos produtos terá de conter todos os produtos que estão registados no sistema.	+++
RF.23	Sempre que a aplicação é fechada repentinamente terminará a sessão, o mesmo pode recuperar ou descartar todas as configurações feitas anteriormente após iniciar a sessão novamente.	++
RF.24	O administrador pode configurar um temporizador de notificação para confeccionar cada produto.	+
RF.25	Sempre que um pedido exceder o temporizador para confeccionar o empregado da cozinha receberá uma notificação.	+
RF.26	Sempre que o utilizador quiser fechar a aplicação aparecerá uma mensagem de confirmação e nenhum dado é guardado.	++
RF.27	A aplicação quando aberta pela primeira vez pelo administrador obriga o	+++

	utilizador a fazer as configurações básicas (colocar pelo menos um produto, o número de mesas, a cadastrar a equipa e o tempo limite para cada pedido).	
RF.28	Se o administrador não quiser fazer imediatamente as configurações básicas pode terminar a sessão e continuar ou recomeçar noutro dia.	++
RF.29	Se nenhuma configuração básica for feita, todas as funcionalidades do empregado de mesa e empregado da cozinha/balcão estarão bloqueadas até que as mesmas sejam feitas e aparecerá uma mensagem de aviso a dizer que o administrador precisa de fazer as configurações básicas.	+++
RF.30	Se o pedido não for para levar o empregado tem de escolher uma mesa obrigatoriamente.	+++
RF.31	Se houver um pedido com vários produtos diferentes o temporizador associado ao pedido será igual ao produto que tiver maior temporizador.	+
RF.32	Não dá para fazer novamente um pedido associado a uma mesa que já tenho um que ainda não foi pago.	+++
RF.33	O empregado de mesa tem de confirmar o pagamento.	+++
RF.34	O administrador inicialmente entrará no sistema com um username e uma password default que será obrigatoriamente mudada.	+++
RF.35	O administrador tem acesso a todos os usernames e passwords e poderá criar mais dados.	+++
RF.37	O empregado de mesa pode registar as gorjetas por cartão bancário ou em numerário.	+
RF.37	O empregado pode ver qual é a lotação de cada mesa antes de seleccionar.	+

3.2. Restrições e requisitos não funcionais

Para além dos requisitos funcionais foram definidos vários requisitos não funcionais tais como:

- requisitos de interface e facilidade de uso;
- requisitos de segurança, privacidade e integridade dos dados;
- requisitos de interface com sistemas externos e ambientes de execução.

Todos eles serão descritos de forma mais detalhada a seguir. Estes tipos de requisitos demonstram a qualidade das funcionalidades que irão ser implementadas.

3.2.1. Requisitos de interface e facilidade de uso

Neste tipo de aplicação o quão intuitivo é e a facilidade em navegar nela é crucial uma vez que o tempo entre ouvir o cliente e registar o pedido tem de ser o menor possível. Assim será pensada para que somente com alguns cliques e sem grandes falhas possa o utilizador aprender a utilizá-lo e atender os vários consumidores. Para ajudar ainda mais, todas as interfaces criadas têm de obedecer a alguns princípios de IHC, referidos na seguinte tabela.

Logo a interface não pode ser um entrave para quem estiver a usar pois aumentará a insatisfação no seu local de trabalho levando ao erro e à angústia.

Tabela 3. Princípios de IHC utilizados no projeto

Princípios de IHC
Reduzir a carga de memória
Esforçar-se pela consistência
Prevenção de erros / Reverter a ação
Naturalidade / Simplicidade
Lembrar os utilizadores e refrescar a sua memória

Com base nestes princípios foram descritos vários requisitos não funcionais que estão na seguinte tabela. Alguns destes não estão relacionados com os requisitos funcionais uma vez que não estão diretamente conectados com funcionalidades do utilizador mas sim com os princípios mencionados acima.

Tabela 4. Requisitos de interface e facilidade de uso

Ref^a	Requisitos de interface e facilidade de uso	Req. funcional relacionado

RInt.1	Todas as interfaces têm de ter um botão para voltar para o menu inicial, exceto o menu inicial e a interface de iniciar sessão.	
RInt.2	Todas as interfaces têm de ter um botão para voltar para a interface anterior, exceto o menu inicial.	
RInt.3	O utilizador só pode terminar a sessão manualmente no menu inicial, através de um botão.	
RInt.4	Todas as interfaces serão escritas em português.	
RInt.5	Sempre que um pedido para confeccionar exceder o tempo definido aparecerá uma notificação a avisar o empregado de cozinha.	<u>RF.24</u> e <u>RF.25</u>
RInt.6	Sempre que o administrador iniciar sessão pela primeira vez aparecerá uma única interface para obrigar o utilizador a fazer as configurações básicas.	<u>RF.27</u>
RInt.7	Se o administrador não fizer as configurações básicas e uns dos funcionários quiser utilizar a aplicação aparecerá impedido por um aviso e a tela bloqueada.	<u>RF.29</u>
RInt.8	Todas as interfaces têm que ser o mais intuitivo e o mais rápido de utilizar para melhorar a performance de cada empregado.	
RInt.9	Ao escolher a funcionalidade de fazer o pedido a interface aparece com vários números (das mesas) e a opção de levar para casa.	<u>RF.3</u>
RInt.10	Na página de pedidos para confeccionar os pedidos estão divididos em duas categorias (uma para levar e outra para consumir no local).	<u>RF.3</u>
RInt.11	Sempre que um pedido for finalizado aparecerá uma interface com todos os dados do pedido (preço, produtos selecionados e a opção de consumo)	<u>RF.5</u>

3.3. Requisitos de segurança e integridade dos dados

Nesta secção foram definidos vários requisitos de segurança, ver [tabela 4](#) que são muito importantes para que a aplicação não deixe escapar nenhum dado relativo à empresa ou aos empregados. Além disso, também é importante manter a base de dados num estado de integridade e se possível na sua 3FN.

Para que estes dois requisitos sejam cumpridos foi decidido criar uma API que fará o controle da passagem de pedidos para aceder à base de dados. Assim sendo a API irá permitir ou não que esses sejam realizados. Para além disso, irá verificar se cada dado enviado para escrever ou alterar a BDs é válido ou não.

Tabela 5. Requisitos de segurança, privacidade e integridade dos dados

Refª	Requisito de segurança, privacidade e integridade dos dados	Rel. funcionais relacionados
RSeg.1	Se houver dois pedidos a serem feitos ao mesmo tempo, não pode haver sobreposição de dados.	
RSeg.2	Os dados têm de estar inscritos na base de dados, sendo a encriptação feita pelo SGB.	
RSeg.3	Para cada administrador e para cada tipo de funcionário existe um username e password diferente.	
RSeg.4	Para cada nome de utilizador só existe uma palavra-passe válida.	
RSeg.5	Não pode conseguir entrar no sistema com um nome de utilizador e palavra-passe associado que não esteja na base de dados.	
RSeg.6	Todas as relações têm de respeitar a 3FN.	
RSeg.7	Nenhum dos funcionários pode conseguir alterar os produtos, os seus preços nem ter acesso ao total de dinheiro ganho pela empresa em qualquer período de tempo.	
RSeg.8	Nenhum menu pode ser guardado se só tiver um único produto	RF.18

	associado.	
RSeg.9	Nenhum produto pode ser guardado na BD se não tiver nome, preço, um número, data (dia e mês) e uma categoria associada.	<u>RF.12</u>
RSeg.10	Sempre que o funcionário entrar na página dos produtos o SGBD envia um pedido para obter todos os produtos.	<u>RF.22</u>
RSeg.11	Nenhum menu pode ser criado a partir de produtos que não existem.	<u>RF.18</u>
RSeg.12	Não pode haver dois menus ou categorias com o mesmo nome.	
RSeg.13	Se o utilizador não concluir as alterações que pretendia fazer e fechar a aplicação a sua sessão irá ser terminada, os dados serão guardados num ficheiro temporário na BDS.	
RSEg.14	Sempre que for feita alguma alteração todos os dados serão guardados num ficheiro temporário.	
RSeg.15	Todos os ficheiros temporários serão encriptados e guardados no Disco.	
RSeg.16	Se não houver nenhuma configuração básica na base de dados nenhum pedido pode ser adicionado.	<u>RF.29</u>
RSeg.17	Se não houver nenhum pedido feito, o SGBD impede qualquer pedido para ver os pedidos.	<u>RF.29</u>
RSeg.18	O SGBD é que gera o número de pedidos para cada pedido.	<u>RF.10</u>
RSeg.19	Após fazer o pedido o SGBD associará a data do pedido que será igual à data do sistema operativo.	
RSeg.20	Se o pedido não for para levar terá obrigatoriamente uma mesa associada.	<u>RF.33</u>
RSeg.21	Sempre que for enviado algum dado para a base de dados o ficheiro temporário será removido do sistema.	

RSeg.2 2	Cada pedido tem de ter uma categoria associada (para levar ou para consumir no local).	<u>RF.3</u>
RSeg.2 3	Não pode ser registado o valor do pagamento do cliente menor que o valor do pedido.	
RSeg.2 4	Inicialmente a base de dados tem um username e uma password default para o administrador que será substituída por outra na primeira utilização.	
RSeg.2 5	O administrador pode alterar e/ou criar nomes de utilizadores e palavras-passes.	

3.4. Requisitos de interface com sistema externos e ambientes de execução

A aplicação por ambiente da disciplina irá ser programa numa linguagem orientada a objetos e neste caso foi escolhido java. Consequentemente, por ser multi-plataforma, não é necessário definir que sistema operativos pode a aplicação ser executada. No entanto é crucial ter instalado o java no computador para que possa executar o *software*.

Tabela 6. Requisitos de interface com sistemas externos e ambientes de execução

Ref ^a	Requisitos de interface com sistemas externos e ambientes de execução
RSA.1	O sistema tem de ter acesso ao armazenamento do sistema operativo.
RSA.2	O sistema operativo tem de ser desktop.
RSA.3	O sistema operativo tem de ter instalado a versão 11 ou mais recente do java.
RSA.4	A base de dados utilizada é estga-dev.clientes.ua.pt

4. Modelo de casos de utilização

Depois de definir todo o projeto serão feitos diagramas de casos de uso para entender que funcionalidades têm cada utilizador. Cada caso de uso terá um caso de utilização que descreve os atores envolvidos, pré-condições e os requisitos funcionais. E como o tempo é limitado cada caso de utilização tem uma prioridade associada.

4.1. Visão geral

Numa visão geral existem casos de utilização cruciais para que todo o sistema funcione, como por exemplo o registar o pedido ou registar o pagamento. No entanto, cada caso de uso principal tem vários casos de uso associados que têm uma prioridade elevada ou não. Por isso que todos os requisitos funcionais têm uma prioridade associada para que seja melhor compreendido o que se irá implementar em primeiro.

Entretanto foram definidos quais eram os atores do sistema e as funcionalidades associadas como se pode ver a seguir.

4.2. Atores

Neste projeto existem 4 atores: o chefe da empresa, o funcionário de cozinha/balcão, o empregado de mesa, o próximo sistema que interage diretamente com ele mesmo. No entanto, apesar de o cliente do estabelecimento não interagir diretamente ele é uma peça fundamental para que todo o sistema funcione. Uma vez que o sistema inicia e acaba consoante as decisões do cliente.

Tabela 7. Descrição dos atores do sistema

Ator	Descrição
Administrador	Pode ser o chefe da empresa ou mesmo o gerente que irá configurar o sistema de acordo com o estabelecimento e gerir os produtos do sistema. Para além disso tem acesso a todo o dinheiro obtido pela empresa.
Funcionário de cozinha/balcão	Irá confeccionar todos os pedidos registados consoante o que o consumidor pretende.
Empregado de mesa	Irá atender todos os clientes do estabelecimento e registar os seus pedidos no sistema. Também irá entregar cada pedido ao respectivo

	consumidor. Será o único que irá interagir diretamente com ele.
O sistema	O sistema comunicará entre as interfaces do empregado de mesa e o funcionário de cozinha/balcão avisando quando há pedidos novos e quando os mesmos estão prontos para serem entregues.

4.3. Diagramas de casos de uso

Os diagramas de caso de uso estão relacionados a cada ator e a suas funcionalidades na aplicação. Para cada ator foi feito um diagrama como se pode ver a seguir, que representa um determinado papel que pode empregado de cozinha, empregado de mesa ou administrador (ver [tabela 7](#)). No entanto, sem um contexto estes diagramas são inúteis para a compreensão do produto, por isso são descritos vários casos de utilização que explicam melhor cada caso de uso e as sequências alternativas.

4.4. Descrição dos casos de utilização

Neste primeiro diagrama representa todas as funcionalidades do empregado de cozinha que começa por iniciar sessão para que tenha acesso à aplicação. O caso de uso de “iniciar sessão” é exatamente igual para o empregado de mesa no sentido que para usufruir terá da aplicação terá de inserir o seu nome de utilizador e palavra-passe. Por isso somente irá ser representado pela seguinte tabela.

Tabela 8. Descrição do caso de utilização para iniciar sessão por parte do empregado de mesa e do empregado de cozinha

Nome: Iniciar sessão
Atores: Empregado de mesa ou empregado de cozinha
Prioridade(1/3): +++
Finalidade: Ter acesso às funcionalidades do sistema
Requisitos funcionais: Início de sessão para funcionário da cozinha/balcão e empregado de mesa
Pré-condições: Ter o nome de utilizador e palavra-passe na base de dados

Sumário: O utilizador insere o nome de utilizador e a palavra-passe na aplicação e terá acesso ao menu inicial.

O diagrama de casos de uso a seguir representa tudo o que o empregado de mesa consegue fazer dentro da aplicação após iniciar a sessão. Este caso de uso como referido acima está na [tabela 8](#) por razões já mencionadas. Para além disso este diagrama tem dois grandes casos de uso que são registar pedido e regista o pagamento que são descritos mais minuciosamente nas tabelas seguintes. O primeiro faz-se o registo no sistema do que o cliente quer consumir para que o empregado de cozinha comece a confeccionar (ver [tabela 9](#) e o diagrama de atividades da [figura 2](#)) sem ter de estar à espera do empregado de mesa. O segundo regista o valor dado pelo cliente uma vez que este pode dar gorjeta e assim fica tudo no sistema.

Em suma, o empregado de mesa tem extrema importância uma vez que será este que começará e acabará todo o ciclo de satisfazer o cliente.

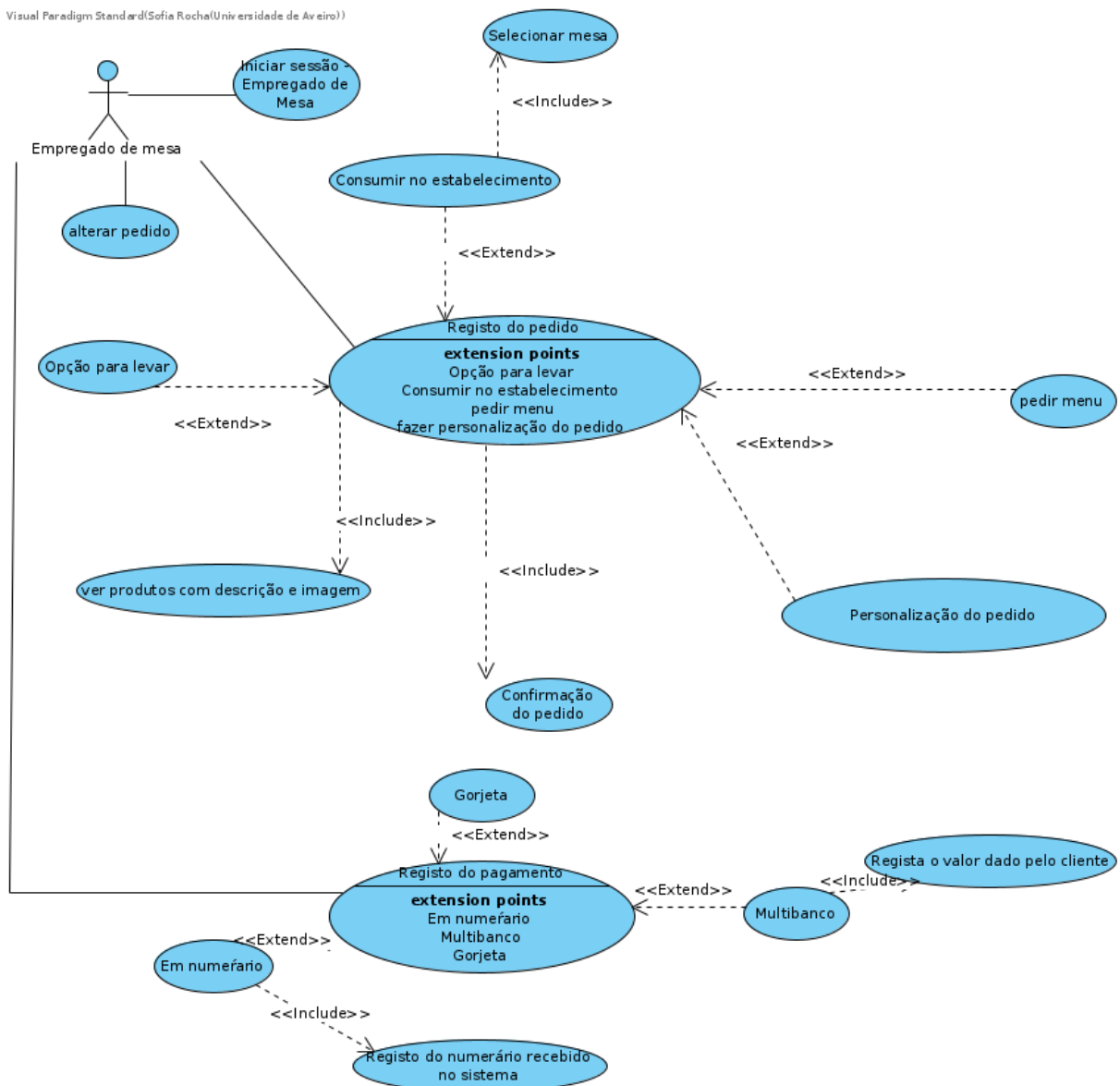


Figura 2. Diagrama de casos de uso do empregado de mesa

Tabela 9. Descrição do caso de uso do registo do pedido por parte do empregado de mesa

Nome: Registo do pedido
Atores: Empregado de mesa
Prioridade(1/3): +++
Finalidade: Registrar o pedido do cliente
Requisitos funcionais: RF.2 , RF.3 , RF.5 , RF.12
Pré-condições: Ter iniciado sessão na aplicação

Sumário: Assim que o cliente tiver escolhido uma mesa, o empregado de mesa seleciona-a na aplicação para que esta esteja associada ao pedido. Após isso seleciona todos os produtos ou menus da categoria correspondente e confirma o pedido para que este seja registado na base de dados e enviado para a cozinha. No entanto, o pedido pode ter personalizações como extras ou ter sido retirado algum ingrediente de algum produto específico se possível. Para além disso o cliente pode escolher para levar e consequentemente não é escolhido nenhuma mesa na aplicação mas sim a opção de *take-away*.

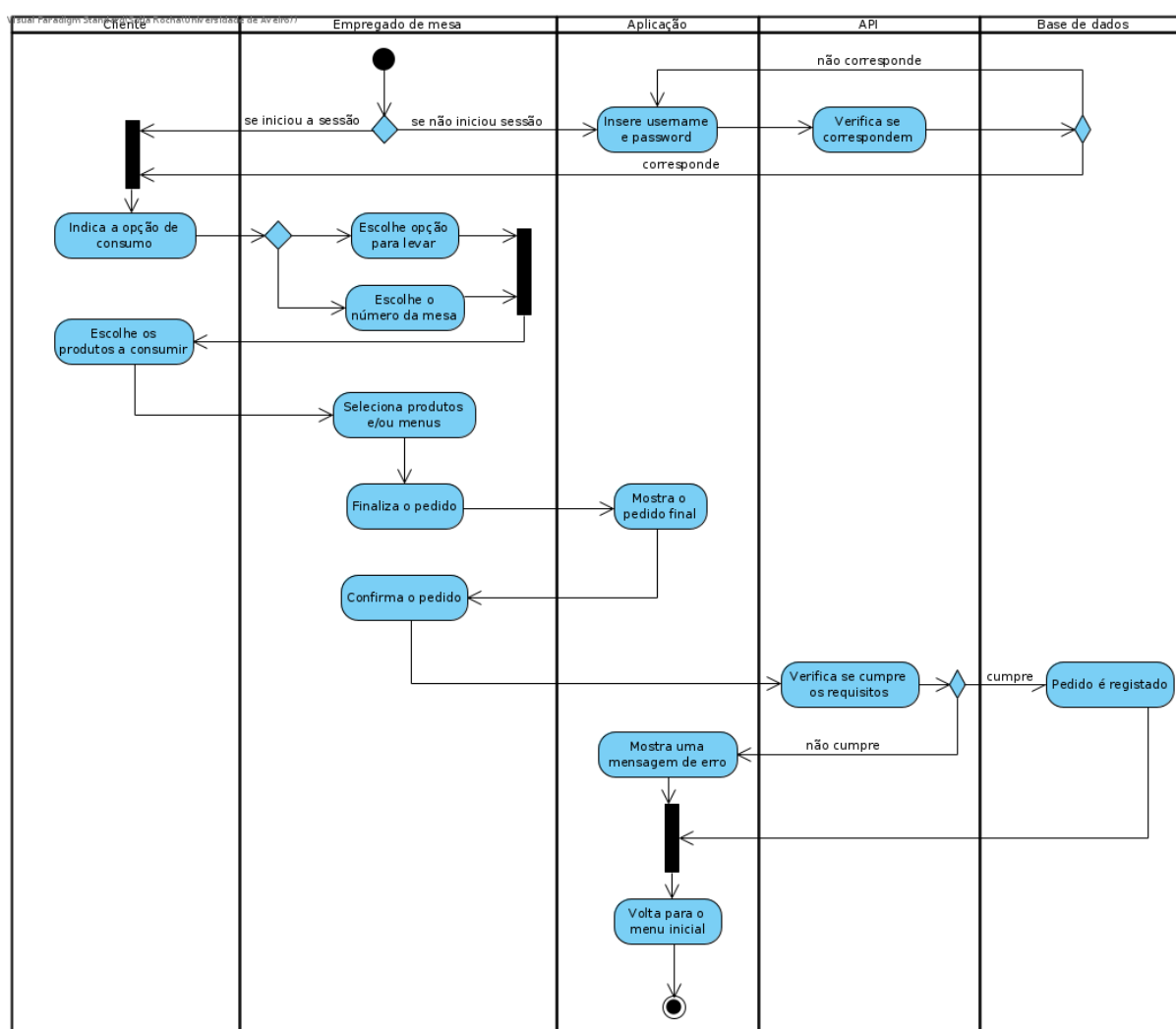


Figura 3. Diagrama de atividades do registo do pedido por parte do empregado de mesa

Tabela 10. Descrição do caso de uso de entregar o pedido por parte do empregado de mesa

Nome: Entregar pedido

Atores: Empregado de mesa
Prioridade(1/3): +++
Finalidade: Entregar pedido ao cliente
Requisitos funcionais: <u>RF.14</u>
Pré-condições: Ter recebido uma notificação de pedido pronto
Sumário: Após o empregado de cozinha acabar de confeccionar o pedido este será entregue ao cliente.

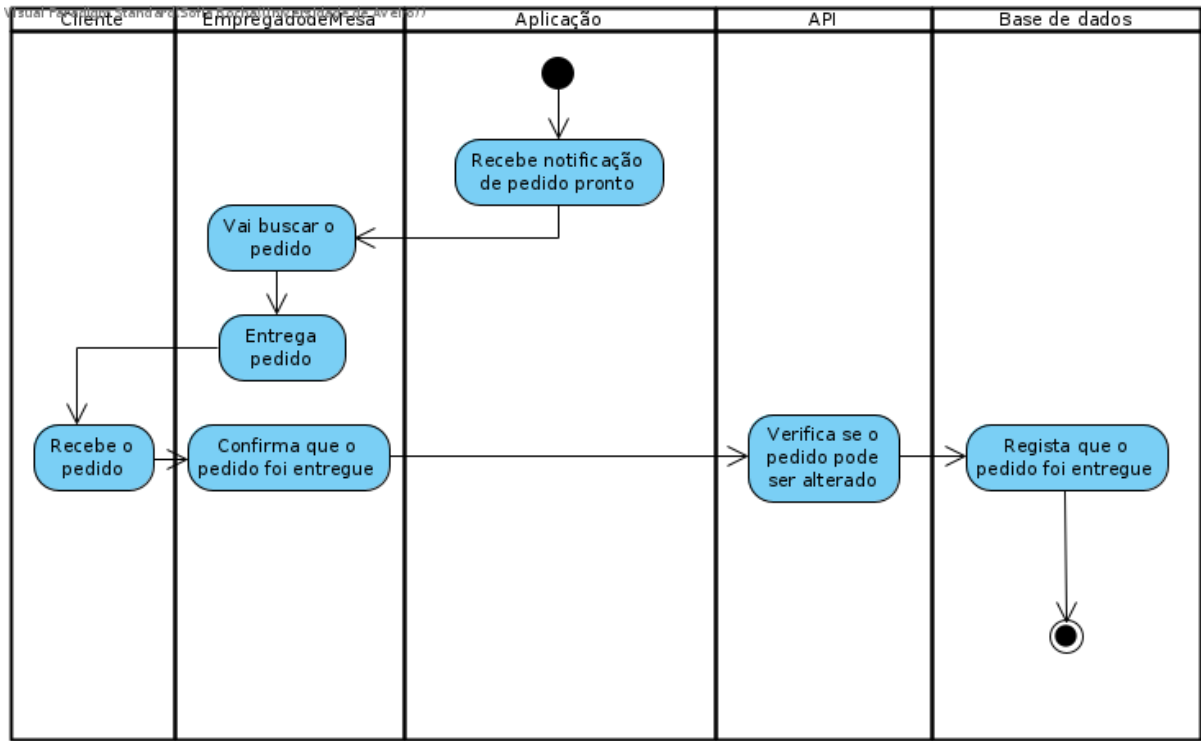


Figura 4. Diagrama de atividades de entrega do pedido por parte do empregado de mesa

Tabela 11. Descrição do caso de uso de registo do pagamento por parte do empregado de mesa

Nome: Registo do pagamento
Atores: Empregado de cozinha
Prioridade(1/3): +++

Finalidade: Registrar todo o valor ganho por cada pedido e se houver as gorjetas.

Requisitos funcionais: RF.37

Pré-condições: Ter sido registado um pedido no sistema e ter sido entregue ao cliente.

Sumário: Quando for efetuado o registo do pagamento o empregado de mesa escolhe a mesa em que foi feito o registo e seleciona entre pagamento com cartão bancário ou em numerário. Se for com cartão bancário a validação do pagamento é realizada pelo TPA no entanto o valor é registado no sistema. Se for com numerário o empregado de mesa coloca o valor dado pelo cliente no sistema para que seja feito o troco e/o recebido as gorjetas.

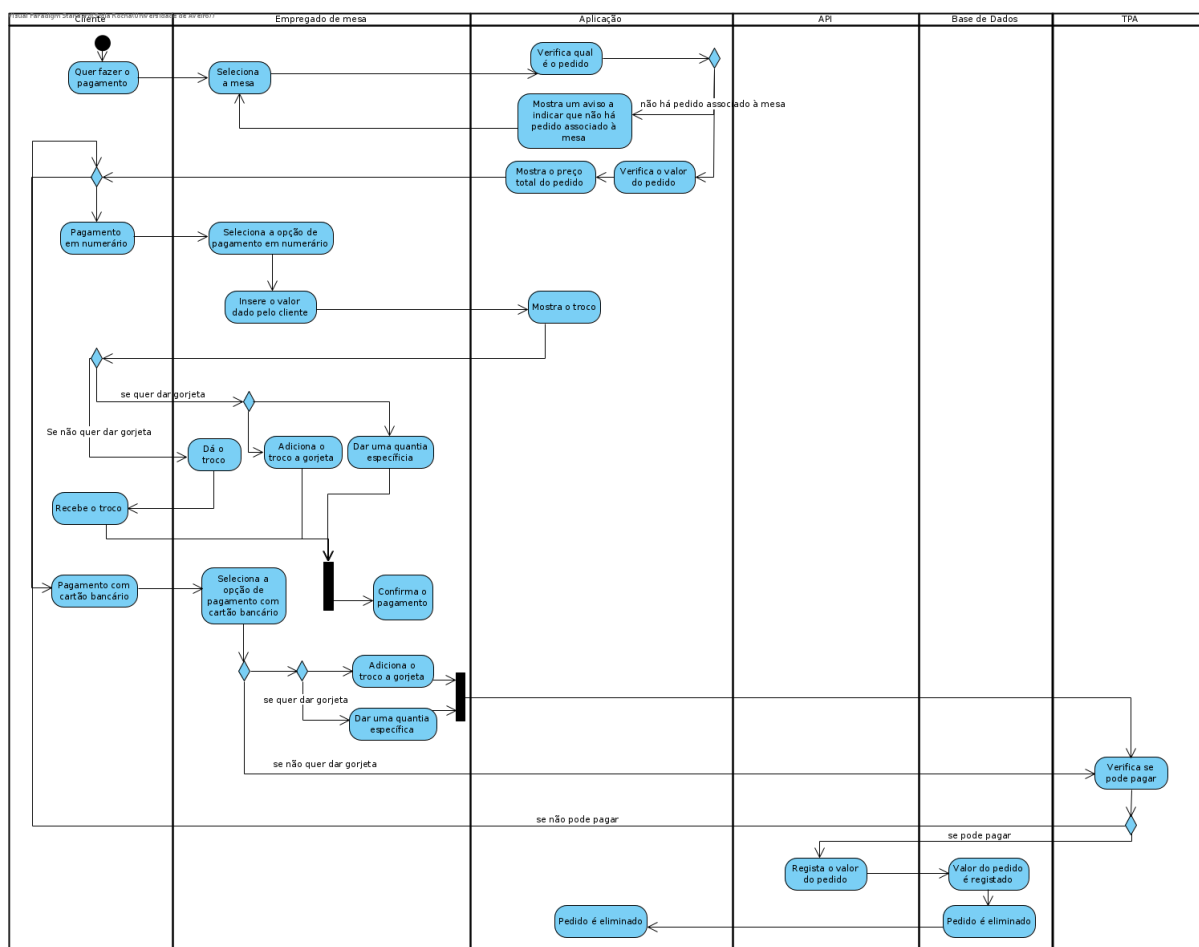


Figura 5. Diagrama de atividade do registo do pagamento por parte do empregado de mesa

Após a descrição de todos os casos de utilização dedicados ao empregado de mesa, a seguir serão dos do empregado de cozinha que estão na tabela 10 que preparará todos os

pedidos e avisar o sistema quando este está pronto a ser entregue. Para complementar esta descrição na figura 9 está um diagrama de atividades que representa todo o processo.

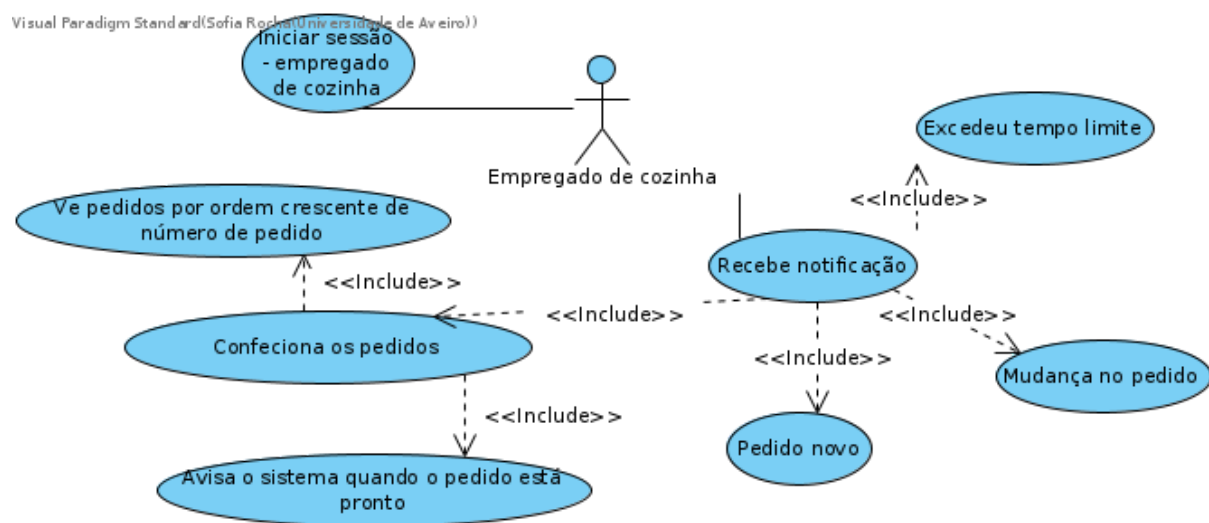


Figura 6. Diagrama do caso de utilização de receber notificações do empregado de cozinha

Tabela 13. Descrição do caso de utilização de receber notificações do empregado de cozinha

Nome: Recebe notificação
Atores: Empregado de cozinha
Prioridade(1/3): +++
Finalidade: Avisar o empregado de cozinha que há pedidos novos para confeccionar ou quando houver alguma alteração no pedido. Também avisar quando exceder o tempo limite proposto para o pedido. No fim o empregado terá de avisar o sistema que o pedido já está pronto.
Requisitos funcionais: RF.8 , RF.11 , RF.25 , RF.36
Pré-condições: Ter sido registado um pedido no sistema.
Sumário: O empregado de cozinha após receber uma notificação de um pedido novo começará imediatamente a confeccionar o pedido. Enquanto isso um temporizador

começará a contar de ordem decrescente até zero. Se o temporizador chegar a zero a aplicação mostra um aviso de que o pedido está atrasado. No entanto se isso não acontecer antes de o empregado avisar o sistema que o pedido está pronto a ser entregue então o temporizador é eliminado e nenhum aviso aparece. No entanto se entretanto o pedido for mudado o empregado de mesa recebe uma notificação de que há alterações num determinado pedido e a alteração aparece em destaque. Todos os pedidos podem ser vistos por ordem crescente por número de pedido.

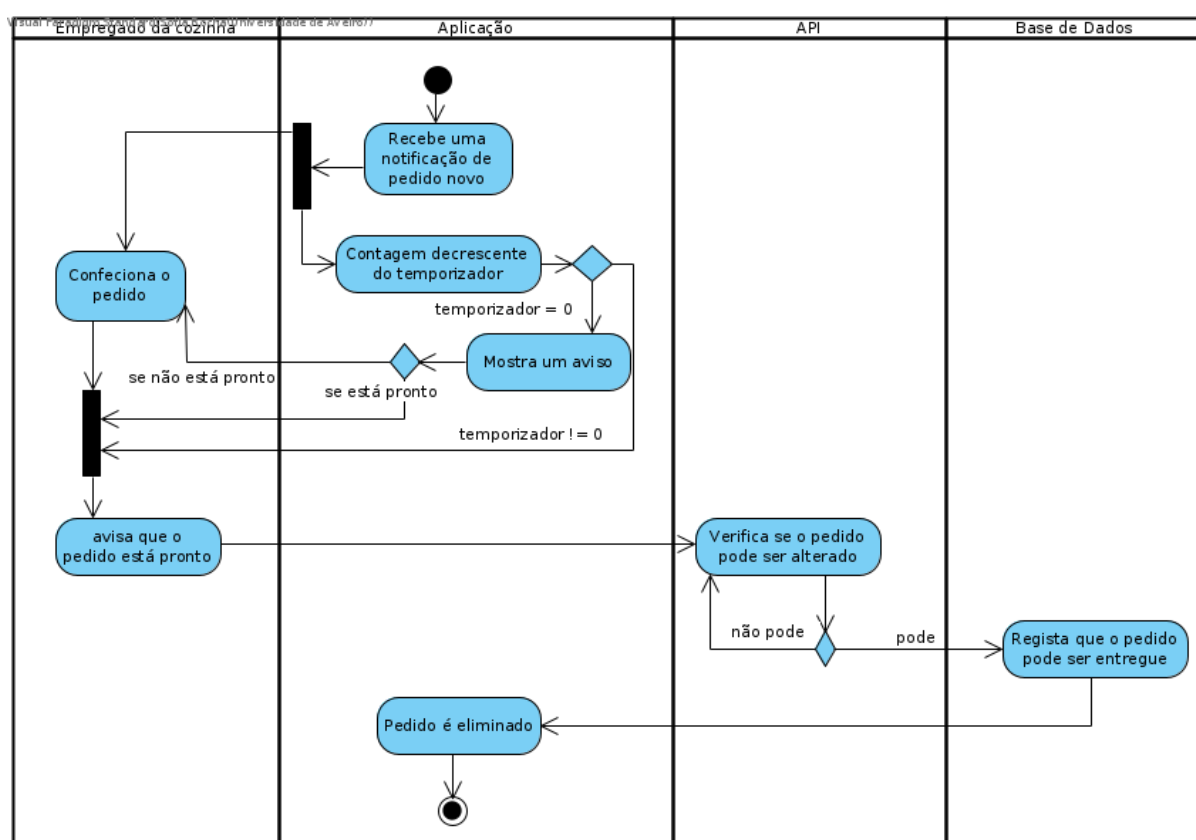


Figura 7. Diagrama de atividades do caso de uso de receber notificações por parte do empregado de cozinha

Por fim serão descritos todos os casos de uso do administrador. Este tem como função configurar todo o sistema para que os seus funcionários possam usufruir. Se nenhuma configuração for feita nenhum dos funcionários pode registar o pedido ou sequer aceder às funcionalidades do sistema mencionadas acima. Por isso quando acede com um nome de utilizador e palavra-passe, para além de ser obrigado a mudar, é obrigado a fazer as configurações básicas do sistema.

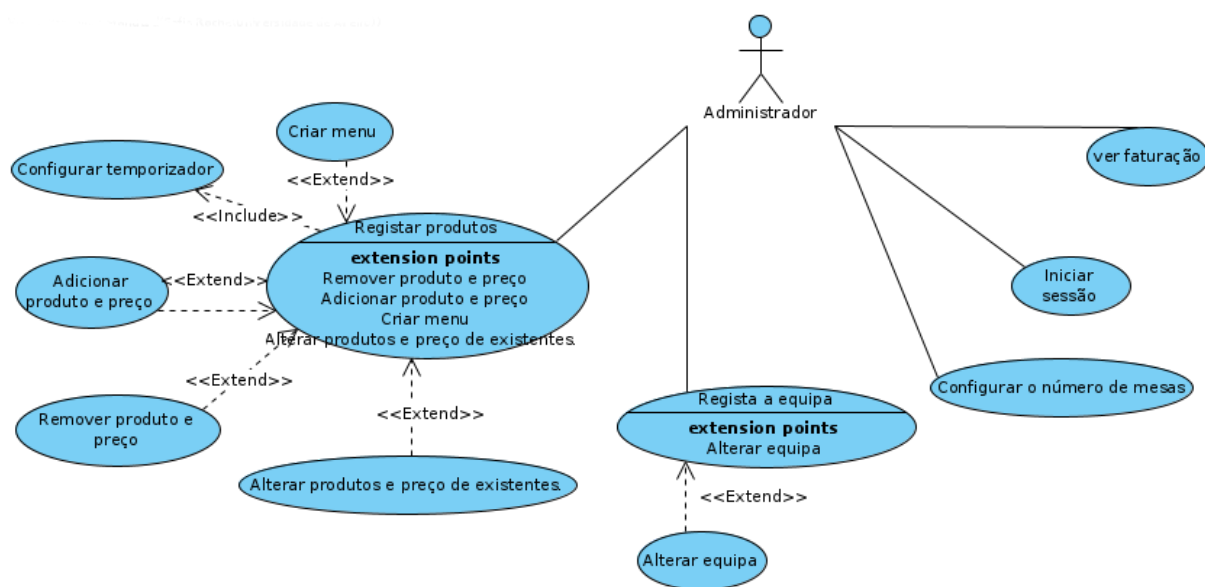


Figura 8. Diagrama de casos de uso do administrador

Tabela 14. Descrição do caso de utilização de iniciar sessão, configurar o número de mesas e registar a equipa por parte do administrador

Nomes: Iniciar sessão, configurar o número de mesas e registar a equipa
Atores: Administrador
Prioridade(1/3): +++
Finalidade: Configurar o sistema para os funcionários
Requisitos funcionais: <u>RF.1</u> , <u>RF.7</u> , <u>RF.21</u> , <u>RF.27</u>
Pré-condições: Ter sido registado um nome de utilizador e uma palavra-passe padrão na base de dados
Sumário: Para configurar o sistema pela primeira vez o administrador entra com o nome de utilizador e palavra-passe padrão que irá imediatamente mudada na base de dados para uma que o administrador configurar. Para além disso é obrigado a configurar o

número de mesas existentes no estabelecimento e também os membros da equipa. Cada funcionário terá associado um nome de utilizador e uma palavra-passe. No entanto se o administrador posteriormente quiser eliminar algum membro simplesmente ficará inativo na base de dados para guardar constantemente esses dados. Continuando com a configuração obrigatória o administrador tem de adicionar pelo menos um produto. Após isso pode adicionar categorias, mais produtos e até menus.

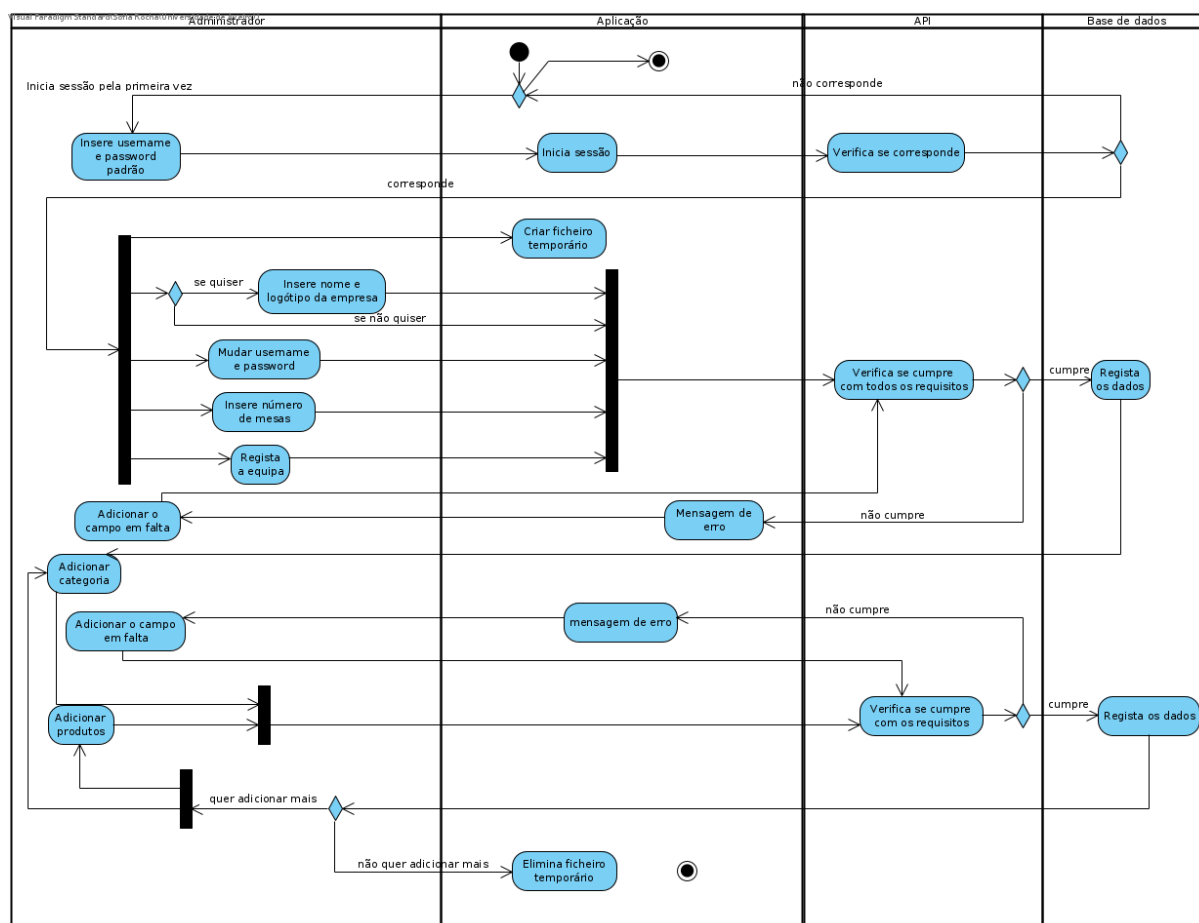


Figura 9. Diagrama de atividade para o início de sessão pela primeira vez do administrador

Tabela 15. Descrição do caso de utilização de ver a faturação por parte do administrador

Nome: Ver faturação
Atores: Administrador
Prioridade(1/3): +
Finalidade: Visualizar o dinheiro ganho pela empresa
Requisitos funcionais: <u>RF.16</u>

Pré-condições: Ter sido registado pagamentos dos pedidos
Sumário: O administrador pode ver em gráfico quanto dinheiro ganhou num determinado período de tempo escolhido pelo o mesmo.

Tabela 16. Descrição do caso de utilização de registar os produtos por parte do administrador

Nome: Registrar produtos
Atores: Administrador
Prioridade(1/3): +++
Finalidade: Registrar os produtos no sistema para venda
Requisitos funcionais: <u>RF.9</u> , <u>RF.17</u> , <u>RF.24</u>
Pré-condições: Ter iniciado sessão na aplicação
Sumário: O administrador pode registar quantos produtos quiser mas tem de ter uma imagem, uma categoria associada, preço e um temporizador. Com esses produtos pode criar menus. Também pode remover cada um deles e alterar consoante o que precisar.

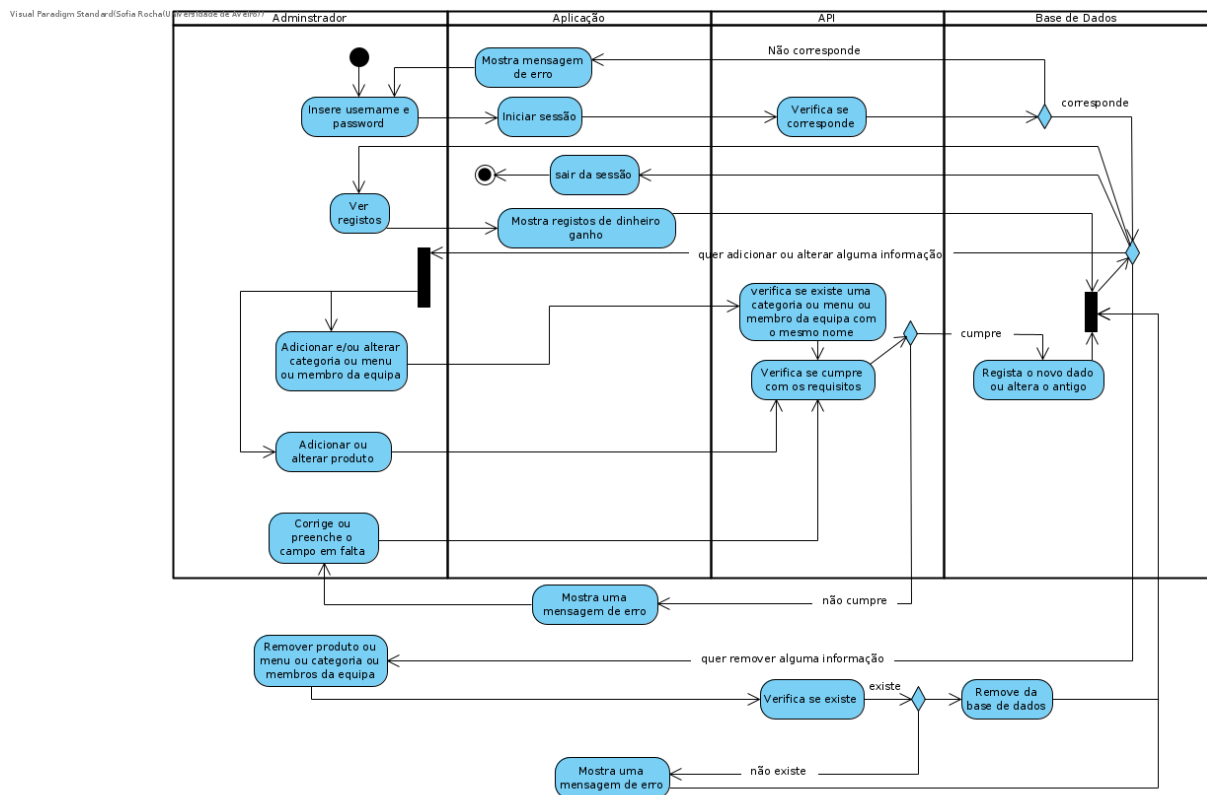


Figura 10. Diagrama de atividades das funcionalidades do administrador

5. Modelo dinâmico

Este modelo consiste em mostrar quais são as interações entre os vários atores, o sistema e a base de dados. Todos os diagramas estão divididos em funcionalidades diferentes para facilitar a compreensão das mesmas. Primeiramente irá ser falado sobre a configuração do sistema, seguido de:

- confeccionar o pedido;
- pagamento do pedido;
- registo do pedido.

5.1. Configuração do sistema

Antes de o empregado de mesa poder fazer algum registo dos pedidos o administrador tem de configurar o sistema de modo que tenha as configurações básicas para que o sistema esteja adaptado ao estabelecimento. Por isso o diagrama a seguir representa o início de sessão por uma palavra-passe e nome de utilizador padrão por parte

do administrador. Assim que é aceite é obrigado a alterar os seus dados por uma questão de segurança. Logo após tem indicar quantas mesas tem o estabelecimento e a sua lotação e registar toda a equipa criando nome de utilizadores e palavra-passes para cada um.

No fim pode criar produtos, sendo obrigado a criar pelo menos um, menus, quantas vezes quiser associadas a categoria também criadas por ele. Se depois não quiser alterar mais nada então poderá terminar a sessão.

Todos os dados escritos antes de serem enviados para a base de dados são válidos pela API para que evite confusões entre a aplicação e a base de dados.

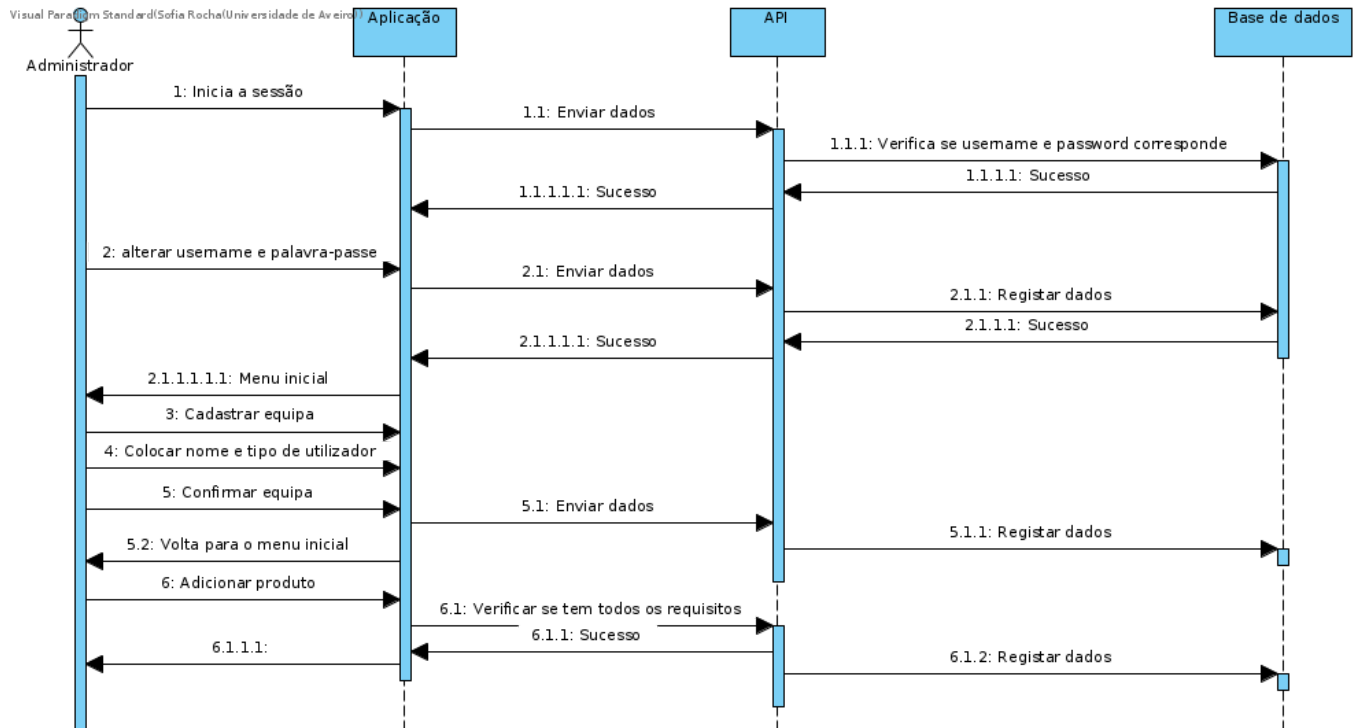


Figura 11. Diagrama de sequência da configuração básica

5.2. Registo do pedido do cliente

Depois de todo o sistema ter sido configurado, o empregado de mesa pode registar o pedido do cliente. No próximo diagrama existem várias interações com o cliente uma vez que será este que fará o pedido e o empregado de mesa irá registá-lo. Antes de o fazer é escolhido entre escolher a mesa ou escolher para levar, consoante a opção de consumo do cliente. Após isso é escolhido todos os produtos que o cliente quer consumir e após a finalização e confirmação do pedido este é registado na base de dados. No fim a aplicação mostra o menu inicial com todas as mesas pronto para fazer um novo pedido.

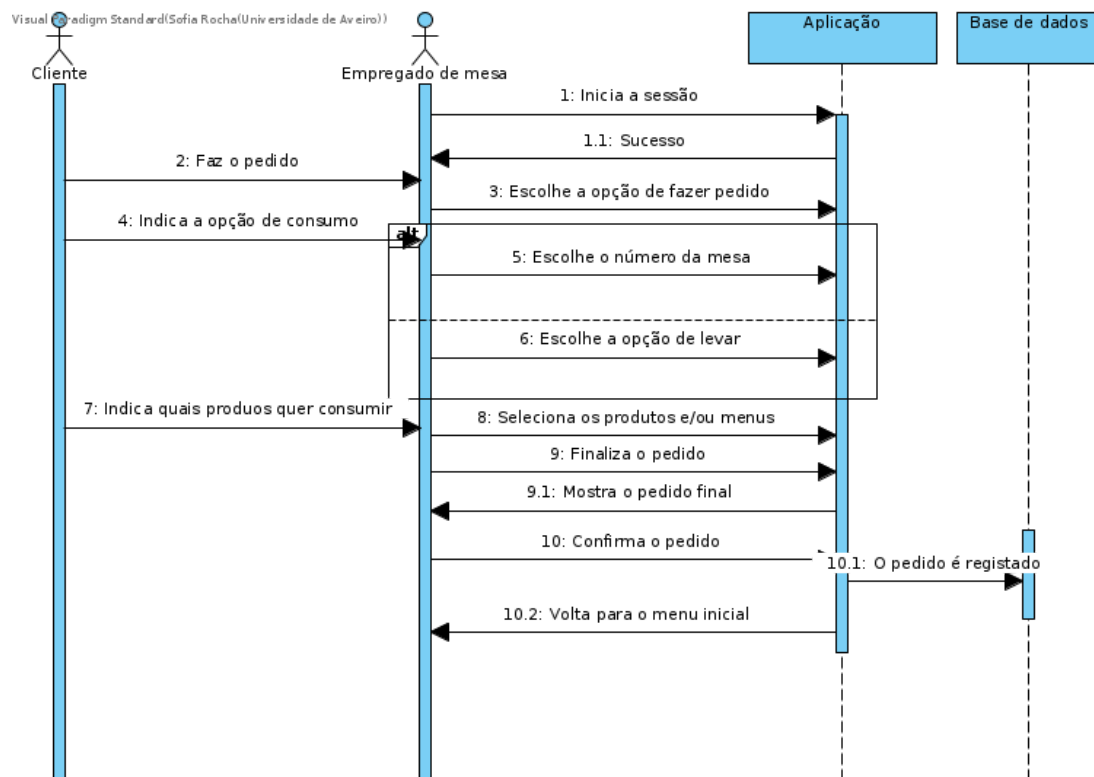


Figura 12. Diagrama de sequência do registo do pedido no sistema

5.3. Confeccionar o pedido

No diagrama seguinte há diversas interações que vão desde a aplicação receber o pedido para confeccionar até ser entregue ao cliente. Este diagrama começa com uma notificação da aplicação que tem como objetivo avisar que foi feito um registo de um pedido novo. Após isso, o empregado de cozinha começa imediatamente a confeccioná-lo. Assim que estiver pronto o empregado avisa a aplicação que o pedido está pronto a ser entregue.

Imediatamente a seguir essa informação é registada na base de dados e o empregado de mesa recebe uma notificação. Consequentemente, o pedido é eliminado da interface do empregado de cozinha. No fim o pedido é entregue ao cliente e é registado na base de dados que o pedido foi entregue.

No entanto, se por algum motivo o empregado da cozinha demorar mais tempo que o previsto então a aplicação mostra um aviso de que o pedido está atrasado.

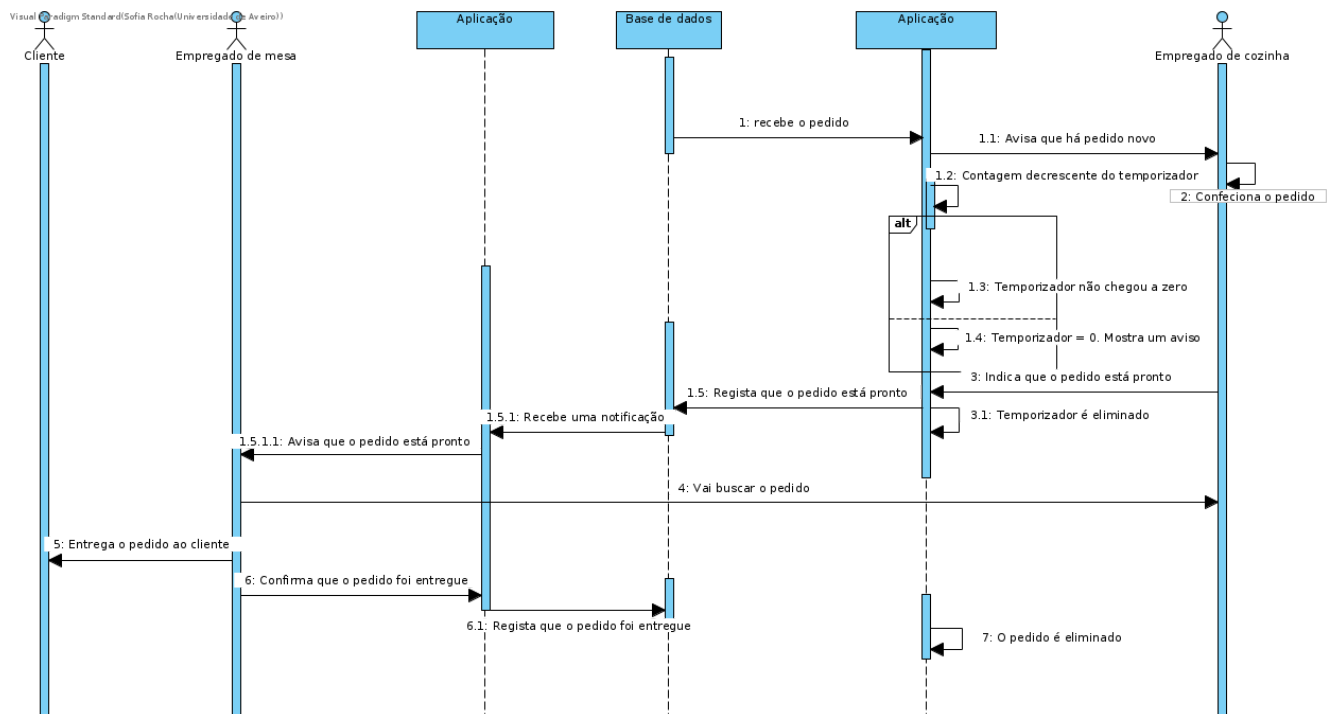


Figura 13. Diagrama de sequência de confeccionar o pedido por parte do empregado de cozinha

5.4. Pagamento

No ato do pagamento o empregado de mesa só necessita de seleccionar a mesa em que o pedido do cliente foi registado para depois saber o valor e seleccionar a opção de pagamento. Este pagamento tem duas opções: uma em numerário e outra em cartão bancária e as duas dão para dar gorjeta se o cliente assim o entender. Assim que é confirmado o pedido é eliminado da base de dados ficando somente o valor deste.

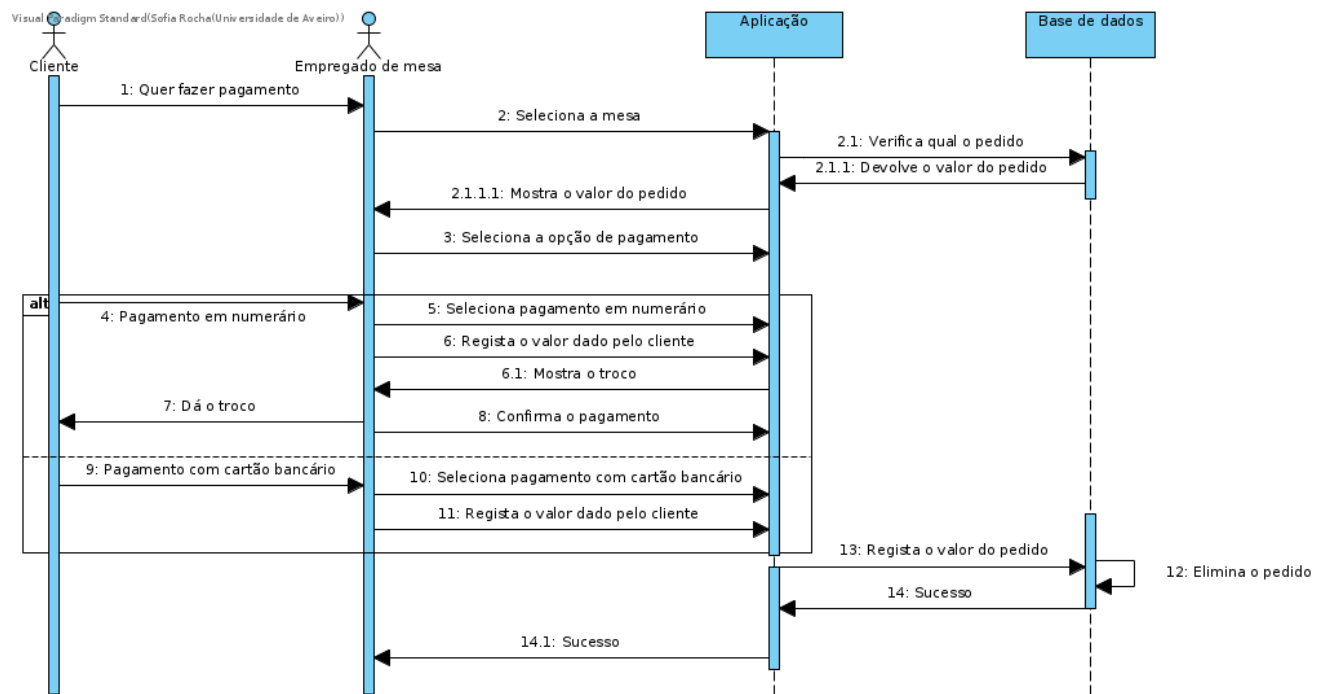


Figura 14. Diagrama de sequência do pagamento do pedido

6. Modelo estrutural

Por fim foi criado um diagrama de classes para entender a estrutura de todo o *software*. Este diagrama representa todos os dados guardados na base de dados para além de todas as dependências.

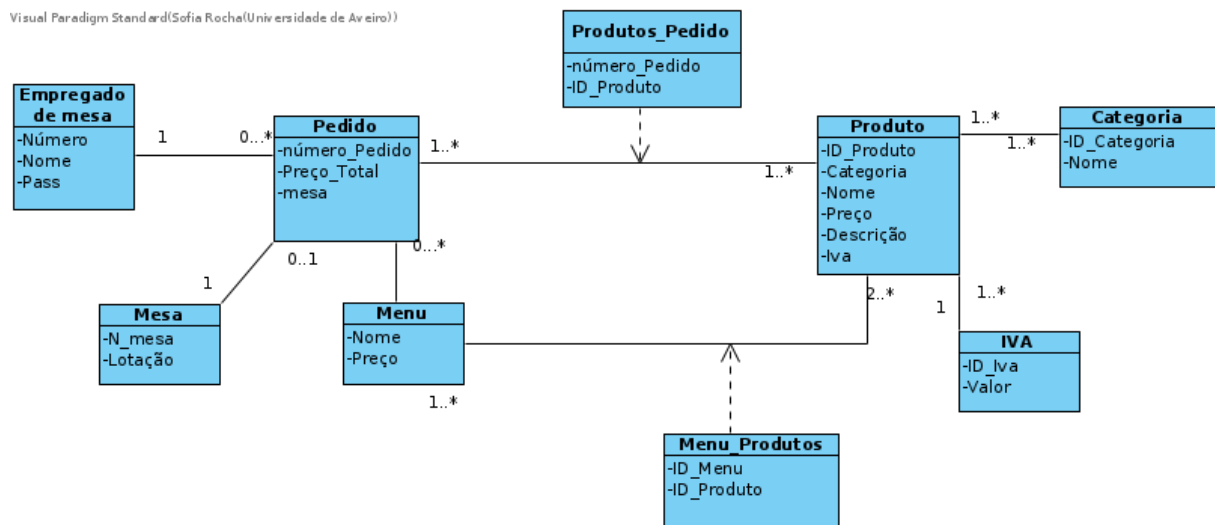


Figura 15. Diagrama de classes

Neste diagrama é visível as várias classes que irão ser criadas ao longo da implementação. Assim por exemplo para a classe produto terá vários atributos e duas classes associadas a do IVA e da categoria, uma vez que estas classes têm dados importantes para o pedido mas para evitar estar a reescrever sempre que um destes dados for alterado então fez-se desta forma.

Para além disso a cada pedido estará obrigatoriamente associada a pelo menos um produto, uma mesa e um empregado de mesa que registará o pedido. Para que quando o pedido for confeccionado saber que produtos o cliente quer consumir e também se houver algum problema como por exemplo não há *stock* poder falar com o empregado para refazer o pedido e talvez oferecer alguma sugestão. O menu também é uma opção a consumir e por isso está associado ao pedido. Para cada menu são necessários no mínimo dois produtos e um preço final que pode ser diferente dos produtos associados.

No entanto, antes mesmo de fazer um pedido o empregado de mesa tem de seleccionar a mesa e claro que associado a esta classe estará a lotação máxima de cada uma para poder facilitar a sua escolha.

Por fim, cada empregado de mesa terá um nome para se poder identificar, um nome de utilizador e uma palavra-passe diferenciado de todos os outros e assim poder aceder às funcionalidades do sistema.

7. Atividade desenvolvidas

7.1. API

7.2. Conexão à base de dados

Para a conexão à base de dados inicialmente pensou-se em usar uma API mas por falta de experiência e conhecimento isto tomou mais tempo do que estava previsto. Então usou-se o driver JDBC para uma comunicação mais “direta”. Este driver foi adicionado ao netBeans e de seguida ao projeto.

7.3. Interfaces centradas

7.4. Iniciar sessão

Antes de qualquer utilizador poder utilizar as suas funcionalidades, estes terão de iniciar a sessão escolhendo inicialmente qual a sua função dentro do estabelecimento e de seguida este tem de introduzir o seu nome de utilizador e palavra-passe correspondente. Para facilitar o utilizador a função "Administrador" já está selecionada por padrão.



The screenshot shows a Java Swing window titled "Login" with standard window controls (minimize, maximize, close). The main content area has a light gray background. At the top center, the word "Restauração" is displayed in a large, dark font. Below this, there are three radio buttons for selecting a user role: "Administrador" (which is selected), "Empregado de mesa", and "Empregado de Cozinha". To the right of the "Administrador" radio button is a text input field with the placeholder text "Nome de utilizador". To the right of the "Empregado de Cozinha" radio button is a password input field filled with dots, followed by a "Mostrar" checkbox. At the bottom center of the window is a blue button with the text "Entrar".

Se o nome do utilizador e a palavra-passe não coincidirem aparecerá uma mensagem de erro igual à figura abaixo.

O nome de utilizador e a palavra-passe não coincidem.

7.5. Administrador

Após o administrador iniciar sessão pela primeira vez com um nome de utilizador palavra-passe padrão este terá de configurar toda a aplicação para estar de acordo com o seu estabelecimento. Essa configuração está dividida em várias partes , pela seguinte ordem, que começa com:

- Alterar o nome de utilizador e palavra-passe padrão [ver [Configurar Conta](#)]. A aplicação, por uma questão de segurança, não deixará inserir o nome de utilizador nem a palavra-passe padrão.
- Configuração de mesas [ver [Configuração das mesas](#)] - o administrador indica quantas mesas tem o estabelecimento;
- Adicionar produtos, categorias e personalizações - o administrador tem de adicionar pelo menos um produto preenchendo os vários campos. Nesta interface também pode adicionar categorias e personalizações para a seguir agregar ao produto.
- Cadastrar equipa - nesta equipa o administrador adiciona toda a sua equipa, criando as várias credenciais.
- Menu inicial - por fim irá aparecer o menu, que será nas seguintes sessões, a interface principal.

7.5.1. Configurar Conta

Assim como foi dito acima, o utilizador será obrigado a mudar as credenciais padrão, por uma questão de segurança. Para além disso, não deixa inserir na base de dados nomes de utilizadores iguais e verifica se as palavras-passes são iguais, uma vez que a função **verNomeUtilizadorRepetido** retorna *true* se houver um nome de utilizador igual registado ou *false* se não houver. Se alguma destas condições não acontecer aparecerá um erro correspondente com a situação como se pode ver nas figuras seguinte.

Configuração da conta

Novo nome de utilizador

Nova palavra-passe

Repita a palavra-passe

Esse nome já existe.

Cancelar
0%
Próximo

```

if ("".equals(escreverNomeUtilizador.getText())) {
    erro.setVisible(true);
    erro.setText("Tem de inserir um nome de utilizador.");
} else if ("".equals(escreverNovaPasse.getText())) {
    erro.setVisible(true);
    erro.setText("Tem de inserir uma palavra-passe.");
} else if ("".equals(repetirPasse.getText())) {
    erro.setVisible(true);
    erro.setText("Tem de repetir a palavra-passe.");
} else if (escreverNomeUtilizador.getText().toLowerCase().equals("admin")) {
    erro.setVisible(true);
    erro.setText("Tem de escolher um nome de utilizador diferente do padrão.");
} else if (escreverNovaPasse.getText().toLowerCase().equals("admin")) {
    erro.setVisible(true);
    erro.setText("Tem de escolher uma palavra-passe diferente do padrão.");
} else if (!identidade.verNomeUtilizadorRepetido(escreverNomeUtilizador.getText(), "Admin")) {
    erro.setVisible(true);
    erro.setText("Esse nome já existe.");
} else if (!escreverNovaPasse.getText().equals(repetirPasse.getText())) {
    erro.setVisible(true);
    erro.setText("As palavras-passes não correspondem.");
}

```

Assim que o utilizador conseguir corrigir os erros a barra de progresso percorrerá até aos 100%, para que o utilizador saiba que está a acontecer algo, as credenciais irão ser mudadas e imediatamente a seguir irá aparecer uma nova interface. Estes dois processos são conduzidos por duas threads independentes.

```

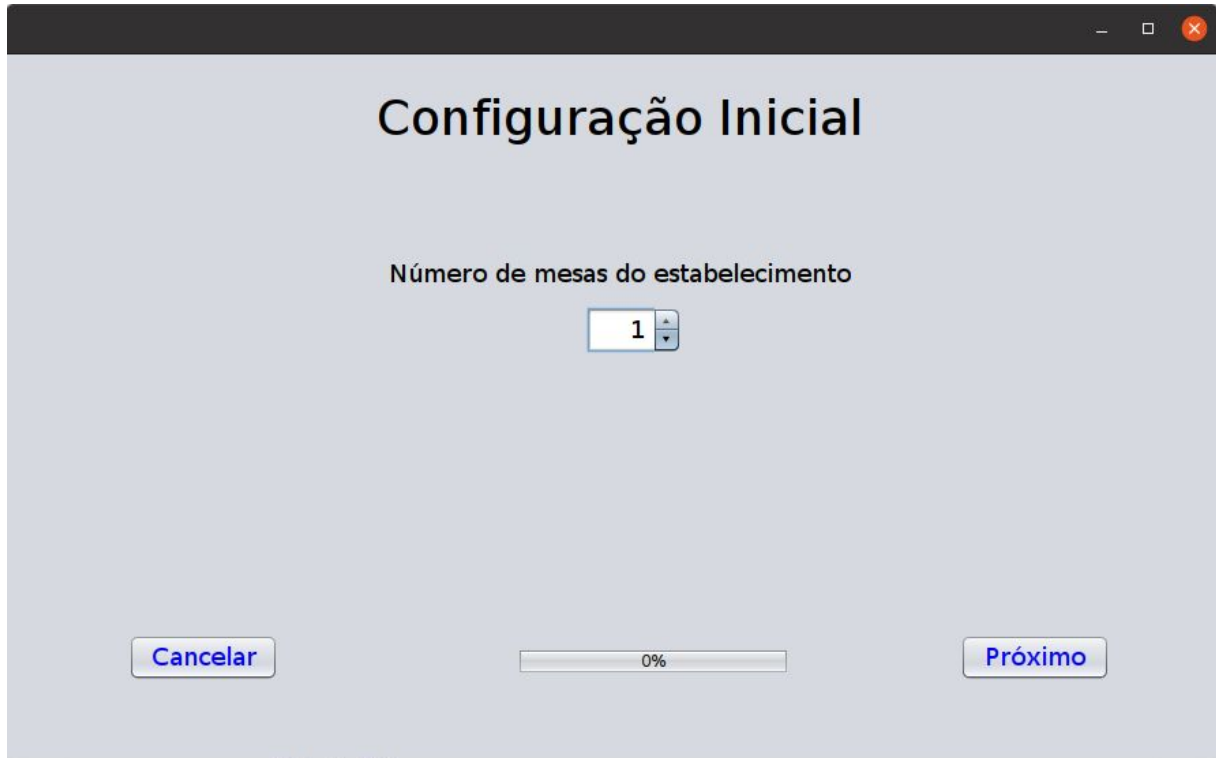
erro.setVisible(false);
identidade.mudarNomePasse(escreverNomeUtilizador.getText(), escreverNovaPasse.getText());

ProgressBar m = new ProgressBar(progressBar);
MostrarInterface mi = new MostrarInterface(this, new ConfiguracaoInicial());
m.start();
mi.start();

```

7.5.2. Configuração das mesas

Nesta etapa o utilizador insere o número de mesas existentes no estabelecimento e este será guardado na base de dados.



```
ConfiguracaoEstabelecimento ce = new ConfiguracaoEstabelecimento();  
  
int num = (Integer) spinner.getValue(); //valor do tipo int  
  
try {  
    ce.inserirNumMesas(num); // insere o numero de mesas na base de dados  
    MostrarInterface mi = new MostrarInterface(this, new AdicionarProduto());  
    mi.start();  
}
```

7.5.3. Adicionar produtos

Após inserir o número de mesas, o administrador tem de adicionar pelo menos um produto para que o empregado de mesa consiga realizar algum pedido. Para adicionar um produto este tem de ter um nome diferente de todos os outros, diferente do que estava inicialmente planeado. Esta mudança de planos deveu-se a não se conseguir distinguir os produtos na lista de produtos adicionados, que se fala mais à frente. Por isso podem aparecer dois tipos de erros: “Tem de adicionar um nome”, para que não seja adicionado um produto com um nome em branco e “Este nome já existe” para não repetir nenhum outro nome de algum produto.

Assim, para adicionar um produto é necessário colocar o seu nome, preço, tempo estimado para confeccionar, iva, categoria e opcionalmente descrição ou personalização.

A informação das *combobox* apresentadas são obtidas diretamente da base de dados e atualizadas assim que se clica, ou seja se for adicionada uma categoria ou personalização a *combobox* é atualizada, sem ter de fechar e abrir a interface, com o seguinte código (o código é semelhante para a *combobox* de personalização).

```
public void comboboxCategoria() throws ClassNotFoundException, SQLException {
    Object o = comboboxCategoria.getSelectedItemAt();
    comboboxCategoria.removeAllItems();

    String[] categorias = produto.verDados("Categoria").split("\n");
    for (String n : categorias)
        comboboxCategoria.addItem(n);

    comboboxCategoria.setSelectedItemAt(o);
}
```

Para além disso, a categoria (ou personalização) selecionada anteriormente será guardada como objeto para que na próxima que o utilizador clicar na *combobox* a mantenha selecionada. No entanto, se a categoria (ou personalização) for apagada da base de dados será selecionado o primeiro item.

7.5.4. Adicionar categorias ou personalizações

Se ao adicionar produtos o utilizador quiser adicionar alguma categoria ou personalização clica no botão “alterar”, da interface da figura x, e aparecerá um *popup* com

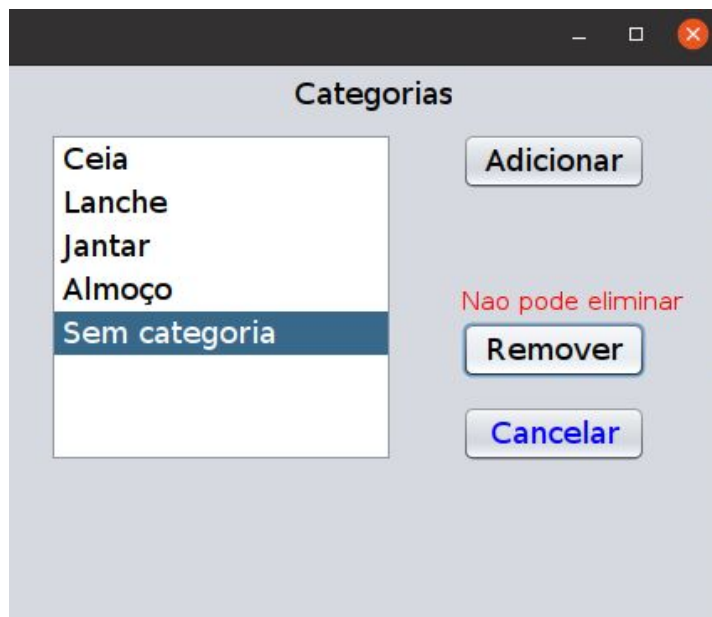
duas opções “categoria” e “Personalização”. Em cada uma surgirá uma nova interface onde se pode adicionar ou remover elementos.



O utilizador pode adicionar quantas categorias ou personalizações quiser com a condição de todos terem nomes diferentes ou não terem somente um espaço em branco. Para além disso se por algum motivo o utilizador sem querer adicionar uma categoria sem ter modificado o texto padrão este mostrado uma mensagem de aviso e não será adicionado. Assim ,se estas condições não forem respeitadas irá aparecer algum desses erros:

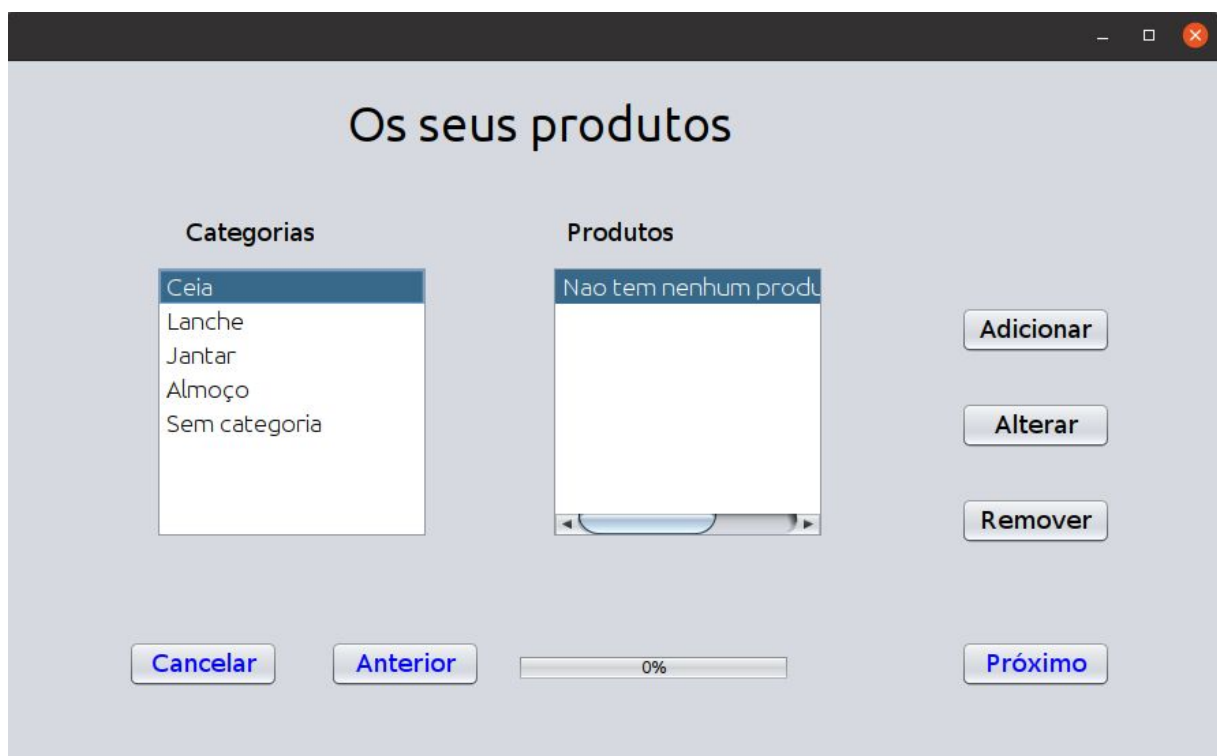
```
if (selecao == 2 && escreverCategoria.getText().equals("ex: Lanche")) {  
    erro.setVisible(true);  
    erro.setText("Nada foi alterado.");  
    selecao = 1;  
} else if ("".equals(escreverCategoria.getText())) {  
    erro.setVisible(true);  
    erro.setText("Tem de adicionar um nome");  
    selecao = 1;  
} else if (selecao == 2) {  
    selecao = 1;  
    erro.setVisible(true);  
    erro.setText("A categoria já existe.");  
}
```

No entanto, se o utilizador não quiser associar os produtos a uma categoria ou personalização específica haverá sempre a opção de “Sem categoria” ou “Sem personalização” que não pode ser removida. E também caso haja produtos que estejam associados a uma categoria ou personalização e que seja eliminada todos os produtos são associados a “Sem categoria” ou “Sem personalização”, respetivamente.



7.5.5. Produtos adicionados

Após o utilizador ter adicionado um produto este pode ver a lista de produtos associados às categorias. Nesta fase pode adicionar mais produtos com o botão “Adicionar”, pode alterar o produto se tiver enganado ou até mesmo remover.



Ao alterar um produto específico a aplicação irá “buscar” todos os dados à base de dados através do ID do produto obtido a partir do nome e coloca na interface da figura x.

A seguir todos os dados irão ser guardados na base de dados com as alterações ocorridas. Mesmo que mude o nome do produto, as alterações serão feitas corretamente uma vez que este ID é retirado antes do mudar o nome, como se pode no exemplo abaixo. Neste caso mudou-se a categoria, nome, descrição e o tempo de confecção.

ID	ID_Categoria	Nome	Preco	Descricao	ID_Iva	ID_Personalizacao	Tempo
6	35	Sopa	2		2	2	10
7	35	Bacalhau	5	Pode ser acompanhado com batatas	1	11	4

ID	ID_Categoria	Nome	Preco	Descricao	ID_Iva	ID_Personalizacao	Tempo
6	35	Sopa	2		2	2	10
7	34	Bacalhau com natas	5	Excelente escolha para duas pessoas	1	11	8

7.5.6. Cadastrar a equipa

Por último, o administrador tem de registar a equipa para que possam entrar na aplicação e atender os clientes. Para que isto aconteça o utilizador tem de inserir o nome do funcionário, que irá aparecer associado aos pedidos (ver [Empregado de Cozinha](#)), um nome de utilizador, uma palavra-passe e a sua função que pode ser uma de três: administrador, empregado de cozinha e empregado de mesa. O nome do funcionário é validado através de *regex*, para que só tenha letras e letras acentuadas, da seguinte forma:

```
else if (!nomeFuncionario.getText().matches("[a-zA-ZÁÂÃÄÅÇÈÉÊËÌÍÎÏÐÑÒÓÔÕÖ×ØÙÚÛÜÝàáâãäåçêëìíîïðñòóôõö\\s]*$")) {
    erro.setVisible(true);
    erro.setText("Só pode inserir letras");
}
```

Para que o utilizador consiga colocar mais de uma palavra esta validação termina em \s que permita que tenha um espaço no final da palavra.

Para além disso como é um nome próprio todas as primeiras letras serão guardadas como maiúsculas independentemente se o utilizador escreve dessa forma, exceto os “de” ou outras palavras que contenham menos de duas letras. Essa mudança de minúscula para maiúscula é feita da seguinte forma:

```
String[] partes = nomeFuncionario.getText().split(" ");
StringBuilder sb = new StringBuilder();
for (String word : partes) {
    if (word.length() > 2) {
        word = word.substring(0, 1).toUpperCase() + word.substring(1);
    }
    sb.append(" ").append(word);
}

String nome = sb.toString().replaceFirst(" ", "");
```

Neste caso a *string* é dividida em várias partes e dentro dessas partes é separado a primeira letra (word.substring(0,1) para ser maiúscula e de seguida junta-se o resto da palavra (word.substring(1)). Por fim tudo é transformado numa string com o replaceFirst(``, ``). No entanto, apesar destas verificações e ao contrário do nome de utilizador(por ser utilizado para iniciar sessão na aplicação), podem existir vários nomes iguais, uma vez que no mundo existem várias pessoas com o mesmo nome.

Adicionar empregado

Nome: Nome de Utilizador:

Tipo: Palavra-Passe:

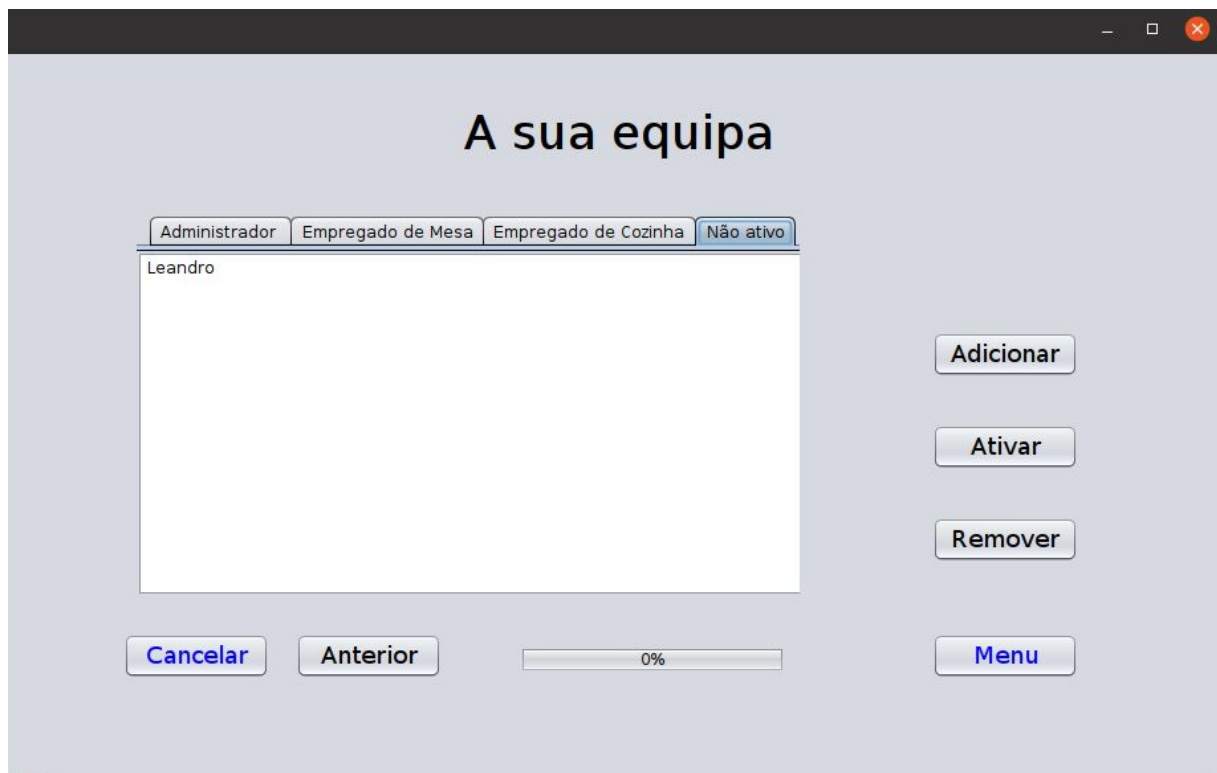
Repetir:

0%

Depois de preencher os dados do funcionário e carregar no botão “Próximo” irá poder ver toda a sua equipa. Por padrão os funcionários adicionados terão as contas com “Ativas”.

7.5.7. Ver a equipa

Nesta etapa o utilizador pode ver todos os seus funcionários ativos e não ativos. Os funcionários ativos estão separados pela sua função sendo apresentados pelo nome, enquanto que os não ativos estão todos na mesma aba e por isso são apresentados os nomes de utilizador, neste caso, para que ao ativar seja possível distinguir os vários funcionários. Nesta interface pode adicionar um novo elemento, alterar as suas características ou até mesmo desativar a conta (uma vez que nenhuma conta é apagada da base de dados). Se quiser voltar a ativar a conta terá de escolher o botão “Ativar” na tab “Não ativos”. Este botão nas outras tabs está como “Alterar” para poder modificar dados das contas.



Na figura mostra que o funcionário “Leandro” tem a conta desativada mas ao carregar no botão “Ativar” esta irá voltar a estar “Ativa” e assim poderá iniciar sessão com a mesma.

ID	Nome_Utilizador	Palavra_passe	Funcao	Nome	Situacao
3	José	Jose21	Mesa	Maria	Ativo
9	sofia	sofia123	Cozinha	Sofia	Ativo
14	Bruna	bruna23	Admin	Bruna	Ativo
15	Leandro	leoSilva	Admin	Leandro	Não ativo

ID	Nome_Utilizador	Palavra_passe	Funcao	Nome	Situacao
3	José	Jose21	Mesa	Maria	Ativo
9	sofia	sofia123	Cozinha	Sofia	Ativo
14	Bruna	bruna23	Admin	Bruna	Ativo
15	Leandro	leoSilva	Admin	Leandro	Ativo

Para alterar os dados de uma conta o processo é muito idêntico a alterar produtos. Irão ser colocados todos os dados dos funcionários na interface e o administrador pode alterar o que pretender.

Adicionar empregado

Nome: Nome de Utilizador:

Tipo: Palavra-Passe:

Repetir:

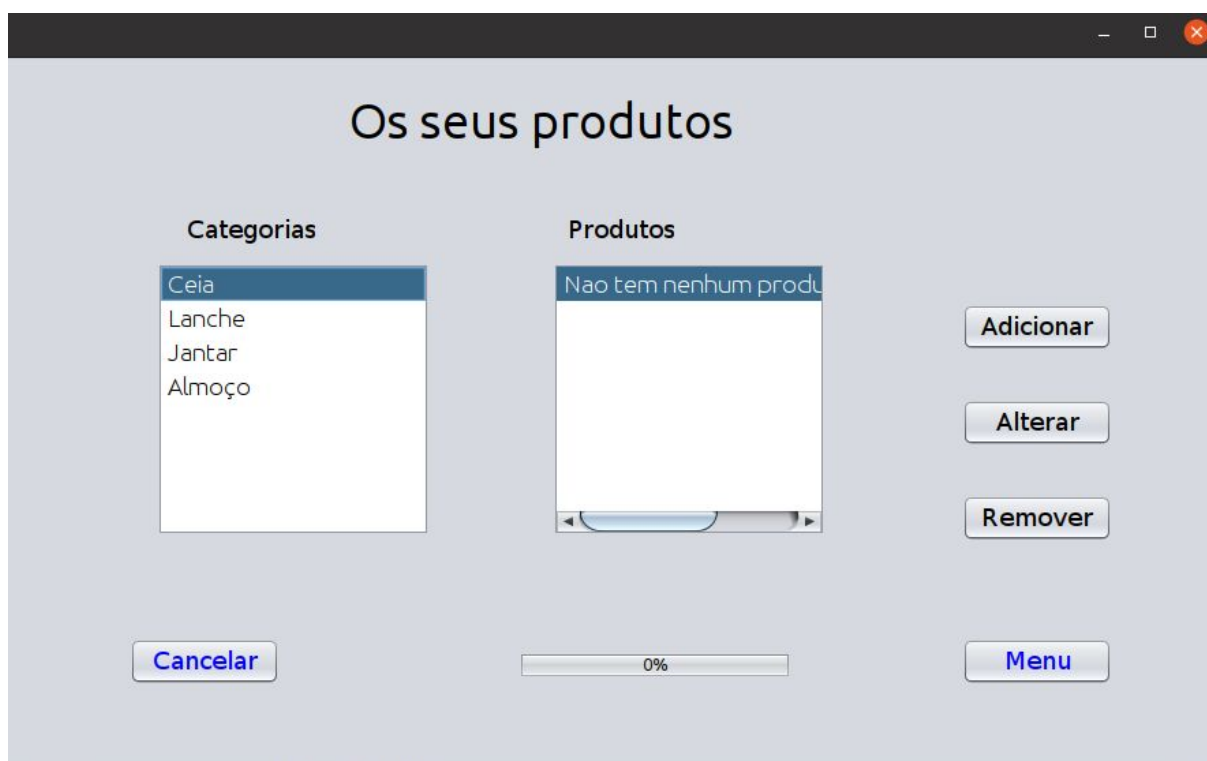
0%

7.5.8. Menu inicial

Depois de concluir toda a configuração inicial o menu será a interface principal do utilizador. A partir deste poderá administrar todo o estabelecimento através da aplicação. Poderá ver os seus produtos, equipa ou categorias. A faturação apesar de estar no menu não está implementada.

Menu inicial

Para reutilização de código não foram criadas novas interfaces e por isso foi utilizada uma variável booleana que permite identificar se o utilizador já completou as configurações básicas. Esta variável permite que consoante a situação possamos alterar um pouco os botões de “Próximo” que muda para “Menu” e colocar invisível o botão “Anterior”, uma vez que já não existe uma sequência de interfaces.



7.6. Empregado de Mesa

O empregado de mesa terá de aceder à página `ver_Mesas`. Nessa página, ele terá os números de todas as mesas existentes na base de dados. As mesas irão aparecer como botões, nos quais o empregado pode clicar, e ao clicar, será enviado para a página `pedir_pedidos`, esses botões estão diferenciados com cores, ou seja, os botões a vermelho indicam que a mesa está a ser ocupada, já os botões a cinza indicam que a mesa está livre para ser utilizada.

Nessa página aparecerão botões com os nomes de todas as categorias de produtos. Ao clicar num desses botões, será impresso na lista da esquerda, todos os produtos

referentes a essa categoria. Nessa lista, o empregado pode clicar no produto que quer e, clicando no botão “adicionar”, irá adicionar o produto selecionado na lista da esquerda. Existe também um botão remover, no qual o empregado pode remover produtos da lista da direita.

No final, para confirmar o pedido, o utilizador clica no botão “Confirmar Pedido”, para criar o pedido e armazená-lo na base de dados.

7.7. Empregado de Cozinha

Apenas existe uma página destinada ao empregado de cozinha, sendo que o nome é Ver Pedidos.

Nessa página irão aparecer três listas, uma delas representando os pedidos pelo número da mesa, outra mostrando os produtos a serem preparados depois de clicar num dos pedidos e por último uma lista que mostra as personalizações de produtos caso o pedido contenha alguma.

Também mostrará um temporizador que fará uma contagem decrescente que indica quanto tempo têm os empregados para realizar o pedido.

Por fim, a página terá dois botões nos cantos inferiores, um com o texto “Sair” que serve para o empregado fazer logout da aplicação e outro botão com o texto “Pronto” que serve para indicar que o pedido já foi confeccionado.



8. Comparação do planeamento real com o previsto

A seguir estão os requisitos que não foram concebidos com sucesso, sendo que os requisitos relacionados com o ficheiro temporário não foram adicionados nesta lista pois foi uma escolha do grupo não utilizar o mesmo pois seria irrelevante para o projeto porque seria utilizada uma base de dados onde todas as funcionalidades que o ficheiro temporário nos dava a base de dados também dava.

Requisitos	
RF.2	O pedido pode ter personalizações ou um ou mais menus
RF.3	Cada pedido tem duas opções de consumo: consumir no local ou levar.
RF.4	O empregado quando faz o pagamento tem a opção de escolher entre numerário e multibanco.
RF.8	Há sempre uma notificação quando há um pedido novo e quando o pedido está pronto.
RF.13	O administrador no final do dia vê o dinheiro ganho num período de tempo escolhido pelo mesmo
RF.14	Todos os empregados de mesa têm acesso a todos os pedidos já confeccionados.
RF.15	Ao fazer o pagamento o funcionário tem de colocar o dinheiro recebido pelo cliente para a aplicação poder fazer o troco.
RF.16	O administrador pode ver um gráfico de barras com toda a faturação do mês.
RF.26	Sempre que o utilizador quiser fechar a aplicação aparecerá uma mensagem de confirmação e nenhum dado é guardado.
RF.33	O empregado de mesa tem de confirmar o pagamento.
RF.37	O empregado de mesa pode registar as gorjetas por cartão bancário ou em numerário.

RF.37	O empregado pode ver qual é a lotação de cada mesa antes de selecionar.
	Reduzir a carga de memória
RInt.5	Sempre que um pedido para confeccionar exceder o tempo definido aparecerá uma notificação a avisar o empregado de cozinha.
RInt.9	Ao escolher a funcionalidade de fazer o pedido a interface aparece com vários números (das mesas) e a opção de levar para casa.
RInt.10	Na página de pedidos para confeccionar os pedidos estão divididos em duas categorias (uma para levar e outra para consumir no local).
RInt.11	Sempre que um pedido for finalizado aparecerá uma interface com todos os dados do pedido (preço, produtos selecionados e a opção de consumo)
RSeg.2	Os dados têm de estar inscritos na base de dados, sendo a encriptação feita pelo SGB.
RSeg.8	Nenhum menu pode ser guardado se só tiver um único produto associado.
RSeg.18	O SGBD é que gera o número de pedidos para cada pedido.
RSeg.22	Cada pedido tem de ter uma categoria associada (para levar ou para consumir no local).
RSA.1	O sistema tem de ter acesso ao armazenamento do sistema operativo.

9. Conclusão

Damos assim como concluído este relatório. Esperamos ter abordado todos os tópicos aos quais nos propusemos falar, e esperamos também ter sido claros e sucintos o suficiente na explicação dos mesmos. Este projeto foi uma atividade que se estendeu durante todo este primeiro semestre, e acompanhou toda a nossa aprendizagem, tanto na área de planeamento, que envolveu a disciplina de Engenharia de Software, como também na parte mais prática que envolveu a disciplina de Sistemas de Bases de Dados.

Durante todo o tempo do projeto fomos fazendo uso das matérias lecionadas nas disciplinas supra-referidas, o que fez com que conseguíssemos concluir este projeto.

Durante este projeto lidamos com várias dificuldades, uma delas sendo a disponibilidade de tempo dos elementos do grupo. Sendo que inicialmente éramos um grupo com cinco elementos, a meio do projeto um dos nossos colegas abandonou o grupo, sendo que a partir daí passámos apenas a ser quatro. Como a maioria dos elementos do grupo são trabalhadores-estudantes, a atenção para o projeto sempre foi menor do que o necessário.

Devido a essa falta de atenção dada, tivemos também algumas dificuldades na implementação de dados no projeto, tendo ficado por implementar algumas das funcionalidades que tinham sido inicialmente projetadas. Em projetos futuros, esperamos não ter de lidar com o mesmo tipo de problemas, pois aprendemos da pior maneira, que não dar a atenção necessária ao projeto se paga caro no futuro.

Desta forma, podemos concluir que este projeto foi algo extremamente necessário e importante, para cada um dos elementos deste grupo, pois não só nos fez melhorar as nossas dinâmicas de trabalho em grupo, tanto física como remotamente, mas também porque foi um elemento chave na nossa melhor aprendizagem das matérias que fomos aprendendo.