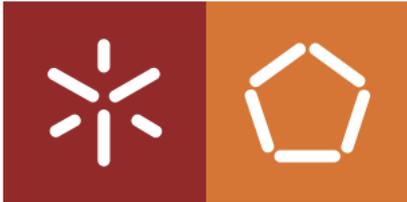


PROJETO EM ENGENHARIA INFORMÁTICA



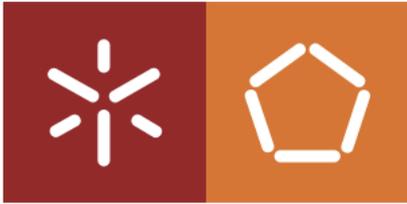
Universidade do Minho
Escola de Engenharia

Nite

Equipa: Bazinga

1 de Fevereiro de 2021

PROJETO EM ENGENHARIA INFORMÁTICA



Universidade do Minho
Escola de Engenharia

Nite

Equipa: Bazinga

ID Documento	RT-20210201-PEI2020
Versão	2.0
Acesso	Restrito
Data de emissão	1 de Fevereiro de 2021
Autores	Ana Pereira Bruno Pereira Diogo Sobral Francisco Freitas Hugo Guião Maria Dias Mariana Pereira Nuno Silva Pedro Freitas
Destinatário	PEI2020

Conteúdo

1	Equipe de trabalho	4
2	Sumário executivo	6
3	Introdução	7
3.1	Oportunidades	7
3.2	A Ideia	8
3.3	Requisitos	9
3.3.1	Cartão Digital	9
3.3.2	Pagamentos	12
3.3.3	Segurança	13
3.3.4	Venda de Bebidas	14
3.3.5	RP's Digitais	17
3.3.6	Registo e <i>Login</i>	18
3.4	Estrutura do relatório	21
4	Modelo	22
5	Estrutura aplicacional	24
6	Interface	28
6.1	App do Consumidor	28
6.1.1	Login e Registo	29
6.1.2	Home	29
6.1.3	Consultar e Alterar Perfil	30
6.1.4	Consultar Histórico	31
6.1.5	Consultar Consumo e Efetuar Pagamento	31
6.1.6	Estado do Cartão	33
6.1.7	Adicionar Código de GuestList	33
6.1.8	Cardápio do Estabelecimento	34
6.2	App do Funcionário do Estabelecimento	35
6.2.1	Login	35

6.2.2	Menu Principal	36
6.2.3	Ativar Cartão	36
6.2.4	Cardápio de Produtos	37
6.2.5	Bloquear Cartão	38
6.2.6	Dados do Cartão	39
6.2.7	Perfil do Utilizador	40
6.3	<i>Backoffice</i> da aplicação (<i>WEB App</i>)	42
6.3.1	<i>Login</i> e Registo	42
6.3.2	<i>Home Page</i>	43
6.3.3	Funcionários	44
6.3.4	Produtos	44
6.3.5	Cardápios	45
6.3.6	Relações Públicas	46
6.3.7	Estatísticas dos Relações Públicas	47
6.3.8	Diálogos de edição	47
6.3.9	Navegação	49
6.3.10	Conexão ao <i>backend</i>	50
7	Lógica de Controlo	51
7.1	Cartões Físicos e Digitais	51
7.2	Criação de Cartões Digitais	52
7.3	Associar Cartão ao Cliente	53
7.4	Creditação de Produtos	54
7.4.1	Obter Informação do Cartão Digital	54
7.4.2	Creditar Produto	55
7.5	Pagamento de cartões	57
7.5.1	Pedido de pagamento	57
7.5.2	Confirmar pagamento	58
8	Instalação e manutenção	60
8.1	Interfaces	61
8.2	Back-end	61
8.3	Base de dados	62
9	Conclusão	63

1

Equipe de trabalho

Caraterização dos elementos da equipe *Bazinga*.

Responsáveis

- **Coordenador** : Diogo Sobral
Contacto: a82523@alunos.uminho.pt

Membros da equipa

- Maria Dias
Contacto: a81611@alunos.uminho.pt
- Nuno Silva
Contacto: a78156@alunos.uminho.pt
- Hugo Gião
Contacto: pg41073@alunos.uminho.pt
- Bruno Pereira
Contacto: pg37148@alunos.uminho.pt
- Ana Pereira
Contacto: a81712@alunos.uminho.pt

- Mariana Pereira

Contacto: a81146@alunos.uminho.pt

- Pedro Freitas

Contacto: a80975@alunos.uminho.pt

- Francisco Freitas

Contacto: a81580@alunos.uminho.pt

2

Sumário executivo

O *Nite* é um produto inovador que pretende levar a componente tecnológica dos bares e estabelecimentos noturnos do nível obsoleto à vanguarda da tecnologia. Desta forma, o produto descrito neste relatório propõe-se a resolver dois problemas que retiram felicidade aos clientes destes negócios, a insegurança presente nos cartões de consumo e as enormes filas de pagamento que retardam as saídas destes estabelecimentos.

Através do seu sistema completo, o *Nite* substitui os cartões de consumos em papel por cartões *NFC* combinados com cartões de consumo digitais capacitando os consumidores para se divertirem mais, os funcionários para serem eficazes e os gestores para digitalizarem os seus serviços. Estes objetivos são alcançados com o uso das suas 3 aplicações desenvolvidas para serem intuitivas e de fácil utilização.

Finalmente, o *Nite* possui uma arquitetura baseada em micro-serviços desenvolvida com o objetivo de garantir a qualidade do serviço, a escalabilidade e a segurança dos seus utilizadores.

No presente relatório, encontram-se descritos os processos internos para tornar este serviço funcional e para resolver os problemas a que se compromete.

3

Introdução

Cada vez mais o mundo precisa de produtos inovadores e capazes de solucionar problemas reais e que tornem as nossas vidas melhores. Numa era em que corremos, sistematicamente, contra o tempo, é importante agilizar os processos do dia-a-dia, de forma a torná-los rápidos e cómodos. Tempo é um recurso escasso e, por isso, devemos tirar o maior proveito do mesmo.

Muitos dos estabelecimentos ou eventos noturnos sofrem com a existência de processos demorosos. Para além dos custos monetários associados a este problema, quem acaba por sofrer mais são os consumidores finais, que vêm a qualidade das suas experiências e dos serviços prestados severamente reduzida. A digitalização de um serviço aparece muitas vezes como uma forma de combater estes problemas e, estando este setor atrasado do ponto de vista tecnológico, esta abordagem pode ser a resposta que precisamos.

O *Nite* aparece como uma solução para resolver dois problemas essenciais dos estabelecimentos noturnos: as enormes filas para pagar e a insegurança que os cartões de consumo acarretam. Através do desenvolvimento de uma plataforma unificada e inovadora, o nosso produto procede à digitalização de alguns serviços de forma a solucionar os problemas que pretendemos combater.

3.1 Oportunidades

A tecnologia desempenha um papel importante em qualquer negócio. Acompanhar de perto o progresso e as necessidades tecnológicas dos clientes é fundamental para assegurar o bom desempenho de qualquer projeto. Ao contrário de muitos setores, grande parte dos estabe-

lecimentos noturnos estão subdesenvolvidos neste sentido, o que abre uma janela de oportunidade muito aliciante para novas oportunidades.

A nível temporal, vivemos numa época atípica. Todos os dias surgem notícias a relatar o fecho de estabelecimentos neste setor. Ainda que o número de futuros clientes esteja a diminuir, o número de consumidores destes negócios está a sofrer precisamente o efeito contrário. A cada dia que passa, a predisposição deste público para frequentar locais está a aumentar e, no momento em que o setor reabrir, existirá um excedente de pessoas. Se até aqui os problemas como as filas de pagamento já eram evidentes, então, daqui para a frente, serão ainda mais problemáticos devido a este excedente.

Os grandes negócios que prosperaram durante a pandemia possuíam todos uma componente digital muito forte. As limitações impostas à circulação da população obrigaram os consumidores a adaptar-se a estas novas plataformas de forma a manterem a sua qualidade de vida. Esta adaptação, por sua vez, aumentou a predisposição e a dependência do uso de meios de pagamentos digitais. Mais que nunca, as pessoas procuram utilizar estes meios no seu dia-a-dia, recorrendo cada vez menos a dinheiro vivo. Assim como outros setores, os estabelecimentos noturnos têm de acompanhar esta evolução, visto que é uma necessidade por parte dos seus clientes.

Por fim, a paragem deste setor é uma oportunidade única para a introdução de um novo sistema de funcionamento. Sempre que surgem novos produtos, existe alguma resistência à mudança, o que muitas vezes representa um fator decisivo na adoção de novos sistemas. O fecho e reabertura do setor pode ser visto como uma oportunidade para reduzir esta resistência e para facilitar a adoção de um sistema inovador, quer por parte dos clientes e quer pelos seus consumidores.

3.2 A Ideia

O *Nite* é um sistema digital de consumo cujo objetivo é digitalizar os cartões físicos que são usados por grande parte dos estabelecimentos noturnos. Ao criar um cartão de consumo digital, é possível não só combater a insegurança adjacente ao uso de cartões normais, mas também resolver as enormes filas para pagar, com pagamentos por telemóvel.

Através do uso de cartões com a tecnologia NFC e de uma aplicação desenvolvida por nós, um utilizador conseguiria associar o cartão que recebe à entrada destes estabelecimentos à sua conta na nossa plataforma. O cartão NFC teria um cartão digital associado e seria utilizado para interagir com o resto do sistema.

A partir do momento em que a associação é feita, um utilizador consegue, através do seu

telemóvel, consultar todas as informações do seu cartão, controlar de forma remota o seu estado, efetuar o pagamento do mesmo e até consultar o cardápio do estabelecimento.

A nossa ideia passa por criar uma plataforma comum que permita agregar vários estabelecimentos no mesmo sistema. Desta forma, os consumidores finais poderiam usufruir dos nossos serviços, em todos os estabelecimentos aderentes, utilizando a mesma aplicação.

3.3 Requisitos

Nesta secção apresentamos os requisitos funcionais do nosso produto nas suas diferentes vertentes. Estes requisitos representam funcionalidades a disponibilizar aos utilizadores do sistema.

As secções são apresentadas pela ordem de prioridade no sistema. Os requisitos apresentados em cada secção também se encontram organizados do mais importante para o menos importante. Note-se que utilizamos o termo *funcionário* de forma geral, não fazendo distinção entre diferentes cargos, como *bartender* e segurança.

3.3.1 Cartão Digital

Req. #: 3.3.1.1 | **Use case:** Identificar um cartão

Descrição: Um funcionário registado deve ser capaz de fazer scan de um cartão de consumo físico na sua aplicação.

Razão: Desta forma o funcionário identifica o cartão do consumidor e pode creditar bebidas no mesmo.

Req. #: 3.3.1.2 | **Use case:** Identificar um cartão

Descrição: Um consumidor registado na aplicação deve ser capaz de fazer scan do seu cartão de consumo físico, associando o respetivo cartão virtual à sua conta.

Razão: Desta forma o utilizador passa a poder usufruir das funcionalidades da aplicação.

Req. #: 3.3.1.3 | Use case: O cartão virtual simula um cartão de bebidas

Descrição: O cartão virtual do consumidor deve conseguir simular um cartão de bebidas, onde normalmente se escreve à mão, que contém o consumo, quantidade, e total a pagar. O cartão pode conter outras informações como sexo do utilizador e código de guest list.

Razão: Deste modo, as bebidas ficam creditadas digitalmente.

Req. #: 3.3.1.4 | Use case: Consultar o estado do cartão virtual

Descrição: O consumidor deve ser capaz de visualizar todos os seus consumos e respetivo valor total a pagar.

Razão: O cliente tem controlo sobre o que consumiu no seu cartão desde a sua associação até ao fecho do mesmo.

Req. #: 3.3.1.5 | Use case: Um cartão está associado a um estabelecimento

Descrição: Um cartão virtual deve estar associado ao estabelecimento a que diz respeito.

Razão: Desta forma o cartão só pode realizar operações que digam respeito ao seu estabelecimento.

Req. #: 3.3.1.6 | Use case: Todo o consumidor tem um cartão físico

Descrição: É necessário existir um cartão físico para as pessoas que não possuem a aplicação poderem usufruir do estabelecimento.

Razão: O sistema deve suportar, simultaneamente, clientes com e sem aplicação móvel.

Req. #: 3.3.1.7 | **Use case:** Correspondência entre cartão físico e cartão virtual

Descrição: O cartão físico deve possuir uma forma única de o identificar e pode ou não estar associado a um cartão virtual. O cartão físico só deve estar associado caso o cartão virtual correspondente esteja ativo.

Razão: Desta forma o cartão físico permite a identificação do cartão virtual e todas as suas operações.

Req. #: 3.3.1.8 | **Use case:** Desvirtualização do cartão virtual

Descrição: O cartão virtual deve, em todos os momentos, ter um cartão físico associado.

Razão: Um cartão virtual não pode existir sem estar associado a um cartão físico.

Req. #: 3.3.1.9 | **Use case:** Identificadores físicos e digitais

Descrição: O cliente deve poder visualizar não só o identificador único do cartão físico, mas também o identificador digital. O identificador físico também deve estar presente no cartão.

Razão: Os identificadores permitem ao utilizador tirar maior proveito, não só do sistema, mas também das funcionalidades de segurança.

3.3.2 Pagamentos

Req. #: 3.3.2.1 | Use case: Efetuar pagamentos

Descrição: O utilizador deve ser capaz de realizar o pagamento do cartão através da aplicação.

Razão: Desta forma o utilizador pode pagar através do seu telemóvel e não precisa de ir para a fila.

Req. #: 3.3.2.2 | Use case: Consultar consumo

Descrição: Em qualquer momento, um utilizador deverá poder visualizar a lista de produtos consumidos (com a quantidade e preço de cada um) e montante total.

Razão: Assim o utilizador poderá verificar se não existiu algum lapso e se está a pagar por aquilo que realmente consumiu.

Req. #: 3.3.2.3 | Use case: Consultar histórico de consumo

Descrição: Um utilizador deve conseguir visualizar os movimentos associados à sua conta com informação detalhada (data, hora, lista de consumo)

Razão: Desta forma os movimentos do utilizador estão guardados na sua conta e podem ser consultados pelo mesmo.

3.3.3 Segurança

Req. #: 3.3.3.1 | **Use case:** Definir os estados do cartão virtual

Descrição: Um cartão de consumo, no sistema, pode estar num dos seguintes estados: Ativo, bloqueado ou inativo. Para além disso, pode estar pago ou não pago a qualquer momento.

Razão: Desta forma conseguimos identificar claramente o estado do cartão de consumo, a cada momento.

Req. #: 3.3.3.2 | **Use case:** Fechar um cartão

Descrição: Um consumidor deve conseguir fechar o seu cartão, efetuando o pagamento do que consumiu.

Razão: Desta forma o pagamento do cartão fica efetuado e o consumidor pode sair do estabelecimento.

Req. #: 3.3.3.3 | **Use case:** Bloquear o cartão

Descrição: O consumidor deve conseguir bloquear o seu cartão a qualquer momento.

Razão: Em caso de o perder ou este ser roubado, não lhe serão creditados mais consumos. Oferece maior segurança e tranquilidade ao consumidor.

Req. #: 3.3.3.4 | **Use case:** Fazer *reset* de um cartão

Descrição: Um funcionário registado deve ser capaz de fazer *reset* num cartão de consumo.

Razão: Desta forma o cartão de consumo fica pronto a ser utilizado por outra pessoa.

Req. #: 3.3.3.5 | Use case: Ativar um cartão

Descrição: Um funcionário registado deve ser capaz de ativar um cartão de consumo.

Razão: Desta forma o cartão fica pronto para ser utilizado.

Req. #: 3.3.3.6 | Use case: Bloquear um cartão

Descrição: Um funcionário registado deve ser capaz de bloquear um cartão de consumo.

Razão: Desta forma um consumidor pode ter o seu cartão bloqueado, mesmo que não tenha acesso à aplicação.

3.3.4 Venda de Bebidas

Creditação de itens

Req. #: 3.3.4.1 | Use case: Adicionar item ao cartão

Descrição: Um funcionário registado deve ser capaz de adicionar um item num cartão de consumo.

Razão: Desta forma o consumo do item fica registado no cartão.

Gestão do cardápio

Req. #: 3.3.4.2 | **Use case:** Criar um cardápio digital

Descrição: Assim que um estabelecimento cria conta na aplicação, deve ser criado um cardápio digital para o mesmo.

Razão: Desta forma, o cardápio do estabelecimento fica disponível para visualização na aplicação.

Req. #: 3.3.4.3 | **Use case:** Criar um cardápio digital

Descrição: Um gestor de um estabelecimento deve ser capaz de criar um cardápio digital na aplicação.

Razão: Um estabelecimento pode ter vários cardápios criados na aplicação, apesar de apenas um deles poder estar visível para os consumidores.

Req. #: 3.3.4.4 | **Use case:** Eliminar um cardápio digital

Descrição: Um gestor de um estabelecimento deve ser capaz de eliminar um cardápio digital na aplicação.

Razão: Desta forma um gestor pode eliminar um cardápio que esteja obsoleto.

Req. #: 3.3.4.5 | **Use case:** Configurar um cardápio digital

Descrição: Um gestor de um estabelecimento deve ser capaz de configurar o cardápio digital que fica visível na aplicação do consumidor numa dada noite.

Razão: Desta forma um gestor pode escolher, de entre vários cardápios, aquele que quer que fique visível para os seus clientes.

Req. #: 3.3.4.6 | Use case: Informações do cardápio

Descrição: Os itens presentes no cardápio devem conter as seguintes informações: nome, preço, descrição e tipo.

Razão: Desta forma os clientes do estabelecimento ficam a par das informações dos itens que constam no cardápio.

Req. #: 3.3.4.7 | Use case: Adicionar item ao cardápio

Descrição: Um gestor de um estabelecimento deve ser capaz de adicionar um item ao cardápio do estabelecimento.

Razão: Desta forma, o gestor consegue adicionar um item ao cardápio do estabelecimento.

Req. #: 3.3.4.8 | Use case: Alterar informações de item

Descrição: Um gestor de um estabelecimento deve ser capaz de alterar um item no cardápio do estabelecimento.

Razão: Desta forma, o gestor consegue alterar informações relativas aos itens presentes no cardápio.

Req. #: 3.3.4.9 | Use case: Remover item do cardápio

Descrição: Um gestor de um estabelecimento deve ser capaz de remover um item ao cardápio do estabelecimento.

Razão: Desta forma, o gestor consegue manter o cardápio atualizado com os itens disponíveis para consumo.

Req. #: 3.3.4.10 | Use case: Consultar cardápio

Descrição: Um consumidor do estabelecimento deve ser capaz de consultar o cardápio do estabelecimento.

Razão: Desta forma, o consumidor consegue saber que itens estão disponíveis para consumo no estabelecimento.

Req. #: 3.3.4.11 | Use case: Consultar item

Descrição: Um consumidor do estabelecimento deve ser capaz de consultar as informações de um item do cardápio do estabelecimento.

Razão: Desta forma, o consumidor consegue consultar as informações relativas a um item, como o nome, preço e tipo.

3.3.5 RP's Digitais

Req. #: 3.3.5.1 | Use case: Gerar código para um promotor

Descrição: Os gestores de estabelecimentos deverão ter a capacidade de gerar códigos para RP's

Razão: Desta forma os promotores podem ser identificados pelo seu código único.

Req. #: 3.3.5.2 | Use case: Adicionar código de *guest list*

Descrição: Um cliente deve poder inserir o código de um promotor na aplicação para ter acesso aos benefícios de *guest list* de uma festa.

Razão: Desta forma um utilizador consegue usufruir de uma *guest list*.

Req. #: 3.3.5.3 | Use case: Visualizar estatísticas

Descrição: Os gestores de estabelecimentos devem ter acesso às estatísticas dos promotores dos seus estabelecimentos.

Razão: Assim os gestores conseguem saber quais os promotores com mais influência nos seus estabelecimentos.

3.3.6 Registo e *Login*

Req. #: 3.3.6.1 | Use case: Efetuar registo na aplicação

Descrição: Um consumidor deverá ser capaz de criar uma conta na plataforma, indicando email, password, data de nascimento e nº telefone.

Razão: Desta forma o consumidor pode usufruir da nossa aplicação.

Req. #: 3.3.6.2 | Use case: Efetuar registo na aplicação

Descrição: Um gestor de um estabelecimento deve ser capaz de criar uma conta no sistema, indicando nome, email e password.

Razão: Desta forma o gestor poderá usufruir do sistema.

Req. #: 3.3.6.3 | Use case: Criar conta de funcionário

Descrição: Um gestor de um estabelecimento deverá ser capaz de criar conta para um funcionário, especificando nome, data de nascimento, função que desempenha no estabelecimento e password.

Razão: Desta forma o funcionário tem acesso à aplicação.

Req. #: 3.3.6.4 | Use case: Identificar um funcionário

Descrição: Quando um funcionário é registrado, o sistema deve gerar um identificador único para este funcionário.

Razão: Desta forma é possível identificar inequivocamente um funcionário do estabelecimento.

Req. #: 3.3.6.5 | Use case: Alterar email

Descrição: Um funcionário ou consumidor registrado no sistema deve ser capaz de alterar o seu email.

Razão: O funcionário/consumidor poderá ter trocado de email.

Req. #: 3.3.6.6 | Use case: Alterar password

Descrição: Um funcionário ou consumidor registrado no sistema deve ser capaz de alterar a sua password.

Razão: O funcionário/consumidor poderá ter-se esquecido da password.

Req. #: 3.3.6.7 | Use case: Eliminar uma conta

Descrição: Um gestor de um estabelecimento deverá ser capaz de eliminar um funcionário do sistema.

Razão: Despedimento.

Req. #: 3.3.6.8 | Use case: Alterar registo de um funcionário

Descrição: Um gestor de um estabelecimento deve ser capaz de alterar a informação que consta na ficha de um funcionário.

Razão: Alteração ou correção dos dados do funcionário.

Req. #: 3.3.6.9 | Use case: Repor uma password

Descrição: O sistema deve ser capaz de enviar um email de reposição de password.

Razão: O funcionário/consumidor poderá ter-se esquecido da password.

Req. #: 3.3.6.10 | Use case: Autenticar um utilizador

Descrição: Um utilizador registado deve ser capaz de autenticar-se na aplicação, fornecendo email e password.

Razão: Desta forma, a conta do utilizador está protegida.

Req. #: 3.3.6.11 | Use case: Fechar uma conta

Descrição: Um administrador do sistema deverá ser capaz de fechar uma conta de um consumidor.

Razão: Má conduta por parte dos utilizadores.

3.4 Estrutura do relatório

Estando introduzida a ideia do produto *Nite*, as próximas secções do presente documento abordam as várias componentes do desenvolvimento deste projeto, da modelação à implementação.

Na secção 4 descrevemos a estrutura aplicacional do nosso serviço e as decisões tomadas no desenho da arquitetura utilizada. Na secção 5 é exposto o processo de modelação do sistema. Na secção 6 apresentamos as interfaces das diferentes aplicações que integram o nosso sistema, descrevendo o seu propósito e as funcionalidades que as mesmas disponibilizam. Na secção que se segue é abordada uma parte da lógica de controlo do nosso sistema, com foco principal na funcionalidade fulcral do sistema, os cartões digitais. Na secção 8 é apresentado o diagrama de instalação do nosso sistema, fazendo referência aos serviços utilizados para tornar o nosso produto disponível. Por último, na secção 9, é feita uma conclusão acerca do trabalho realizado e uma discussão do trabalho futuro.

4

Modelo

A partir dos requisitos descritos decidimos desenvolver um modelo domínio para o nosso produto, que pode ser observado na Figura 4.1. O nosso objetivo passou por tentar descrever as principais entidades do sistema que queríamos montar e perceber como é que estas entidades interagem umas com as outras. Até aqui, a ideia do que pretendíamos construir era bastante ambiciosa e, ao falar com o nosso orientador, chegamos à conclusão de que deveríamos priorizar alguns requisitos fundamentais. Assim, o nosso modelo domínio exprime exhaustivamente o *core* do nosso produto, todavia alguns pontos menos relevantes estão apresentados de uma forma mais restritiva, mas simples.

Com o modelo de domínio feito, concluímos que no nosso sistema necessitaríamos de 3 atores diferentes.

1. **Utilizador** - Representa o consumidor final de um estabelecimento
2. **Funcionário** - Representa um trabalhador de um dado estabelecimento
3. **Gestor** - Representa o responsável pela gestão de um estabelecimento

Estes três atores têm todas funções distintas que, quando conjugadas, asseguram o funcionamento do nosso serviço. Um gestor realiza as operações de controlo aos acessos dos funcionários, de gestão dos produtos vendidos aos consumidores e dos cardápios disponibilizados para cada noite. Por fim, tem ainda a seu encargo a gestão dos códigos de relações públicas.

Um funcionário é responsável por efetuar as operações que respondem aos pedidos dos consumidores. Cabe aos funcionários creditar bebidas nos cartões dos clientes, ativar e bloquear cartões, na entrada de um consumidor no estabelecimento e em caso de perda de um cartão,

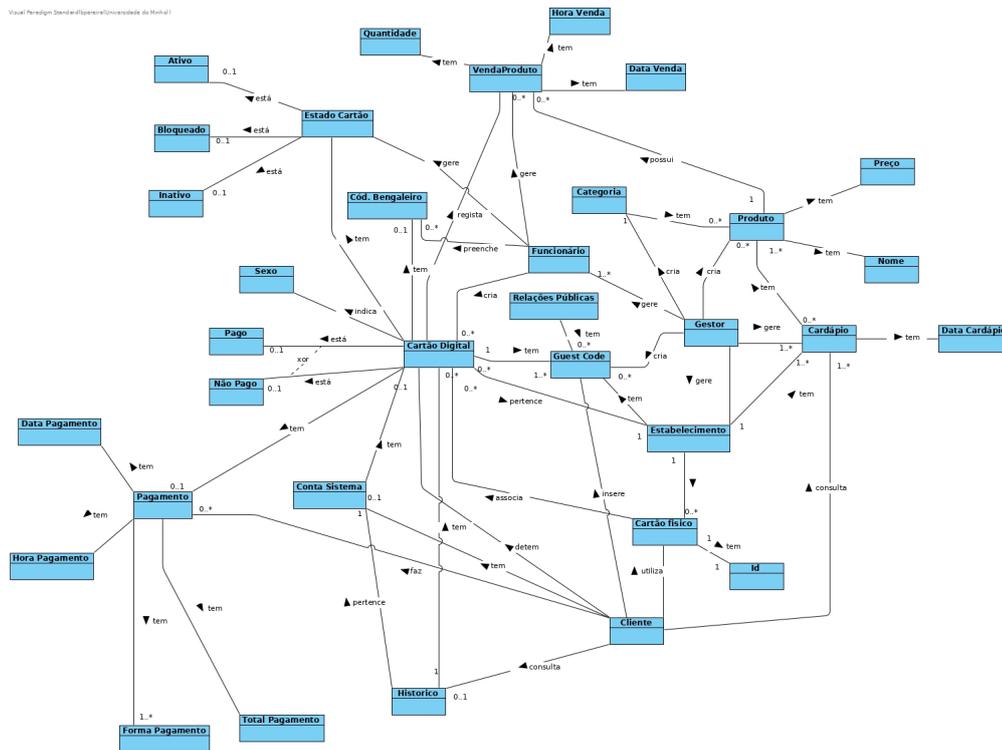


Figura 4.1: Modelo domínio

respetivamente. Este ator tem ainda a capacidade de consultar o estado de um cartão, para ver as bebidas creditadas ou verificar se o cartão já se encontra pago. Finalmente, pode atribuir o estado de pago a um cartão caso um cliente efetue o pagamento no balcão.

O utilizador executa operações de consulta de informação, pagamentos e controlo do cartão. No fim de uma noite, um utilizador tem a possibilidade de pagar pela nossa aplicação, controlar de forma remota o estado do seu cartão e consultar algumas informações de um estabelecimento.

Assim, uma vez que cada ator tem funções muito distintas, cada um deles necessita de uma forma para conseguir interagir com o nosso sistema. Para o utilizador e o funcionário, o tipo de interface mais adequado é uma aplicação móvel que lhes permita utilizarem os seus telemóveis. Como as operações que o gestor realiza requerem a visualização de uma quantidade moderada de dados, a nossa escolha foi uma aplicação web.

Para finalizar o modelo, o componente que nos falta é um servidor. Uma vez que as interfaces sozinhas não respondem às nossas necessidades, é preciso arranjar uma forma de as interligar. Deste modo, desenvolvemos um servidor seguindo uma arquitetura em micros-serviços capaz de dar resposta aos nossos problemas.

5

Estrutura aplicacional

Em termos de estrutura aplicacional, optamos por uma arquitetura de micro-serviços.

Visto que estamos a falar de uma sistema web, optamos por usar micro-serviços de forma a repartir os processos por vários serviços diferentes cada um com o seu propósito bem definido.

Uma vantagem associada ao uso da arquitetura de micro-serviços relativamente à arquitetura monolítica assenta em facilitar o desenvolvimento e distribuição do trabalho por diferentes membros da equipa, o que permite um desenvolvimento mais rápido e posteriormente uma integração mais rápida das mudanças realizadas nos componentes individuais.

Similarmente, permite o teste dos mesmos antes da sua entrada em produção e conduz a uma maior resiliência do sistema permitindo que a falha de componentes individuais não conduza a uma falha total do sistema.

Outras vantagens desta abordagem assentam na possibilidade de uso de diferentes tecnologias no processo de desenvolvimento dos componentes de acordo com requisitos, conhecimentos das equipas responsáveis pelos seus desenvolvimento e o ambiente em que estas possam ser corridas.

Num modo geral, a estrutura aplicacional que obtivemos pode ser representada pela imagem em baixo:

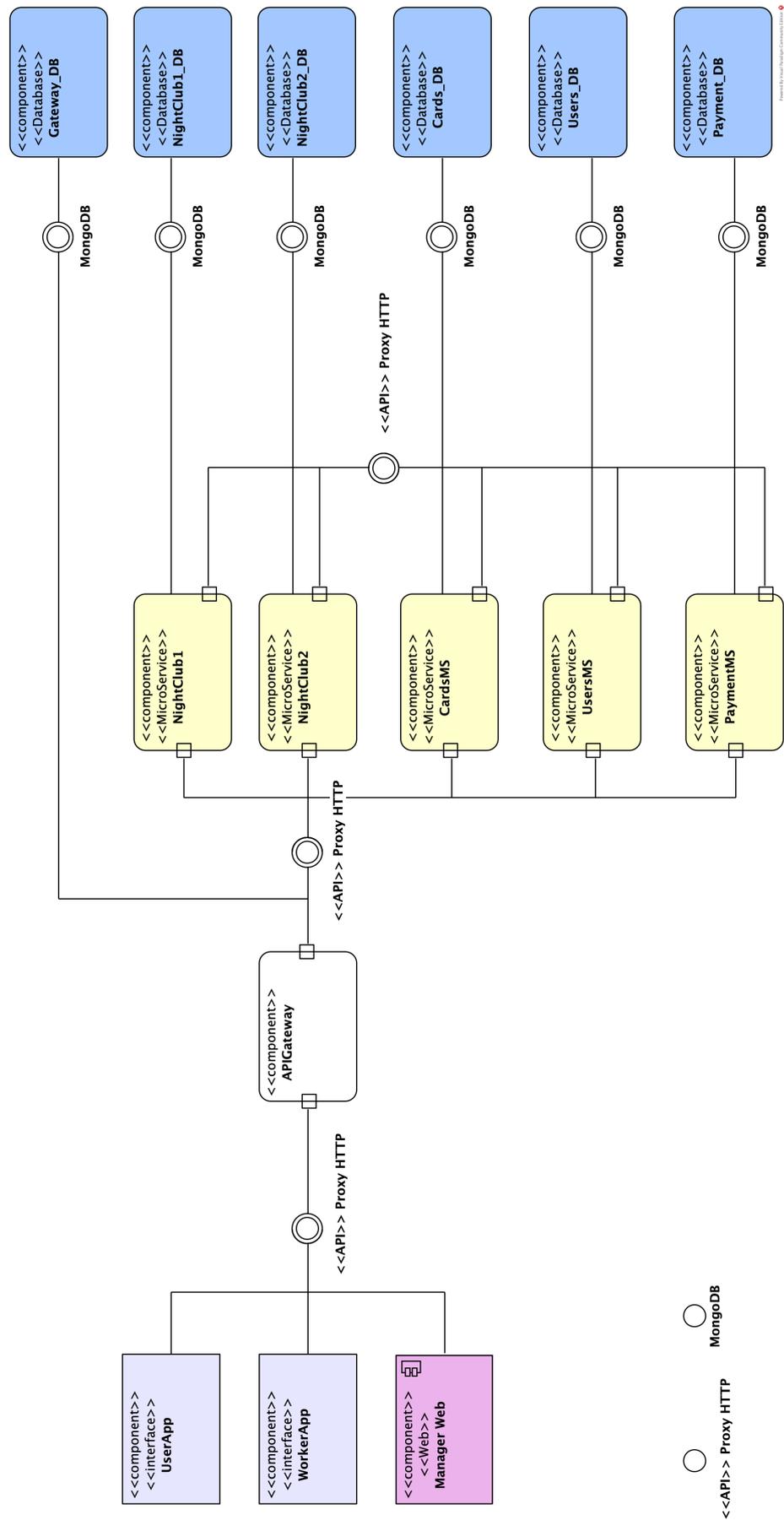


Figura 5.1: Arquitectura

O nosso sistema pretende solucionar problemas que os consumidores de estabelecimentos noturnos encontram quando frequentam esses mesmos locais. Para tal, vamos precisar de três componentes para as diferentes entidades que interagem no nosso sistema. Assim, iremos ter uma aplicação *mobile* para os consumidores - **UserApp** - que suportará todas as funcionalidades que dizem respeito a um consumidor, sendo esta a principal componente do nosso sistema; uma aplicação *mobile* para os funcionários do estabelecimento - **WorkerApp** - que será o novo sistema dos funcionários para registar entradas e produtos consumidos por um cliente; e, por fim, uma interface web - **Manager Web** - que permitirá a um gestor de um estabelecimento fazer toda a gestão dos funcionários, cartões e outras funções que lhe são inerentes.

Por outro lado, as funcionalidades do nosso sistema foram divididas em 4 componentes diferentes, sendo as quatro micro-serviços:

- **UsersMS** - Micro-Serviço que trata de todas as operações relacionadas com os utilizadores (de qualquer tipo) como por exemplo: *registo, login, alteração de dados*, etc.
- **PaymentMS** - Micro-Serviço que trata das operações relacionadas com os pagamentos, isto é, gera referências, processa pedidos de pagamentos, etc.
- **CardsMS** - Micro-Serviço que gere operações relacionadas com os cartões físicos, como a criação dos mesmos, editar informação, etc.
- **NightClubMS** - Micro-Serviço que gere operações relacionadas a um estabelecimento. Isto inclui funcionalidades com o cartão digital, CRUD dos produtos, categorias e menus e ainda funcionalidades que incluem RP's.

A quarta componente diz respeito às funcionalidades que envolvem o estabelecimento - **NightClub**. Esta componente será replicada por cada estabelecimento que adira ao nosso sistema e os seus dados e informações serão exclusivos a apenas ele.

A decisão de replicar esta componente deu-se pelo facto da informação utilizada por este micro-serviço ser relativa para cada um dos estabelecimentos e esta decisão, apesar de possuir custos acrescidos de desenvolvimento, traz vantagens consideráveis, aumentando a robustez do sistema em termos de segurança e fiabilidade.

Em termos de segurança, as vantagens derivam da isolamento dos dados de cada estabelecimento, isto é, cada um tem uma base de dados associada e os seus dados estão privados a essa instância do micro-serviço; em termos de fiabilidade, as vantagens devem-se ao facto de que, para além do **APIGateway**, este é o micro-serviço que mais afeta a experiência dos consumidores e estabelecimentos, devido ao facto de que este, contrariamente aos micro-serviços **PaymentMS** e **CardsMS**, é imprescindível durante a maioria do uso da aplicação, o que o torna susceptível a possíveis falhas que poderiam levar a uma interrupção do serviço total da

aplicação. Desta forma no caso de uma replicação deste micro-serviço, as restantes não ficarão comprometidas.

Os restantes serviços não serão replicados para cada um dos estabelecimentos pois possuem dados pertinentes ao funcionamento da aplicação em vários estabelecimentos, estes sendo os cartões, que serão manufaturados pela equipa do *Nite*. Apesar de serem criados para um estabelecimento específico, ao guardar a sua informação numa localização centralizada, facilitamos a procura do cartão virtual associado ao mesmo no momento da sua leitura pelos consumidores.

No caso do **PaymentMS**, a criação de um único componente facilita o acesso ao histórico de pagamentos de um utilizador. O mesmo se dá para o **UserMS**, em que a informação de um utilizador é relevante para a sua experiência nos vários estabelecimentos integrados no sistema, bem como na sua experiência pós uso da aplicação.

De forma a conectarmos as interfaces ao *back-end*, criamos uma outra componente - **API-Gateway**. Esta componente é responsável por redirecionar os pedidos provenientes da interface para o micro-serviço correto. Visto que uma das componentes será replicada, foi necessário acrescentar mecanismos para quando houver pedidos para um estabelecimento, descobrirmos qual o estabelecimento em questão e redirecioná-lo corretamente.

Cada micro-serviço terá uma base de dados associada, sendo que cada estabelecimento terá uma base de dados própria, e a **API-Gateway** terá de ter uma base de dados associada também para poder usufruir do mecanismo de redirecionamento das discotecas.

6

Interface

Neste capítulo serão apresentadas as interfaces das aplicações móveis e da plataforma web desenvolvidas até ao momento. As aplicações móveis foram implementadas com **React Native** enquanto que a aplicação *web* foi desenvolvida usando **VueJS**.

6.1 App do Consumidor

De modo a poder utilizar o nosso produto, o consumidor de um dado estabelecimento necessita de utilizar a aplicação móvel. Após efetuar a associação do cartão à sua conta, o consumidor poderá:

- Consultar o cardápio do estabelecimento;
- Verificar o que consumiu;
- Pagar o seu cartão;
- Consultar e alterar o estado do cartão.
- Inserir o código de uma *guestlist*.

Para além disso, desde que tenha a aplicação poderá a qualquer momento ver e fazer alterações ao seu perfil, bem como consultar o seu histórico.

6.1.1 Login e Registo

Para utilizar a aplicação, o consumidor necessita de se registar na mesma. No momento do login, caso não se recorde da palavra-passe, poderá inserir o seu email de modo a efetuar o *reset* da mesma, como podemos observar na Figura 6.3.

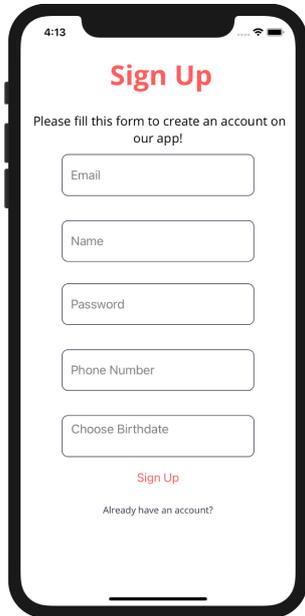


Figura 6.1: Registo de Utilizador

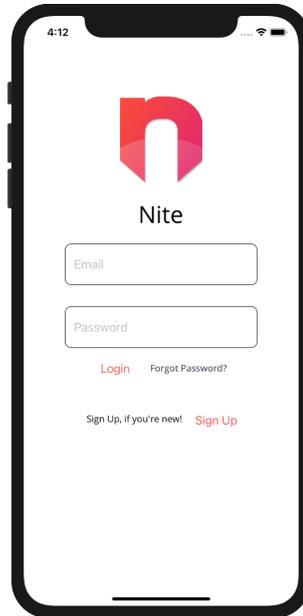


Figura 6.2: Login do Utilizador

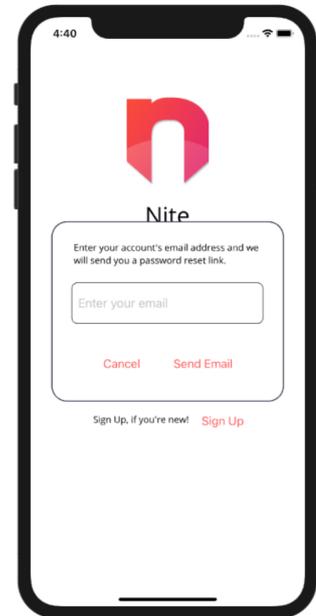


Figura 6.3: Reset da Password

6.1.2 Home

Ao efetuar o login na aplicação, o utilizador é redirecionado para o ecrã inicial, onde é possível fazer scan do cartão físico e, assim, associar o correspondente cartão digital à sua conta. Após a associação, são lhe apresentadas um número de opções, presentes na Figura 6.5, podendo dirigir-se para o pagamento, para o cardápio do estabelecimento, ou adicionar o código de uma *guestlist*.

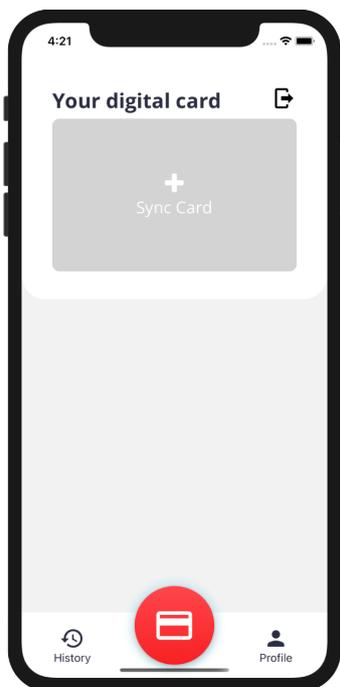


Figura 6.4: Página Home

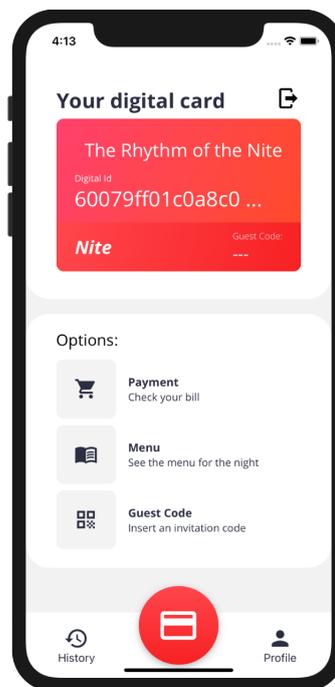


Figura 6.5: Página Home após (2)

6.1.3 Consultar e Alterar Perfil

Independentemente de ter um cartão digital associado à sua conta ou não, é sempre possível para o consumidor consultar o seu perfil e efetuar as alterações que ache necessárias. Podemos observar nas Figuras 6.6 a 6.8 o modo de consulta, o modo de edição do perfil e ainda a alteração da palavra-passe, respetivamente.

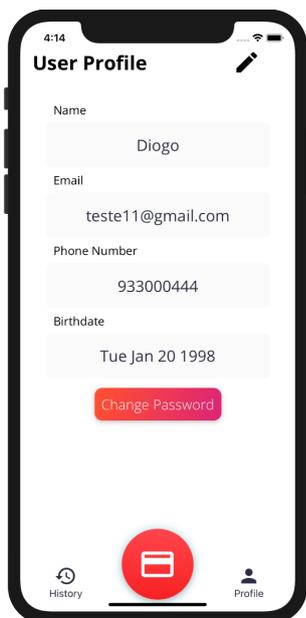


Figura 6.6: Consultar Perfil

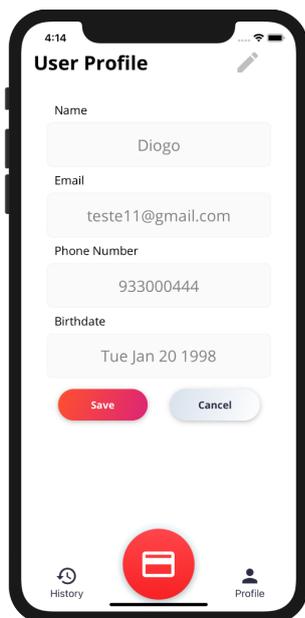


Figura 6.7: Editar Perfil

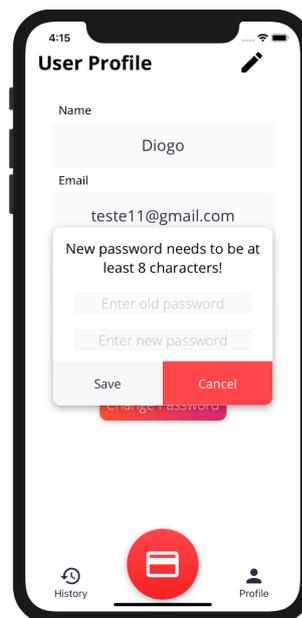


Figura 6.8: Alterar Password

6.1.4 Consultar Histórico

Tal como no perfil, a qualquer momento, quer tenha um cartão associado ou não, o consumidor pode consultar o seu histórico. A Figura 6.9 corresponde ao histórico do utilizador mostrando, do mais recente para mais antigo, para cada saída, uma fatura com o que consumiu e o que pagou no estabelecimento.

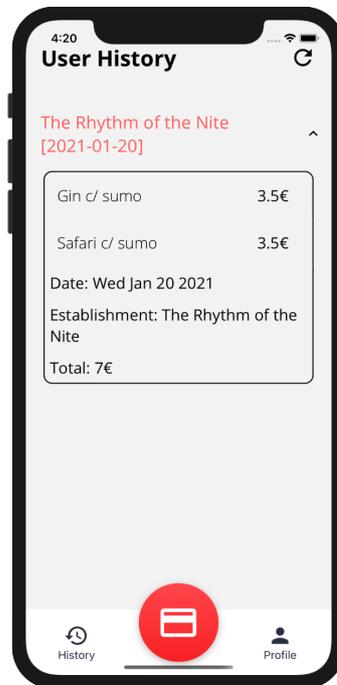


Figura 6.9: Consulta do Histórico

6.1.5 Consultar Consumo e Efetuar Pagamento

Quando escolhe uma das opções do ecrã inicial, é apresentada ao consumidor uma barra na parte inferior do ecrã, com as operações que pode realizar após associar um cartão à sua conta, podendo regressar ao ecrã inicial a qualquer momento.

Selecionando a opção de consulta da conta, *Billing*, o consumidor é capaz de averiguar o que consumiu, quanto tem de pagar e proceder ao pagamento.

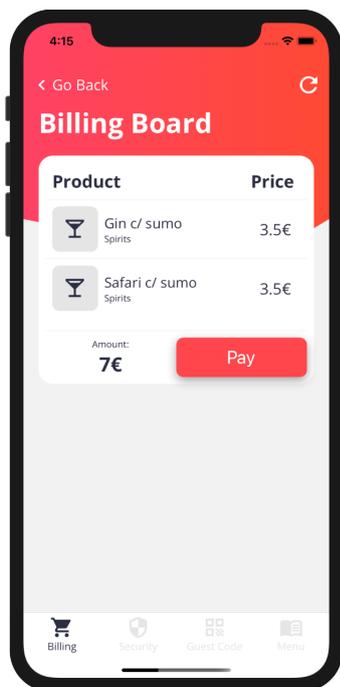


Figura 6.10: Consulta dos consumos do cartão

Nas Figuras 6.11 a 6.12 podemos observar os ecrãs referentes às opções de pagamento que o consumidor poderá usar, bem como o pagamento por cartão de crédito. De notar que, neste momento, apenas está disponível a opção de pagamento por cartão de crédito.

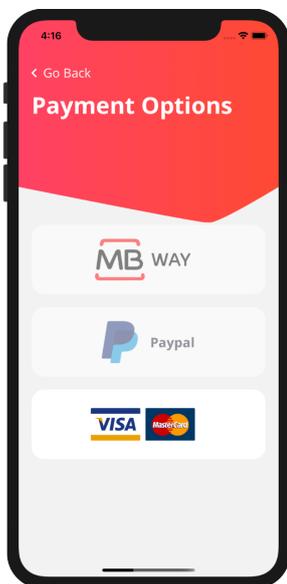


Figura 6.11: Opções de Pagamento

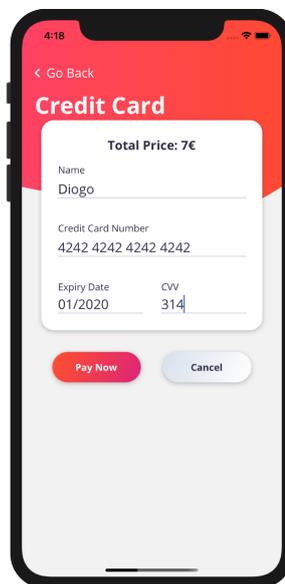


Figura 6.12: Pagamento por Cartão de Crédito

6.1.6 Estado do Cartão

O painel de segurança apresenta ao consumidor os dados do cartão, indicando o estado do mesmo. Aqui, o consumidor poderá alternar o estado do cartão entre bloqueado ou ativo, tendo assim controle sobre o seu cartão a qualquer momento.

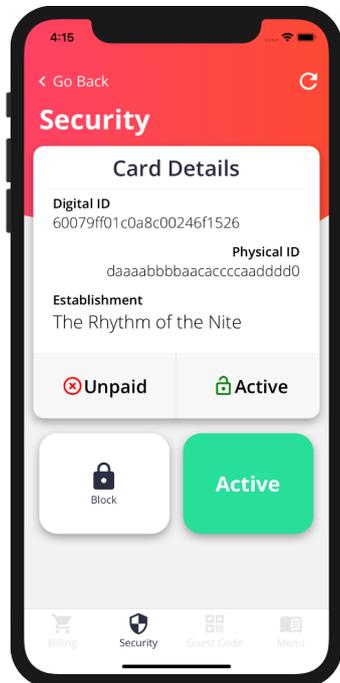


Figura 6.13: Cartão ativo

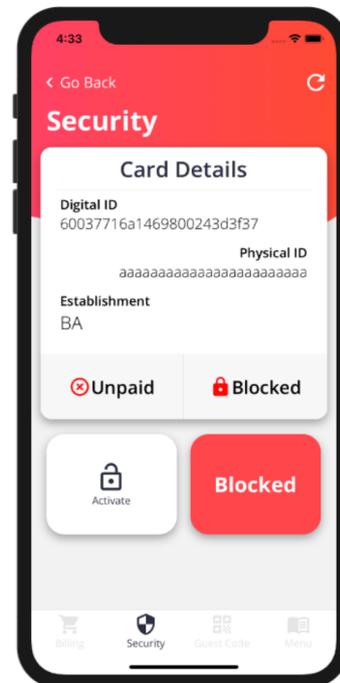


Figura 6.14: Cartão bloqueado

6.1.7 Adicionar Código de GuestList

Caso o consumidor esteja na lista de convidados de um RP, terá acesso ao código desse RP, podendo inserir o mesmo na aplicação. Após adicionar o código, esse ficará associado a esse cartão digital, não sendo possível ao consumidor alterar ou inserir novo código.

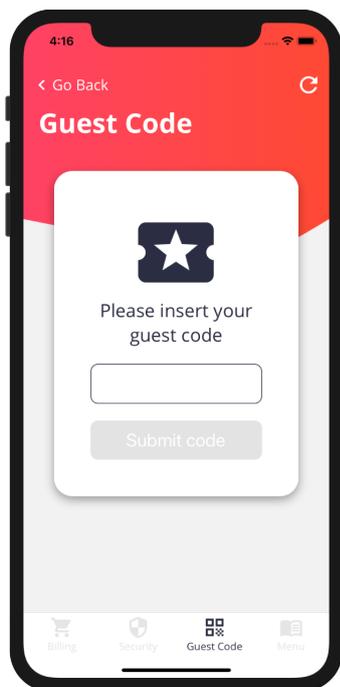


Figura 6.15: Adicionar código de uma guestlist

6.1.8 Cardápio do Estabelecimento

No ecrã referente ao cardápio do estabelecimento, é possível que o consumidor consulte os produtos das várias categorias presentes no cardápio.

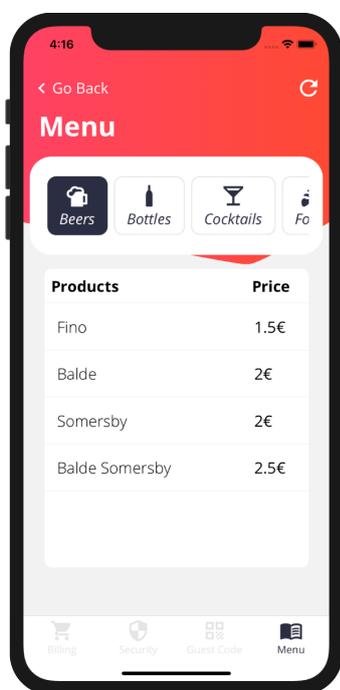


Figura 6.16: Cardápio do estabelecimento

6.2 App do Funcionário do Estabelecimento

Os funcionários de um dado estabelecimento que adote o nosso sistema necessitam de utilizar a aplicação móvel destinada a eles. Esta aplicação possibilita as seguintes ações por parte de um funcionário:

- Ativar um cartão de consumo para que possa ser utilizado por um consumidor;
- Creditar uma bebida num cartão de consumo ativo;
- Bloquear/Desbloquear um cartão de consumo;
- Alterar o estado de um cartão de consumo para pago;
- Desassociar um cartão físico após estar pago.

Para além disso, poderá a qualquer momento ver e fazer alterações ao seu perfil.

6.2.1 Login

Nas Figuras 6.17 a 6.19 podemos ver as interfaces relacionadas com o login do funcionário, o aviso em caso de insucesso no mesmo e o *reset* da palavra passe em caso de esquecimento. Neste último caso é enviado um email para o funcionário para que este possa escolher uma nova palavra-passe.

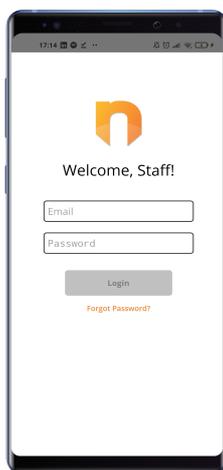


Figura 6.17: Login do Utilizador

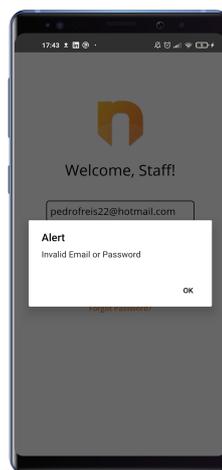


Figura 6.18: Email ou Password inválidos

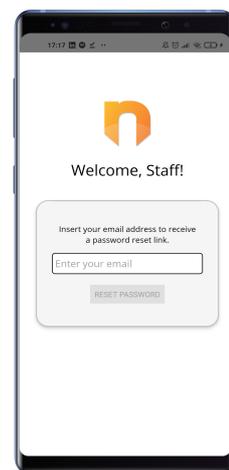


Figura 6.19: Reset da Password

6.2.2 Menu Principal

Na Figura 6.20, podemos ver o menu principal da aplicação que surge após o funcionário estar devidamente autenticado. Este menu contempla as diferentes funcionalidades que cada funcionário necessita de desempenhar num estabelecimento que adote o nosso sistema. Estas funcionalidades serão explicadas com melhor detalhe nas secções que se seguem.



Figura 6.20: Menu Principal

6.2.3 Ativar Cartão

As Figuras 6.21 a 6.24 correspondem às interfaces apresentadas após se seleccionar o botão *Activate* no menu principal. Depois de seleccionar o sexo do consumidor, o funcionário pode aproximar o cartão de consumo ao telemóvel para o mesmo ser lido e ser feito um pedido para o ativar. Tanto nesta funcionalidade como nas próximas, podemos verificar que, após o pedido ser concluído, o funcionário é redireccionado, não para o menu inicial, mas sim de volta à interface inicial da feature seleccionada. Fizemos esta escolha para o *flow* da aplicação, tendo em conta que cada funcionário, geralmente, tem uma tarefa específica que repete continuamente ao longo da noite.

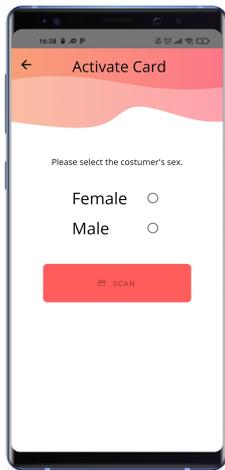


Figura 6.21: Selecionar sexo do consumidor

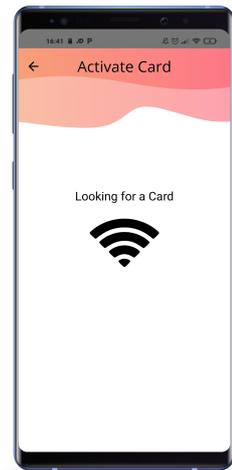


Figura 6.22: Procura de cartão NFC

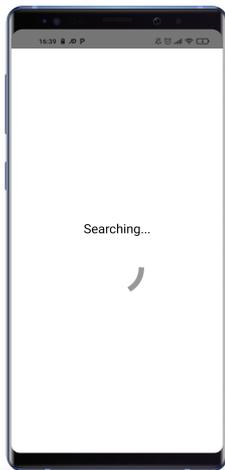


Figura 6.23: Espera pela conclusão do pedido



Figura 6.24: Confirmação de sucesso da ação

6.2.4 Cardápio de Produtos

Quando selecionamos o botão *Menu* no menu principal é feita a leitura do cartão NFC, e, de seguida, é apresentado o menu ativo no estabelecimento, como podemos verificar na Figura 6.27. Na Figura 6.29 observamos o pedido de confirmação de creditação de uma bebida no cartão lido, que surge quando um funcionário seleciona uma das bebidas do cardápio.



Figura 6.25: Procura de cartão NFC

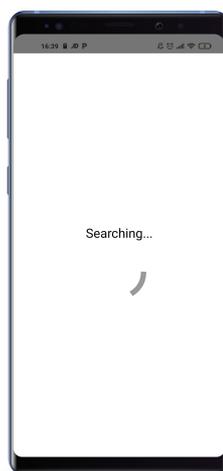


Figura 6.26: Espera pelo pedido



Figura 6.27: Cardápio do Estabelecimento



Figura 6.28: Creditar uma bebida



Figura 6.29: Confirmação da acção

6.2.5 Bloquear Cartão

Quando seleccionado o botão *Lock* no menu principal, o estado do cartão é alterado para bloqueado ou desbloqueado, conforme o estágio prévio. Como podemos ver na Figura 6.32, temos a confirmação por meio de uma notificação que surge no fundo do ecrã, tal como nas outras funcionalidades. Como nas restantes operações, o bloqueio do cartão é precedido pela leitura do cartão NFC.



Figura 6.30: Bloquear o cartão

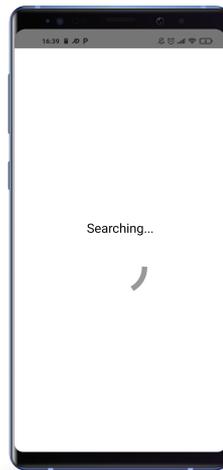


Figura 6.31: Espera pelo pedido



Figura 6.32: Resultado obtido

6.2.6 Dados do Cartão

Na seleção do botão *CardData* é novamente ativado o processo de leitura do cartão NFC, e de seguida é apresentada a informação do mesmo. Se o cartão estiver por pagar, surge a interface que podemos observar na imagem Figura 6.35. Após o cartão ser pago, surge a interface visível na Figura 6.38. Desta forma, dependendo da interface que surge, um funcionário consegue saber se o cartão está pago ou não.



Figura 6.33: Leitura cartão NFC

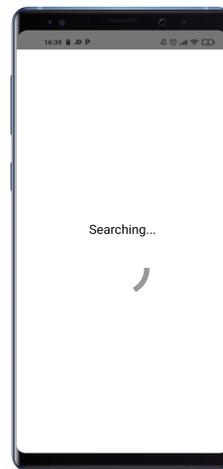


Figura 6.34: Espera pelo pedido

Selecionando o botão de pagamento, surge um *modal* de confirmação. Após confirmação, o estado do cartão muda para pago. Se o funcionário efetuar novamente leitura deste cartão,

pode seleccionar o botão "reset" para desassociar o cartão físico do cartão virtual do consumidor. Desta forma, o cartão fica pronto para ser usado novamente.

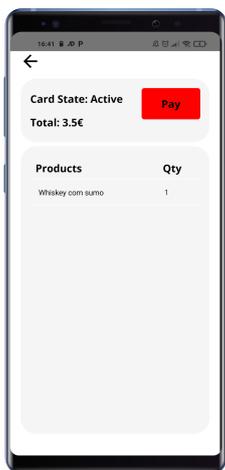


Figura 6.35: Cartão por pagar

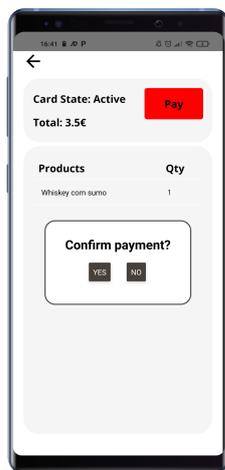


Figura 6.36: Confirmar pagamento

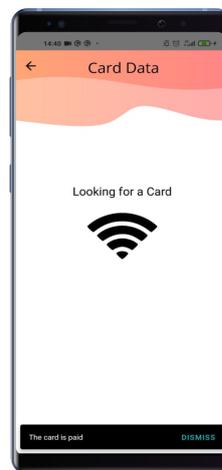


Figura 6.37: Confirmação da operação

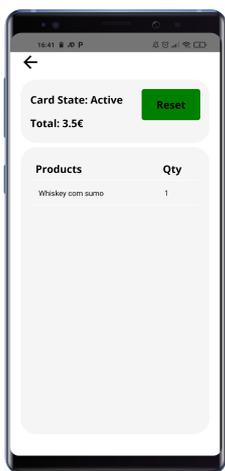


Figura 6.38: Cartão pago

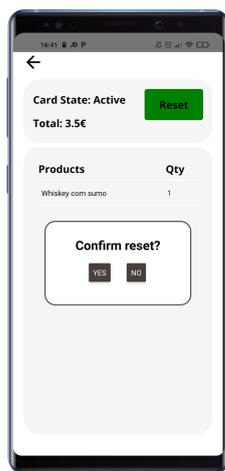


Figura 6.39: Confirmar reset



Figura 6.40: Confirmação

6.2.7 Perfil do Utilizador

Seleccionando o ícone do perfil localizado no canto superior direito do menu principal, é apresentado o perfil do funcionário, como podemos ver na Figura 6.41. O funcionário tem a opção de editar o perfil, clicando no ícone em forma de lápis, e de alterar a sua password, clicando no botão para esse efeito.

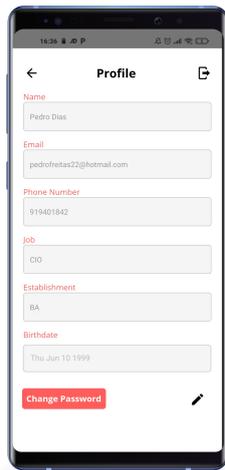


Figura 6.41: Perfil do utilizador

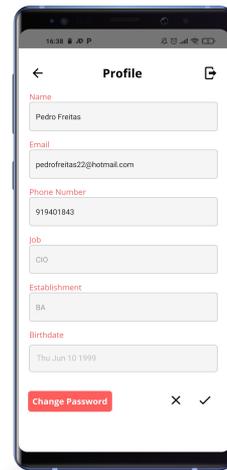


Figura 6.42: Editar perfil

6.3 *Backoffice* da aplicação (*WEB App*)

Com a formulação das possibilidades oferecidas a um consumidor e a um funcionário, resta apresentar todas as restantes logísticas, que serão do encargo de um gestor, tais como metodologias CRUD (*Create, Read, Update and Delete*) de produtos, cardápios, funcionários e relações públicas. Além disso, a funcionalidade de apresentação de estatísticas relativas aos relações públicas do estabelecimento também é fornecida, bem como a capacidade de o gestor editar os seus dados pessoais, endereço de email e palavra-passe. A funcionalidade de recuperação de palavra passe também é fornecida. Note-se que, existe a possibilidade de se registar *online* para o estabelecimento configurado no *backend*.

De forma geral temos:

- Metodologias CRUD para:
 - Funcionários;
 - Produtos;
 - Cardápios;
 - Relações Públicas;

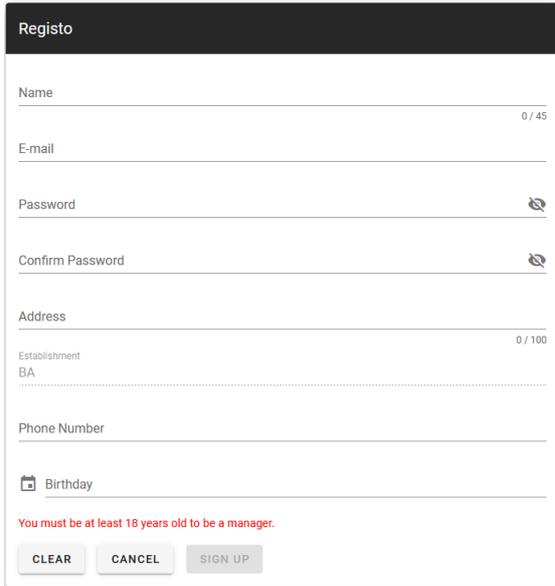
Para elaborar a interface da *web app* escolheram-se tecnologias que permitissem desenvolver a mesma, com vista na usabilidade conforme as características do utilizador específico, neste caso, um gestor do estabelecimento. Numa primeira abordagem, de forma informal, teve-se em conta as necessidades (requisitos) de usabilidade de um gestor, tendo em conta o tipo de utilizador (inexperiente) com a função (necessidade de eficácia e de realização de tarefas de forma inequívoca). Exemplos de elementos ou funcionalidades da interface são: capacidade de fazer *undo*, *tooltips*, facilidade e rapidez de execução, confirmação de ações e indicadores do progresso de pedidos ao servidor, como também indicação de alertas de erro e sucesso.

Com efeito, para o desenvolvimento da interface, escolheu-se a *framework Vuejs*, a biblioteca *Vuetify* para reutilização de componentes e a biblioteca *axios* para comunicação com o *backend*. O *Vuetify* permite usar componentes já testados tendo em vista a usabilidade e funcionalidade, como também usa os princípios do *Google Material Design*.

6.3.1 *Login* e Registo

Como se mencionou, de forma a poder usufruir do produto em questão, um gestor terá de se registar. Para isso existe um formulário de registo com os campos necessários para tal. O ecrã de *login* é fornecido para autenticação perante o sistema.

Figura 6.43: Registo de Gestor



Registo

Name 0 / 45

E-mail

Password 👁

Confirm Password 👁

Address

Establishment 0 / 100

BA

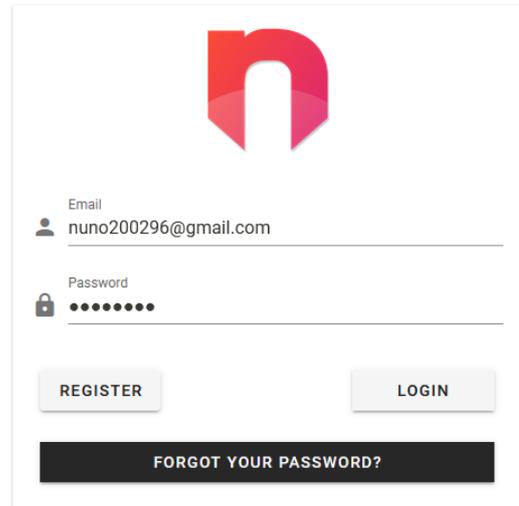
Phone Number

📅 Birthday

You must be at least 18 years old to be a manager.

CLEAR CANCEL SIGN UP

Figura 6.44: Login do Gestor





Email nuno200296@gmail.com

Password ●●●●●●

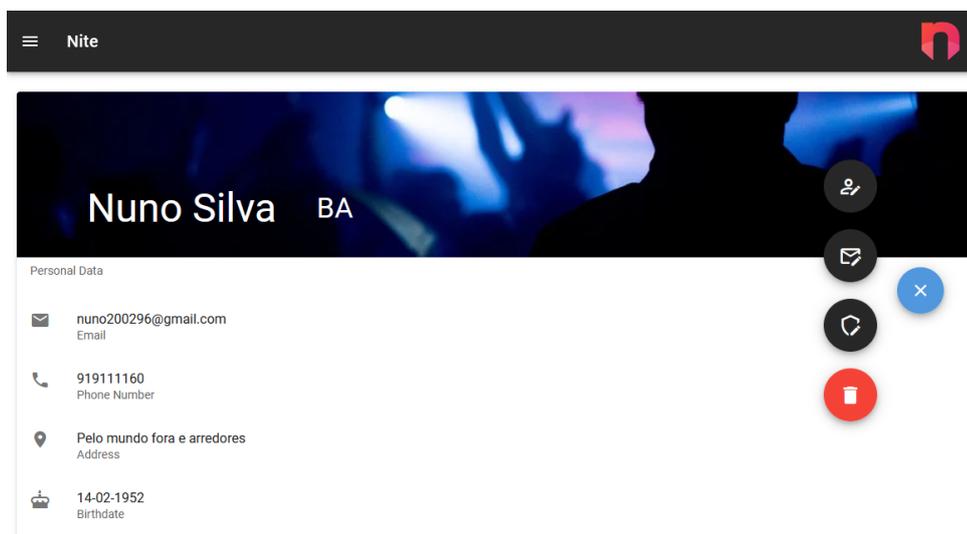
REGISTER LOGIN

FORGOT YOUR PASSWORD?

6.3.2 Home Page

Ao efetuar o *login*, um gestor é redireccionado para a sua área pessoal, onde lhe são apresentadas as suas informações, como nome, email, data nascimento, morada e estabelecimento ao qual está associado. Possibilita ainda a alteração da sua password e email de forma rápida, como também a remoção da sua conta.

Figura 6.45: Área Pessoal de um Gestor



☰ Nite 

Nuno Silva BA

Personal Data

-  nuno200296@gmail.com
Email
-  919111160
Phone Number
-  Pelo mundo fora e arredores
Address
-  14-02-1952
Birthdate



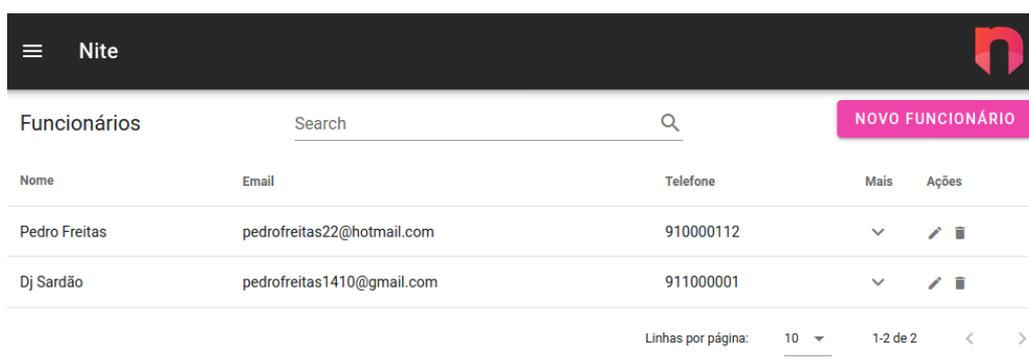




6.3.3 Funcionários

Na área de funcionários é apresentada uma tabela, na qual se encontram listadas informações destes, como o seu nome, email e telefone. A esta se junta metodologias CRUD, ou seja, a capacidade de adicionar, editar, remover funcionários, como também ordenação, pesquisa de termos e paginação da tabela. Note-se que estas últimas funcionalidades são transversais para todos os recursos, i.e., em qualquer tabela é possível ordenar, pesquisar por um termo e navegar pelo demais itens da tabela.

Figura 6.46: Funcionário



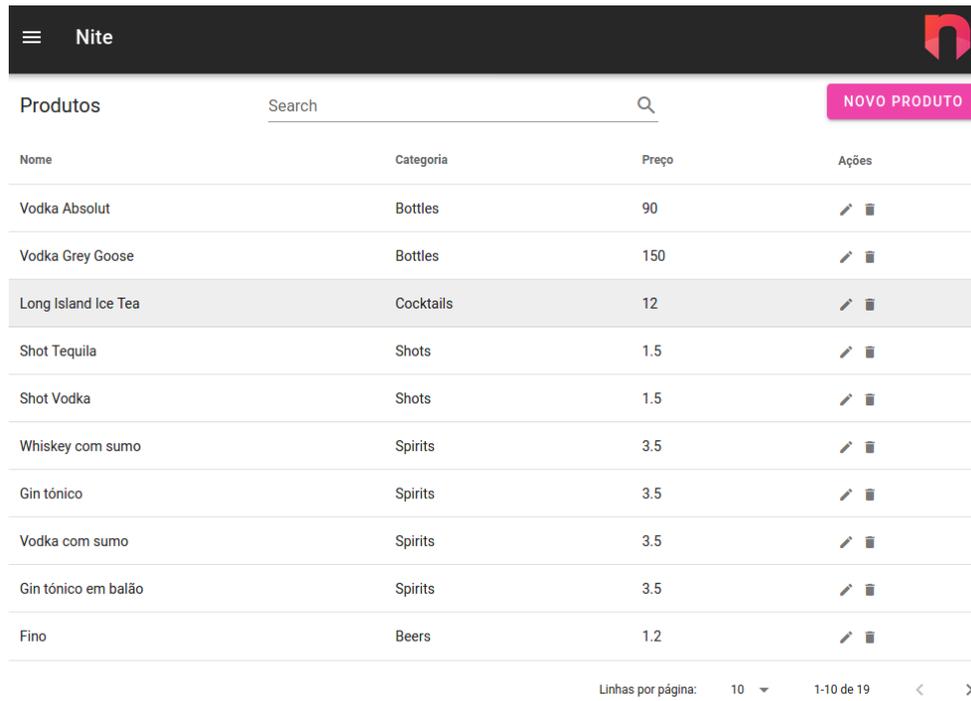
Nome	Email	Telefone	Mais	Ações
Pedro Freitas	pedrofreitas22@hotmail.com	910000112	▼	✎ 🗑
Dj Sardão	pedrofreitas1410@gmail.com	911000001	▼	✎ 🗑

Linhas por página: 10 1-2 de 2 < >

6.3.4 Produtos

A área destinada para os produtos, apresenta a mesma linha de pensamento e capacidade da área apresentada anteriormente. Contudo nesta são apresentadas informações como o nome do produto, a categoria ao qual este se enquadra e o preço.

Figura 6.47: Produtos



Nome	Categoria	Preço	Ações
Vodka Absolut	Bottles	90	 
Vodka Grey Goose	Bottles	150	 
Long Island Ice Tea	Cocktails	12	 
Shot Tequila	Shots	1.5	 
Shot Vodka	Shots	1.5	 
Whiskey com sumo	Spirits	3.5	 
Gin tónico	Spirits	3.5	 
Vodka com sumo	Spirits	3.5	 
Gin tónico em balão	Spirits	3.5	 
Fino	Beers	1.2	 

Linhas por página: 10 1-10 de 19

6.3.5 Cardápios

Relativamente à secção destinada para os Cardápios, aqui será apresentado a um utilizador, duas tabelas. Uma listando todos os cardápios disponíveis, e indicando qual o ativo de momento. E a segunda, destina-se à apresentação dos produtos presentes neste menu, de uma forma semelhante aquela que são apresentados os produtos na sua área.

Figura 6.48: Cardápios

The screenshot shows the 'Nite' application interface. It features a dark header with a menu icon and the text 'Nite', and a red logo on the right. Below the header, there are two main sections:

- Menus:** A table with columns 'Selection', 'Active', 'Name', 'Actions', and 'Description'. It lists four menu items: 'Ladies Night' (active), 'Cuban Night', 'Holden & Zimpel', and 'Marcel Dettmann DJ'. A 'NEW MENU' button is visible in the top right of this section. Below the table is a pagination control showing 'Rows per page: 5' and '1-4 of 4'.
- Menu Products:** A table with columns 'Name', 'Category', and 'Price (€)'. It lists ten products under the 'Ladies Night' menu, including 'Vodka Grey Goose', 'Long Island Ice Tea', 'Shot Tequila', 'Shot Vodka', 'Whiskey com sumo', 'Gin tónico', 'Vodka com sumo', 'Gin tónico em balão', 'Fino', and 'Vodka Absolut'. A pagination control at the bottom shows 'Rows per page: 10' and '1-10 of 10'.

6.3.6 Relações Públicas

Tendo em conta o mercado ao qual o produto final se destina, este engloba um novo tipo de colaborador para além do usual, os Relações Públicas. A área destinada às informações destes, segue a mesma linha de pensamento da área para os funcionários, contudo é apresentado ainda um atributo relativo aos códigos de RP, um identificativo único de um colaborador deste tipo.

Figura 6.49: RP

The screenshot shows the 'Nite' application interface for 'Public Relations'. It features a dark header with a menu icon and the text 'Nite', and a red logo on the right. Below the header, there is a table with columns 'Name', 'PR Shortcode', 'Phone Number', 'Email', and 'Actions'. The table lists six public relations entries. A 'NEW PUBLIC RELATIONS' button is visible in the top right of the table area. Below the table is a pagination control showing 'Rows per page: 10' and '1-6 of 6'.

Name	PR Shortcode	Phone Number	Email	Actions
Diogo Sobral	ds10	917600929	n5y728f7@temporary-mail.net	
Francisco Freitas	ff45	917600928	bc14lysq@temporary-mail.net	
Pedro Freitas	PF10	910000001	email_do_pedro@mail.com	
Nuno Silva	ns25	911111114	email@gmail.com	
Sven Vath	sv33	961111111	xnkd_2he@temporary-mail.net	
Mary Jane	mj0420	916666660	gixzv6e5@temporary-mail.net	

6.3.7 Estatísticas dos Relações Públicas

Numa perspectiva de um gestor, é sempre útil ter estatísticas sobre a noite de um evento. Este espaço será o destino da apresentação destes dados, onde é possibilitada uma filtragem por data e a disponibilização do número de ocorrências de cada código de RP. Ou seja, o número de clientes que cada relações públicas trouxe para o estabelecimento e que utilizou o seu identificativo.

Figura 6.50: Estatísticas RPs

Name	PR Shortcode	Occurrences
Francisco Freitas	ff45	2
Diogo Sobral	ds10	1

6.3.8 Diálogos de edição

Para edição, criação ou remoção de qualquer recurso, é apresentado um *modal* com um formulário com os campos necessários para aquele recurso, como se pode ver nas Figuras 6.51 a 6.53. Note-se que existem elementos destes formulários de forma a facilitar a seleção de produtos, como menus de *dropdown*, *date-pickers*, e *autocomplete*. Para a edição os formulários são preenchidos automaticamente com a informação existente.

Figura 6.51: Criação de novo produto

Figura 6.53: Edição de dados pessoais

Personal Data

Name
Nuno Silva

Address
Pelo mundo fora e arredores

Telephone
919111160

Birthdate
14-02-1952

CANCEL SAVE

Figura 6.52: Edição de novo produto

Edit Product

Name
Vodka Grey Goose

Category
Bottles

Price (€)
150

CANCEL SAVE

A linha de pensamento dos modais presentes nas figuras 6.51 e 6.52, é aplicada ao longo da Web App. Contudo, estes não podem ser aplicados para o caso de um cardápio devido à quantidade de informação que este requiere, assim sendo as figuras 6.54 e 6.55, apresentam a forma de adição e edição de um menu.

Figura 6.54: Criação de novo cardápio

New Menu SAVE

Menu Data

Menu Status
 State: Deactivated

Menu Name
 Name
 Please provide a valid name.

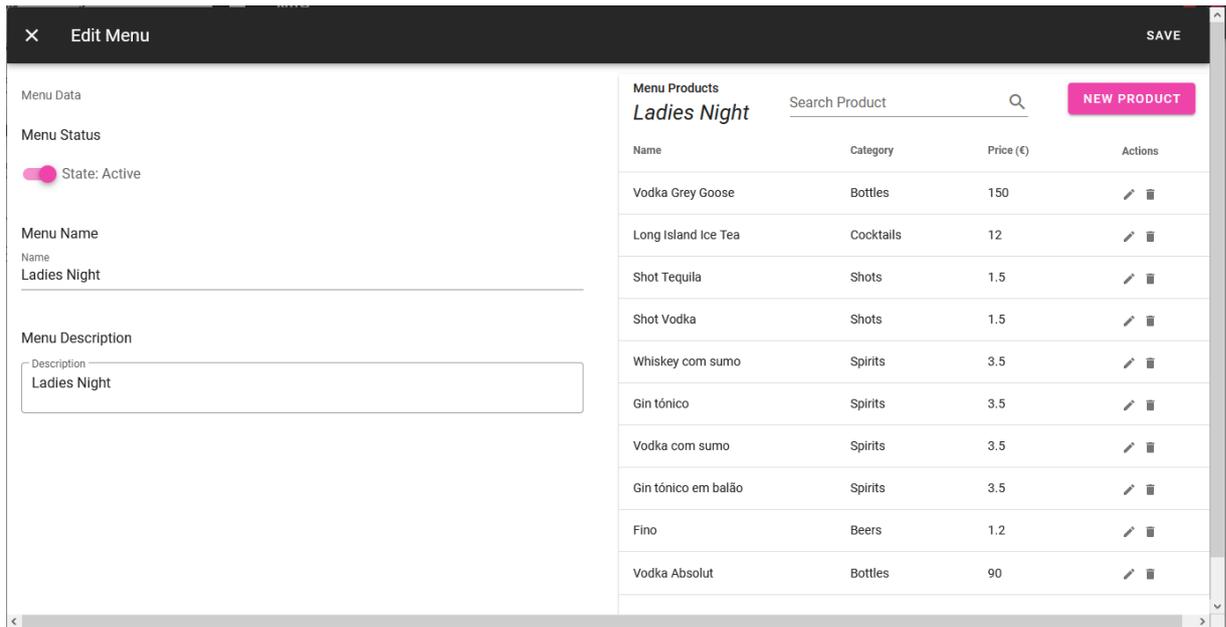
Menu Description
 Description
 Please provide a valid description.

Products

Search Product

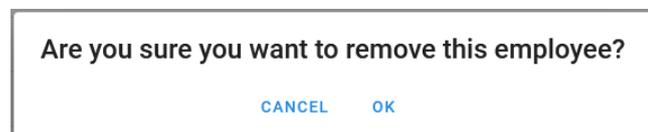
<input type="checkbox"/>	Name	Category	Price (€)
<input type="checkbox"/>	Vodka Grey Goose	Bottles	150
<input type="checkbox"/>	Long Island Ice Tea	Cocktails	12
<input type="checkbox"/>	Shot Tequila	Shots	1.5
<input type="checkbox"/>	Shot Vodka	Shots	1.5
<input type="checkbox"/>	Whiskey Coke	Cocktails	3.5
<input type="checkbox"/>	Gin tonic	Spirits	3.5
<input type="checkbox"/>	Screwdriver	Spirits	3.5
<input type="checkbox"/>	Gin tonic in balloon	Spirits	3.5
<input type="checkbox"/>	Caipirinha	Cocktails	3.5
<input type="checkbox"/>	Caipiroska	Cocktails	4.5

Figura 6.55: Edição de um cardápio



Relativamente à remoção de qualquer tipo dado (e.g. produto, funcionário, menu, etc.), surge um aviso, após a indicação do desejo pretendido. Desta forma é possibilitado a um gestor dar um "passo atrás" nas suas pretensões e manter o estado atual dos dados.

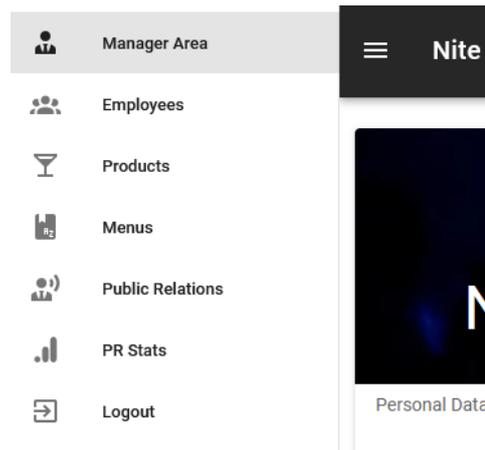
Figura 6.56: Aviso de remoção de um funcionário



6.3.9 Navegação

Um dos elementos importantes para a usabilidade é a barra de navegação entre áreas. O detalhe da mesma pode ser vista na Figura 6.57.

Figura 6.57: Barra de Navegação



6.3.10 Conexão ao *backend*

A comunicação com o *backend* foi efetuada através da biblioteca *axios* para os diversos *endpoints* disponibilizados no *backend* para recursos com o gestor, produtos, cardápios, relações públicas e funcionários, seguindo a convenção do tipo de pedidos - GET, POST, PUT e DELETE - onde os verbos POST, PUT e DELETE serviriam para alterar um único recurso da coleção. A maioria dos GETs serviu para obter a coleção inteira.

Autenticação

Relativamente à autenticação foi exposto um *endpoint* POST para um gestor no *backend*, onde se consegue obter um *token* para autenticação e um *refresh token* para obter um novo *token* no caso do primeiro expirar, ou inválido de outra forma. Note-se que para o *login* há mais informação que se pode obter relativa ao gestor. Todavia, percebeu-se que o armazenamento do *token* no *localStorage* do *browser* seria suficiente, pelo que a informação do gestor seria obtida já com o *token*.

Efetivamente, o *axios* foi configurado para utilizar o *token*, bem como outros cabeçalhos dos pedidos. O mecanismo de autenticação usa o *Vuex* - a *store* para gestão central do estado da aplicação do *Vue*. Todos os pedidos na área autenticada usam o *token* obtido.

De forma a manter a área autenticada segura, i.e., no caso de *logout*, o *token* é removido do *localStorage*. Para impedir o acesso a rotas autenticadas, utilizaram-se guardas nas mesmas, que verificam a condição de autenticado, i.e., que existe um *token* válido. Note-se que a informação do gestor é obtida já com o *token*. Logo, se o *token* for válido, a informação do gestor é obtida sem problemas. Assim, a condição em teste é se existe um *token* e existem dados do utilizador, ou seja existe um *token* válido.

7

Lógica de Controle

Sendo os cartões digitais um dos pontos fulcrais do nosso sistema, neste capítulo descrevem-se quatro funcionalidades focadas neste ponto, nomeadamente, a criação de cartões digitais, a associação de cartões digitais a clientes, a creditação de produtos e o pagamento de cartões.

7.1 Cartões Físicos e Digitais

Neste capítulo existem dois conceitos chave para que o mesmo possa ser compreendido na sua totalidade. No nosso sistema existem cartões físicos e cartões digitais. Os cartões físicos são os cartões NFC que são utilizados pelos consumidores, enquanto que o cartão digital é uma simulação digital de um cartão de consumo em papel onde, tradicionalmente, se acrescentariam bebidas com a escrita à mão.

Os cartões físicos têm dois id: um que o identifica, que é único e nunca muda; outro que faz referência a um cartão digital e que altera sempre que o cartão é ativado por um funcionário. Desta forma, um cartão físico pode ter ou não associado um cartão digital e é utilizado pelo sistema para identificar o cartão digital a que se refere. Quando um cartão físico é reiniciado por um funcionário, o identificador para o cartão digital é destruído.

Os cartões digitais são o componente do nosso sistema que implementa toda a lógica associada à simulação de um cartão de consumo.

7.2 Criação de Cartões Digitais

Como o próprio nome indica, esta *feature* é responsável por criar os cartões digitais de um estabelecimento. Os cartões devem ser criados antes da abertura do estabelecimento para facilitar a associação destes aos consumidores. Na Figura 7.1 é apresentado um diagrama de sequência que demonstra os passos da implementação. A autenticação foi omitida do diagrama para melhorar a sua legibilidade.

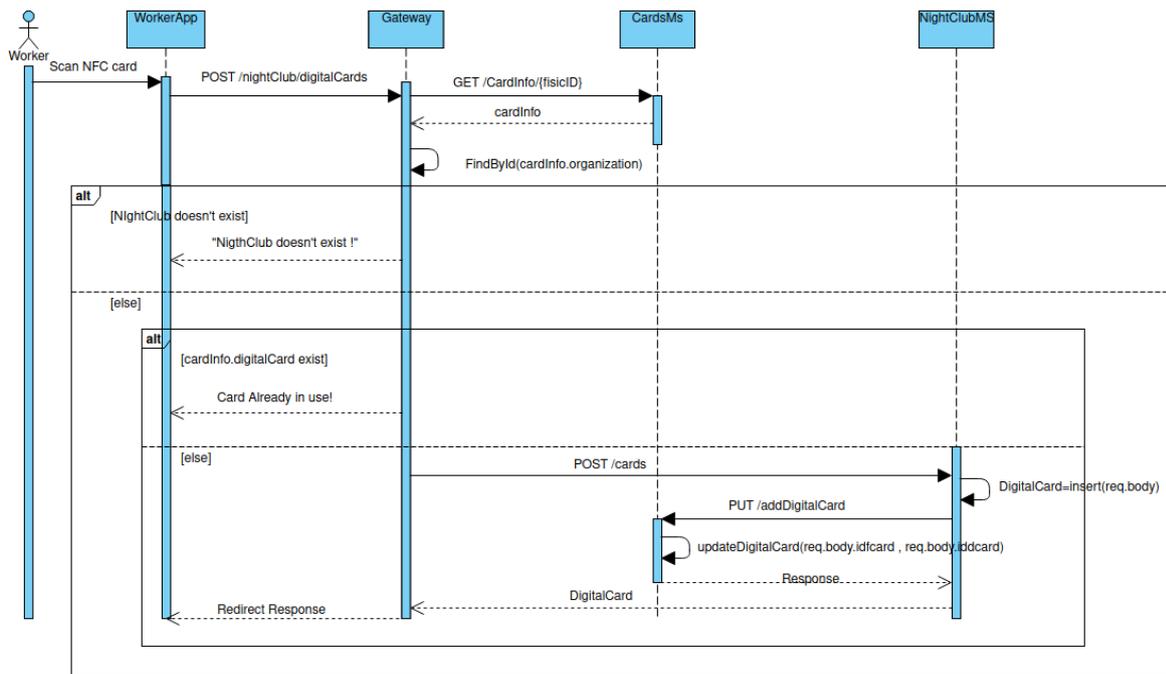


Figura 7.1: Diagrama de sequência de criação de cartões

Este processo inicia-se com o scan do cartão físico por parte do funcionário. O cartão físico contém apenas o seu identificador, que é lido pela aplicação do funcionário com o intuito de realizar um pedido para a criação de um cartão digital associado a esse cartão físico. Antes de ser processado pelo micro-serviço do estabelecimento, o pedido é filtrado no *gateway*. Isto é, o pedido que chega ao *gateway* contém no seu corpo apenas o *id* do cartão físico, no entanto, é necessário identificar o estabelecimento ao qual o cartão pertence e detetar se já existe um cartão digital associado ao cartão físico. Assim sendo, o *gateway* faz um pedido ao micro-serviço dos cartões para pedir as informações do cartão físico. Entre essas informações está o identificador do estabelecimento e, caso exista, o identificador do cartão digital associado.

Depois de obtido o identificador do estabelecimento, é feita uma procura pelo seu *endpoint*. Em caso de insucesso, é imediatamente enviada uma resposta para o funcionário a citar o sucedido; caso contrário, é verificado se existe um cartão digital associado ao cartão físico,

e em caso de sucesso é enviada uma resposta ao funcionário a informar que o cartão digital já existe, no entanto, caso não se verifique esse facto, o *gateway* redirecciona o pedido para o micro-serviço do estabelecimento ao qual o cartão pertence.

No micro-serviço do estabelecimento é criado um cartão digital e de seguida é feito um pedido ao micro-serviço dos cartões para guardar uma referência do cartão digital criado no cartão físico a que este está associado. Por último, é enviado como resposta o novo cartão digital.

7.3 Associar Cartão ao Cliente

Quando um cliente entra no estabelecimento, recebe um cartão físico para efeitos de consumo e, caso tenha a aplicação móvel *Nite* instalada, poderá tirar proveito do nosso sistema, bastando para isso que associe o cartão digital associado ao cartão físico recebido à sua conta. Na Figura 7.2 é apresentado um diagrama de sequência que demonstra os passos da implementação desta *feature*.

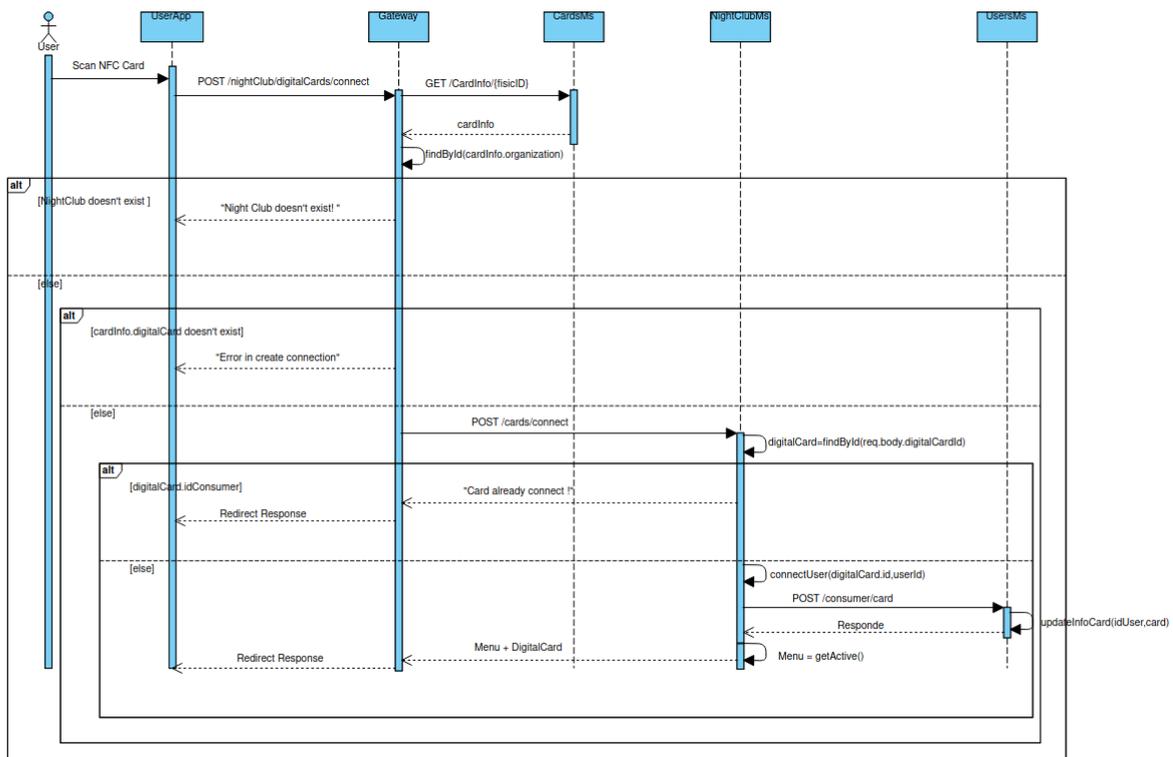


Figura 7.2: Diagrama de sequência de associar um cartão digital

Assim como a *feature* apresentada anteriormente, o processo inicia-se com o scan do cartão, só que desta vez por parte do cliente, dando início à fase de validação do cartão onde é feito

um pedido ao micro-serviço dos cartões para obter as informações do cartão físico. Tal como a *feature* anterior, também é feita uma pesquisa pelo *endpoint* do estabelecimento, e, em caso de sucesso, é verificado se o cartão físico apresenta alguma referência a um cartão digital. Caso não apresente é enviada uma resposta ao cliente a citar o sucedido, caso contrário, o *gateway* redireciona o pedido para o serviço do estabelecimento, onde é feita uma pesquisa pelo cartão digital correspondente ao cartão físico. No caso de não ser encontrado nenhum cartão, é enviado como resposta uma mensagem de erro. Sendo o cartão digital encontrado, é feita a verificação se este já tem um cliente associado e, se for o caso, é enviado como resposta uma mensagem a citar o sucedido. Quando o cartão digital não tem nenhum cliente associado é feita a associação do cliente que fez o pedido, sendo guardado no cartão uma referência ao utilizador e na conta do utilizador é guardada uma referência ao cartão digital para melhorar o desempenho dos pedidos feitos posteriormente. Por último, para evitar um novo pedido por parte do cliente e melhorar a performance, é feita uma pesquisa do menu disponível. Como resposta é enviada a informação do cartão digital bem como a do menu.

7.4 Creditação de Produtos

Uma das grandes *core features* do nosso sistema é a creditação de produtos, uma vez que esta requer a intervenção de todos os utilizadores do sistema. O gestor é responsável pela gestão de produtos e menus, desde a sua criação até ao seu término, os funcionários creditam produtos que estão disponíveis no menu e os clientes consomem produtos e visualizam o menu. Esta *feature* é composta por duas partes: obter a informação do cartão digital e adicionar o produto ao cartão digital.

7.4.1 Obter Informação do Cartão Digital

Como todas as *features* que envolvem o cartão físico, esta também começa com o scan do cartão físico por parte do funcionário com o intuito de creditar a bebida. Na Figura 7.3 podemos visualizar todos os passos para obter a informação do cartão digital.

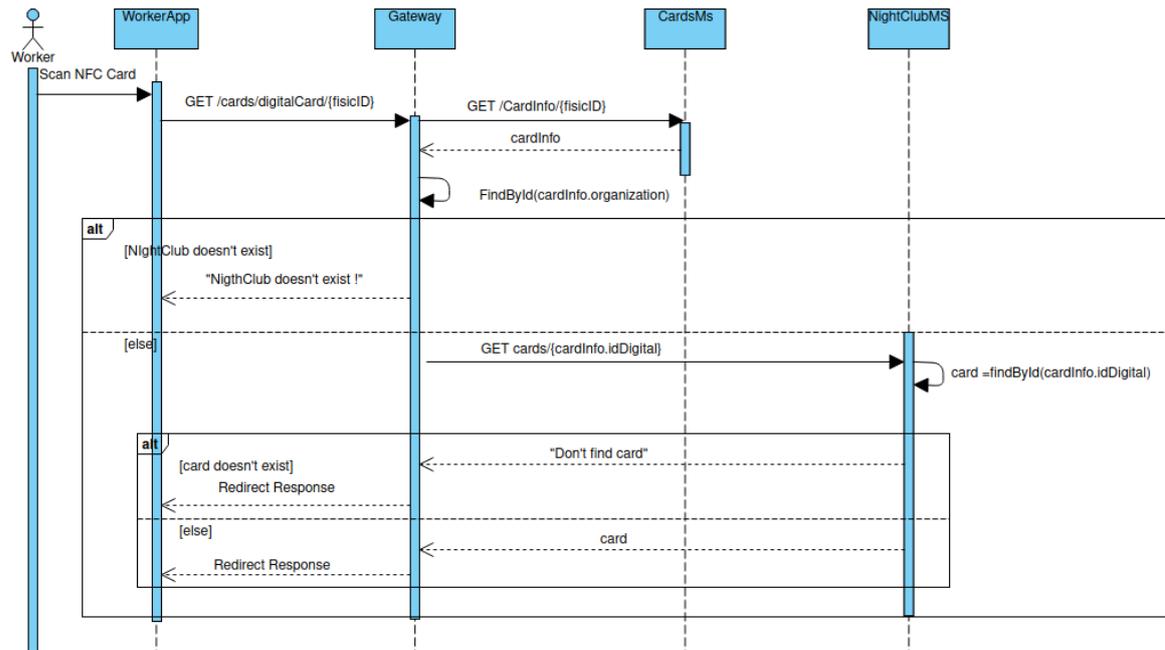


Figura 7.3: Diagrama de sequência de obter informação de cartão digital

Após o scan do cartão físico é feito um pedido para obter informação do cartão digital à qual este tem uma referência. Após passar a filtragem do *gateway* o pedido é redirecionado para o serviço do estabelecimento onde se faz uma procura pelo cartão digital e, em caso de sucesso, são enviadas as informações do cartão para o funcionário.

7.4.2 Creditar Produto

O próximo passo depois de obter a informação do cartão digital é verificar se este está no estado ativo e não pago, e de seguida é feita a creditação no cartão, mas para isso é necessário saber quais os produtos disponíveis. Na Figura 7.4 podemos observar todos os passos para a creditação do produto, desde a obtenção do menu até ao ato de creditação.

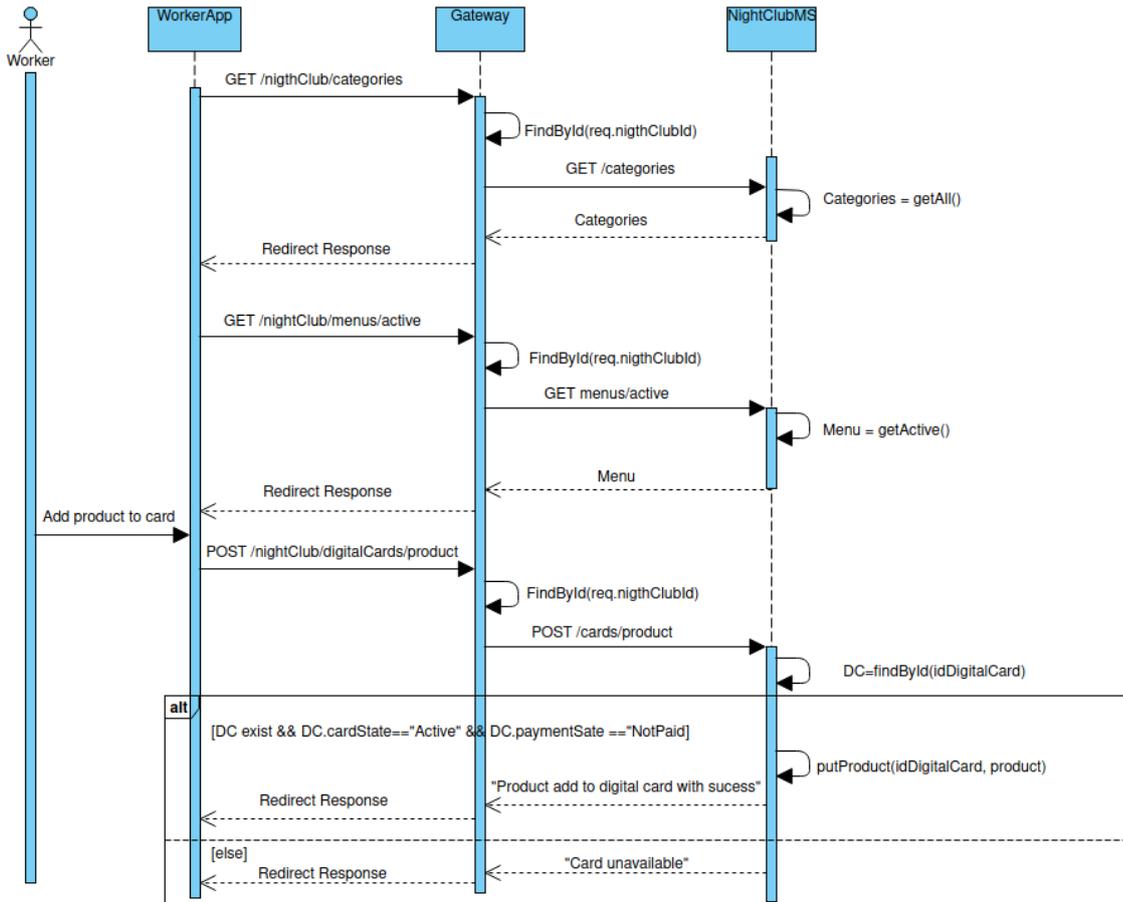


Figura 7.4: Diagrama de sequência de creditar bebidas

Uma vez que o funcionário necessita de visualizar o menu, previamente criado pelo gestor do estabelecimento, para proceder à creditação, a *WorkerApp* faz dois pedidos ao serviço do estabelecimento para obter as categorias dos produtos disponíveis bem como o menu. As categorias são utilizadas no *frontend* para uma melhor organização dos produtos presentes no menu. Após a visualização do menu, na aplicação, o funcionário seleciona o produto que o cliente pretende, o que despoleta um pedido de creditação do produto, indo no corpo deste a informação referente ao produto e o identificador do cartão digital. No serviço do estabelecimento é feita uma procura pelo cartão digital com o intuito de verificar a veracidade deste e o seu estado como também o estado do pagamento, sendo possível creditar o produto apenas quando o cartão estiver no estado "ativo" e "não pago".

7.5 Pagamento de cartões

Sendo o pagamento de cartões, por parte dos utilizadores, uma das proposta de valor do nosso sistema, é imprescindível explicar o algoritmo por detrás desta *feature*. Para que seja possível pagar o cartão através da aplicação *Nite*, é necessário que o cliente tenha associado a sua conta a um cartão digital. No entanto, como cada cartão físico tem um cartão digital associado e existe a possibilidade do cliente não se conectar ao cartão digital, este pode pagar a sua conta através do método convencional (pagar ao balcão). Para isso, e com intuito de manter a consistência dos dados, a aplicação direccionada para os empregados do *Nite* tem a opção de marcar cartões digitais como pagos. Para a explicação do algoritmo assume-se que o cliente quer pagar através da aplicação para consumidores. Este algoritmo é dividido em duas partes: pedido de pagamento e confirmação do pagamento

7.5.1 Pedido de pagamento

O processo de pagamento inicia-se com o cliente a seleccionar uma das opções de pagamento disponíveis. Devido ao curto tempo de realização do projeto, existe apenas a possibilidade de pagar com cartão de crédito através do *Stripe*. Na Figura 7.5 podemos observar os passos para gerar a referência que será associada ao pagamento.

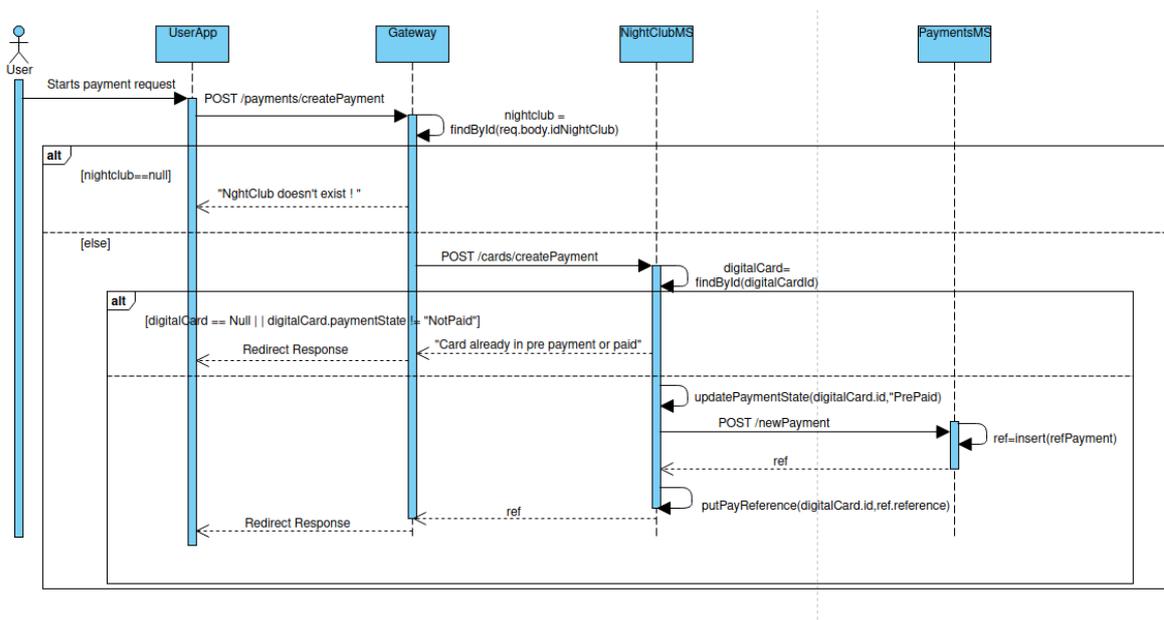


Figura 7.5: Diagrama de sequência de criar um pagamento

A aplicação do utilizador faz o pedido para iniciar o pagamento e este é filtrado no *gateway*. Caso a autenticidade do estabelecimento seja confirmada, o pedido é redireccionado

para o micro-serviço do estabelecimento levando no seu corpo o identificador do cartão digital. De seguida, é feita uma pesquisa pelo cartão digital e verificado se este existe e se está no estado "não pago", visto que, caso isso não se verifique significa que já existe uma referência para o pagamento do cartão digital e, por isso, não é necessário iniciar este processo novamente. Confirmado o estado do cartão, o mesmo é alterado para "pré pago", com o intuito de privar consumos durante o processo de pagamento. Segue-se um pedido ao micro-serviço dos pagamentos para criar uma referência para o pagamento. Essa referência é devolvida para o micro-serviço do estabelecimento para ser guardada juntamente com o cartão digital. O cliente recebe como resposta na sua aplicação a referência criada que inclui o montante a pagar e outras informações pertinentes.

7.5.2 Confirmar pagamento

O cliente, ao receber a referência gerada no passo anterior, tem duas opções: pagar a sua conta fornecendo as informações do seu cartão de crédito ou cancelar o pagamento. Se o cliente optar pelo cancelamento, o seu cartão volta para o estado "não pago" e a associação da referência ao cartão digital é removida. Caso opte por pagar a conta, na Figura 7.6, podemos observar os passos envolvidos nesse processo.

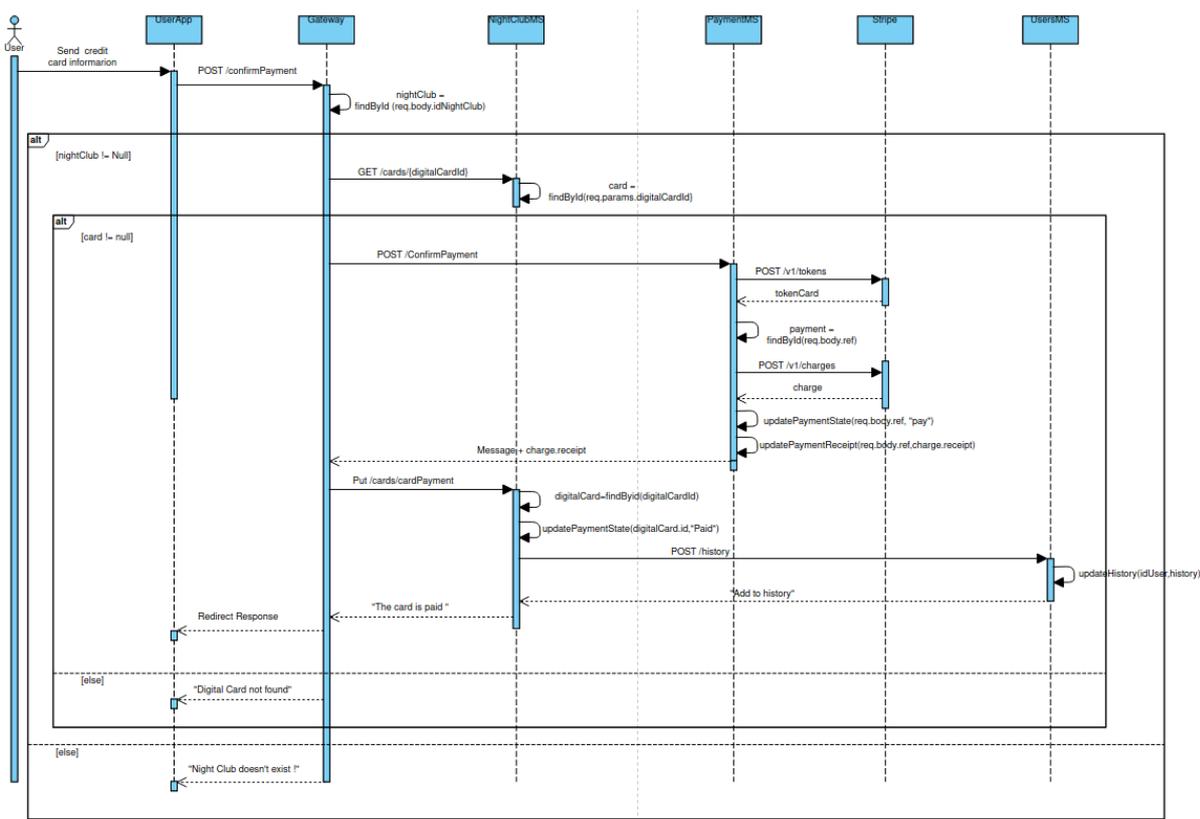


Figura 7.6: Diagrama de sequência de confirmar um pagamento

Depois de o cliente inserir os dados para efetuar o pagamento na aplicação móvel, esta faz um pedido, que contem no seu corpo as informações do cartão de crédito e a referência do pagamento, para que o pagamento seja processado. No *gateway*, para verificar a autenticidade do cartão digital e da referência associada a este, é feito um pedido ao micro-serviço do estabelecimento. Sendo a autenticidade verificada, é feito um pedido ao micro-serviço dos pagamentos que contem a lógica necessária para realizar os pagamentos. Este comunica com o *Stripe*, uma plataforma de pagamentos online. O micro-serviço dos pagamentos faz dois pedidos ao *Stripe*, o primeiro para gerar um *token* que representa o cartão de crédito do utilizador e, de seguida, um pedido para cobrar o dinheiro do cartão de crédito utilizando o *token* gerado. Realizado o pagamento, a referência é marcada como "*paga*" e é armazenado, juntamente com a referência, um comprovativo. De seguida, o estado do cartão digital é marcado como "*pago*" e realiza-se um pedido ao micro-serviço dos utilizadores para guardar no histórico do cliente os seus consumos referentes ao pagamento efetuado.

Instalação e manutenção

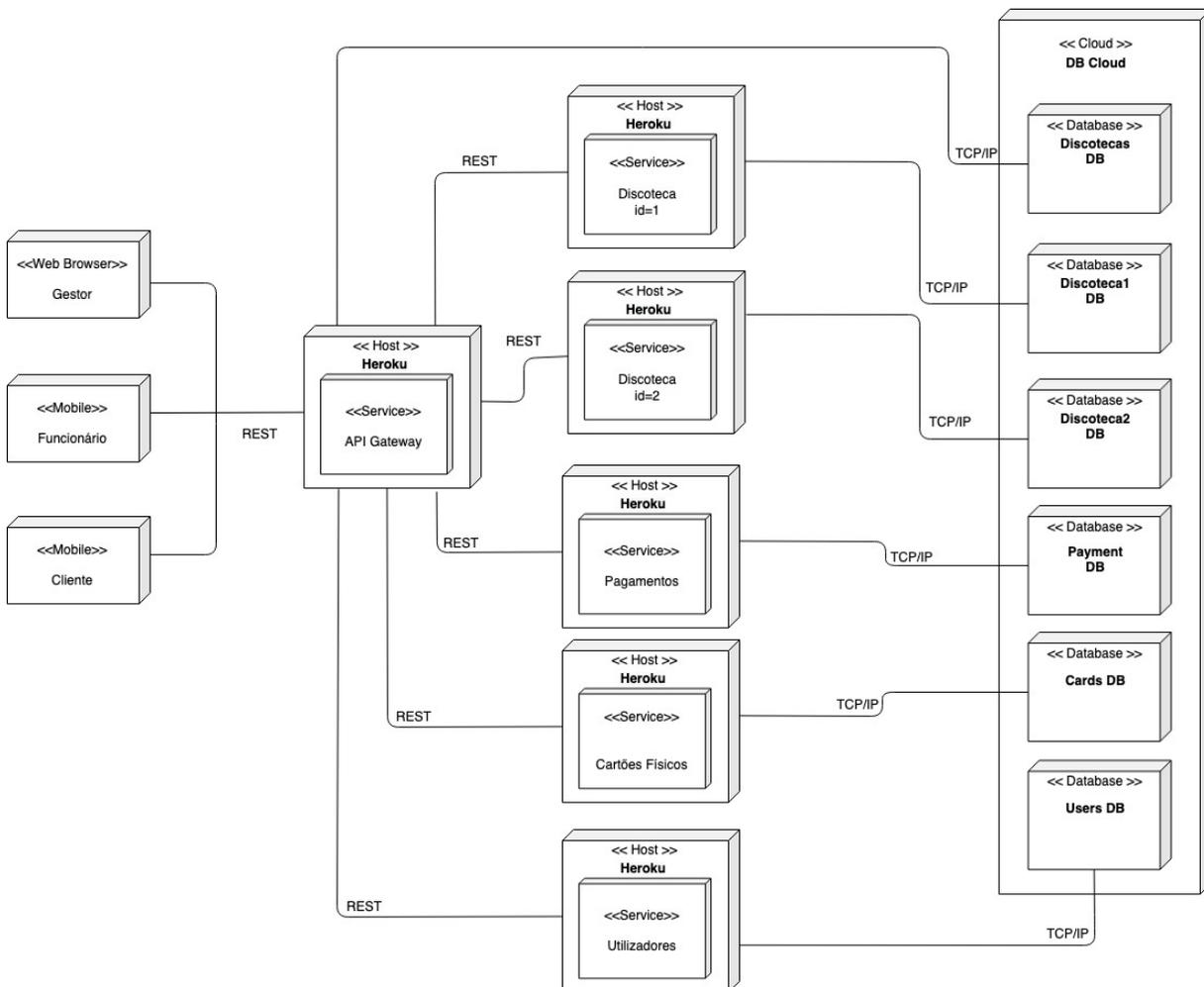


Figura 8.1: Diagrama de instalação

Na figura acima representada podemos ver o diagrama de instalação do nosso sistema.

O nosso sistema consiste em três interfaces, uma para cada tipo de utilizador do sistema, sendo que a do consumidor e a do funcionário são aplicações *mobile* e a do gestor de um estabelecimento é uma *interface Web* acedida através de um *Web Browser*.

Cada componente do back-end está instalado num servidor *Heroku*, incluindo o *APIGateway*, os micro-serviços e os serviços das discotecas.

As bases de dados estão guardada na *Cloud Atlas MongoDB*.

8.1 Interfaces

As interfaces constituem uma parte crucial para o sucesso do nosso sistema. Elas são o nosso único meio que permite aos diferentes utilizadores usufruírem e desfrutarem daquilo que nós podemos oferecer.

As interfaces dos consumidores e dos funcionários são aplicações *mobile* pois, tendo em conta o propósito do sistema, estas são direcionadas a telemóveis. Estas aplicações vão estar disponíveis para *download* no nosso *website* oficial <https://www.nite.pt> em formato *apk*.

Já a interface para o gestor de um estabelecimento pode ser acedida através de *Web Browser* através do link: <https://nite-app.vercel.app/sign-in>. Esta foi instalada na plataforma *Vercel* que permite a instalação ao longo do desenvolvimento, estando integrada com a plataforma *Github*.

8.2 Back-end

Todas as componentes do *back-end* estão instaladas na plataforma *Heroku*.

O *back-end* é composto por $4 + N$ componentes, onde N é o número de discotecas aderentes ao nosso sistema. Cada um destas componentes é instalada na plataforma *Heroku* como um servidor independente. Apesar dos serviços das discotecas apenas diferirem, uns dos outros, na base de dados e no *token* de acesso, este serviço é replicado por cada estabelecimento registado no sistema.

A plataforma *Heroku* permite desenvolvimento e instalações contínuas e automáticas devido à integração com a plataforma *Github*.

8.3 Base de dados

Tal como acontece nos serviços, o número de base de dados também será $4 + N$ componentes, com N o número de estabelecimentos que aderiram ao sistema.

Uma arquitetura de micro-serviços baseia-se na independência entre as várias componentes. Esta permite partir as diferentes ações de um sistema e espalhar por diferentes serviços que funcionam de forma independente dos restantes. Isto não se aplica apenas aos serviços e lógica de *back-end*, mas também às interfaces, bases de dados e correspondentes acessos. Isto é, cada serviço é responsável pelo acesso e persistência dos seus próprios dados.

De forma a manter-nos fiéis a uma arquitetura de micro-serviços, esta camada de dados está separada da lógica de *back-end*, onde cada base de dados comunica apenas com um único serviço, sendo por isso independentes e exclusivas.

As bases de dados são componentes *MongoDB* e todas elas estão instaladas no *MongoDB Atlas*, uma *Cloud* própria do *Mongo* que permite manter bases de dados disponíveis em ambiente *Web*.

9

Conclusão

Com o término deste projeto, a equipa considera como satisfatório o trabalho realizado até ao momento. A concepção do projeto em questão passou por diversas fases, desde um *brainstorming* de ideias e *features* a implementar, um pequeno estudo de mercado de forma a ter conhecimento se a nossa linha de pensamento se enquadrava com as necessidades de um consumidor final, uma pesquisa de tecnologias a utilizar, a formulação da arquitetura, o desenvolvimento das aplicações, e por fim, uma fase de testes de forma a garantir a integridade do produto desenvolvido.

Com a limitação imposta pelo tempo de corrimento da unidade curricular Projeto em Engenharia Informática, nem todas as *features* pretendidas poderiam ser implementadas. Importa referir que o produto desenvolvido corresponde a um MVP (Minimum Viable Product), a partir do qual é possível crescer em termos do serviço disponibilizado. Dessa forma, e tendo uma perspetiva para o futuro e o mercado atual, a equipa pretende a implementação de novas funcionalidades, tais como, o acesso a dados das vendas por noite de cada estabelecimento ao gestor do mesmo, com intuito de assentar a sua posição perante a sua concorrência e atrair o interesse de novos clientes. Depois destas implementações estarem concluídas, pretendemos apostar nas áreas de marketing e publicidade através do estabelecimento de parcerias com grupos de relações públicas, difusão do produto em locais estratégicos, como Universidades, eventos e discotecas aderentes, com intuito de captar novos utilizadores e, por sua vez, clientes para os aderentes dos nossos serviços. Porém, a visão não está restringida a um único tipo de consumidor final, mas também aos locais de eventos. Para estes a visão passa pelo desenvolvimento de material estatístico por festa, como por exemplo, a percentagem de afluência por sexo, as bebidas mais vendidas, horas de maior afluência, entre outros.