

Universidade do Minho

MESTRADO EM ENGENHARIA INFORMÁTICA

PROJETO EM INFORMÁTICA

Relatório técnico - PlannerUp -

Aluno/a:

Ana Filipa Pereira
Ana Luísa Carneiro
Ana Rita Peixoto
Carolina Oliveira
Carolina Santejo
Francisco Peixoto
Luís Pinto
Raquel Costa

Número:

PG46978
PG46983
PG46988
PG47101
PG47102
PG47194
PG47428
PG47600

Janeiro 2023

Conteúdo

1	Introdução	3
1.1	Motivação e contextualização	3
1.2	Caracterização dos utilizador	4
1.2.1	Prestador de serviço	4
1.2.2	Cliente	4
1.3	Fluxo geral da plataforma	5
1.3.1	Prestador de serviço	5
1.3.2	Cliente	5
2	Modelação	6
2.1	Modelo de domínio	6
2.2	Requisitos	7
2.2.1	Requisitos funcionais	7
2.2.2	Requisitos não funcionais	9
2.3	Diagrama de Use Cases	10
2.4	Especificação de Use Cases	11
2.4.1	Registo	11
2.4.2	Adicionar funcionário	13
2.4.3	Adicionar serviço	14
2.4.4	Agendar marcação	15
2.4.5	Consultar estabelecimentos	17
2.4.6	Consultar marcação	18
2.4.7	Reagendar marcação	19
2.5	Protótipos da interface	20
2.5.1	Web	20
2.5.2	Mobile	30
2.6	Máquina de estados	47
2.6.1	Web	48
2.6.2	Mobile	49
2.7	Modelo de dados	50
2.7.1	Identificação e caracterização das entidades e respetivos atributos	50
2.7.2	Modelo lógico da base de dados	53
2.8	Apresentação das tabelas e do espaço ocupado por registo	54
2.9	Diagrama de componentes	59
2.10	Diagrama de <i>Packages</i>	60
2.11	Diagrama de classes	61
3	Arquitetura da Aplicação	62
4	Implementação	63
4.1	Frontend	64

4.1.1	<i>Web</i>	64
4.1.2	<i>Mobile</i>	66
4.2	Backend	69
4.2.1	<i>Controllers</i>	69
4.2.2	<i>Beans</i>	69
4.2.3	Base de dados	71
5	Deployment	71
5.1	Deployment Base de dados	71
5.1.1	Imagens	71
5.2	Deployment Backend	71
5.3	Deployment Frontend	72
5.3.1	WEB	72
5.3.2	Mobile	72
6	Anexos	72
6.1	Protótipos	72
6.1.1	<i>Web</i>	72
6.1.2	<i>Mobile</i>	83

Listas de Figuras

1.1	Modelo de domínio	6
1.2	Modelo de domínio	11
1.3	Especificação do <i>use case</i> - Registar prestador de serviço	12
1.4	Especificação do <i>use case</i> - Registar cliente	13
1.5	Especificação do <i>use case</i> - Adicionar funcionário	14
1.6	Especificação do <i>use case</i> - Adicionar serviço	15
1.7	Especificação do <i>use case</i> - Agendar marcação (Prestador de serviço)	16
1.8	Especificação do <i>use case</i> - Agendar marcação (Cliente)	17
1.9	Especificação do <i>use case</i> - Consultar estabelecimentos	18
1.10	Especificação do <i>use case</i> - Consultar marcação	19
1.11	Especificação do <i>use case</i> - Reagendar marcação	20
1.12	Máquina de estados <i>web</i>	48
1.13	Máquina de estados <i>mobile</i>	49
1.14	Modelo lógico da base de dados	54
1.15	Diagrama de Componentes	59
1.16	Diagrama de <i>Packages</i>	60
1.17	Diagrama de classes	62
1.18	Arquitetura da aplicação	63
1.19	Diagrama de <i>Packages</i>	66
1.20	Diagrama ilustrativo das <i>notificações-push</i>	68

1 Introdução

1.1 Motivação e contextualização

É inegável que a evolução da tecnologia nas últimas décadas veio facilitar bastante o quotidiano da população. Atividades que outrora eram entediantes e maçadoras, são, atualmente realizadas de forma automática e quase instantaneamente. No entanto, verifica-se, que existem diversos setores, nomeadamente no mercado português, que apenas recentemente começaram a usufruir das vantagens e consequentes melhorias que a era digital lhes pode fornecer. Exemplos desta gradual introdução da tecnologia no nosso dia a dia é a utilização recente das caixas automáticas nos supermercados, investimento em impressoras 3D, que podem ser utilizadas em campos como a medicina e indústria, ou ainda a substituição dos funcionários reais por assistentes virtuais.

A verdade é que apesar da tecnologia ser uma aliada essencial da mudança ainda se verifica, nomeadamente em Portugal, uma aversão à mesma, visto que uma parte significativa dos pequenos médios negócios prefere recorrer a métodos manuais, embora tenham à sua disposição um leque variado de ferramentas que permitem a automatização de tarefas. Isto poderá dever-se ao envelhecimento acelerado da população e ao facto de que a grande maioria das ferramentas no mercado não apelam todas as gerações, nomeadamente as mais antigas, a fazer a transição para a era digital.

Um dos setores em Portugal onde se verifica uma crescente necessidade de automatização e digitalização são os serviços que exigem marcações prévias, tais como salões de estética, consultórios médicos, loja do cidadão ou a segurança social. Para usufruir destes serviços o cliente deverá dirigir-se ao estabelecimento correndo o risco de encontrar fila e não ter vaga para a hora pretendida ou então terá que contactar diretamente o estabelecimento desejado para realizar uma marcação. Este último caso, revela-se também penoso para o próprio prestador do serviço, pois este terá que dispendar do seu tempo de trabalho ou contratar uma recepcionista para gerir a agenda. Adicionalmente, a gestão manual dos agendamentos possui um risco inerente de erro humano, podendo levar à sobreposição de marcações ou ainda ao esquecimento por parte do funcionário como dos próprios clientes. Consequentemente, de forma a evitar estes problemas, os prestadores de serviços recorrem a chamadas telefónicas ou mensagens de texto para relembrar os seus clientes das marcações que terão brevemente, sendo este um método demorado e bastante incomodativo para ambas as partes.

Após uma análise cuidada destes modelos de negócio, concluimos também que outra dificuldade que os serviços enfrentam são os processos de reagendamento e cancelamento de marcações, visto que o cliente terá de contactar novamente o estabelecimento. No caso de reagendamento haverá ainda a necessidade de verificar quais as próximas vagas disponíveis para esse cliente. Consequentemente, as vagas que ficam livres, possuem uma baixa probabilidade de serem novamente ocupadas, uma vez que não existe um método de divulgação através do qual o estabelecimento consiga notificar automaticamente todos os outros clientes. Isto poderá resultar num período de tempo inutilizado para o prestador de serviço o que implica perda monetária.

Tendo-se estudado várias soluções existentes no mercado português, concluiu-se que estas não são amplamente divulgadas nem utilizadas por múltiplas razões tais como o

facto de não serem *user friendly*, não possuirem uma gama de serviços alargada e os que têm não estão bem organizados ou ainda não condensarem todas as funcionalidades numa única plataforma.

Deste problema surge a necessidade de criar uma plataforma com duas vertentes, uma direcionada aos clientes e outra aos prestadores de serviços, de forma a facilitar a comunicação entre ambas as partes. De forma a responder a este desafio a plataforma criada deverá automatizar os processos de agendamento, remarcação e cancelamento, mitigando a possibilidade de erros que surgem quando estas ações são realizadas de forma manual. Assim sendo, o cliente poderá manusear as suas marcações como preferir e de acordo com a sua disponibilidade, não sendo necessário que o prestador de serviço dispenda do seu tempo de trabalho para lidar com estas alterações. Complementarmente, a plataforma deverá também garantir que são enviadas notificações automáticas para o cliente, relembrando-o das suas marcações mais próximas. Por outro lado, esta ferramenta deverá ser capaz de fornecer alternativas que incentivem o seu uso a todas as pessoas, mesmo aquelas que não estão familiarizadas com soluções tecnológicas deste género, como por exemplo as gerações mais antigas.

1.2 Caracterização dos utilizador

De modo a concluir a fundamentação da fase inicial do desenvolvimento do projeto, tratamos de fazer um pequeno estudo para tentar perceber a possível abrangência da aplicação.

A aplicação prevê dois tipos de utilizadores, clientes e os prestadores de serviço, cada um deles com a sua respetiva interface.

Recorremos assim à caracterização de duas *personas* que passamos a apresentar de seguida:

1.2.1 Prestador de serviço

Leonor Veigas, 23 anos, Porto

Leonor inaugurou, recentemente, o seu primeiro salão de estética. Como ainda não tem uma base de clientes frequentes, é a única pessoa a trabalhar no estabelecimento. Uma das grandes dificuldades da Leonor é conciliar os serviços que está a realizar com as chamadas que recebe de clientes que pretendem realizar marcações, dado que a maior parte das vezes tem de parar o que está a fazer para atender outros clientes.

Outra preocupação que Leonor tem é a exposição do seu salão a novos clientes, procurando por essa razão uma forma de divulgar o seu negócio.

1.2.2 Cliente

Joana Pereira, 28 anos, Lisboa

Joana foi recentemente contratada para um novo emprego com um horário muito rígido e completo. Devido a este novo horário, Joana passou a ter menos tempo li-

vre, saindo na maioria do tempo muito tarde do trabalho e tendo pouco tempo de almoço. Por esta razão, várias tarefas básicas, como ligar à sua esteticista passam a ser complicadas, devido ao facto de que nas horas em que se encontra livre, a maioria dos estabelecimentos já estão encerrados e não sendo, por isso, capazes de atender chamadas.

Assim, Joana viu-se na necessidade de procurar uma alternativa que permitisse realizar as suas marcações de forma remota e sem ter de contactar diretamente os estabelecimentos.

1.3 Fluxo geral da plataforma

O fluxo geral da plataforma deve ser simples e minimalista. A plataforma deverá possuir como funcionalidades base o **registo** e **login** tanto de clientes como de prestadores de serviço, o **agendamento** de marcações por parte dos dois tipos de utilizadores e a **gestão** da agenda dos prestadores de serviço.

1.3.1 Prestador de serviço

Qualquer estabelecimento pode **registar-se** na aplicação como prestador de serviço, sendo necessário indicar as seguintes informações: nome da empresa, email, palavra-passe, NIF, setor de negócio, telefone, cidade, morada, código postal, horário do estabelecimento, fazer *upload* do ficheiro CAE e opcionalmente de fotos dos serviços prestados. De forma complementar, o prestador tem de indicar qual a subscrição que deseja, o seu método de pagamento, adicionar pelo menos um funcionário e um serviço.

A qualquer momento o prestador pode **configurar os seus funcionários e serviços**, podendo adicionar, remover e alterar as informações de um funcionário/serviço já existente. Uma vez autenticado, o prestador pode aceder à sua agenda onde consegue **consultar a sua agenda**, onde pode ver todas as suas marcações e filtrá-las por funcionário. Adicionalmente, selecionando uma das marcações pode **cancelar**, **reagendar** ou **marcá-la como concluída**. Se desejar, pode também a possibilidade de **agendar novas marcações**.

Complementarmente, pode **acceder ao seu perfil**, onde pode consultar e alterar as suas informações, plano, serviços, funcionários e avaliações, aceder à **lista de contactos dos seus clientes** e ao **dashboard de estatísticas** relativas à *performance* do negócio.

1.3.2 Cliente

Para usufruir de um serviço da aplicação, um cliente deverá **registar-se** na aplicação *mobile*. Para tal, é necessário fornecer o seu *email*, nome, *password*, número de telemóvel, género e cidade.

Depois de autenticados os clientes, na sua página inicial, poderão **consultar** o perfil de múltiplos prestadores de serviço, podendo mesmo filtrar estes mesmos segundo vários critérios. Para **agendar uma marcação**, o cliente depois de selecionar o estabelecimento a que pretende ir, deverá selecionar o serviço, o funcionário e a data e hora que

pretende para a sua marcação. Adicionalmente, o cliente terá acesso à sua **agenda de marcações** onde pode consultar todas as marcações que agendou até ao momento. Selecionando uma das suas marcações, o cliente tem oportunidade de a **cancelar ou reagendar**.

De forma complementar, o cliente recebe uma **notificação** para cada um dos seguintes cenários: o cliente ou prestador de serviço agendar/cancelar uma marcação, faltar x tempo para uma marcação agendada e o prestador de serviço dar uma marcação como concluída.

Depois de um prestador de serviço assinalar uma marcação como concluída, o cliente tem opção de **avaliar** o prestador de serviço.

2 Modelação

2.1 Modelo de domínio

Em primeiro lugar, de modo a conseguirmos apresentar o funcionamento básico da plataforma, foi decidido desenvolver um modelo de domínio que a representá-se. Deste modo, através de um *brainstorming* de ideias realizado entre os elementos da equipa foi possível levantar as principais entidades do sistema em estudo. Resultando, assim, num conhecimento mais aprofundado do contexto do problema entre todas as partes envolventes.

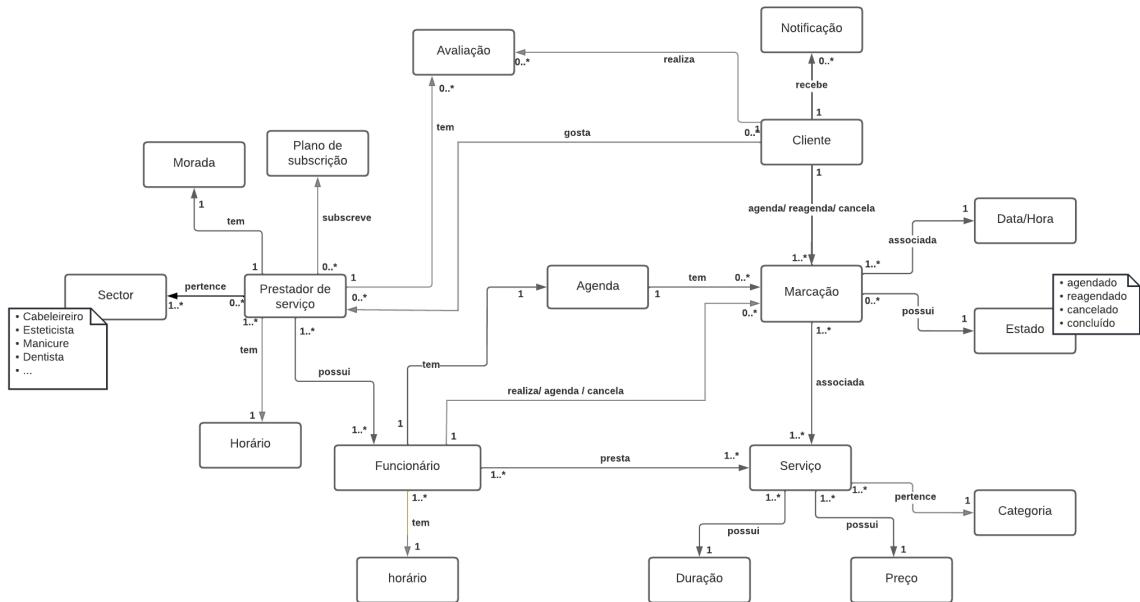


Figura 1.1: Modelo de domínio

A plataforma apresenta duas vertentes, uma direcionada para os **prestadores de serviços** e a outra para os **clientes**.

Um prestador de serviços, na altura do registo, tem de indicar qual o seu **setor**, **morada**, **plano de subscrição** e **horário** de funcionamento. Cada prestador de serviço tem os seus **funcionários** possuindo cada um o seu próprio **horário**. Adicionalmente, cada funcionário está associado ao(s) **serviço(s)** que presta. Por sua vez, cada serviço pertence a uma **categoria**, tem um **preço** e **duração**. Cada funcionário possui a sua própria **agenda**, composta pelas suas **marcações** realizadas e ainda por realizar.

A outra vertente da plataforma destina-se aos clientes dos estabelecimentos que podem realizar marcações nos diversos estabelecimentos, indicando qual ou quais os serviços que pretendem, o funcionário e a **data e hora**. Cada marcação terá um **estado**, podendo este ser agendado, reagendado, cancelado ou concluído. Quando uma marcação se encontrar concluída, o cliente poderá **avaliar** o prestador de serviço. Para além disto, o cliente receberá **notificações** quando, por exemplo, tiver uma marcação a acontecer num futuro próximo ou quando uma das suas marcações for dada como concluída.

2.2 Requisitos

Para realizar o levantamento de requisitos, como foi visto anteriormente, foi crucial investigar o domínio de aplicação. Para tal, procurou-se saber o que é que os potenciais utilizadores, tanto clientes como prestadores de serviços, procuram numa plataforma destas.

Para este efeito, numa primeira fase, foi feita uma análise da concorrência, quais as funcionalidades que dispõem e que são bem recebidas pelo público alvo. Adicionalmente, também se procurou funcionalidades que diferenciassem a plataforma da concorrência.

Após a análise da concorrência, a equipa fez entrevistas a diversos prestadores de serviço, como cabeleireiros, esteticistas, etc. Desta forma, conseguimos perceber o que faz falta no mercado português e quais as funcionalidades mais importantes, as quais foram designadas principais que devem estar no sistema e a sua ordem de prioridade.

2.2.1 Requisitos funcionais

Foi feita uma divisão dos requisitos funcionais por tipo de utilizador – clientes ou prestadores de serviços –, e quanto à sua importância – principais e secundários.

Cliente

Principais:

1. Permitir **registro** na aplicação.
2. Permitir o **Login** na aplicação.
3. Permitir **consultar os estabelecimentos**.
4. Permitir **filtrar as lojas** por disponibilidade, localização, reviews, serviço, categoria.

5. Permitir aceder ao perfil de uma loja.
6. Permitir **agendar** uma marcação.
7. Permitir aceder ao histórico de marcações.
8. Permitir **consultar** marcação.
9. Permitir **cancelar** marcações.
10. Permitir **filtrar** as marcações por estado.
11. Permitir **reagendar** marcações.
12. Permitir aceder ao histórico de notificações.

Secundários:

1. Permitir fazer **pesquisa** através do nome da loja.
2. Permitir **aceitar ou recusar** reagendamentos propostos pelo prestador de serviços.
3. Permitir **avaliar** prestador de serviço.
4. Permitir **editar** perfil.
5. Permitir **adicionar** loja aos favoritos.
6. Permitir **consultar** as lojas assinaladas como favoritas.
7. Permitir **configurar** as notificações que deseja receber.

Prestador de Serviços

Principais:

1. Permitir **registro** no website.
2. Permitir **configurar** o plano de pagamento.
3. Permitir **definir** método de pagamento.
4. Permitir **adicionar** funcionário.
5. Permitir **adicionar** os serviços prestados no estabelecimento.
6. Permitir o **Login** no website.
7. Permitir **aceder à sua agenda**.

8. Permitir **filtrar agenda por funcionários**.
9. Permitir **cancelar marcação**.
10. Permitir **assinalar a conclusão de uma marcação**.
11. Permitir **agendar uma marcação**.
12. Permitir **aceder à lista de contactos de clientes**.

Secundários:

1. Permitir **aceder às suas estatísticas e dashboards**.
2. Permitir **aceder às avaliações**.
3. Permitir **reagendar marcação**.
4. Permitir **editar serviço**.
5. Permitir **eliminar serviço**.
6. Permitir **editar funcionário**.
7. Permitir **eliminar funcionário**.
8. Permitir **configurar as informações** do estabelecimento
9. Permitir **optar entre vista diária ou semanal**
10. Permitir **enviar uma mensagem** ao cliente relativa à sua marcação.
11. Permitir **indicar o período de férias do funcionário**.

2.2.2 Requisitos não funcionais

1. O sistema deverá ser responsivo, adaptando a sua apresentação dependendo do dispositivo utilizado.
2. A interface do utilizador não deve conter linguagem que possa ser considerada discriminatória e ofensiva.
3. Na primeira utilização da aplicação, sem ajuda, 90% dos utilizadores deverá conseguir efetuar um pedido em menos de 7 minutos.
4. O sistema deve ser capaz de suportar vários pedidos em simultâneo.
5. O sistema deve validar os dados inseridos pelos utilizadores para prevenir erros.
6. O sistema deve responder às ações do utilizador em menos de 2 segundos, desprezando eventuais atrasos da rede.

7. A linguagem utilizada não deve recorrer aos termos técnicos relacionados com a implementação da plataforma.
8. O código fonte da aplicação deve conter comentários de modo a facilitar a sua manutenção.
9. Novas funcionalidades deverão ser adicionadas durante um horário em que se verifique um menor número de utilizadores ativos.

2.3 Diagrama de Use Cases

Identificadas as principais entidades e funcionalidades do sistema, foi altura de identificar e desenvolver o diagrama de *use cases*. A equipa optou por apresentar apenas os casos de uso relativos aos requisitos principais para simplificar a interpretação do diagrama. Coloridos a azul, encontram-se identificados os *use cases* que foram considerados mais importantes e que na próxima secção serão expostos com maior detalhe.

Como já foi referido, existem duas vertentes da plataforma, sendo que uma é orientada a prestadores de serviços e a outra para os clientes. Ambos estes atores podem realizar o **registo, login, agendar e cancelar marcações**.

Os Prestadores de serviço podem ainda **adicionar serviços** que ofereçam, **adicionar funcionários**, **aceder à sua agenda**, tendo a possibilidade de a **filtrar por funcionários**, **configurar o plano de pagamento**, **definir o método de pagamento**, **concluir uma marcação** e **aceder à lista de contactos dos seus clientes**.

De forma complementar os clientes podem **consultar os estabelecimentos** registados no sistema, **filtrar lojas** por disponibilidade, localização, reviews, serviço e categoria, **aceder ao perfil de uma loja**, **aceder ao histórico de marcações**, **filtrar marcações pelo seu estado**, **consultar marcações**, **reagendar marcação** e **aceder ao seu histórico de notificações**.

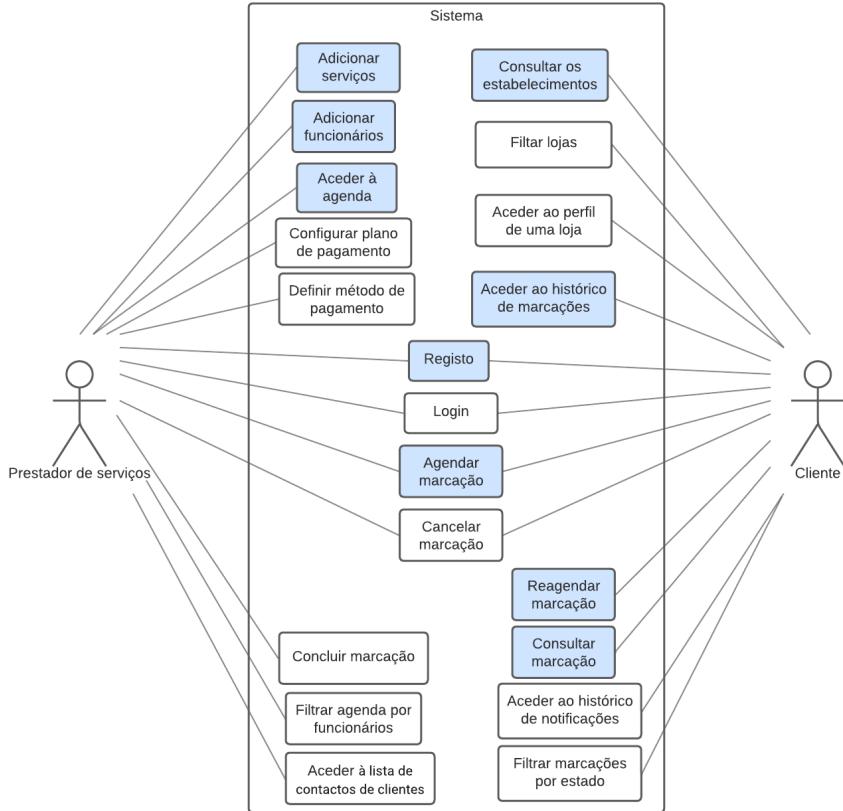


Figura 1.2: Modelo de domínio

2.4 Especificação de Use Cases

Como referido anteriormente, nesta secção será apresentada a especificação dos principais casos de uso que se consideraram mais complexos, de forma a dar um maior entendimento sobre o funcionamento da plataforma.

2.4.1 Registo

Dado que este caso de uso tem um fluxo diferente dependendo de qual o utilizador que está a utilizar o sistema

Prestador de serviço

Qualquer estabelecimento legalizado pode registrar-se, a qualquer momento, na plataforma, tendo de indicar o seu nome, email, palavra passe, NIF, setor de negócio, telefone, cidade, morada, código postal, horário de funcionamento, fazer upload do comprovativo CAE e aceitar os termos e condições. Para concluir o seu registo, o prestador de serviço deverá ainda escolher qual o plano de pagamento que vai subscrever, definir o seu método de pagamento e adicionar pelo menos um funcionário e um serviço que disponibilize.

Use case		
Use Case:	Registo	
Ator:	Prestador de serviços	
Pré-condição:	—	
Pós-condição:	O prestador de serviço foi registado no sistema	
	Autor	Sistema
Fluxo normal	1. Indica que se quer registar	
		2. Apresenta a página de registo
	3. Insere nome da empresa, email, palavra passe, NIF, setor de negócio, telefone, cidade, morada, código postal, horário, faz <i>upload</i> do ficheiro CAE e aceita os termos e condições	
	4. Selecciona o botão de registar	
		5. Verifica o preenchimento de todos os campos
	6. <i>Include <<Configurar plano de pagamento>></i>	
	7. <i>Include <<Definir método de pagamento>></i>	
	8. <i>Include <<Adicionar funcionário>></i> uma ou mais vezes	
	9. <i>Include <<Adicionar Serviço>></i> uma ou mais vezes	
		10. Regista o ator
Fluxo exceção 1: [Ator quer cancelar o registo] (Passo 3, 6, 7, 8 e 9)	3.1/ 6.1/ 7.1/ 8.1/ 9.1 A operação de registo é cancelada	
		3.2/ 6.2/ 7.2/ 8.2/ 9.2 Apresenta a página inicial
Fluxo alternativo 1: [Os campos não estão devidamente preenchidos] (Passo 5)		5.1 Indica o(s) campo(s) mal preenchido(s)
	5.2 Preenche novamente os campos assinalados	
	5.3 Volta a 4	
Fluxo alternativo 2: [O email já está registado no sistema] (Passo 5)		5.1 Informa que o email já se encontra registado
	5.2 Volta a preencher o email	
	5.3 Volta a 4	

Figura 1.3: Especificação do *use case* - Registar prestador de serviço

Cliente

Já um cliente, no caso de se querer registar na plataforma, deverá indicar o seu email, palavra passe, nome, número de telemóvel, género e cidade.

Use case		
Use Case:	Registo	
Ator:	Cliente	
Pré-condição:	—	
Pós-condição:	O cliente foi registado no sistema	
	Autor	Sistema
Fluxo normal	1. Indica que se quer registrar	2. Apresenta a página de registo
	3. Indica o email e palavra passe, nome, número de telemóvel, género e cidade	
		4. Verifica o preenchimento de todos os campos
		5. Regista o ator
	3.1 A operação de registo é cancelada	
Fluxo alternativo 2: [O email já está registado no sistema] (Passo 4)		4.1. Informa que o email já se encontra registado
	4.2. Volta a preencher o email	4.3 Volta a 4
		4.1 Indica o(s) campo(s) mal preenchido(s)
Fluxo alternativo 1: [Os campos não estão devidamente preenchidos] (Passo 4)	4.2 Preenche novamente os campos assinalados	
	4.3 Volta a 4	

Figura 1.4: Especificação do *use case* - Registar cliente

2.4.2 Adicionar funcionário

Um prestador de serviço, tanto durante o registo como quando já estiver autenticado, poderá adicionar um funcionários à lista dos funcionários que trabalham no estabelecimento. Para tal terá de indicar o nome, email, número de telemóvel, profissão que exerce e o respetivo horário de trabalho.

Use case		
Use Case:	Adicionar um funcionário	
Autor:	Prestador de serviço	
Pré-condição:	O ator está a registar-se ou encontra-se autenticado	
Pós-condição:	O funcionário foi registado no sistema	
	Autor	Sistema
Fluxo normal	1. Indica que quer adicionar um funcionário	
		2. Apresenta a página de registar um novo funcionário
	3. Indica o nome, email, número de telemóvel, profissão do funcionário e horário de trabalho	
	4. Selecciona o botão de gravar	
		5. Verifica o preenchimento de todos os campos
		6. Regista o novo funcionário
Fluxo Exceção 1: [Autor não quer registar o funcionário](Passo 3)	3.1 A operação de registo do funcionário é cancelada	3.2 Apresenta a página anterior
Fluxo alternativo 1: [Os campos não estão devidamente preenchidos] (Passo 5)		5.1 Indica o(s) campo(s) mal preenchido(s)
	5.2 Preenche novamente os campos assinalados	
	5.3 Volta a 4	

Figura 1.5: Especificação do *use case* - Adicionar funcionário

2.4.3 Adicionar serviço

De forma semelhante ao uso anterior, um prestador de serviço pode adicionar um serviço que o estabelecimento preste, indicando o nome, a categoria a que pertence, como por exemplo depilação, uma breve descrição se desejar, a sua duração, preço e quais os funcionários que realizem o serviço em questão.

Use case		
Use Case:	Adicionar um serviço	
Ator:	Prestador de serviço	
Pré-condição:	O ator está a registar-se ou encontra-se autenticado	
Pós-condição:	O serviço foi registado no sistema	
	Autor	Sistema
Fluxo normal	1. Indica que quer adicionar um serviço	
		2. Apresenta a página de registrar um novo serviço
	3. Indica o nome, categoria, descrição, duração média, preço e associa os funcionários que oferecem aquele serviço	
	4. Selecciona o botão de gravar	
		5. Verifica o preenchimento de todos os campos
		6. Regista o novo serviço
Fluxo Exceção 1: [Atores não querem registrar o serviço](Passo 3)	3.1. A operação de registo do serviço é cancelada	
		3.2 Volta à página anterior
Fluxo alternativo 1: [Os campos não estão devidamente preenchidos] (Passo 5)	5.1 Indica o(s) campo(s) mal preenchido(s)	
	5.2 Preenche novamente os campos assinalados	
	5.3 Volta a 4	

Figura 1.6: Especificação do *use case* - Adicionar serviço

2.4.4 Agendar marcação

Novamente, ambos os tipos de utilizadores podem realizar marcações, sendo o processo muito semelhante para os dois, havendo alguns aspetos que os distinguem.

Prestador de serviço

É importante que um prestador tenha a capacidade de agendar uma marcação para, por exemplo, adicionar marcações que tenham sido agendadas no final de uma outra marcação e/ou que tenha sido feita por clientes que não tenham acesso à aplicação e desta forma, manter a sua agenda sempre atualizada. Para agendar uma marcação, o utilizador deve indicar o cliente, serviços desejados, o funcionário e a data e hora da marcação. No caso desse cliente se encontrar registrado no sistema, este irá receber uma notificação relativa a esta marcação.

Neste processo, com o objetivo de evitar o erro de existirem marcações sobrepostas, é importante que depois de selecionado um funcionário, se tenha a consideração a sua disponibilidade e seja permitido apenas fazer a marcações em horas que o funcionário esteja livre.

Use case		
Use Case:	Agendar marcação	
Autor:	Prestador de serviço	
Pré-condição:	O ator encontra-se autenticado e encontra-se na sua agenda	
Pós-condição:	A marcação foi registada no sistema	
	Autor	Sistema
Fluxo normal	1. Indica que quer fazer uma marcação	
		2. Apresenta a página de agendar nova marcação
	3. Indica o cliente, categoria, serviço, funcionário, data e hora, com base na disponibilidade do funcionário escolhido	
	4. Selecciona o botão de agendar	
		5. Regista a marcação no sistema
		6. Se o cliente está registado na plataforma, envia uma notificação com a nova marcação
Fluxo Exceção 1: [ator não quer realizar uma marcação](Passo 3)	3.1. A operação de marcação é cancelada	

Figura 1.7: Especificação do *use case* - Agendar marcação (Prestador de serviço)

Cliente

Um cliente que esteja autenticado e deseje agendar uma marcação, depois de escolher o estabelecimento, deve selecionar os serviços que procura, o funcionário pelo qual quer ser atendido, a data e hora da marcação.

Neste cenário é novamente necessário estar presente o cuidado de não existirem marcações sobreposas.

Use case		
Use Case:	Agendar marcação	
Ator:	Cliente	
Pré-condição:	O cliente encontra-se autenticado e encontra-se no perfil do estabelecimento em que pretende fazer marcação	
Pós-condição:	A marcação foi registada no sistema	
	Ator	Sistema
Fluxo normal	1. Indica que quer fazer uma marcação	
		2. Apresenta a página de agendar nova marcação
	3. Selecionar um ou mais serviços, um funcionário e data e hora da marcação, com base na disponibilidade do funcionário escolhido	
	4. Selecciona o botão de agendar	
		5. Regista a marcação no sistema
Fluxo Exceção 1: [ator não quer realizar uma marcação](Passo 3)	3.1. A operação de marcação é cancelada	

Figura 1.8: Especificação do *use case* - Agendar marcação (Cliente)

2.4.5 Consultar estabelecimentos

Um cliente que esteja autenticado e que deseje consultar um estabelecimento, depois de indicar que quer ir para a página inicial, deverá procurar e selecionar o estabelecimento que deseja consultar. Em seguida, ser-lhe-á apresentado o perfil do estabelecimento escolhido.

Adicionalmente, se assim o desejar, o cliente tem a opção de filtrar a lista de estabelecimentos por vários critérios como cidade, categoria, serviços prestados, entre outros.

Use case		
Use Case:	Consultar estabelecimentos	
Autor:	Cliente	
Pré-condição:	O ator encontra-se autenticado	
Pós-condição:	O cliente consultou o estabelecimento	
	Autor	Sistema
Fluxo normal	1. Indica que quer consultar estabelecimentos	2. Apresenta a página inicial
	3. Seleciona o estabelecimento que pretende consultar	4. Apresenta o perfil do estabelecimento
	3.1 Indica que quer consultar a página de marcações ou o seu o perfil	3.2 Apresenta a página de marcações ou perfil
	3.1. Indica o filtro que pretende aplicar aos estabelecimentos	3.2. Apresenta apenas os estabelecimentos que respeitam o filtro escolhido pelo ator
	5.3 Volta a 3	

Figura 1.9: Especificação do *use case* - Consultar estabelecimentos

2.4.6 Consultar marcação

Um cliente que esteja autenticado e que deseje consultar uma marcação agendada, depois de indicar que quer consultar as suas marcações, deverá indicar, selecionando no calendário apresentado, a data da marcação que deseja procurar. Em seguida, entre as marcações existentes para o dia selecionado, deverá escolher aquela que pretende consultar.

Use case		
Use Case:	Consultar marcação	
Autor:	Cliente	
Pré-condição:	O ator encontra-se autenticado	
Pós-condição:	O cliente consultou a marcação	
	Autor	Sistema
Fluxo normal	1. Indica que quer consultar as marcações	2. Apresenta a página marcações
	3. Seleciona no calendário qual o dia da marcação que quer consultar	
		4. Apresenta as marcações do dia selecionado
	5. Indica a marcação que deseja consultar	
		6. Apresenta a página de consulta da marcação
Fluxo Exceção 1: [Atores não querem consultar marcação] (Passo 1/3/5)	1.1/ 3.1/ 5.1 Indica que quer consultar a página inicial ou o seu perfil	
		1.2/ 3.2/ 5.2 Apresenta a página inicial ou perfil
Fluxo alternativo 1: [O ator enganou-se a selecionar a data] (Passo 5)	5.1. Seleciona, novamente, no calendário qual o dia da marcação que quer consultar	
		5.2. Apresenta as marcações do dia selecionado
	5.3 Volta a 5	

Figura 1.10: Especificação do *use case* - Consultar marcação

2.4.7 Reagendar marcação

Um cliente que esteja autenticado e pretenda reagendar uma marcação, deverá consultar a mesma e depois selecionar a opção de reagendar. Como previamente o serviço já foi escolhido, não é possível alterar o mesmo, se assim quisesse, teria de cancelar a marcação e agendar uma nova. Para reagendar terá de escolher novamente o funcionário, assim como a data e hora para a qual quer alterar a marcação.

Use case		
Use Case:	Reagendar marcação	
Autor:	Cliente	
Pré-condição:	O ator encontra-se autenticado e a consultar uma marcação	
Pós-condição:	O cliente reagendou a marcação	
	Autor	Sistema
Fluxo normal	1. Indica que quer reagendar a marcação	
		2. Apresenta a página de reagendamento
	3. Selecciona o funcionário e a nova data e hora da marcação	
	4. Selecciona guardar o reagendamento	
		5. Regista as alterações da marcação
		6. Apresenta confirmação de reagendamento
Fluxo Exceção 1: [Atores não quer reagendar marcação] (Passo 3 ou 4)	3.174.1 Indica que quer cancelar reagendamento	
		3.2 Apresenta a página de consulta de marcação
Fluxo alternativo 1: [O ator selecionou mal o funcionário ou data/hora] (Passo 3)	3.1. Volta a selecionar o funcionário e/ou data e hora que pretende	
	3.2 Volta a 4	

Figura 1.11: Especificação do *use case* - Reagendar marcação

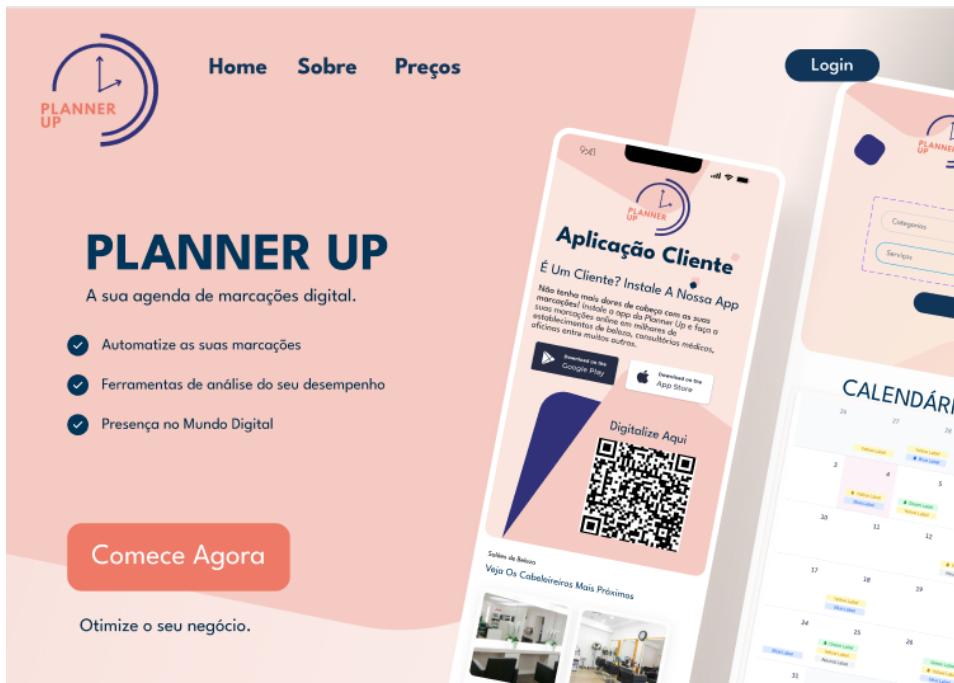
2.5 Protótipos da interface

Ainda nesta fase, foram elaborados protótipos, de baixa fidelidade, da interface gráfica da aplicação, recorrendo à ferramenta Figma.

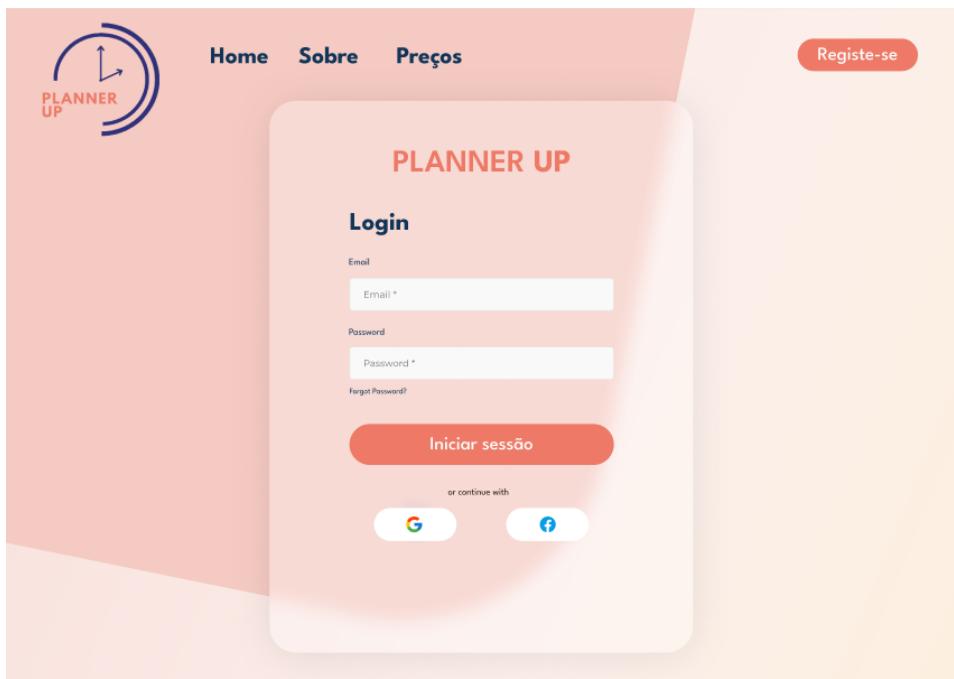
Ao longo do desenvolvimento dos mesmos, tivemos atenção aos princípios de usabilidade. Em seguida, encontram-se apresentados alguns dos protótipos desenvolvidos, os restantes, assim como o próprio *flow* entre páginas, podem ser consultados [aqui](#) ou nos anexos ([6.1](#)).

2.5.1 Web

Página inicial



Login



Registo

PLANNER UP

Home **Sobre** **Preços** **Login**

Nome do Empresa **Telefone** **Ficheiro CAE**

Email **Cidade**

Palavra-passe **Morada**

NIF **Código de Postal**

Setor de Negócio

Setor

Cabeleireiro

Esteticista

Monicure

Dentista

Horário de funcionamento

I agree with [Terms](#) and [Privacy](#)

Gravar

Horário de funcionamento
Preencha os seguintes campos

2^a Feira 09:00 até 13:00 hh:mm até hh:mm + -

3^a Feira Início hh:mm até Fim hh:mm + -

4^a Feira Início hh:mm até Fim hh:mm + -

5^a Feira Início hh:mm até Fim hh:mm + -

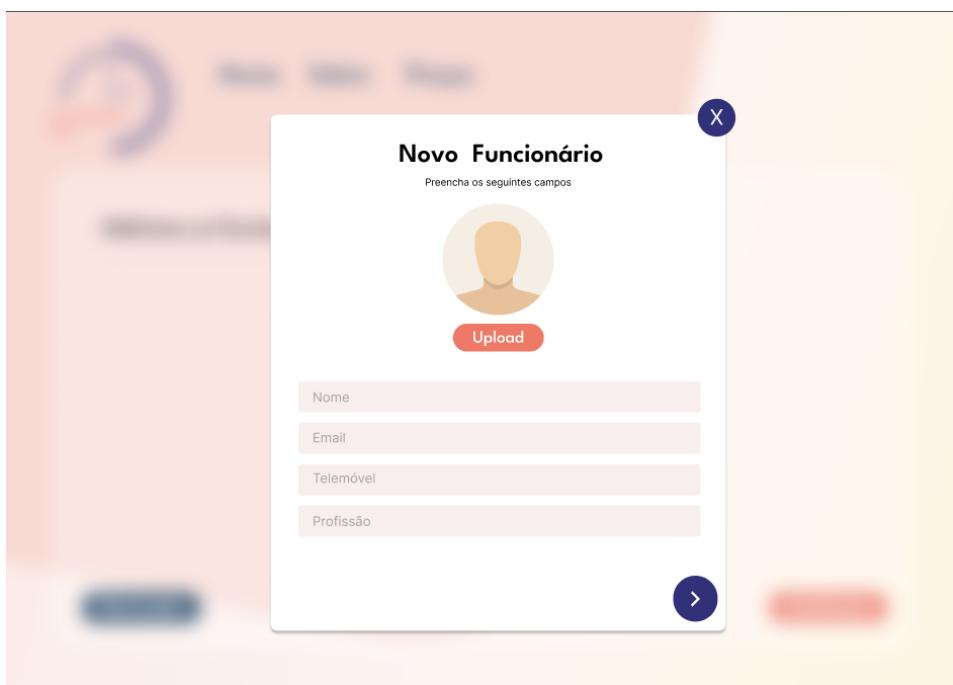
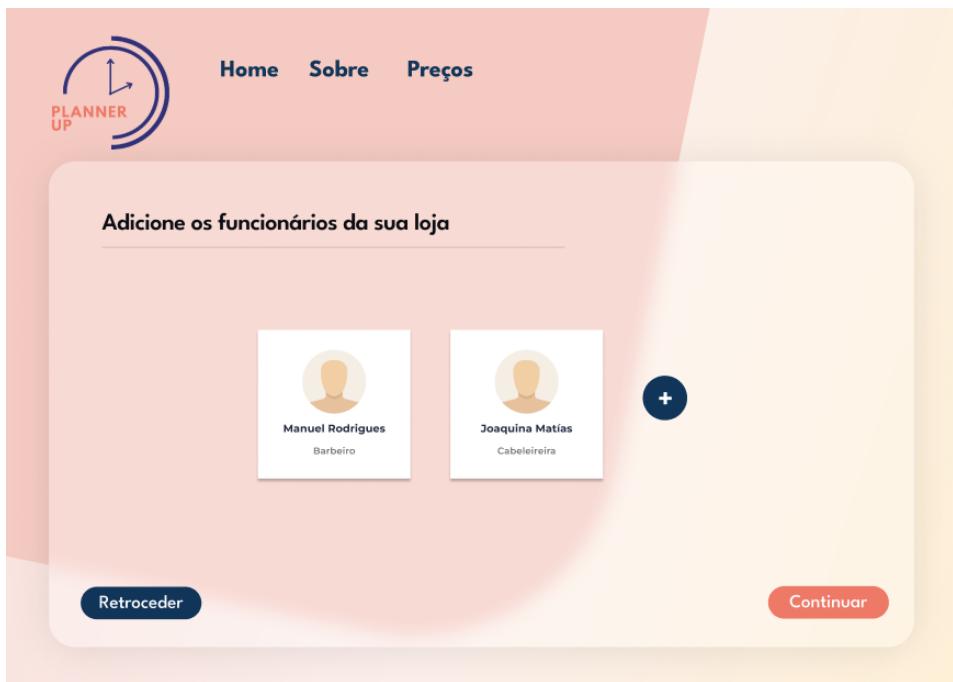
6^a Feira Início hh:mm até Fim hh:mm + -

Sábado Início hh:mm até Fim hh:mm + -

Domingo Início hh:mm até Fim hh:mm + -

Pronto

Adicionar funcionário



X

João Miranda

Indique o horário semanal e a disponibilidade do funcionário

2ª Feira	09:00	até	13:00	hh:mm	até	hh:mm	+	-
3ª Feira	Inicio hh:mm	até	Fim hh:mm	+	-			
4ª Feira	Inicio hh:mm	até	Fim hh:mm	+	-			
5ª Feira	Inicio hh:mm	até	Fim hh:mm	+	-			
6ª Feira	Inicio hh:mm	até	Fim hh:mm	+	-			
Sábado	Inicio hh:mm	até	Fim hh:mm	+	-			
Domingo	Inicio hh:mm	até	Fim hh:mm	+	-			

Pronto



Home **Sobre** **Preços**

Adicione os funcionários da sua loja



Manuel Rodrigues
Barbeiro



Joaquina Matias
Cabeleireira

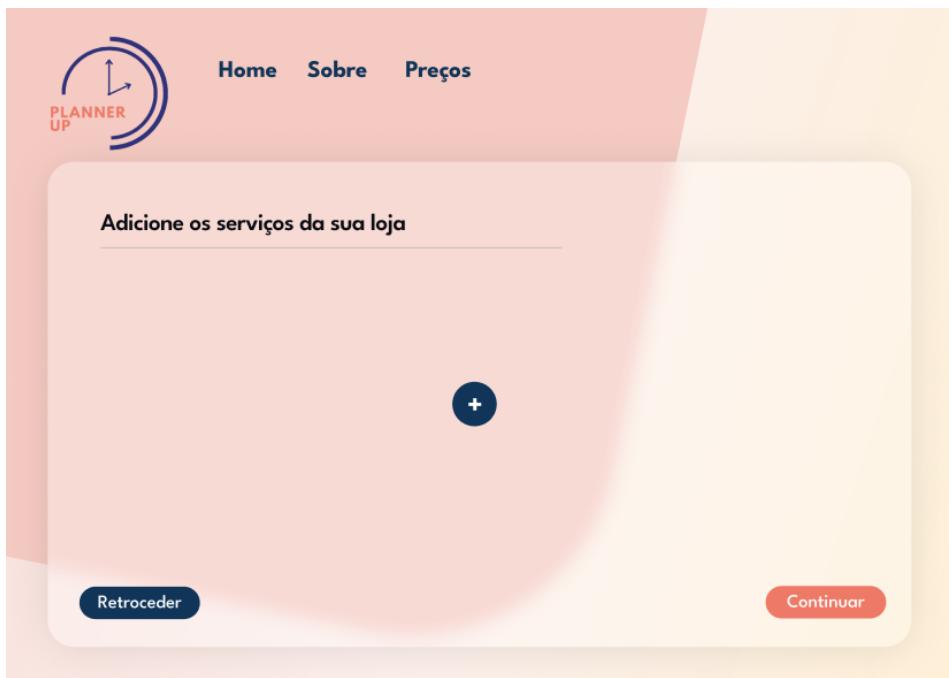


João Miranda
Estéticista

+

Retroceder **Continuar**

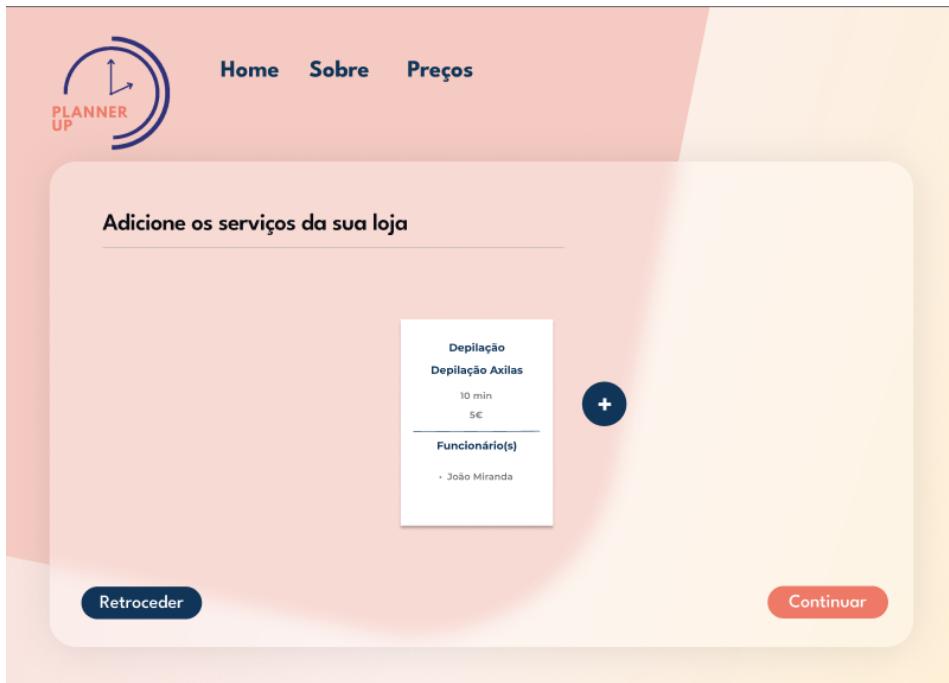
Adicionar serviço



This screenshot displays a modal window titled "Novo Serviço" (New Service). The title is centered above a sub-instruction: "Preencha os seguintes campos". The form consists of several input fields:

- Nome***: A text input field with placeholder text "Ex: Depilação cera Axila".
- Categoria***: A text input field with placeholder text "Ex: Depilação".
- Descrição**: A text input field with placeholder text "Descrição".
- Duração média em minutos***: A text input field with placeholder text "Duração".
- Preço (€€)***: An input field with a radio button selected next to "Ex: 5€", followed by a range slider with two handles and the word "até" between them.

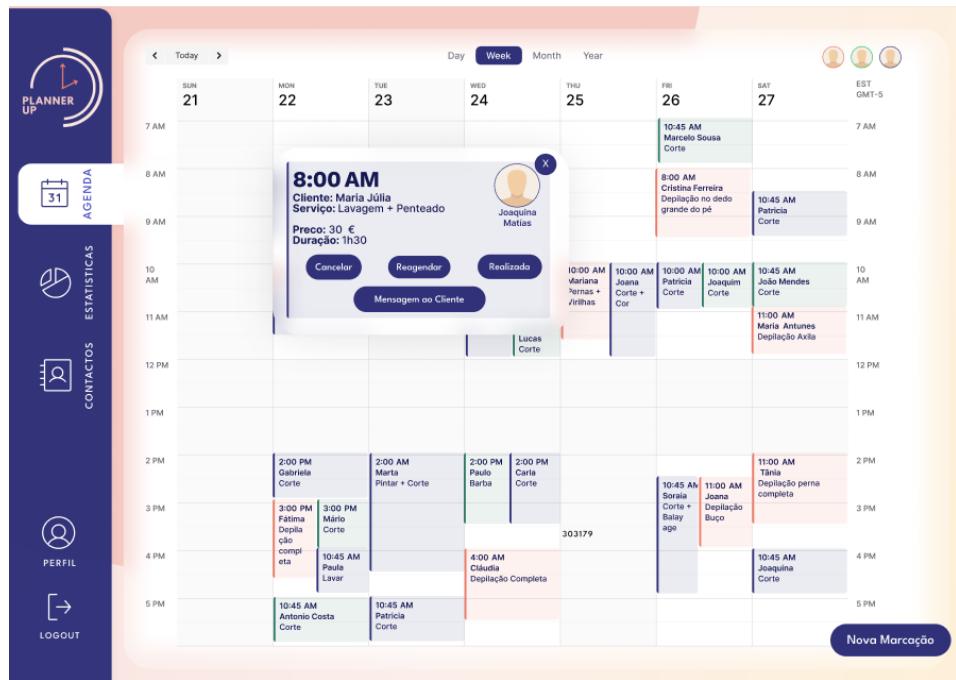
Below these fields is a section titled "Associar Funcionários*" (Associate Staff*) with a dropdown menu labeled "Selecione um funcionário" (Select a staff member). The dropdown shows three names: João Miranda, Joaquim Matias, and Manuel Rodrigues. A small blue circular arrow icon is located to the right of the dropdown.



Agenda

SUN	MON	TUE	WED	THU	FRI	SAT	EST GMT-5	
21	22	23	24	25	26	27		
7 AM					10:45 AM Marcelo Sousa Corte		7 AM	
8 AM	8:00 AM Maria Julia Lavagem + Penteado				8:00 AM Cristina Fernanda Depilação no dedo grande do pé	10:45 AM Patrícia Corte	8 AM	
9 AM			9:00 AM Mariana Depilação Completa				9 AM	
10 AM	10:00 AM Mariana - Buço	10:45 AM Patrícia Corte	10:00 AM Patrícia Cor	10:00 AM Luciana Axilas	10:00 AM Mariana Pernas + Vitilígas	10:00 AM Joana Corte + Cor	10:45 AM João Mendes Corte	10 AM
11 AM				11:00 AM Lucas Corte			11:00 AM Maria Antunes Depilação Axila	11 AM
12 PM								12 PM
1 PM								1 PM
2 PM	2:00 PM Gabriela Corte	2:00 AM Marta Pintar + Corte	2:00 PM Paulo Barba	2:00 PM Carla Corte			11:00 AM Tânia Depilação perna completa	2 PM
3 PM	3:00 PM Fátima Depilação Completa	3:00 PM Mário Corte			10:45 AM Sandra Corte + Balayage	11:00 AM Isaura Depilação Buçó		3 PM
4 PM		10:45 AM Paula Lavar		4:00 AM Cláudia Depilação Completa			10:45 AM Joaquina Corte	4 PM
5 PM	10:45 AM Antonio Costa Corte	10:45 AM Patrícia Corte						5 PM

Consultar marcação



Agendar nova marcação

Nova Marcação

Nome do Cliente

Selecione um Cliente

- Maria Júlia
- Cristina Ferreira
- Fátima

Categoria

Selecione uma Categoria

- Cabelos
- Depilação
- Barbearia

Serviço

Selecione um Serviço

- Depilação Axilas
- Depilação Completa
- Depilação Perna

Associar Funcionário

Selecione um funcionário

- João Miranda
- Joaquina Matias
- Manuel Rodrigues

Hora

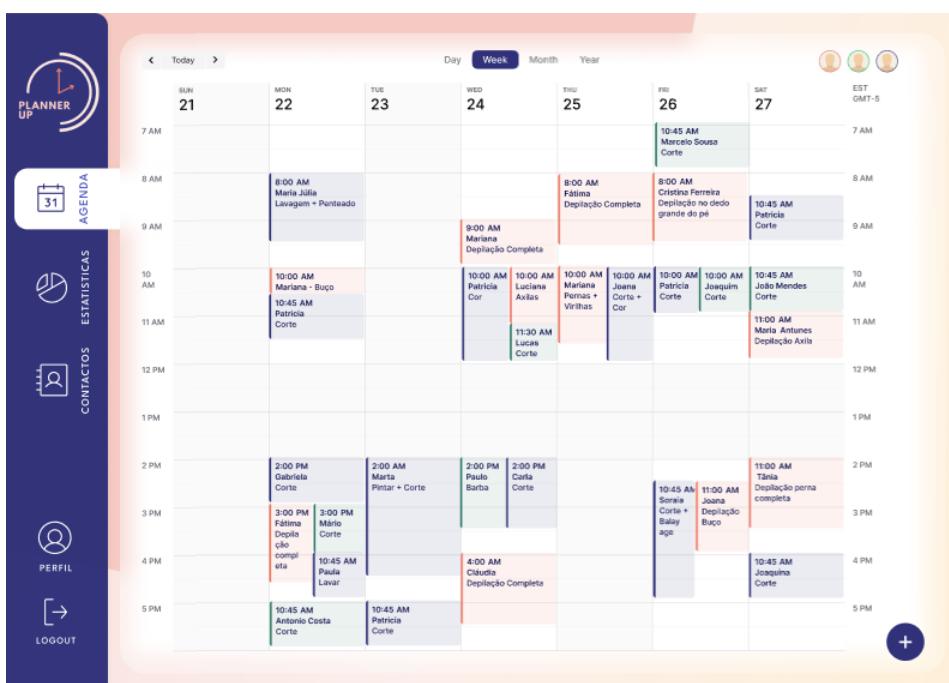
Selecione a hora

AM

PM

Retroceder

Continuar



Perfil

Cabeleireiro Leila

Serviços [View all >](#)

- Corte
- Pintar
- Barba
- Depilação Axilas

Funcionários [View all >](#)

- Joaquina Matias
- Manuel Rodrigues
- João Miranda

Indisponibilidades

Avaliações [View all >](#)

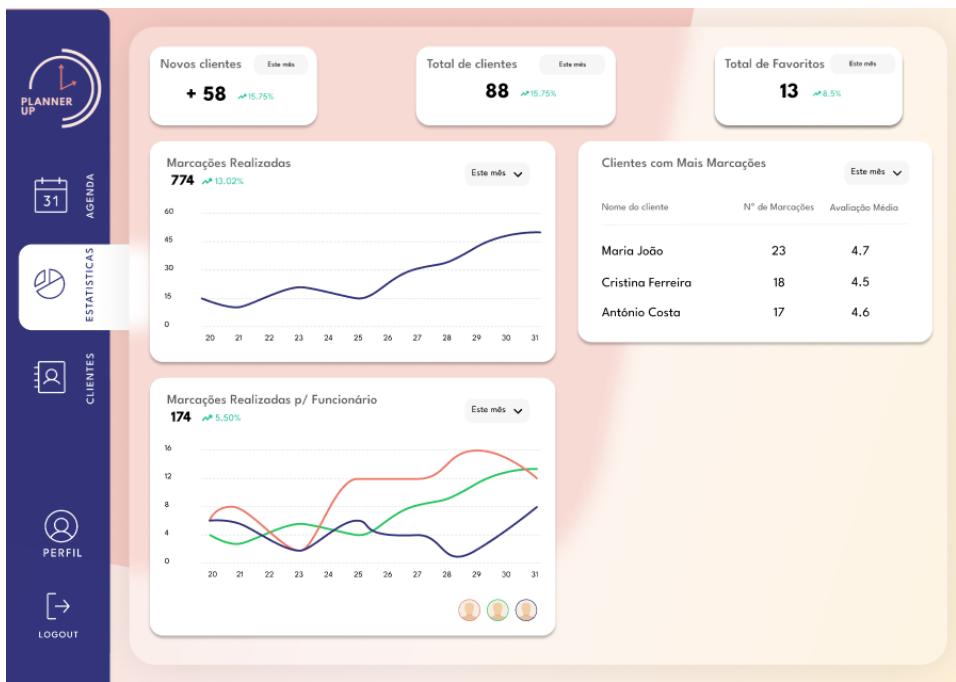
Cliente	Data	Rate	Type
Maria João	Feb 12	★ 4.8	CORTE
Cristina Ferreira	Feb 14	★ 4.6	DEPILAÇÃO
Manuel Silva	Feb 16	★ 4.5	BARBA
António Costa	Feb 18	★ 4.8	SOBRANCEIRA

UPGRADE TO PREMIUM

Lista de contactos

Fatima fatima@gmail.com 965783231	Marcelo Souca prMarco@gmail.com 967345627	Mariana mariana@gmail.com 962345345	Mário mario@gmail.com 961352653
Lucas lucas@gmail.com 964572453	Patrícia patricia@gmail.com 912354890	Gabriela gabi@gmail.com 924587921	António Costa costaMP@gmail.com 921189045
Paula paula@gmail.com 912467985	Cristina Ferreira cf@gmail.com 916534987	Maria Julia maria_julia@gmail.com 910267983	Joaquim joaquim@gmail.com 917809345

Estatísticas



2.5.2 Mobile

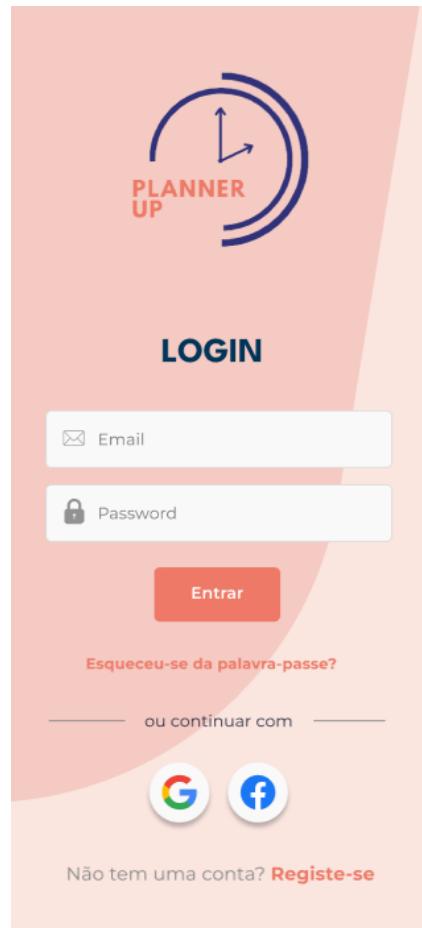
Início



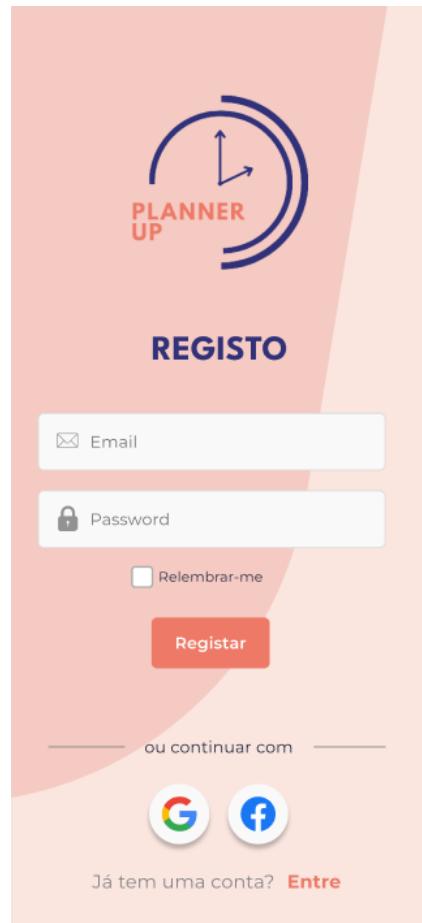


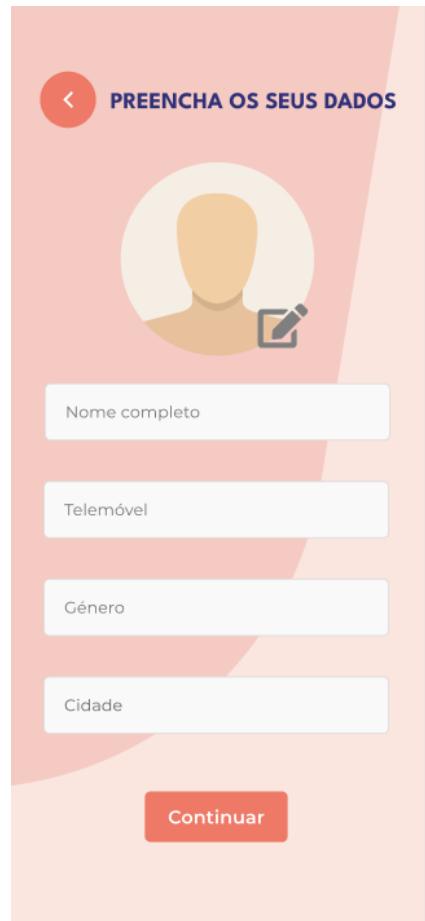
Login





Resgisto





Perfil



Histórico de marcações







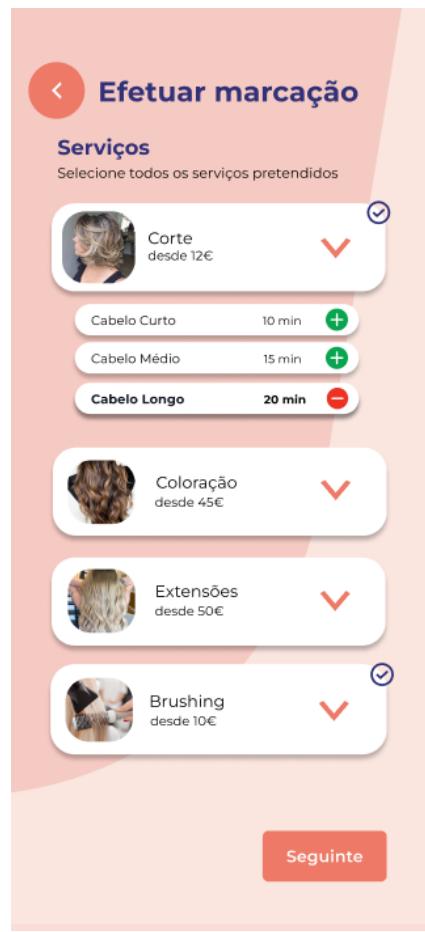
Página inicial



Perfil prestador de serviço



Agendar marcação









Histórico notificações



Consultar marcação



2.6 Máquina de estados

Em seguida, para definir o fluxo entre as várias interfaces da aplicação foram desenhados dois diagramas máquina de estados, um para a aplicação do prestador (*web*) e outro para a aplicação cliente (*mobile*).

2.6.1 Web

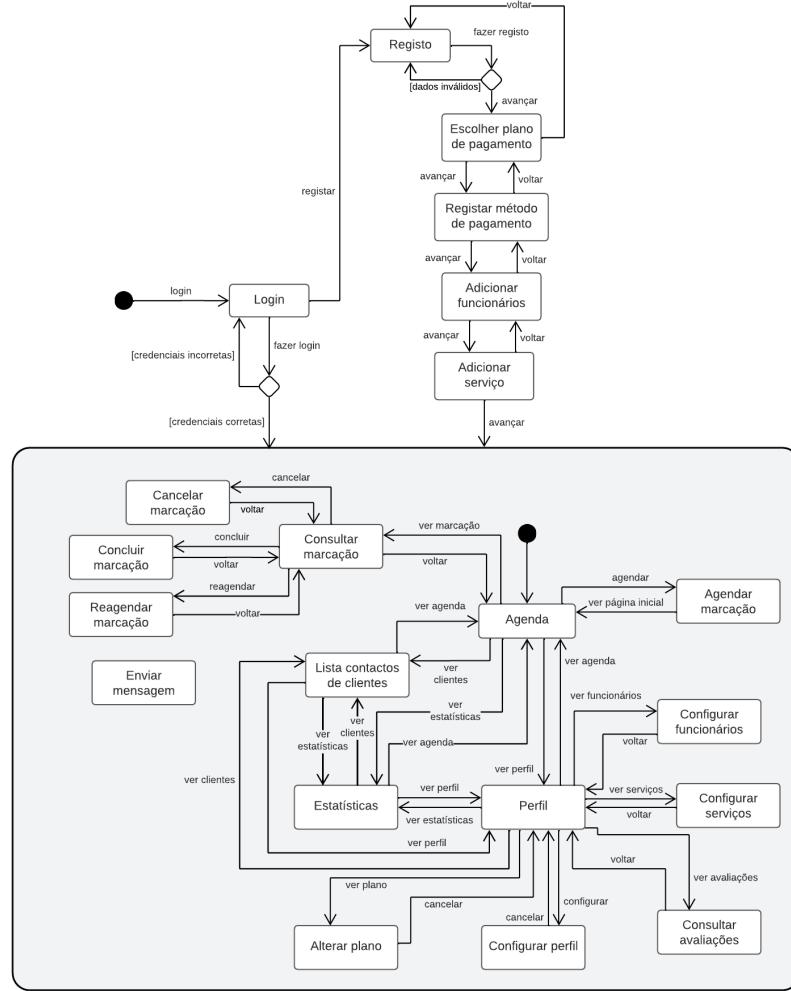


Figura 1.12: Máquina de estados *web*

Para aceder às funcionalidades da aplicação, é necessário começar por fazer o *login*, caso o prestador não esteja registado pode-o fazer, tendo de passar por várias páginas como escolher plano de pagamento, método de pagamento e adicionar funcionários e serviços. Assim que o utilizador estiver autenticado, este é redirecionado par a sua agenda.

As páginas principais da aplicação *web* são a agenda, perfil, lista de contacto dos clientes e as estatísticas, estando todas elas à distância de um cliqueumas das outras.

A partir da agenda é possível agendar uma nova marcação ou consultar uma já existente. Ao consultar uma marcação, o prestador pode realizar três ações, cancelar, concluir e reagendar uma marcação.

Por sua vez, accedendo ao perfil, o utilizador poderá:

- configurar funcionários, onde é possível adicionar, eliminar ou editar funcionários;

- configurar serviços, onde é possível adicionar, eliminar ou editar serviços;
 - consultar avaliações feitas por clientes;
 - configurar o perfil;
 - alterar o plano de pagamento mensal;

2.6.2 *Mobile*

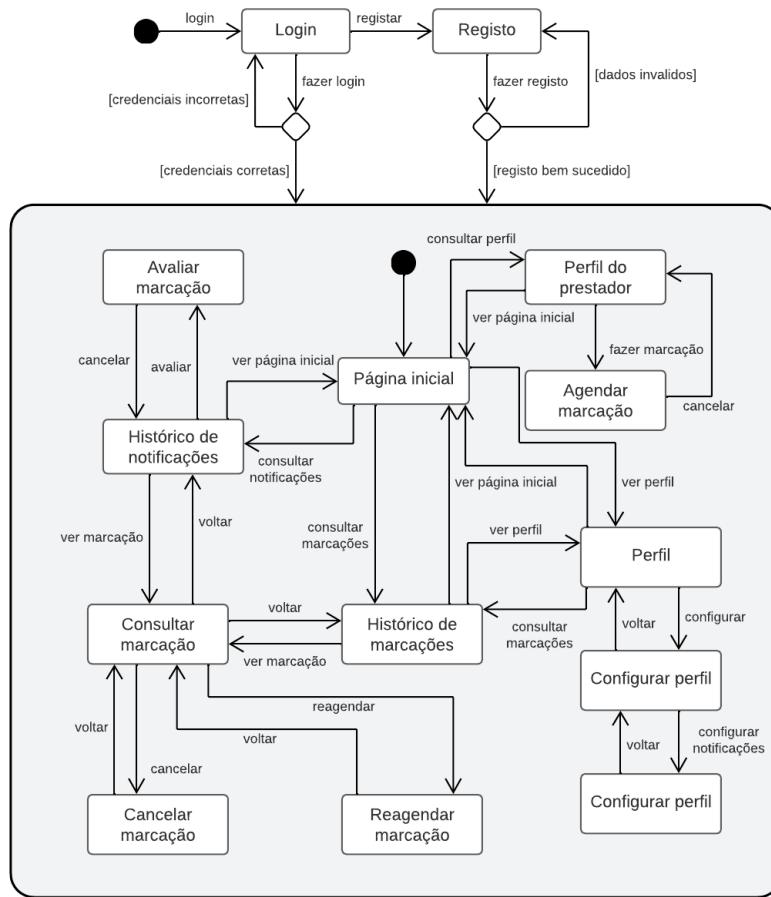


Figura 1.13: Máquina de estados *mobile*

Inicialmente é sempre apresentada a página de *login*, caso o utilizador ainda não esteja registado na aplicação, pode proceder ao mesmo, selecionando a opção de registo. Se o registo ou *login* forem realizados com êxito, será redirecionado para a página inicial.

Uma vez na página inicial, será capaz de optar entre:

- consultar o perfil de um estabelecimento, onde poderá agendar uma marcação;
 - consultar o seu perfil, a partir do qual poderá configurar o mesmo, assim como fazer a configuração das notificações que recebe;

- aceder ao histórico de marcações, onde pode consultar todas as suas marcações;
- aceder ao histórico de notificações, no qual poderá fazer a avaliação de uma marcação concluída ou consultar uma marcação que tenha agendado.

Adicionalmente, quando um cliente se encontra na página de consultar uma marcação, pode cancelar ou reagendar a mesma.

2.7 Modelo de dados

Como seguimento à modelação realizada até ao momento, procedeu-se ao desenvolvimento do modelo de base de dados do sistema, que será nas próximas secções apresentado em detalhe.

2.7.1 Identificação e caracterização das entidades e respetivos atributos

Nesta secção, irão ser apresentadas as entidades do modelo de dados, assim como os respetivos atributos e a sua descrição.

business

- Representa um prestador de serviço.

Atributo	Domínio	Descrição
business_email	varchar(255)	Sequência de caracteres que identificam o email do prestador de serviço.
address	varchar(255)	Sequência de caracteres que identificam o endereço do estabelecimento.
city	varchar(255)	Sequência de caracteres que identificam a cidade do estabelecimento.
name	varchar(255)	Sequência de caracteres que identificam o nome do prestador de serviço.
nif	integer	Sequência de dígitos que identificam o número de contribuinte do estabelecimento.
password	varchar(255)	Sequência de caracteres que identificam a <i>password</i> do prestador de serviço.
phone_number	integer	Sequência de dígitos que identificam o número de telemóvel do estabelecimento.
postal_code	varchar(255)	Sequência de caracteres que identificam o código postal do estabelecimento.

client

- Representa um cliente.

Atributo	Domínio	Descrição
client_email	varchar(255)	Sequência de caracteres que identificam o email do cliente.
city	varchar(255)	Sequência de caracteres que identificam a cidade do cliente.
gender	integer	Sequência de dígitos que identificam o género do cliente.
name	varchar(255)	Sequência de caracteres que identificam o nome do cliente.
password	varchar(255)	Sequência de caracteres que identificam a <i>password</i> do cliente.
phone_number	integer	Sequência de dígitos que identificam o número de telemóvel do cliente.

employee

- Representa um funcionário de um dos prestadores de serviço.

Atributo	Domínio	Descrição
employee_id	integer	Valor inteiro que representa o identificador único de um funcionário.
email	varchar(255)	Sequência de caracteres que identificam o email do funcionário.
job	varchar(255)	Sequência de caracteres que identificam a profissão do funcionário.
phone_number	integer	Sequência de dígitos que identificam o número de telemóvel do funcionário.

notification

- Representa a notificação que um cliente recebe.

Atributo	Domínio	Descrição
id	integer	Valor inteiro que representa o identificador único de uma notificação.
description	varchar(255)	Sequência de caracteres que representam a descrição da notificação.
title	varchar(255)	Sequência de caracteres que representam o título da notificação.

payment_plan

- Representa um dos planos de pagamento que o prestador de serviço pode escolher.

Atributo	Domínio	Descrição
id_plan	integer	Valor inteiro que representa o identificador único de um plano de pagamento.
name	varchar(255)	Sequência de caracteres que representam o nome do plano de pagamento.
price	integer	Valor inteiro que representa preço do plano de pagamento.

service

- Representa um serviço que o prestador de serviço pode oferecer.

Atributo	Domínio	Descrição
service_id	integer	Valor inteiro que representa o identificador único de um serviço.
average_duration	integer	Valor inteiro que representa a duração média do serviço.
category	varchar(255)	Sequência de caracteres que indicam a categoria do serviço.
name	varchar(255)	Sequência de caracteres que representam o nome do serviço.
price	real	Valor real que representa o preço do serviço.

appointment

- Representa uma marcação feita por um cliente.

Atributo	Domínio	Descrição
id_appointment	integer	Valor inteiro que representa o identificador único de uma marcação.
end_date	timestamp	Representa a data e hora de início da marcação.
price	real	Valor real que representa o preço da marcação.
start_date	timestamp	Representa a data e hora de fim da marcação.
status	varchar(255)	Sequência de caracteres que representam o estado da marcação.

2.7.2 Modelo lógico da base de dados

Na imagem abaixo pode-se consultar o modelo lógico da base de dados construída.

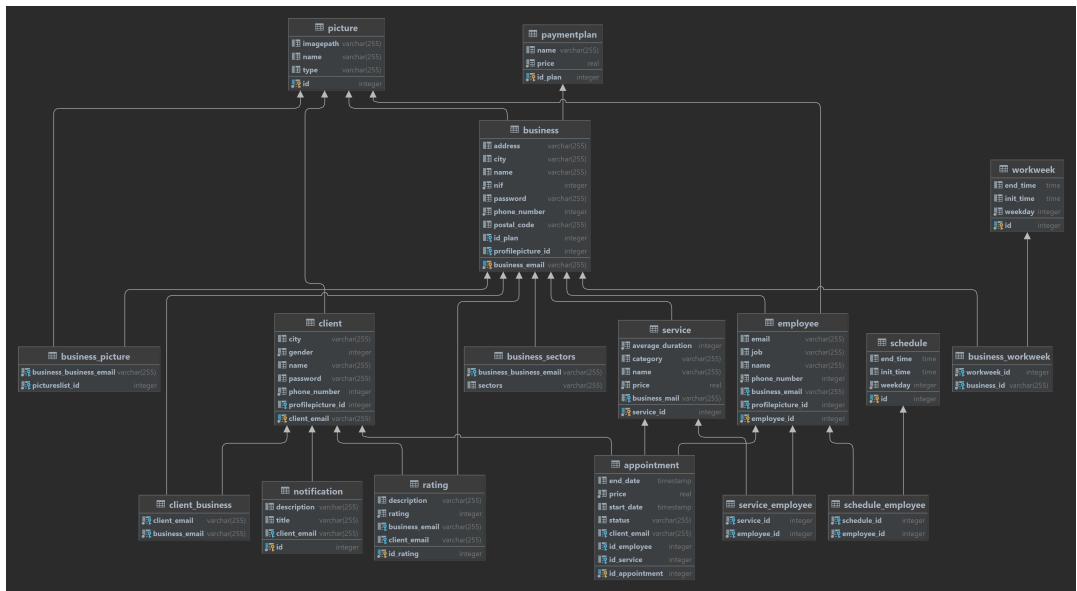


Figura 1.14: Modelo lógico da base de dados

2.8 Apresentação das tabelas e do espaço ocupado por registo

appointment

Atributo	Domínio	Tamanho em bytes
id_appointment	integer	4
end_date	timestamp	8
price	real	4
start_date	timestamp	8
status	varchar(255)	1 a 255
client_email	varchar(255)	1 a 255
id_employee	integer	4
id_service	integer	4

Tamanho, máximo, por registo: 542 bytes

business

Atributo	Domínio	Tamanho em bytes
business_email	varchar(255)	1 a 255
address	varchar(255)	1 a 255
city	varchar(255)	1 a 255
name	varchar(255)	1 a 255
nif	integer	4
password	varchar(255)	1 a 255
phone_number	integer	4
postal_code	varchar(255)	1 a 255
id_plan	integer	4
profilepicture_id	integer	4

Tamanho, máximo, por registo: 1546 bytes

business_picture

Atributo	Domínio	Tamanho em bytes
business_business_email	varchar(255)	1 a 255
picturelist_id	integer	4

Tamanho, máximo, por registo: 259 bytes

business_sectors

Atributo	Domínio	Tamanho em bytes
business_business_email	varchar(255)	1 a 255
sectors	varchar(255)	1 a 255

Tamanho, máximo, por registo: 510 bytes

business_workweek

Atributo	Domínio	Tamanho em bytes
business_id	varchar(255)	1 a 255
workweek_id	integer	4

Tamanho, máximo, por registo: 259 bytes

client

Atributo	Domínio	Tamanho em bytes
client_email	varchar(255)	1 a 255
city	varchar(255)	1 a 255
gender	integer	4
name	varchar(255)	1 a 255
password	varchar(255)	1 a 255
phone_number	integer	4
profilepicture_id	integer	4

Tamanho, máximo, por registo: 1032 bytes

client-business

Atributo	Domínio	Tamanho em bytes
client_email	varchar(255)	1 a 255
business_email	varchar(255)	1 a 255

Tamanho, máximo, por registo: 510 bytes

employee

Atributo	Domínio	Tamanho em bytes
employee_id	integer	4
email	varchar(255)	1 a 255
job	varchar(255)	1 a 255
phone_number	integer	4
business_email	varchar(255)	1 a 255
profilepicture_id	integer	4

Tamanho, máximo, por registo: 777 bytes

notification

Atributo	Domínio	Tamanho em bytes
id	integer	4
description	varchar(255)	1 a 255
title	varchar(255)	1 a 255
client_email	varchar(255)	1 a 255

Tamanho, máximo, por registo: 769 bytes

payment_plan

Atributo	Domínio	Tamanho em bytes
id_plan	integer	4
name	varchar(255)	1 a 255
price	integer	4

Tamanho, máximo, por registo: 263 bytes

picture

Atributo	Domínio	Tamanho em bytes
id	integer	4
imagepath	varchar(255)	1 a 255
name	varchar(255)	1 a 255
type	varchar(255)	1 a 255

Tamanho, máximo, por registo: 769 bytes

rating

Atributo	Domínio	Tamanho em bytes
id_rating	integer	4
description	varchar(255)	1 a 255
rating	integer	4
business_email	varchar(255)	1 a 255
client_email	varchar(255)	1 a 255

Tamanho, máximo, por registo: 773 bytes

schedule

Atributo	Domínio	Tamanho em bytes
id	integer	4
end_time	timestamp	8
init_time	timestamp	8
weekday	integer	4

Tamanho, máximo, por registo: 24 bytes

schedule_employee

Atributo	Domínio	Tamanho em bytes
schedule_id	integer	4
employee_id	integer	4

Tamanho, máximo, por registo: 8 bytes

service

Atributo	Domínio	Tamanho em bytes
service_id	integer	4
average_duration	integer	4
category	varchar(255)	1 a 255
name	varchar(255)	1 a 255
price	real	4
business_email	varchar(255)	1 a 255

Tamanho, máximo, por registo: 777 bytes

service_employee

Atributo	Domínio	Tamanho em bytes
service_id	integer	4
employee_id	integer	4

Tamanho, máximo, por registo: 8 bytes

workweek

Atributo	Domínio	Tamanho em bytes
id	integer	4
end_time	timestamp	8
init_time	timestamp	8
weekday	integer	4

Tamanho, máximo, por registo: 24 bytes

2.9 Diagrama de componentes

Após o levantamento das responsabilidades do sistema, tanto através da exploração do domínio como também através da elaboração de *mockups* e modelos de tarefas que, por sua vez, permitiram uma melhor compreensão dos requisitos principais que o sistema deverá cumprir, foi necessária a determinação dos subsistemas que seriam implementados no trabalho. Isto foi necessário pois a criação de subsistemas permite uma maior organização, uma vez que haverá o agrupamento dos métodos pelos subsistemas onde melhor se adequam.

Além disto, esta prática é fundamental na questão do encapsulamento uma vez que cada subsistema implementa uma interface com os métodos que lhe foram atribuídos, impedindo isto o acesso ao seu "core" (o subsistema poderá conter métodos que não estão na interface e que não convém que "entidades" exteriores tenham acesso).

Como se pode verificar, no componente relativo à lógica de negócios (PlannerUpLN), foram definidos dois subsistemas, sendo que cada um deles contém os métodos correspondentes a cada um dos dois tipos de utilizador da nossa aplicação, respetivamente.

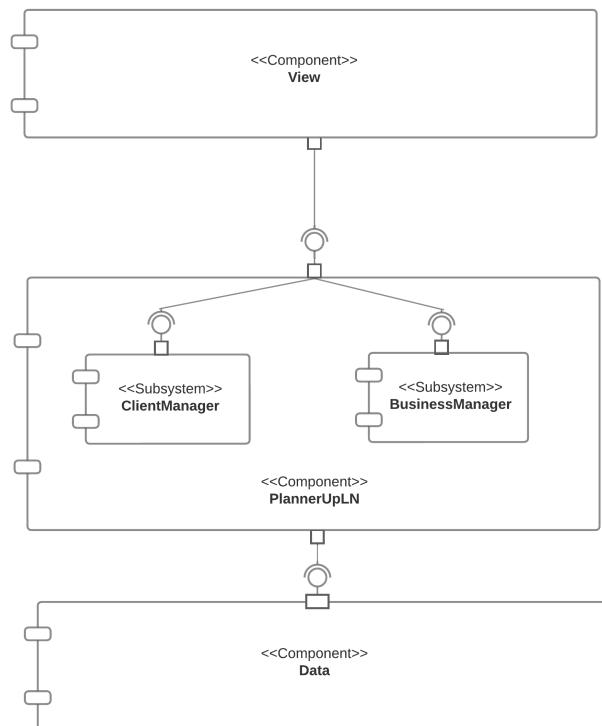


Figura 1.15: Diagrama de Componentes

Componentes	Descrição
View	Componente responsável pela interface gráfica do utilizador.
PlannerUpLN	Componente com a lógica de negócios da aplicação.
Data	Componente que representa a base de dados.

Subsistemas	Descrição
ClientManager	Subsistema da lógica de negócios encarregue de todas as funcionalidades relativas ao utilizador 'Client' da aplicação.
BusinessManager	Subsistema da lógica de negócios encarregue de todas as funcionalidades relativas ao utilizador 'Business' da aplicação.

2.10 Diagrama de *Packages*

Em seguida, após desenvolvimento do diagrama de subsistemas, foi elaborado o diagrama de pacotes, que permite organizar a estrutura e dependências do programa.

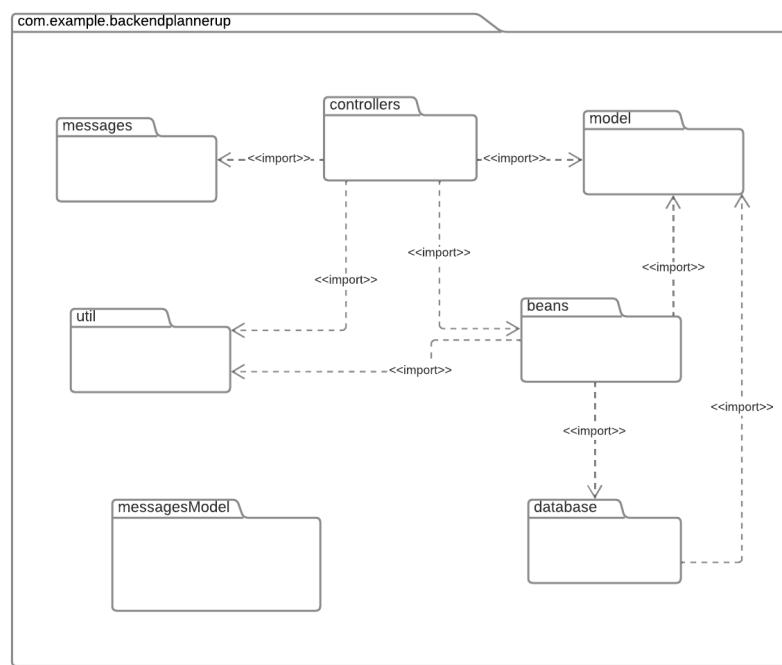


Figura 1.16: Diagrama de *Packages*

Package	Descrição
backendplannerup	Pacote que engloba todos os outros pacotes.
controllers	Contém as classes que estão responsáveis por fazer a comunicação entre os <i>frontends</i> e o <i>backend</i> .
model	Contém todas as classes que correspondem a entidades que são persistidas na base de dados.
messages	Contém as classes auxiliares relativas a mensagens que o <i>backend</i> recebe e envia de e para o <i>frontend</i> .
beans	Este pacote tem as classes que implementam as funcionalidades da aplicação, fazendo a ponte entre os controladores e a base de dados.
util	Contém as classes auxiliares ao desenvolvimento da aplicação, como classes para ajudar no <i>debug</i> .
database	As classes que pertencem a este pacote são responsáveis por aceder à base de dados, mais conhecidas como DAOs (<i>data access objects</i>).
messagesModel	Contém classes auxiliares para a comunicação com o <i>backend</i> , têm classes simplificadas que servem para enviar ao <i>frontend</i> .

2.11 Diagrama de classes

Em seguida será apresentado o diagrama de classes PSM do sistema a ser desenvolvido. Com ele pretende-se destacar como os diferentes componentes do sistema se vão relacionar entre si, bem como o tratamento de cada entidade abordada.

Na secção 4.2 será exposto em maior detalhe as classes e quais as suas funcionalidades.

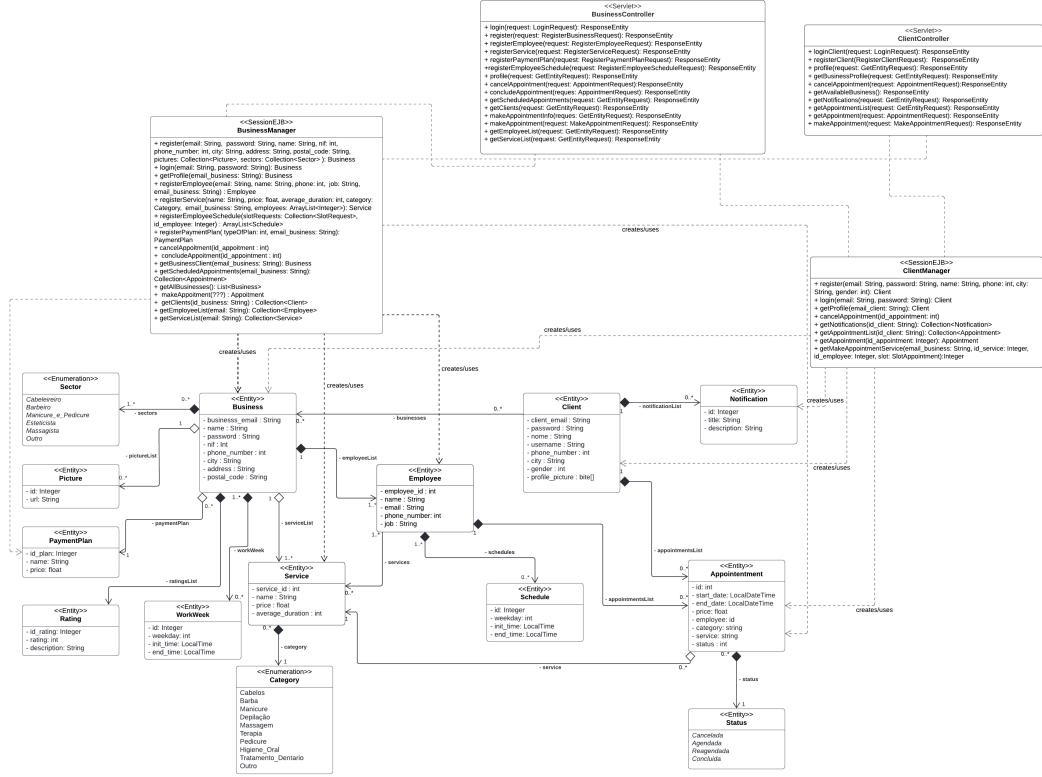


Figura 1.17: Diagrama de classes

3 Arquitetura da Aplicação

Terminada a fase de modelação, passamos ao desenvolvimento da aplicação. Como primeiro passo, começamos por escolher quais as tecnologias e *frameworks* que vamos utilizar para o desenvolvimento da aplicação.

Assim sendo, o grupo começou por traçar a arquitetura da aplicação. Posto do princípio que iríamos implementar uma arquitetura MVC, seriam precisas à partida 3 tecnologias: uma direcionada ao *frontend*, uma para a lógica de negócios e uma terceira para a camada de dados. No entanto, em relação ao *frontend*, uma vez que estamos perante dois tipos diferentes, isto é, uma interface mobile e outra web o grupo decidiu utilizar tecnologias diferentes para cada uma.

Começando pela camada de dados, a escolha do **Hibernate** foi direta, dada a sua grande utilização no mercado e visto que alguns elementos do grupo já trabalharam com a mesma anteriormente.

Já para a lógica de negócios, depois de uma pesquisa sobre as várias *frameworks* que poderíamos utilizar no projeto, optamos por utilizar **Spring**, devido a fatores como a sua importância no mercado, à sua curva de aprendizagem e a existirem elementos da equipa que já tinham experiência com a mesma.

Já para o *frontend*, no que toca, à interface Web, recorremos à utilização do **Vue.js**,

devido à sua baixa curva de aprendizagem. Relativamente à interface mobile, foi escolhida a framework **React Native**, uma vez que é uma tecnologia muito utilizada para este tipo de interfaces e possui uma documentação bastante completa, o que se torna útil visto que nenhum elemento do grupo tinha experiência com a mesma.

Assim, para sumarizar, a arquitetura que nos propomos implementar é a seguinte:

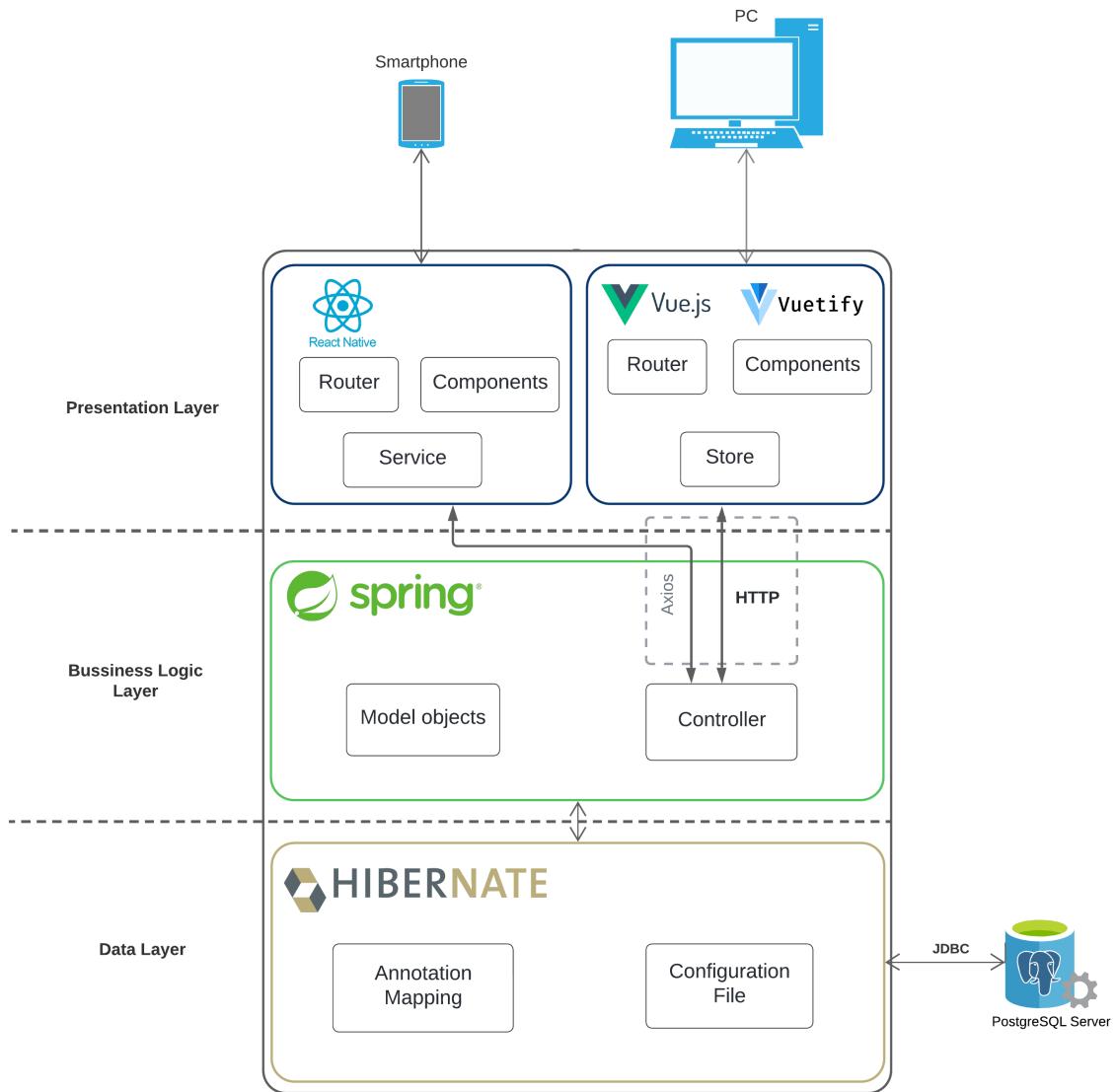


Figura 1.18: Arquitetura da aplicação

4 Implementação

Antes de mais, foi necessário escolher as linguagens de programação a utilizar. Para a camada de negócio, optou-se por utilizar **Java**, devido a se tratar de uma linguagem estável, robusta e bastante utilizada na indústria, nomeadamente, em aplicações web.

Já na camada de apresentação, decidiu-se usar **JavaScript** devido à familiaridade de alguns membros da equipa com a mesma, assim como a sua curva de aprendizagem e comunidade ativa.

É de notar que o grupo, dividiu o sistema em três aplicações distintas, uma destinada ao **backend** e as outras duas ao **frontend**, *mobile* e *web*. Por este meio, foi possível que o grupo trabalhasse paralelamente em cada uma das partes da aplicação. Além disso, permite que futuramente, caso o grupo ache necessário, seja possível mudar a tecnologia de qualquer uma das camadas sem impactar as restantes. É também de realçar que a arquitetura implementada garante a capacidade de escalabilidade da aplicação, sendo possível fazer réplicas dadas componentes que forem necessárias, deste modo, evita-se a replicação total.

Nas secções seguintes será abordado em maior detalhe o processo de desenvolvimento de cada uma das componentes.

4.1 Frontend

4.1.1 Web

Para o desenvolvimento da camada de apresentação optamos pela framework em **Vue.js**.

Views

Para cada página gerou-se views independentes, que utilizam também os componentes gerados.

- *login-form.vue* - form de login.
- *login-page.vue* - página de login.
- *forgot-password-page.vue* - página de "esqueci a minha palavra-passe".
- *reset-password-page.vue* - página para dar reset à palavra-passe.
- *business-agenda.vue* - página da agenda da empresa.
- *business-new-appointment.vue* - página para criação de uma nova marcação.
- *business-profile.vue* - página de perfil da empresa.
- *business-statistics.vue* - página das estatísticas da empresa.
- *contact-page.vue* - página dos contactos da empresa.
- *faq-page.vue* - página das perguntas frequentes.
- *features-page.vue* - página das features do serviço.
- *home-page.vue* - homepage.
- *pricing-page.vue* - página do preçoário do serviço.

- *add-employers.vue* - página onde são adicionados os funcionários da empresa.
- *add-services.vue* - página onde são adicionados os serviços que a empresa pode prestar aos seus clientes.

Componentes

Durante o desenvolvimento da interface, de modo a evitar código repetido entre diferentes páginas recorreu-se à utilização de **componentes**. Estes componentes representam blocos de código que podem ser utilizados em qualquer página sem a necessidade de repetir o código do mesmo. Desta forma os componentes utilizados foram:

- **footer-tab**: Representa o *footer* da interface que está presente em todas as páginas.
 - **nav-bar**: Representa a barra de navegação superior presente na página inicial, login e registo da aplicação
 - **side-bar**: Representa a barra lateral que é apresentada nas páginas após o *business* completar o seu registo ou login.
 - **card-profile**: uma estrutura de formato específico, que pode ser usada para representar um título e certas features.
 - **store-banner**: um banner personalizável para a App Store e a Play Store.
- Entre outros.

Router

Uma das funcionalidades que o *Vue.js* fornece é a utilização do **router**, que permite navegar entre páginas de forma fácil, associando a cada *view* um *path*. Para além disto, o *router* possibilita que seja criado um histórico de páginas pelas quais o utilizador navegou, assim como implementar a autorização dos utilizadores relativamente ao acesso a determinadas páginas, como por exemplo, utilizadores não autenticados acederem a páginas específicas do *business*.

Store

Recorrendo à biblioteca **Vuex** recorreu-se à utilização da sua *Store* para gerir o estado da aplicação. Desta forma foi possível armazenar todos as informações necessárias, nomeadamente dados relativos ao utilizador autenticado como o seu email ou *token* os quais serão essenciais em chamadas ao *backend*. Para além disso, de modo a garantir que os dados são guardados de forma persistente recorreu-se ao *plugin vuex-persistedstate*.

Pedidos ao Backend

De modo implementar a conexão ao *backend*, utilizou-se a biblioteca **Axios**. Esta permitiu efetuar pedidos *http* o que possibilitou aceder aos métodos presentes na *REST API* disponibilizada pelo *backend*.

4.1.2 Mobile

Nesta secção será descrito quais as principais decisões que a equipa tomou relativamente à implementação da aplicação *mobile*, bem como a lógica por trás desta. Além disso, será apresentada a estrutura adotada para a organização do código.

Considerações Iniciais

Para a construção da aplicação em React Native foi usada a framework Expo. Esta contém várias ferramentas que têm como objetivo auxiliar os testes feitos ao código e que também providenciam um desenvolvimento mais flexível e robusto. Para efetuar o deployment é muito mais simples comparativamente com a aplicações que não usam esta framework. Além disso, é de notar que o código é reutilizável uma vez que tanto pode ser usado para iOS como para android. Caso não usássemos a framework seria necessário fazer código específico para cada um dos sistemas operativos.

Infraestrutura

Para a organização dos *packages* no frontend mobile, seguiu-se a lógica representada na Figura 1.19. Sendo que a pasta ”/assets” contém as figuras e icons necessários; a pasta ”/themes” contém o esquema de cores usado bem como os tipos de letra; a pasta ”/containers” contém os ecrãs que derivam dos ecrãs principais; a pasta ”/stores” contém as variáveis globais que são usadas em toda a aplicação; a pasta ”/routes” contém todos os componentes relacionados com a navegação dentro da aplicação; a pasta ”/screens” contém os ecrãs principais da aplicação; a pasta ”/notifications” contém os métodos relacionados com as notificações; e por fim, a pasta ”/components” contém os componentes da aplicação, tais como botões, *dropdowns*, etc. que são reutilizados pelos vários containers e screens.

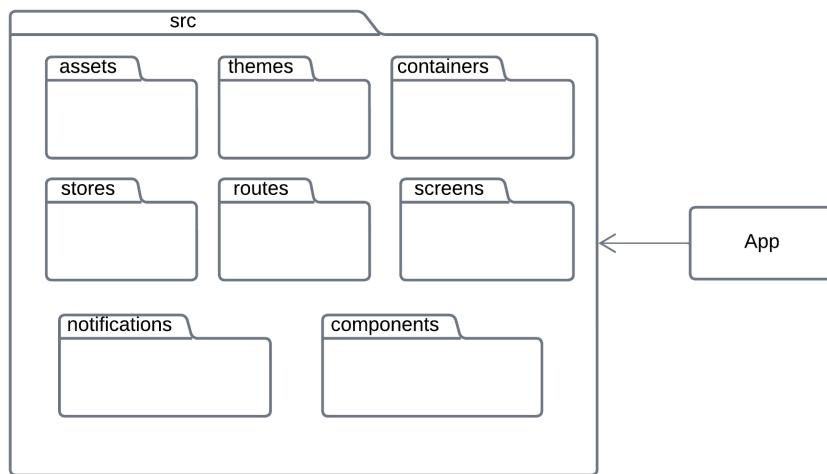


Figura 1.19: Diagrama de *Packages*

Navigation

A navegação entre os ecrãs da aplicação é feita através de uma biblioteca chamada "react-navigation". Os dois componentes usados são o "Tab.Navigator" e o "Stack.Navigator".

O primeiro permite criar uma barra com separadores que permite ao utilizador navegar entre esses ecrãs. Já o segundo, cria uma "stack" de ecrãs permitindo ao utilizador navegar para o ecrã seguinte ou voltar ao anterior.

Store

Este tópico é relativo ao armazenamento de variáveis e valores, aos quais é necessário aceder ao longo do uso da aplicação. Posto isto, é usada então a biblioteca "AsyncStorage", de forma a armazenar valores tais como as credenciais do utilizador que se registou ou fez login, e o token do dispositivo em que o utilizador iniciou sessão, que é usado para enviar notificações push. Assim, é possível identificar qual o utilizador que está a usar a aplicação.

É de ter em atenção que existem dois métodos para gerir a *asyncStorage* da aplicação:

- *storeData*: Este método é responsável por armazenar valores. Neste caso em relação ao identificador do utilizador.

```
export const storeData = async (value) => {
  try {

    await AsyncStorage.setItem("user", value)
  } catch (e) {
    // saving error
  }
}
```

- *getData*: Este método é responsável por ler os valores armazenados. Tal como o caso acima, o exemplo está a ler o valor do identificador do utilizador.

```
export const getData = async () => {
  try {
    const value = await AsyncStorage.getItem("user")
    if (value !== null){
      // We have data!!
      console.log("STORAGE:" + value)

    }
  } catch(e) {
    // error reading value
  }
}
```

Push Notifications

Para o dispositivo ser capaz de receber notificações *push*, foi necessário implementar um *listener* através da biblioteca "expo-notifications". Este é mais um exemplo de como a framework Expo simplifica a implementação de várias funcionalidades.

Posto isto, implementou-se dois listeners, um que fica à escuta para receber uma notificação do backend enquanto a aplicação está a correr em "background" e outro que fica à escuta e é ativado quando o utilizador interage com a notificação, por exemplo, quando clica na notificação é direcionado para uma página da aplicação.

A partir da figura 1.20 é possível entender o funcionamento das notificações push, e qual o processo entre o backend e o frontend. Primeiramente, quando um utilizador inicia sessão na aplicação, o dispositivo é registado e é atribuído um token único a esse mesmo dispositivo. Posteriormente, esse token é armazenado pelo backend, para que depois este consiga enviar notificações específicas para cada utilizador e o dispositivo correspondente.

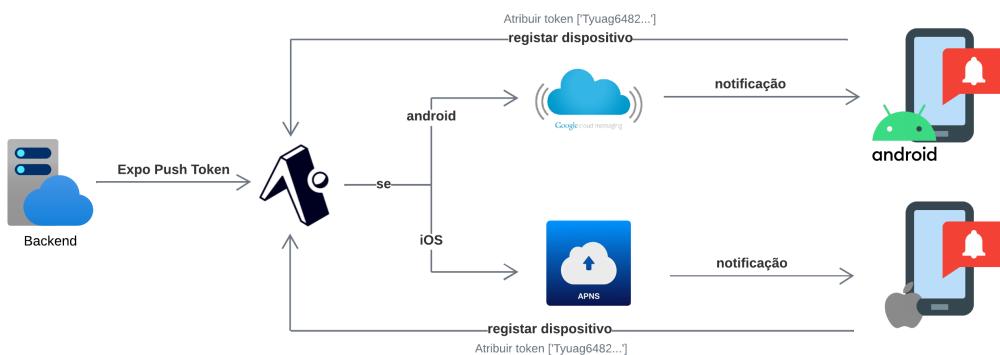


Figura 1.20: Diagrama ilustrativo das *notificações-push*

Pedidos ao Backend

Para fazer os pedidos ao backend, foi utilizada a biblioteca *Axios*. Assim, são efetuados pedidos http, os principais métodos utilizados são o GET e o POST.

De seguida, estão representados dois exemplos destes pedidos:

- GET

```
...() => {
  axios
    .get(url)
    .then( ( response ) => {
      console.log("Result:");
      setResult(response.data);
      setIsLoading(false);
    })
    .catch(error => {
      console.log(error);
      setIsLoading(false);
      setError(error);
    })
}
```

- POST

```
() => {
    axios
        .post(url, {
            email : value
        })
        .then( (response) => {
            console.log("SUCESSO")
            setName(response.data.name);
            setIsLoading(false);
        })
        .catch(error => {
            console.log(error);
            console.log("ERRO RESPOSTA");
            setIsLoading(false);
            setError(error);
        })
}
```

4.2 Backend

Para conectar o frontend com o backend, implementamos uma Rest API, de forma a responder a qualquer pedido que seja feito pelo cliente no frontend. Desta forma, estes componentes são independentes e apenas utilizam uma API conhecida para comunicar.

4.2.1 *Controllers*

Como já foi referido, a framework utilizada para a implementação da lógica de negócio foi a Spring, na qual os *HTTP Request* são tratados por **Rest Controllers**.

A equipa procedeu então à implementação de dois *controllers*, um que recebe pedidos do *frontend web* (*BusinessController*) e outro para o *mobile* (*ClientController*). Estes, para além de receberem os pedidos, fazem também o seu mapeamento no *backend* (*session beans*), de forma a obterem as respostas adequadas e reencaminham-nas para o *frontend* respetivo.

4.2.2 *Beans*

Entity beans

Os entity beans representam entidades guardadas em suporte persistente, armazenadas na base de dados. Os entity beans presentes na aplicação são:

- *Appointment*: Classe que representa a entidade de marcação.
- *Business*: Classe que representa um prestador de serviço.
- *Client*: Classe que representa um cliente.

- *Employee*: Classe que representa um funcionário.
- *Notification*: Classe que representa uma notificação.
- *PaymentPlan*: Classe que representa um plano de pagamento.
- *Picture*: Classe que representa uma foto.
- *Rating*: Classe que representa uma avaliação.
- *Schedule*: Classe que representa um horário de funcionário.
- *Service*: Classe que representa um serviço.
- *WorkWeek*: Classe que representa horário de funcionamento de um prestador de serviço.

Session beans

O outro tipo de EBJ implementado foi **Session beans**. Estes gerem processos ou tarefas, implementam a lógica de negócio, estabelecendo os relacionamentos entre as várias entidades do sistema e fazendo pedidos aos DAOs que comunicam com a base de dados.

Depois de deliberar sobre quantos *session beans* deveriam ser utilizados e quais as responsabilidades de cada um deles, a equipa optou por implementar três, que passam a ser apresentados:

- *BusinessManager*: Classe responsável por todas as operações relacionadas com os prestadores de serviços. Algumas das operações implementadas neste *bean* são: o registo e *login* de um prestador, registo de funcionários e serviços e agendar, cancelar e concluir uma marcação.
- *ClientManager*: Classe responsável por todas as operações relacionadas com os clientes. Algumas das operações implementadas neste *bean* são: o registo e *login* de um cliente, registo de funcionários e serviços, devolver o histórico de marcações de um cliente e agendar, reagendar e cancelar uma marcação.
- *NotificationManager*: Classe responsável pela operações relacionadas com o envio de notificações *push* para a aplicação do cliente. Para implementar esta classe foi utilizado o *Expo Server SDK* que permite enviar notificações *push* a partir de código Java para uma aplicação mobile Expo, onde corre a aplicação do cliente. Esta classe envia todo o tipo de notificações que fazem parte do programa, incluindo os lembretes de marcação que acontecem 1 dia antes da marcação e 1h antes da mesma.
- *SMSManager*: Classe responsável pelas operações relacionadas com o envio de notificações SMS para os clientes que não possuem a aplicação do cliente. Para isso foi utilizado o sistema Twilio que permite o envio de SMS a partir de um número criado pelo o Twilio que representa o nosso sistema. Desta forma os clientes que não possuem a aplicação conseguem receber notificações mesmo não estando registo no sistema.

4.2.3 Base de dados

Relativamente à camada de dados, como já foi dito anteriormente, optamos por utilizar o **Hibernate**, uma ferramenta ORM *open source*. Esta ferramenta é a mais reconhecida no mercado, dado o seu alto desempenho e confiabilidade. A configuração do *Hibernate* pode ser feita utilizando ficheiros de configuração ou anotações. A equipa decidiu utilizar anotações devido a considerar que é a abordagem mais intuitiva.

Para a base de dados, optou-se por utilizar o **PostgreSQL** como sistema de gestão de base de dados (RDBMS).

Na lógica de negócio, todos os acessos aos dados localizados na base de dados são feitos através de **DAOs** (*Data Access Objects*) geridos pelo *Hibernate*.

5 Deployment

Para *deployment* do sistema foi utilizado a **Amazon Web Services (AWS)**, uma plataforma de serviços de computação em nuvem oferecida pela Amazon.com. Desta forma é possível que todo o sistema de base de dados, *backend* e *frontend* seja acedido por qualquer cliente em qualquer localização quando este está online.

5.1 Deployment Base de dados

Para o deployment da base dados foi utilizado o serviço **Relation Database Service** (RDS) da AWS que permite a criação de uma instância de base de dados Postgres que é facilmente acedida através do backend para assim armazenar toda a informação do sistema.

5.1.1 Imagens

As fotos do negócio, dos funcionários e dos clientes são juntamente armazenadas no AWS fora da base de dados para que o acesso a estas seja mais eficiente e separado do acesso aos dados do sistema. Para o *deployment* destas imagens utilizou-se o serviço **Scalable Storage** (S3) que permite a criação de uma pasta para o armazenamento de ficheiros, neste caso de imagens. Estas imagens ficam assim acessíveis partir de um link público que é criado por este serviço S3, sendo este link armazenado na base de dados.

5.2 Deployment Backend

Para que a parte lógica do sistema seja acedido pelos os seus utilizadores é também necessário fazer o *deployment* do *backend*. Para isso foi utilizado o serviço de **Virtual Servers** (EC2) que permite a criação de uma máquina virtual que executa continuamente o *backend*, funcionando como um servidor. O *frontend* faz ligação com esta máquina virtual através de um link fixo que foi criado pelo o EC2 para assim obter os dados e informações do *backend*.

5.3 Deployment Frontend

5.3.1 WEB

Para expor a interface de utilizador do lado da aplicação Web destinada aos prestadores de serviços foi necessário realizar o *deployment* do *frontend* desenvolvido em *Vue.js*. Sendo assim, tal como o *backend* utilizando as máquinas virtuais do *AWS* (*EC2*). Desta forma o utilizador acede à aplicação através do *ip* fixo da máquina onde foi instalado o *frontend* que por sua vez redireciona os pedidos ao *backend* de modo a obter os dados quando for necessário.

5.3.2 Mobile

Como foi usada a framework *Expo*, o deployment da aplicação é muito simples, sendo apenas necessário correr o seguinte comando ”*expo publish*”. A aplicação será então publicada no repositório *Expo*, onde qualquer pessoa tem acesso à mesma, desde que instale a aplicação *Expo-cli* da *playstore*. No final será fornecido, um QR-Code que basta ler para ser direcionado para a aplicação *Planner-Up*.

Mais tarde, após uma versão atualizada da aplicação, a aplicação será publicada na Google Playstore e na Apple AppStore, já que esta está disponível tanto para Android como para iOS.

6 Anexos

6.1 Protótipos

6.1.1 Web

Sobre

The screenshot shows the homepage of the Planner Up website. At the top, there's a large banner with two columns of features. The left column includes sections for 'AGENDA DIGITAL' (with a note about automating agendas), 'PODER DA PLATAFORMA DIGITAL' (mentioning a digital platform for services), 'AVALIAÇÕES' (customer reviews), and 'ESTATÍSTICAS' (activity statistics). The right column includes 'PLATAFORMA DIGITAL' (described as a virtual shop), 'NOTificações' (notifications), 'AVAILABILIDADES' (availability), and 'CALENDÁRIO' (calendar). Below the banner is a navigation bar with links for 'Home', 'Sobre', 'Preços', and social media icons. At the bottom, there's a footer with copyright information, a 'PLANNER UP' logo, and links for 'Terms of Service' and 'Privacy Policy'.

Preços

The screenshot shows the pricing section of the Planner Up website. It features two main plans: 'FREE' on the left and 'Standard' on the right. Both plans have a circular icon above them. The 'FREE' plan is labeled 'FREE' and shows a price of '0 \$ Per Month'. It includes a list of features with green checkmarks: 'Unlimited product updates', 'Unlimited product updates', 'Unlimited product updates', '1GB Cloud storage', and 'Email and community support'. A 'Try for free' button is at the bottom. The 'Standard' plan is labeled 'Standard' and shows a price of '0 \$ Per Month'. It includes a list of features with green checkmarks: 'Organize across all apps by hand', 'Slate helps you see how many more days you need to work to reach your financial goal for the month and year.', and a 'Try for free' button at the bottom.

Plano de pagamento - Registro

The screenshot shows the Pricing section of the Planner Up website. At the top, there are three navigation links: Home, Sobre, and Preços. Below them are two plan options:

- FREE**
Organize across all apps by hand
0 \$ Per Month
 - Unlimited product updates
 - Unlimited product updates
 - Unlimited product updates
 - 1GB Cloud storage
 - Email and community support

[Select](#)

[Retroceder](#)
- Standard**
Organize across all apps by hand
0 \$ Per Month

Slate helps you see how many more days you need to work to reach your financial goal for the month and year.

[Select](#)

[Continuar](#)

Método de pagamento - Registro

The screenshot shows the Payment registration page. At the top, there are three navigation links: Home, Sobre, and Preços. Below them is a heading and form fields:

Pagamento

Cartão
 P M C VISA

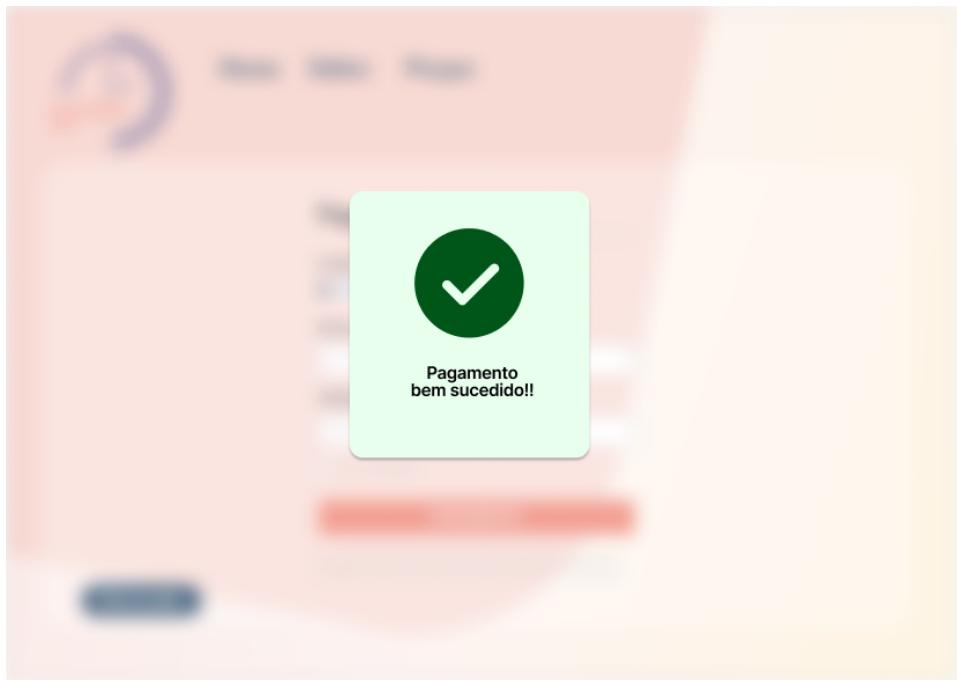
Número de Cartão
1234 5678 9101 1121

Validade **CVV**
MM/YY 123

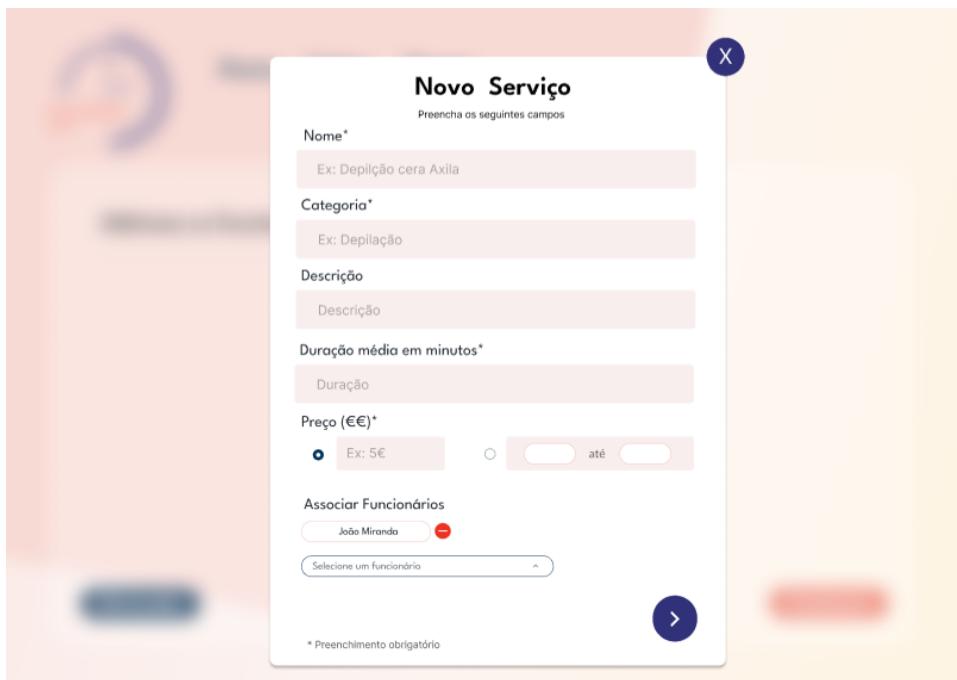
[PAGAMENTO](#)

Your personal data will be used to process your order, support your experience throughout this website, and for other purposes described in our privacy policy.

[Retroceder](#)



Associar funcionário ao serviço



Filtrar agenda por funcionário

The screenshot shows a weekly calendar from Sunday, 21 to Saturday, 27. The agenda includes the following appointments:

- Sunday, 21:** 3:00 PM Fátima - Depilação completa
- Tuesday, 23:** 10:00 AM Mariana - Buço; 9:00 AM Mariana - Depilação Completa
- Wednesday, 24:** 10:00 AM Luciana - Axilas; 10:00 AM Joana - Penteado + Penas + Vírginhas
- Thursday, 25:** 4:00 AM Cláudia - Depilação Completa
- Friday, 26:** 8:00 AM Cristina Ferreira - Depilação no dedo grande do pé; 10:00 AM Sônia - Corte + Salage; 11:00 AM Joana - Depilação Buço
- Saturday, 27:** 8:00 AM Tânia - Depilação perna completa; 10:45 AM Marcelo Sousa - Corte; 11:00 AM Maria Antunes - Depilação Axila; 10:45 AM Joaquina - Corte; 11:45 AM Joálio Mendes - Corte; 11:00 AM Sônia - Corte + Salage; 11:45 AM Joana - Depilação perna completa; 10:45 AM Joaquina - Corte

A "Nova Marcação" (New Booking) button is located at the bottom right.

Cancelar marcação

The screenshot shows a weekly calendar with a booking for "Maria Júlia" highlighted. A modal window asks if the user wants to cancel the booking. The booking details are:

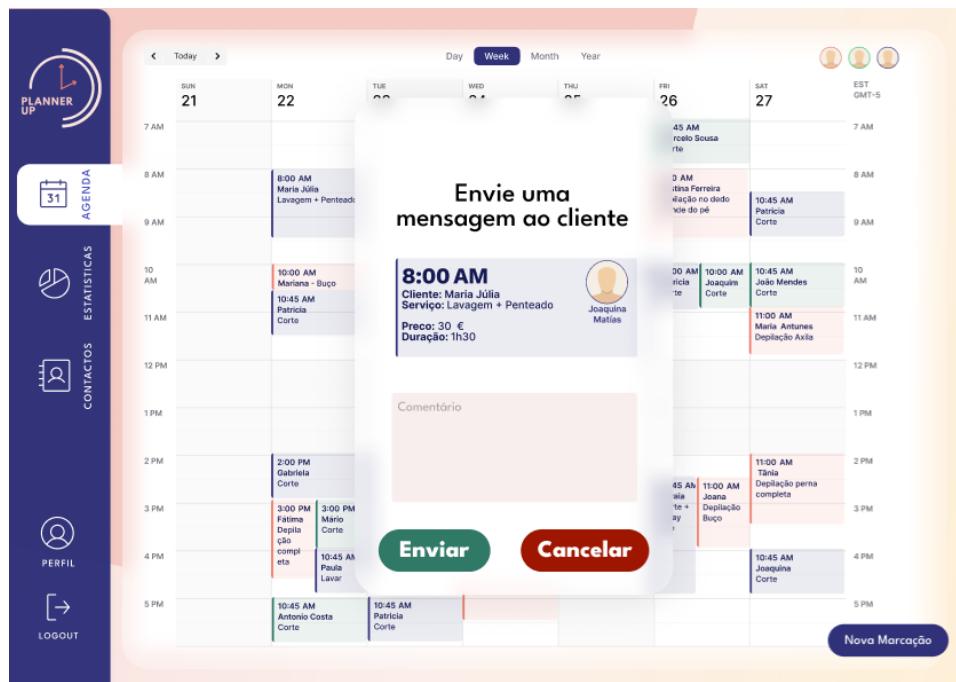
8:00 AM
Cliente: Maria Júlia
Serviço: Lavagem + Penteado
Preço: 30 €
Duração: 1h30

Mensagem de Cancelamento

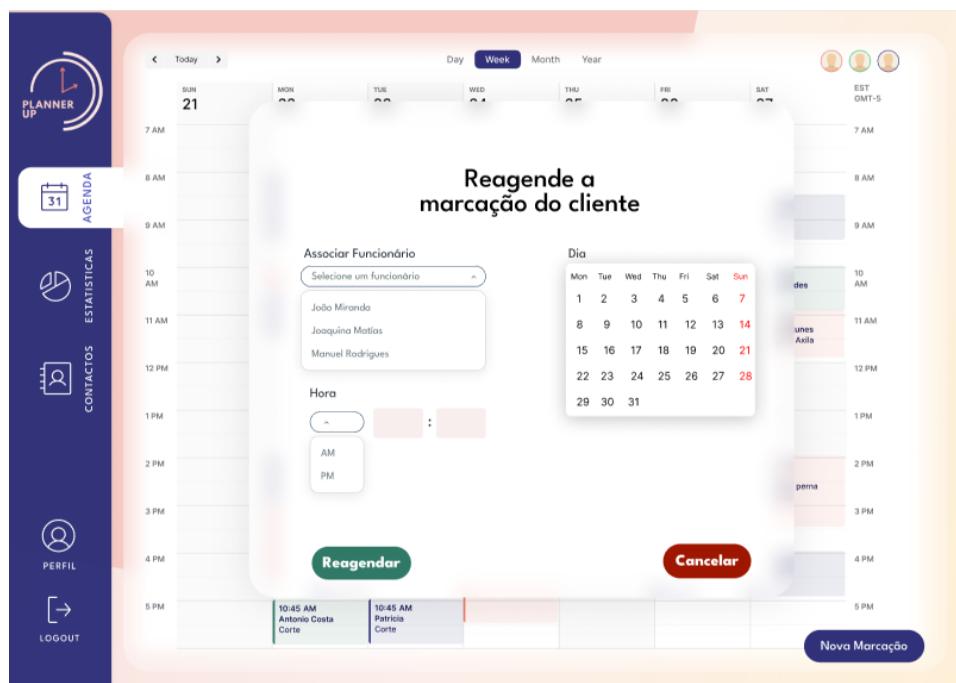
Enviar **Cancelar**

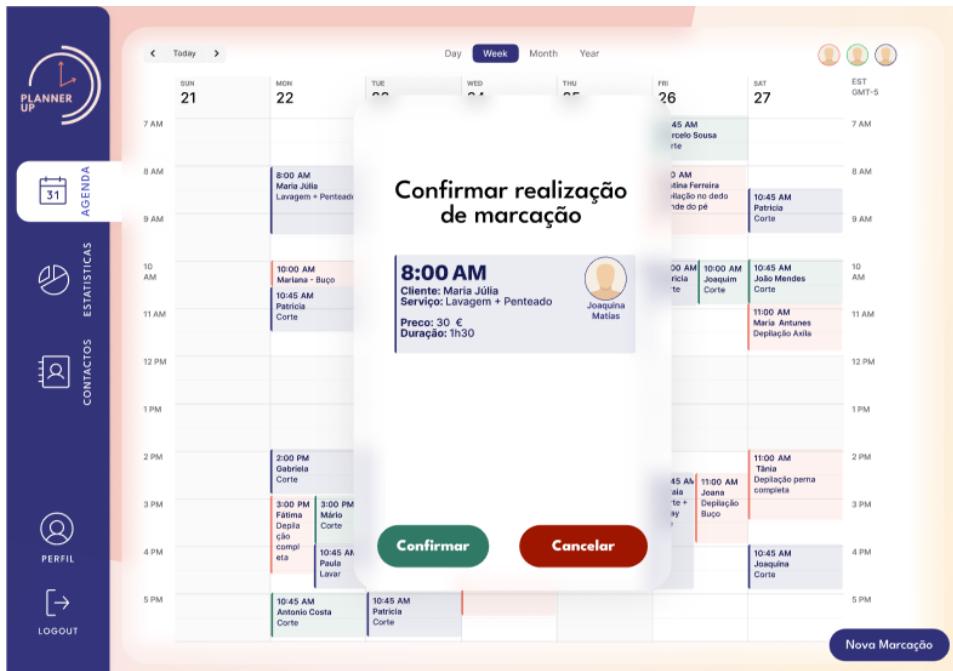
A "Nova Marcação" (New Booking) button is located at the bottom right.

Mensagem marcação

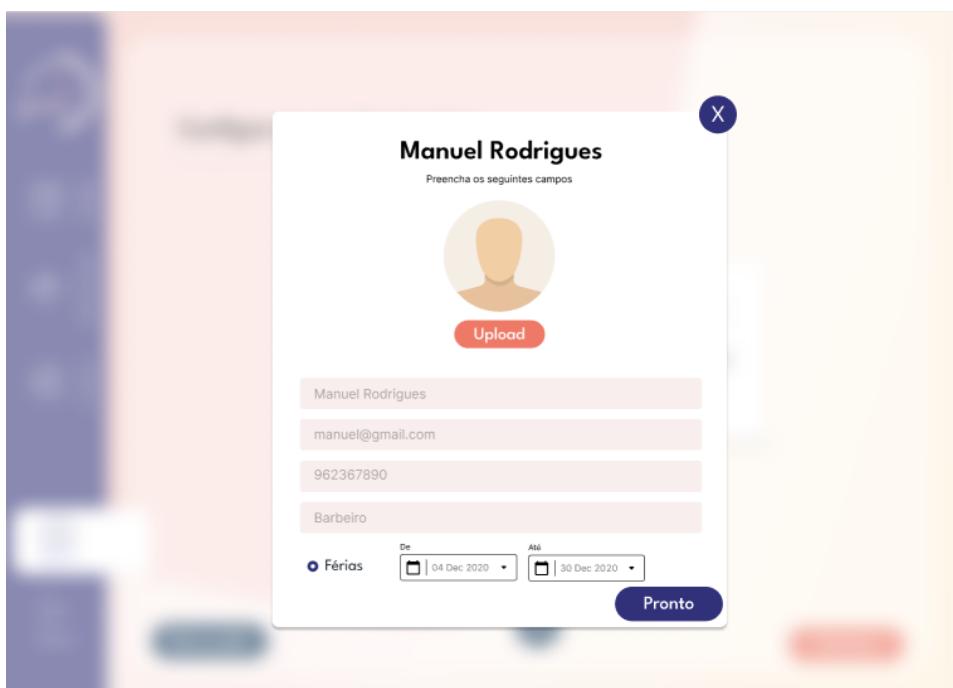


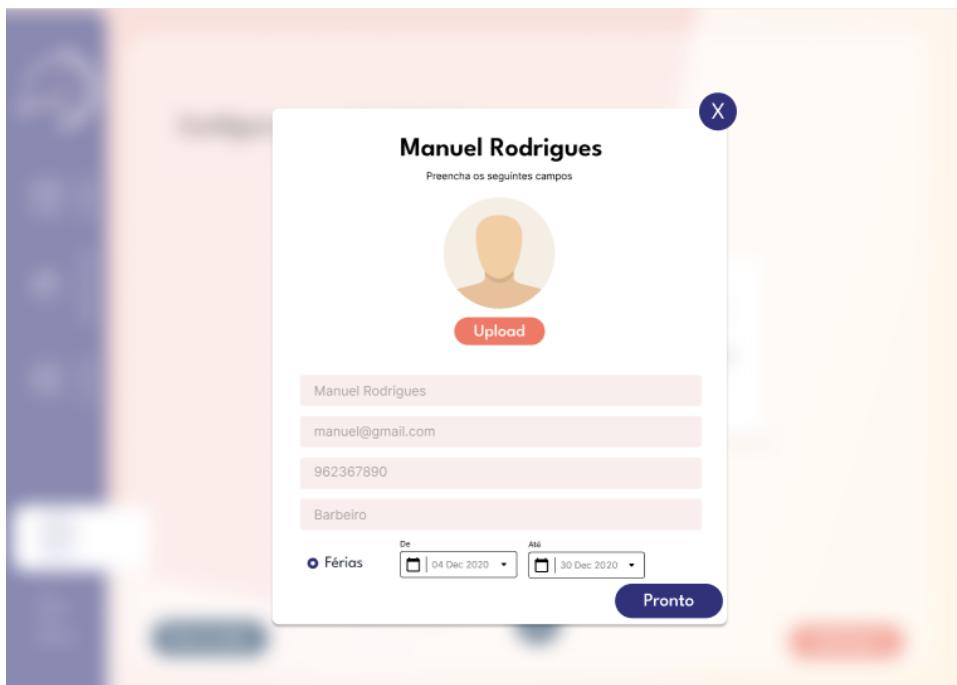
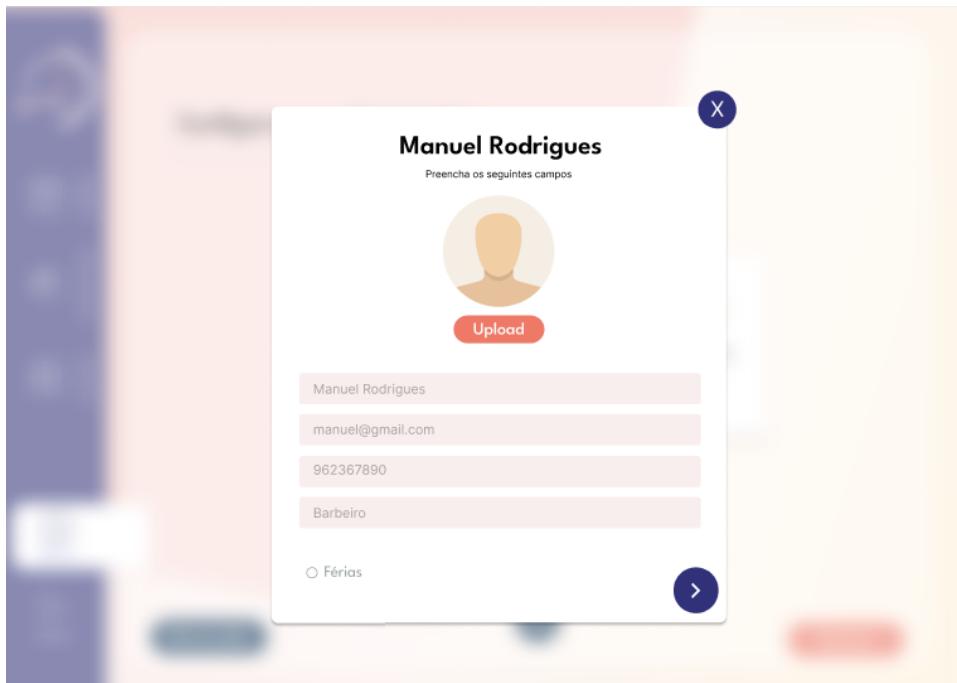
Reagendar marcação

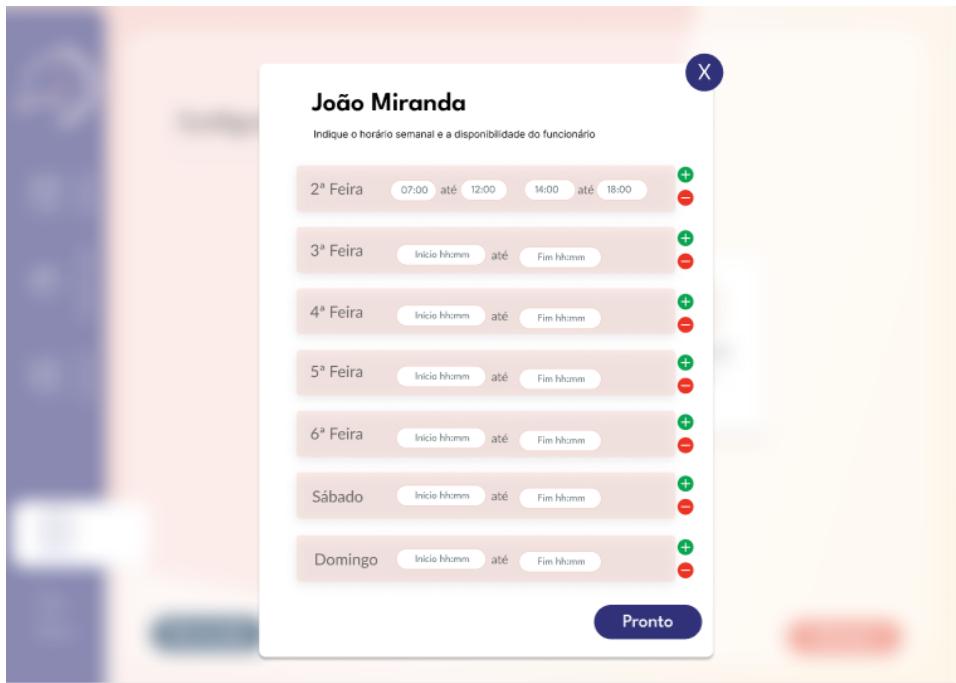




Configurar funcionários







Configurar serviços

The screenshot shows a service configuration interface titled "Configure os seus serviços". It lists various services with their descriptions, times, and costs. Each service has a red trash icon at the top right. Below each service are lists of staff members assigned to it.

Service	Description	Time	Cost	Staff
Cabelos	Corte	1 hora	20€	Joaquina Matias Manuel Rodrigues
Cabelos	Pintar	1:30 hora	60€	Joaquina Matias Manuel Rodrigues
Barbearia	Barba	30 min	20€	Manuel Rodrigues
Barbearia	Barba	30 min	20€	Manuel Rodrigues
Barbearia	Barba	30 min	20€	Manuel Rodrigues
Barbearia	Barba	30 min	20€	Manuel Rodrigues
Depilação	Depilação Axilas	10 min	5€	João Miranda

Buttons at the bottom include "Retroceder", a blue "+", and "Gravar".

Edição Serviço

Preencha os seguintes campos

Nome*
Corte

Categoria*
Cabelos

Descrição
Descrição

Duração média em minutos*
1 hora

Preço (€€)*
20 €

Associar Funcionários*
Joaquina Motas, Manuel Rodrigues

Selecionar Funcionário
João Miranda

Configurar plano de pagamento

FREE

Organize across all apps by hand

0 \$ Per Month

- Unlimited product updates
- Unlimited product updates
- Unlimited product updates
- 1GB Cloud storage
- Email and community support

Selecionada

Standard

Organize across all apps by hand

30 \$ Per Month

Slate helps you see how many more days you need to work to reach your financial goal for the month and year.

Selecionar

Retroceder

Gravar

Configurar informações

Configure a Loja

Ficheiro CAE
Drag & drop files or [Browse](#)
Formatos suportados: PDF
[document-name.PDF](#)

Nome da Empresa
Type something here!

Telefone
Type something here!

Email
Type something here!

Cidade
Type something here!

Nova polavra-passe
Type something here!

Morada
Type something here!

NIF
Type something here!

Código de Postal
Type something here!

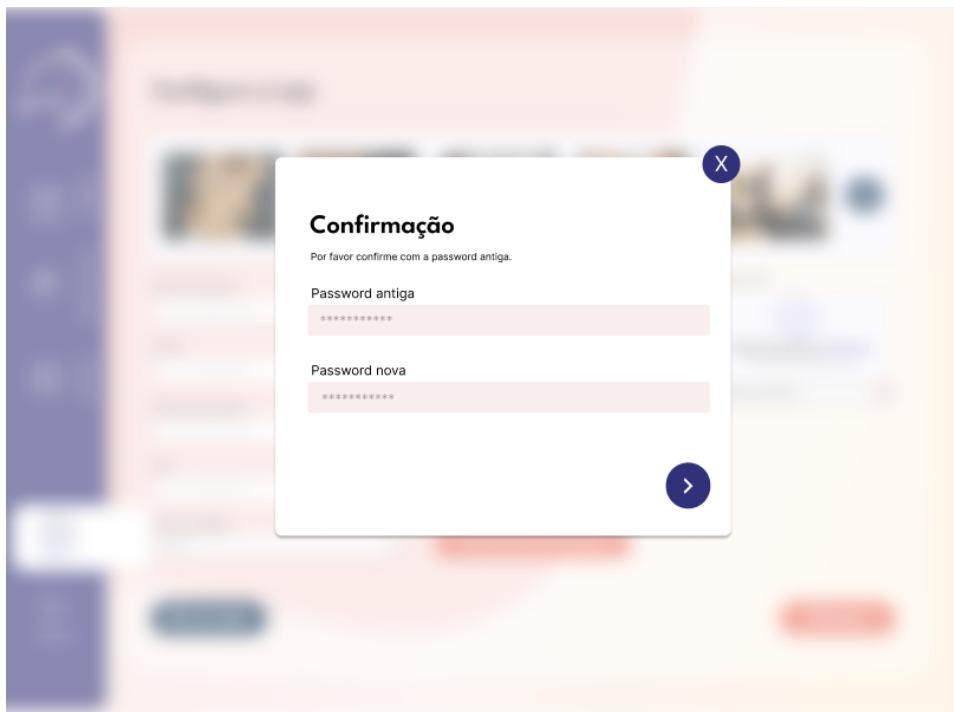
Setor de negócio
Setor

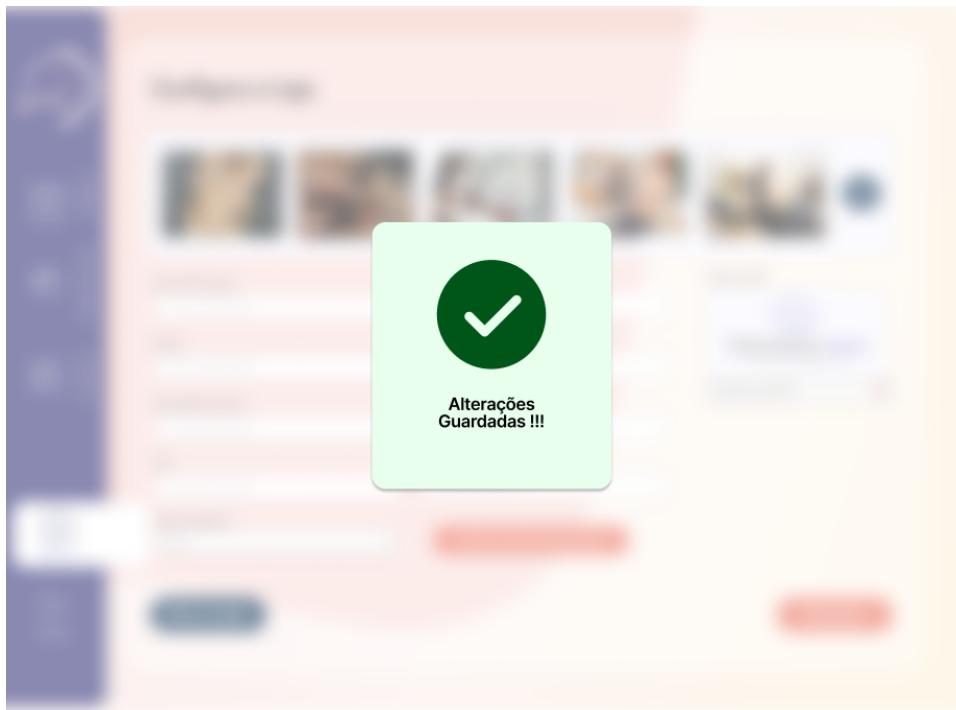
Horário de funcionamento

Alterar Método de Pagamento

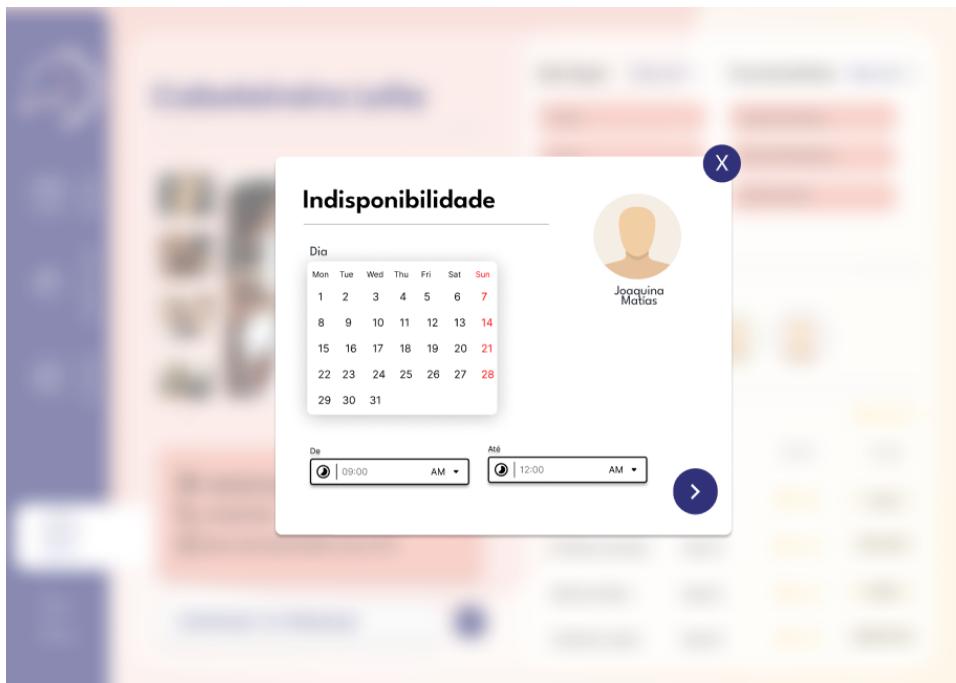
Retroceder

Gravar





Configurar indisponibilidade funcionário



6.1.2 Mobile

Configurar perfil



Configurar notificações

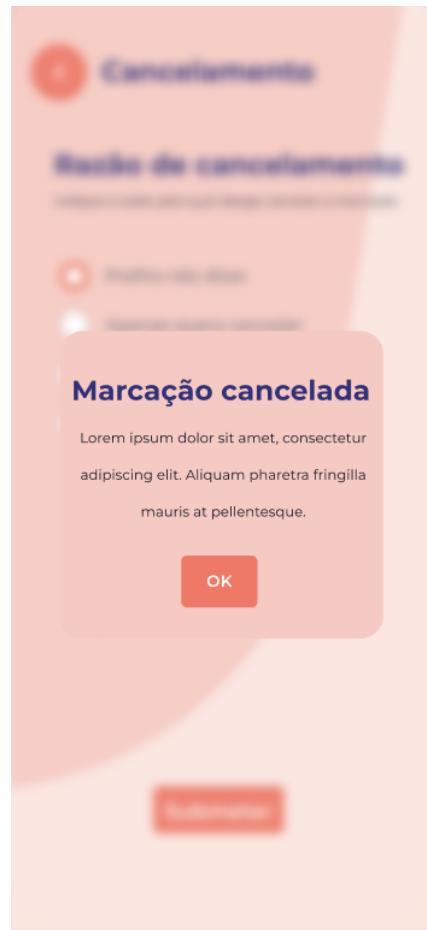


Filtrar página inicial



Cancelar marcação





Reagendar marcação







Avaliar marcação





