****

**INSTITUTO POLITÉCNICO DE LEIRIA**

SnakRestaurant

RELATÓRIO DE projeto em sistemas de informação

Relatório de Projeto em Sistemas de Informação para cumprimento dos requisitos necessários à realização da prova de apresentação de projeto do Curso Técnico Superior Profissional (TeSP) de **Programação de Sistemas de Informação** realizado sob a orientação de (**nome do orientador**)

Claúdio Martins Nº2190760

Tiago Santos Nº2180699

Programação em Sistemas de informação

18 de fevereiro de 2022

**DECLARAÇÃO**

Declaro que este Relatório se encontra em condições de ser apreciada (o) pelo júri a designar.

O estudante 1,

\_Cláudio Martins\_\_\_\_\_\_\_\_\_\_

Leiria, 18 de fevereiro de 2022

Relatório de Projeto em Sistemas de Informação para cumprimento dos requisitos necessários à realização da prova de apresentação de projeto do Curso Técnico Superior Profissional (TeSP) de **Programação de Sistemas de Informação** realizado sob a orientação de (**nome do orientador**)

**DECLARAÇÃO**

Declaro que este Relatório se encontra em condições de ser apreciada (o) pelo júri a designar.

O estudante 2,

\_Tiago Santos\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Leiria, 18 de fevereiro de 2022

# Resumo

**[SnakRestaurant]**

**[Claúdio Martins e Tiago Santos]**

Este relatório descreve o projeto e as suas respetivas componentes tecnológicas, foi desenvolvido no âmbito 1º semestre do TESP de Programação de Sistemas de Informação, no desenvolvimento deste projeto foram utilizadas as seguintes linguagens *Sql, Php e Java*.

SnakRestaurant consiste num projeto desenvolvido especificamente para um restaurante em que o cliente tem acesso a uma aplicação *Web* e *Android* para poder interagir com todas as outras funcionalidades disponíveis.

Começando por uma base de dados usada para armazenar todos os dados do restaurante, esta base de dados ira ser usada pelo website e pela *Api* por sua vez a aplicação Android ira efetuar pedidos para consultar esses dados.

Por sua vez o website possui um sistema de permissões, que dependendo dessas permissões, permite ao utilizador aceder a um *BackOffice* e controlar os dados. Por sua vez o *Website* que o cliente terá acesso será diferente, apenas poderá ter acesso ao *FrontOffice*, pois o cliente não poderá ter acesso aos dados privados do restaurante.

PALAVRAS-CHAVE: TESP, *Sql, Php, Java*, Web, *Android, Api, BackOffice, FrontOffice*.

# Índices

Índice

[Agradecimentos iv](#_Toc96036144)

[Resumo v](#_Toc96036145)

[Índices vi](#_Toc96036146)

[Índice de Figuras vii](#_Toc96036147)

[1. Introdução 2](#_Toc96036148)

[2. Metodologia 3](#_Toc96036149)

[3. Arquitetura do Sistema 4](#_Toc96036150)

[4. Gestão do Projeto 5](#_Toc96036151)

[5. Análise 6](#_Toc96036152)

[6. Desenho 7](#_Toc96036153)

[7. Implementação 13](#_Toc96036154)

[8. Testes 14](#_Toc96036155)

[9. Conclusão e trabalho futuro 15](#_Toc96036156)

[10. Bibliografia 16](#_Toc96036157)

[11. Anexos 17](#_Toc96036158)

# Índice de Figuras

[Figura 1-Modelo de Dados 6](file:///C:\xampp\htdocs\projeto21-22\documentos\relatorio%20final\Modelo_Relatório_Final%20_ProjetoSistemasInformação.docx#_Toc96114281)

[Figura 2-Mokups do login e registo. 7](file:///C:\xampp\htdocs\projeto21-22\documentos\relatorio%20final\Modelo_Relatório_Final%20_ProjetoSistemasInformação.docx#_Toc96114282)

[Figura 3-Mokups da página inicial e página do about. 8](file:///C:\xampp\htdocs\projeto21-22\documentos\relatorio%20final\Modelo_Relatório_Final%20_ProjetoSistemasInformação.docx#_Toc96114283)

[Figura 4- Exemplo de um index do BackOffice 9](file:///C:\xampp\htdocs\projeto21-22\documentos\relatorio%20final\Modelo_Relatório_Final%20_ProjetoSistemasInformação.docx#_Toc96114284)

[Figura 5-Mokup's do login, registo e página principal. 10](#_Toc96114285)

[Figura 6-Mokups do cardápio, página do perfil e lista de compras. 10](#_Toc96114286)

[Figura 7-Mokup carrinho de compras 11](#_Toc96114287)

[Figura 8-Página principal. 16](file:///C:\xampp\htdocs\projeto21-22\documentos\relatorio%20final\Modelo_Relatório_Final%20_ProjetoSistemasInformação.docx#_Toc96114288)

[Figura 9-Página do about com localização. 16](#_Toc96114289)

[Figura 10-Login 17](#_Toc96114290)

[Figura 11-Registo 17](#_Toc96114291)

[Figura 12-Mostrar os dados do utilizador. 18](file:///C:\xampp\htdocs\projeto21-22\documentos\relatorio%20final\Modelo_Relatório_Final%20_ProjetoSistemasInformação.docx#_Toc96114292)

[Figura 13-Perfil. 18](#_Toc96114293)

[Figura 14-Alterar dados do perfil. 19](#_Toc96114294)

[Figura 15-Lista de pedidos feitos. 19](#_Toc96114295)

[Figura 16-Mostra os dados de um pedido específico. 20](#_Toc96114296)

[Figura 17-Página principal do BackOffice. 21](#_Toc96114297)

[Figura 18-Exemplo de uma das páginas de gestão de dados do BackOffice. 21](#_Toc96114298)

[Figura 19-Criar um produto. 22](#_Toc96114299)

[Figura 20-Ver produto. 22](#_Toc96114300)

[Figura 21-Editar um produto. 23](#_Toc96114301)

[Figura 22-Apagar um produto. 23](#_Toc96114302)

[Figura 23-Exemplo dos testes unitários. 24](#_Toc96114303)

[Figura 24-Exemplo dos testes funcionais. 24](#_Toc96114304)

[Figura 25-Pedido feito a Api. 25](#_Toc96114305)

[Figura 26-Aplicação desenvolvida no netbeans. 25](#_Toc96114306)

[Figura 27-Linhas de comando para aceder a aplicação do netbeans. 26](#_Toc96114307)

[Figura 28-Login em Android. 27](file:///C:\xampp\htdocs\projeto21-22\documentos\relatorio%20final\Modelo_Relatório_Final%20_ProjetoSistemasInformação.docx#_Toc96114308)

[Figura 29-Registo em Android. 28](file:///C:\xampp\htdocs\projeto21-22\documentos\relatorio%20final\Modelo_Relatório_Final%20_ProjetoSistemasInformação.docx#_Toc96114309)

[Figura 30-Página principal da app Android. 29](file:///C:\xampp\htdocs\projeto21-22\documentos\relatorio%20final\Modelo_Relatório_Final%20_ProjetoSistemasInformação.docx#_Toc96114310)

[Figura 31-Lista de produtos em Android. 30](file:///C:\xampp\htdocs\projeto21-22\documentos\relatorio%20final\Modelo_Relatório_Final%20_ProjetoSistemasInformação.docx#_Toc96114311)

[Figura 32-Página de perfil. 31](file:///C:\xampp\htdocs\projeto21-22\documentos\relatorio%20final\Modelo_Relatório_Final%20_ProjetoSistemasInformação.docx#_Toc96114312)

[Figura 33-Informação do pedido. 32](file:///C:\xampp\htdocs\projeto21-22\documentos\relatorio%20final\Modelo_Relatório_Final%20_ProjetoSistemasInformação.docx#_Toc96114313)

[Figura 34-Lista de compras efetuadas pelo utilizador. 32](file:///C:\xampp\htdocs\projeto21-22\documentos\relatorio%20final\Modelo_Relatório_Final%20_ProjetoSistemasInformação.docx#_Toc96114314)

[Figura 35-Carrinho de compras. 33](file:///C:\xampp\htdocs\projeto21-22\documentos\relatorio%20final\Modelo_Relatório_Final%20_ProjetoSistemasInformação.docx#_Toc96114315)

[Figura 36-Pedido final para inserir a mesa. 33](file:///C:\xampp\htdocs\projeto21-22\documentos\relatorio%20final\Modelo_Relatório_Final%20_ProjetoSistemasInformação.docx#_Toc96114316)

[Figura 37-Gantt 34](#_Toc96114317)

[Figura 38-Divisão de tarefas. 34](#_Toc96114318)

[Figura 40-Tarefas completas 35](#_Toc96114319)

[Figura 39-Percentagem completo do projeto(os 3% que da em falta são este relatório e o documento de power point) 35](file:///C:\xampp\htdocs\projeto21-22\documentos\relatorio%20final\Modelo_Relatório_Final%20_ProjetoSistemasInformação.docx#_Toc96114320)

# Introdução

Este relatório de projeto surge como trabalho final do curso TeSP de Programação de Sistemas de Informação do Politécnico de Leiria, apresentando assim uma síntese do todo o trabalho realizado ao longo do 1º semestre do 2ºano do ano 2021/22.

O seguinte relatório encontra-se dividido em cinco capítulos, o primeiro capítulo faz referência a introdução. Os capítulos 3, 6, 7 e 8 dedicam-se a explicar como o projeto funciona. Enquanto os capítulos 4 e 5 se dedicam a explicar um pouco a forma como o projeto se desenvolveu.

SnakRestaurant consiste num sistema desenhado para a gestão de um restaurante, este sistema possui duas vertentes um website e android para o utilizador.

O *Website* permite aos funcionários ter acesso a um *BackOffice* (figura 17 a 22) no qual podem controlar os dados inerentes ao bom funcionamento do restaurante e um *FrontOffice* (figura 8 a 16) no qual o cliente pode efetuar diversas operações, mas no qual tem uma vertente mais virada para dar a conhecer e transmitir informação sobre o espaço.

A aplicação android será para uso dos clientes, em que o cliente poderá, entre outras funcionalidades, adicionar produtos ao carrinho de compras e a pedi-los (figura 35). Poderá ainda alterar as suas informações (figura 32) e consultar compras anteriores (figura 34).

Estas funcionalidades, tanto *Website* como o *Android*, só estão disponíveis após o login (figura 10 e 28) caso não tenha conta pode-se registar (figura 11 e 29).

SnakRestaurant ainda possui mais uma vertente a *Api* (figura 25 a 17), é graças a ela que a aplicação *Android* consegue fazer pedidos à base de dados, o que garante uma sincronização de dados entre toda o sistema.

# Metodologia

Deve ser especificada a metodologia adotada no desenvolvimento do Projeto em Sistemas de Informação

Neste projeto adotamos uma metodologia ágil com foco no próprio projeto e no resultado final. Através de reuniões pontuais entre a nossa equipa e os docentes das diferentes unidades curriculares em que o nosso projeto se integra.

Foi através destas reuniões que nos permitiram fazer melhorias e alterações constantes ao projeto e resolver eventuais problemas que nos foram aparecendo.

Através da entrega das primeiras etapas de cada uma das unidades curriculares obtivemos feedback, por parte dos docentes encarregues dessas unidades, o que nos foi muito útil para o futuro da aplicação.

# Arquitetura do Sistema

O projeto SnakRestaurant apresenta 3 componentes tecnológicas, um *Website* (figura 8 a 24), uma aplicação em *Android* (figura 28 a 36) e uma *Api* (figura 25 a 27).

Foram também desenvolvidas uma base de dados em que as componentes tecnológicas se apoiaram para fazer operações *CRUD* (figura 19 a 22) e uma aplicação desenvolvida em java para a mesma unidade curricular da Api (figura 26 e 27).

O *Website* e a *Api* foram ambos desenvolvidos em *Php*, utilizando a *YII framework* como *framework* definida pela unidade curricular Plataformas de Sistemas de Informação e pela Serviços e Interoperabilidade de Sistemas. A aplicação android foi desenvolvida em java na unidade curricular de Acesso Móvel a Sistemas de Informação.

No website o funcionário tem a permissão de aceder ao *BackOffice* (figura 17 a 22), mas apenas lhe é permitido aceder a certas funcionalidades inerentes ao funcionamento diário do restaurante. Por fim o administrador tem as permissões que lhe permitem aceder a tudo no *BackOffice* esta permissão é permite-lhe poder aceder a todas às funcionalidades, por exemplo, atribuir a outro utilizador a permissão de funcionário ou administrador.

O *FrontOffice* (figura 8 a 16), possui um *login* e um registo, apos o *login* o cliente pode alterar as suas informações, no entanto o nosso *FrontOffice* possui um caracter mais informativo onde o cliente pode ficar a conhecer a história do restaurante bem como a sua localização.

A aplicação *Android* (figura 28 a 36), será usada pelos clientes do restaurante, no qual os mesmos podem efetuar *login* ou efetuar registo, ver produtos e efetuar pedidos, alterar os dados do perfil e ver as compras efetuadas.

É através de uma *Api* que a aplicação *Android* consegue obter informação da base de dados, a aplicação *Android* faz pedidos a *Api*, que lhe envia uma resposta em *Json* (figura 25).

# Gestão do Projeto

A distribuição de tarefas e tudo o que diz respeito a gestão do projeto foi feito no trello incluindo o *Gantt Chart* que foi desenvolvido no *trello*.

O diagrama de Gantt (Figura 37) é um gráfico para ilustrar o avanço das diferentes etapas do nosso projeto.

Os dois membros do projeto (Figura 39) foram divididos para várias funcionalidades do projeto, tendo a participação nas 3 componentes tecnológicas que integram este projeto (Figura 38). Ambos os desenvolvedores do projeto possuem cargos de membros da equipa bem como gestor de projeto.

Pode-se aceder ao nosso trello através deste link (<https://trello.com/invite/b/tbpgzO8P/3332d83303186356e827d9eb0293fbee/projeto>), desde que possua conta.

# Análise

A temática da SnakRestaurant consiste no desenvolvimento de um sistema de gestão para um restaurante, tal sistema possui as seguintes componentes um *Website* e uma aplicação *Android*.

Os objetivos principais são cumprir o que o propósito deste sistema, este sendo permitir que o cliente possa usufruir da aplicação e poder fazer os pedidos e receber o que pediu de forma simples e rápida.

A análise de requisitos do projeto em *Web* são *login*, registo, alterar dados pessoais, e gestão de todos os dados dependendo das permissões do utilizador em questão. Em android são o *login*, registo, alterar dados pessoais, ver compras anteriores, ver produtos e efetuar pedidos. Na *Api* são fornecer dados a aplicação *Android* sempre que esta lhe pedir.

As funcionalidades que ficaram por desenvolver são a recuperação da *password* e o alterar a *password*, bem como o acesso a um sistema de pagamentos apos os pedidos na aplicação de *Android*;

# Desenho

Uma imagem com texto

Descrição gerada automaticamente

Figura -Modelo de Dados

*Mokups* do *Website*.

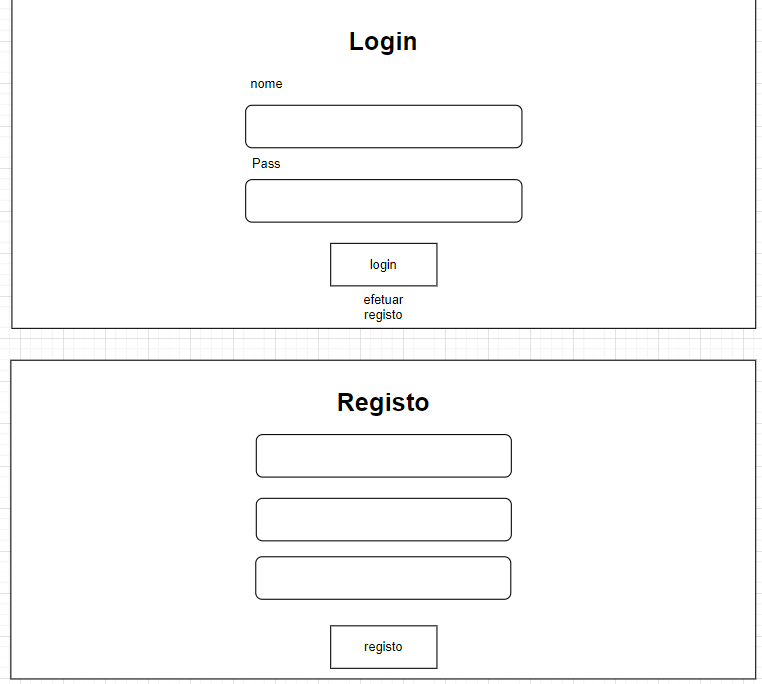


Figura -Mokups do login e registo.

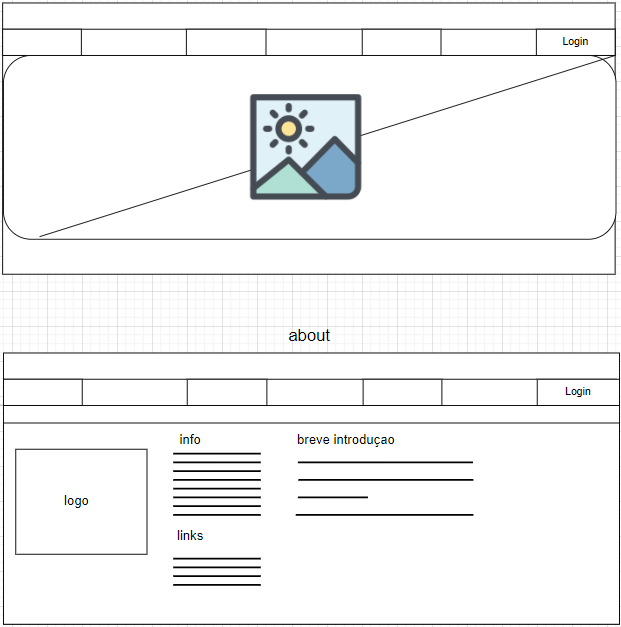


Figura -Mokups da página inicial e página do about.

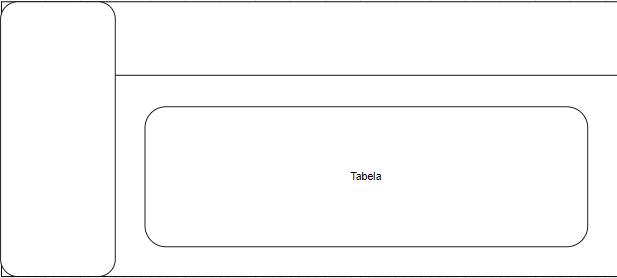


Figura - Exemplo de um index do BackOffice

*Mokups* da aplicação de *Android*:

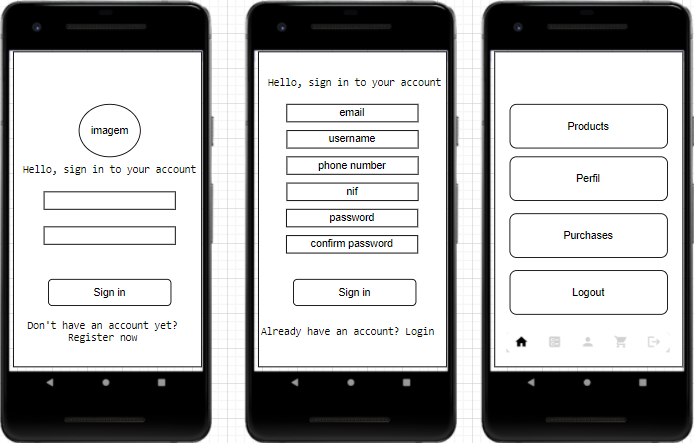


Figura -Mokup's do login, registo e página principal.

Uma imagem com texto, interior, diferente

Descrição gerada automaticamente

Figura -Mokups do cardápio, página do perfil e lista de compras.

Uma imagem com texto, telefone, iPod, telemóvel

Descrição gerada automaticamente

Figura -Mokup carrinho de compras

# Implementação

No início do projeto começamos por desenvolver uma ideia, de seguida elaboramos um documento de proposta.

O próximo passo foi criação da base de dados que iria interligar os diversos componentes tecnológicos, ao longo do desenvolvimento do projeto esta base de dados foi alterada várias vezes, também começamos a discutir como seria o design geral da aplicação, ou seja, os layouts.

No desenvolvimento do *Website* começamos por desenvolver o *login* tanto no *FrontOffice* bem como no *BackOffice*, de seguida acabamos o registo. Após isso, começámos aos poucos a desenvolver as páginas do *FrontOffice*, ao mesmo tempo começamos a implementar os *CRUDS* no *backend*, de modo a puder gerir as informações da base de dados.

A seguir a implementar os *CRUDS* básicos incluídos na ferramenta *GII* da *YII* *framework*, começámos a alterá-los de modo a melhorá-los. O próximo passo foi criar um sistema de perfil para o *FrontOffice* e *BackOffice*.

No desenvolvimento da *RESTful Api* começamos por criar os métodos *GET, POST, PUT* e *DELETE* para todas as tabelas na base de dados, de seguida estabelecemos autenticação para cada um dos métodos referidos acima, e criamos ainda uma aplicação de *messaging* para quando é efetuado um pedido aparecer a informação do pedido numa aplicação desenvolvida à parte em *netbeans*.

No desenvolvimento de *Android*, apos o desenvolvimento dos *layouts* começamos por desenvolver um sistema de *login*, ao mesmo tempo começávamos a fazer a ligação a *Api* para obter os produtos. Apos concluímos, começamos a desenvolver o registo e o perfil. Para concluir desenvolvemos um carrinho de compras e uma lista com todas as compras efetuadas pelo utilizador.

# Testes

Durante o desenvolvimento do projeto realizamos testes, nomeadamente no nosso website, para assegurar um bom funcionamento.

No desenvolvimento do website em *YII framework* utilizámos a ferramenta de testes *Codeception*, que permite fazer 3 tipos de testes, testes funcionais, unitários, e de aceitação. Destes 3 tipos realizamos testes unitários e funcionais presentes no *backend*.

Os testes funcionais são responsáveis por testar funções ou ações do website, tal como o teste humano faria. Os testes unitários testão os models do *Website*, e da maneira como eles se conectam à base de dados, através de inserções ou apagar registos ou até mesmo editar.

Para executar os vários testes têm que ser introduzidos os códigos seguintes num terminal do Windows.

Para executar os testes unitários (Figura 23) pode-se usar a linha de comando mais em baixo:

vendor/bin/codecept run unit -c backend

Os testes unitários disponíveis são ConsumoTest e ProductsTest. Para correr apenas um teste basta colocar o nome após a palavra unit como no exemplo.

Exemplo: vendor/bin/codecept run unit -c backend

Para executar os testes funcionais (Figura 24) pode-se usar a linha de comando mais em baixo:

vendor/bin/codecept run unit -c backend

Os testes funcionais disponíveis são LoginCest e testeCest. Para correr apenas um teste basta colocar o nome após a palavra unit como no exemplo.

Exemplo: vendor/bin/codecept functional -c backend

# Conclusão e trabalho futuro

Neste projeto conciliamos conhecimentos aprendidos ao longo do semestre, tais conhecimentos permitiram-nos criar o sistema descrito ao longo deste relatório.

Apos termos a nossa ideia e elaborarmos a proposta de ideia para o projeto, criamos uma base de dados e planeamos todos os passos até chegarmos a elaboração deste relatório e a entrega final, também ganhamos e renovamos conhecimentos em *Php* e *Java*. Tais experiências permitiram-nos ganhar conhecimentos e experiências para trabalhos futuros.

# Bibliografia

* <https://codeception.com/> - utilizado para correr os testes da aplicação da YII *Framework*
* <https://www.yiiframework.com/doc/guide/2.0/en> - é o guia oficial da *Framework* de YII
* <https://trello.com> – site para gestão de projeto
* <https://developer.android.com/docs> - guia oficial do *android*

# Anexos

Anexo A:

O anexo A diz respeito ao desenvolvimento do website.

Uma imagem com texto, interior, captura de ecrã

Descrição gerada automaticamente

Figura -Página principal.

Uma imagem com texto

Descrição gerada automaticamente

Figura -Página do about com localização.

Este anexo diz respeito a parte da autenticação no FrontOffice:

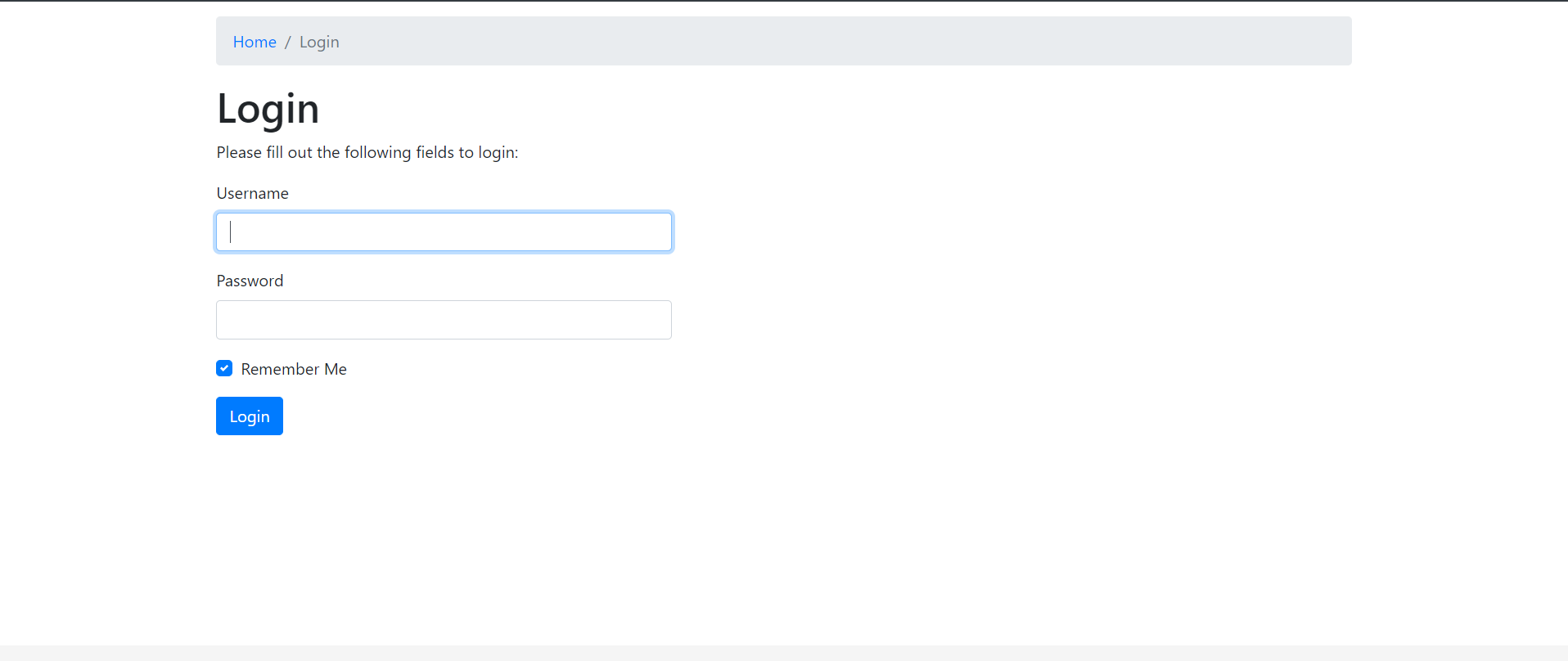


Figura -Login

Esta figura diz respeito a forma de como o cliente efetua o registo:

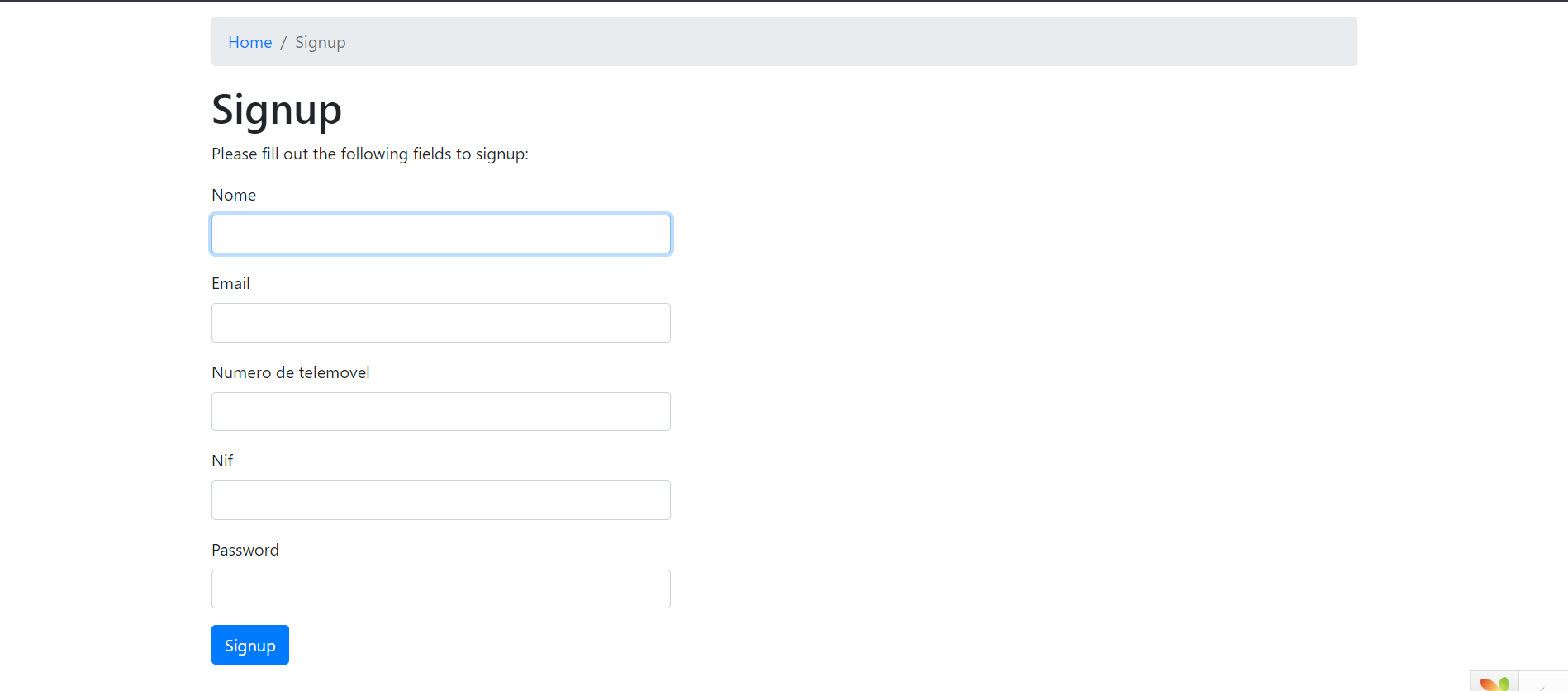
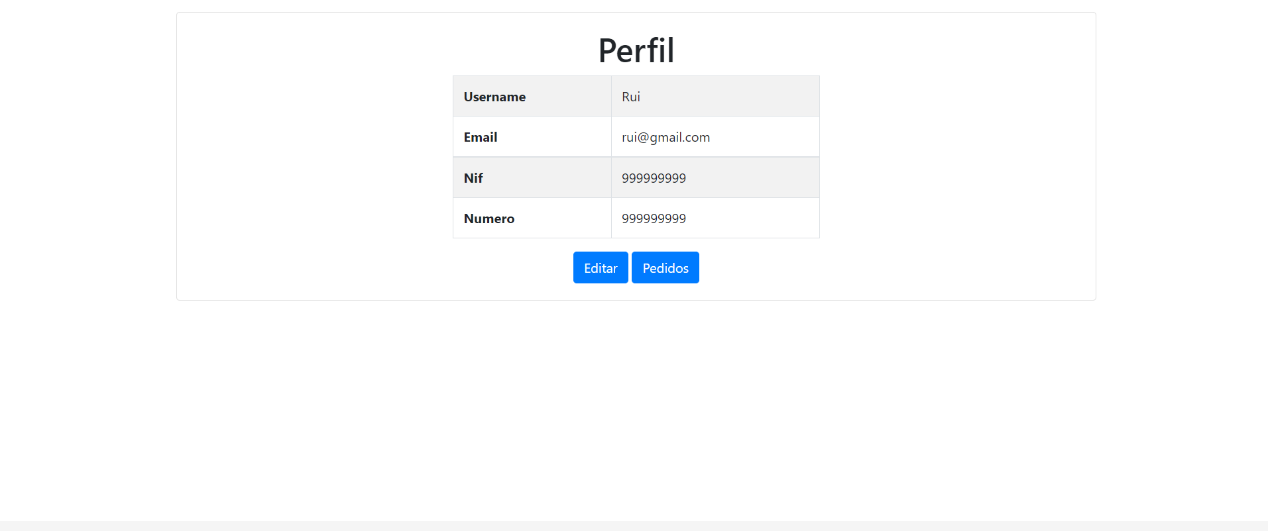


Figura -Registo



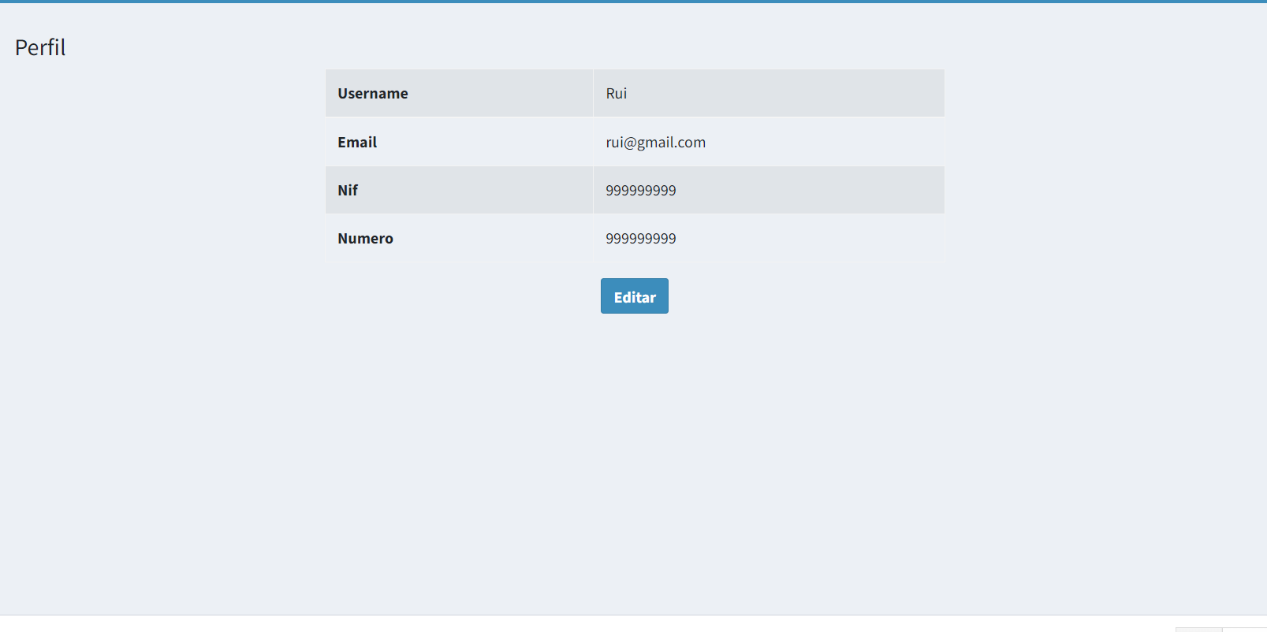
Figura -Perfil.

Figura -Mostrar os dados do utilizador.

Uma imagem com texto

Descrição gerada automaticamente

Figura -Alterar dados do perfil.

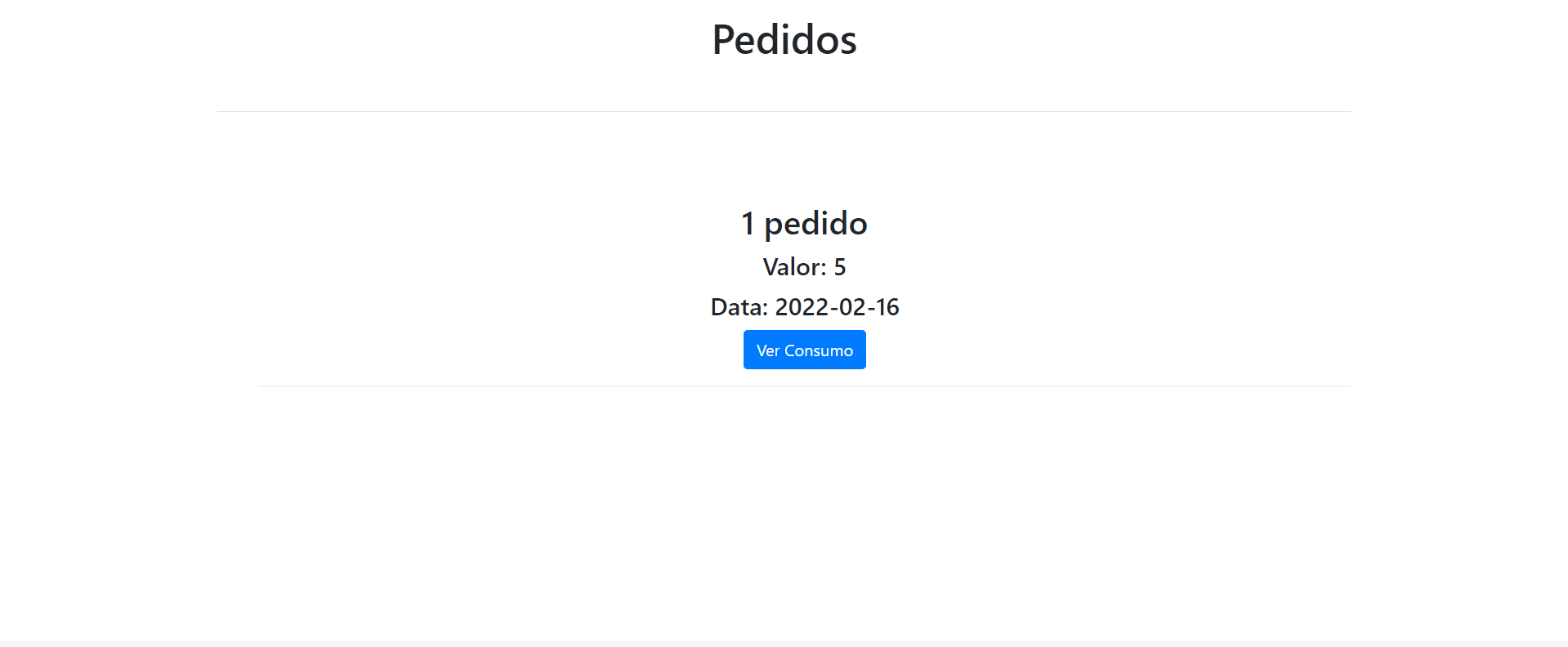


Figura -Lista de pedidos feitos.

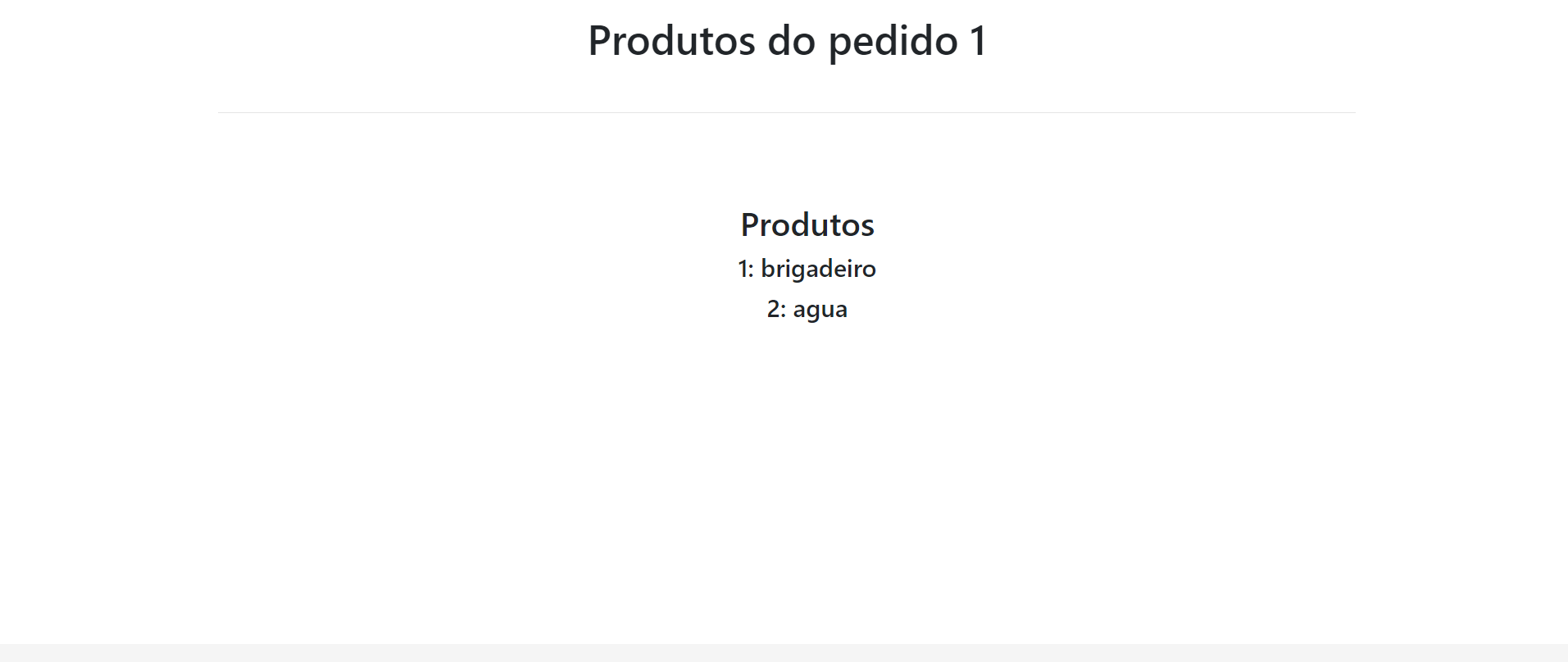


Figura -Mostra os dados de um pedido específico.

As seguintes figuras dizem respeito ao desenvolvimento do BackOffice do website.

Uma imagem com texto, chávena, café

Descrição gerada automaticamente

Figura -Página principal do BackOffice.

Uma imagem com texto, captura de ecrã, interior

Descrição gerada automaticamente

Figura -Exemplo de uma das páginas de gestão de dados do BackOffice.

Estas figuras falam sobre as diversas operações CRUD:

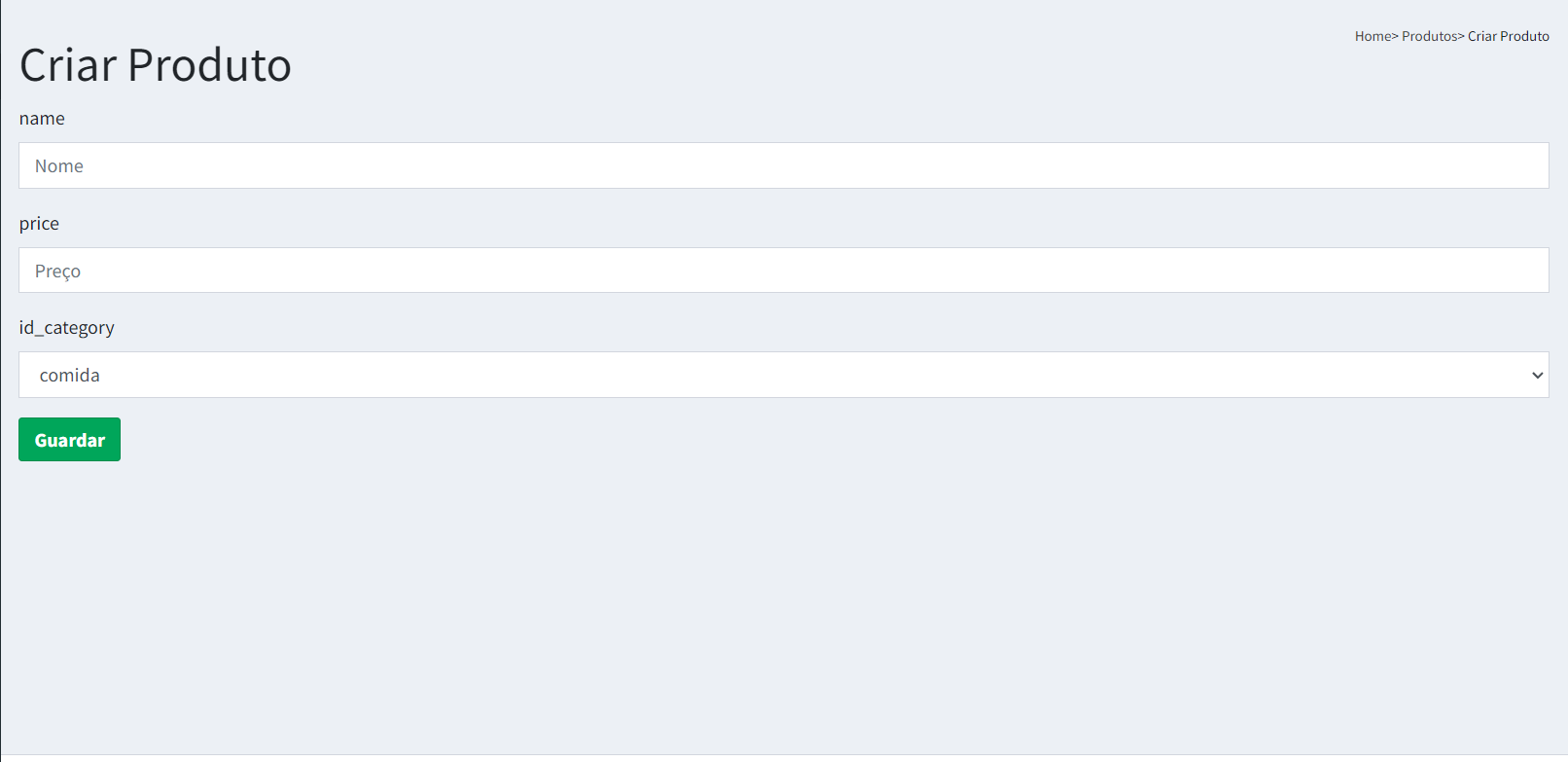


Figura -Criar um produto.



Figura -Ver produto.

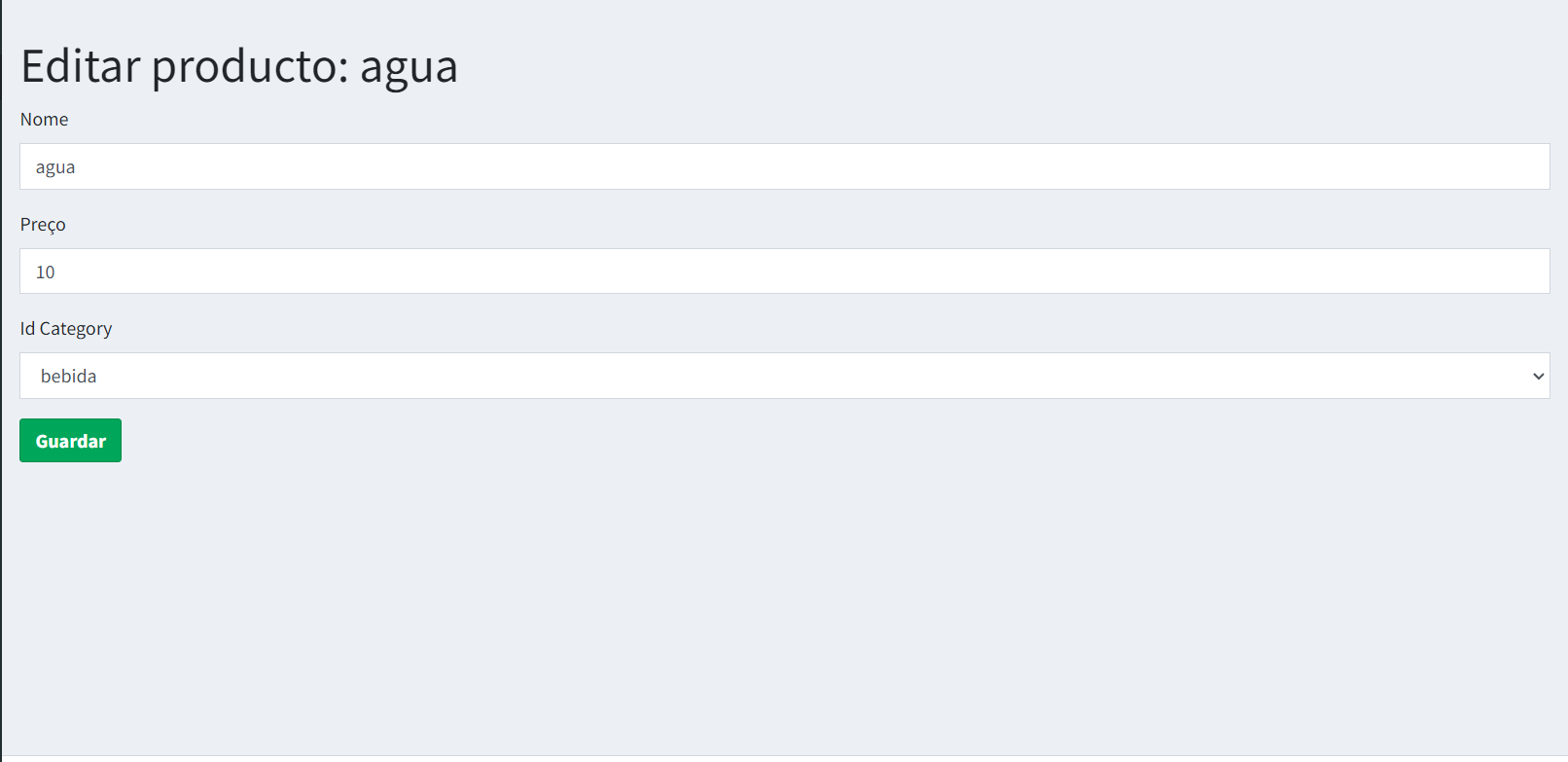


Figura -Editar um produto.

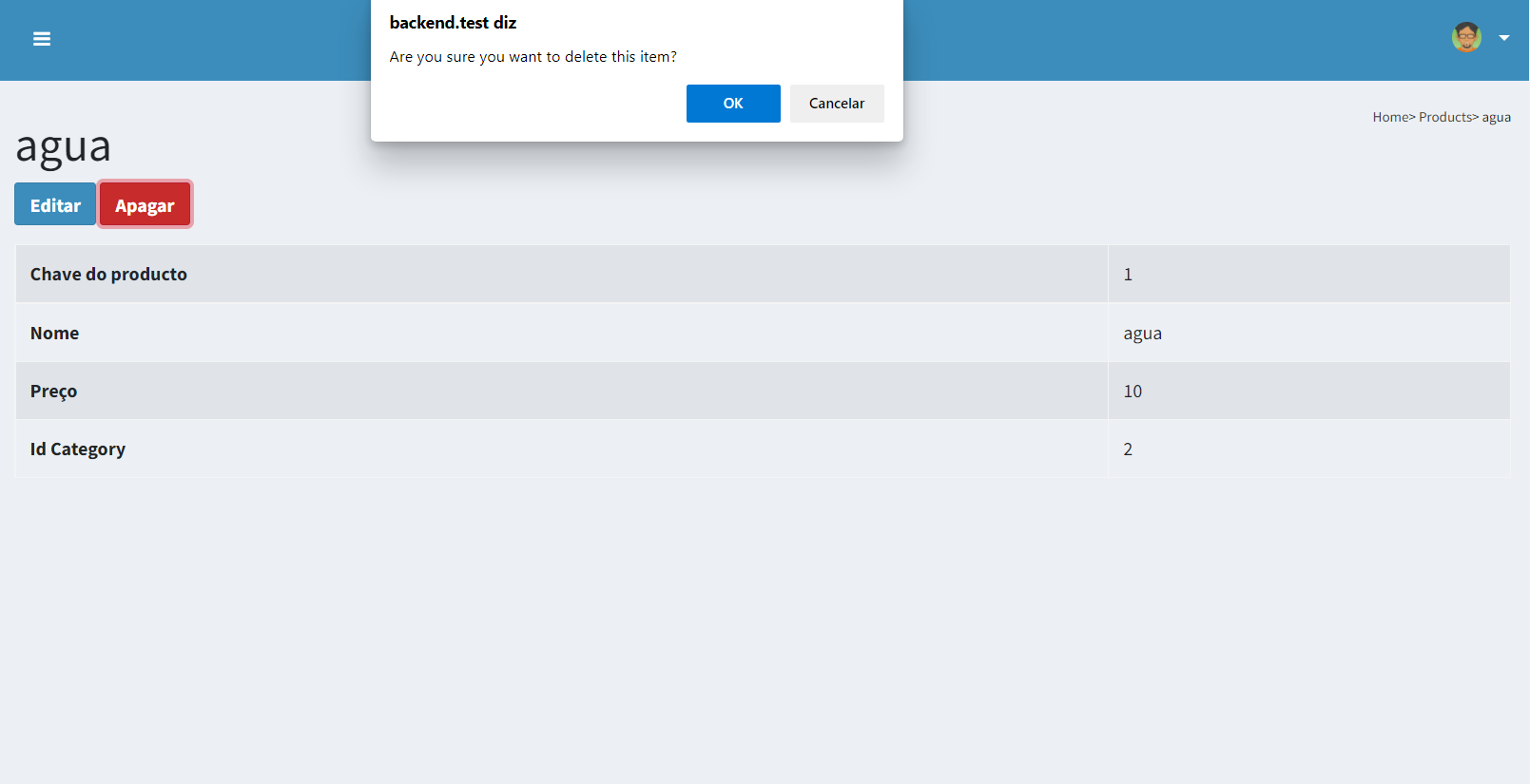


Figura -Apagar um produto.

Ilustração dos testes realizados no BackOffice.

Uma imagem com texto

Descrição gerada automaticamente

Figura -Exemplo dos testes unitários.

Uma imagem com texto, mesa

Descrição gerada automaticamente

Figura -Exemplo dos testes funcionais.

Anexo B:

Este anexo mostra os pedidos efetuados a Api e a respetiva resposta:

Uma imagem com texto

Descrição gerada automaticamente

Figura -Pedido feito a Api.

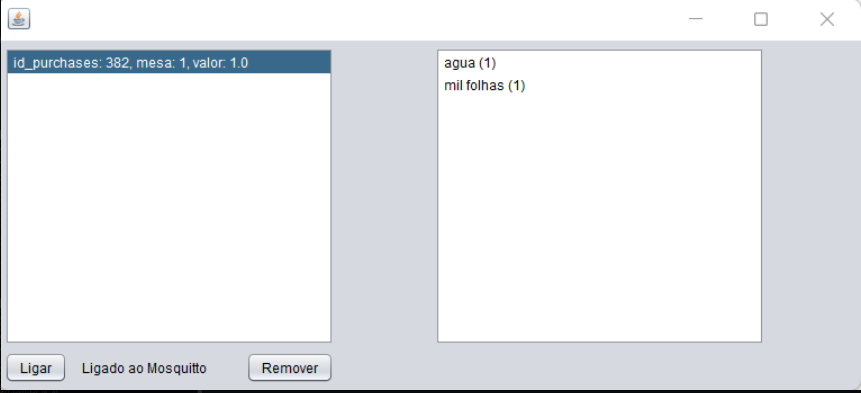


Figura -Aplicação desenvolvida no netbeans.

Uma imagem com texto

Descrição gerada automaticamente

Figura -Linhas de comando para aceder a aplicação do netbeans.

Anexo C:

Este anexo demostra a aplicação android:

Esta parte diz respeito a parte da autenticação:

Uma imagem com texto

Descrição gerada automaticamente

Figura -Login em Android.

Registo:

Figura -Registo em Android.

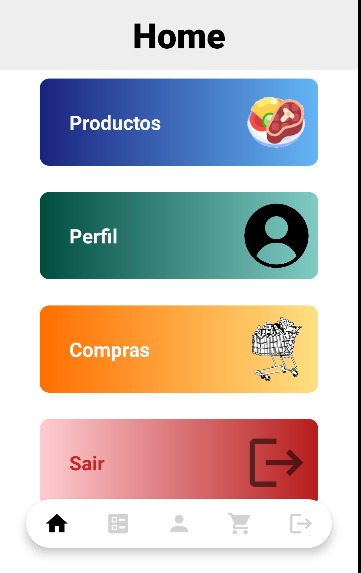


Figura -Página principal da app Android.



Figura -Lista de produtos em Android.

Uma imagem com texto

Descrição gerada automaticamente

Figura -Página de perfil.

:

Uma imagem com texto

Descrição gerada automaticamente

Figura -Informação do pedido.

Figura -Lista de compras efetuadas pelo utilizador.



Figura -Carrinho de compras.

Uma imagem com texto

Descrição gerada automaticamente

Figura -Pedido final para inserir a mesa.

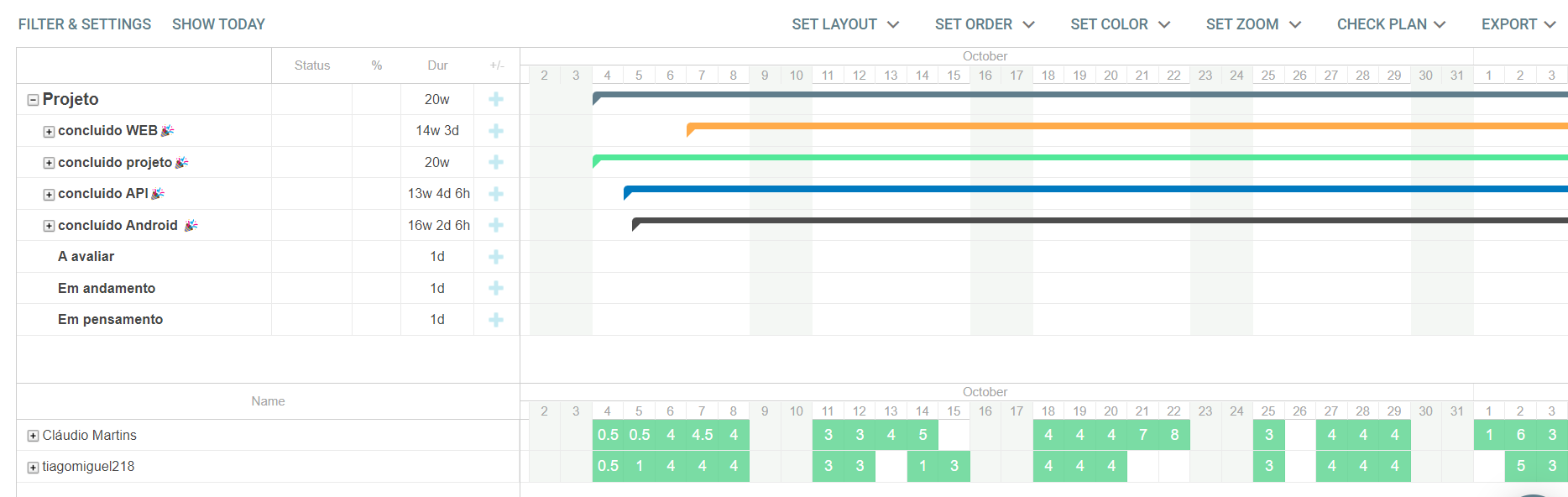


Figura -Gantt (imagem demasiado grande para o relatório, consulta no trello)

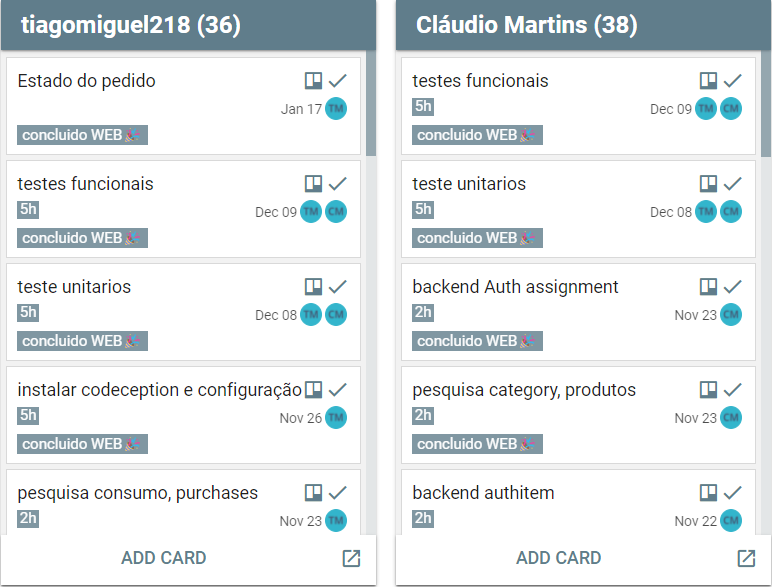


Figura -Divisão de tarefas.

Uma imagem com texto

Descrição gerada automaticamente

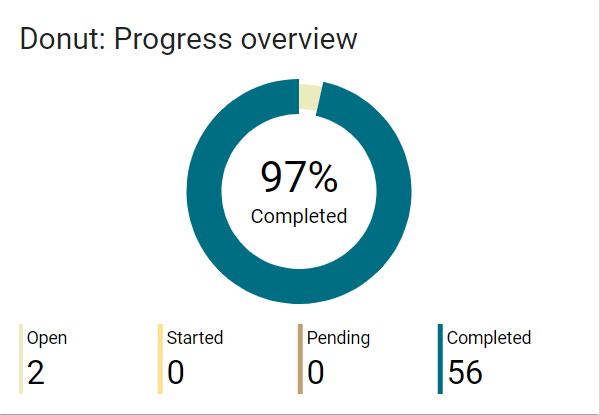
Figura -Tarefas completas

Figura -Percentagem completo do projeto(os 3% que da em falta são este relatório e o documento de power point)