

**Mestrado Integrado em Engenharia Informática e Computação**

**5º Ano**

**1º Semestre**

**Shop and Pay App**

**Computação Móvel**

**2017/2018**

**Paulo Jorge Silva Ferreira,** [**up201305617@fe.up.pt**](mailto:up201305617@fe.up.pt)

**7 de novembro de 2017**

Índice

[Introdução 3](#_Toc497825396)

[Principais Funcionalidades 4](#_Toc497825397)

[Arquitetura do Sistema 5](#_Toc497825398)

[Esquema de Dados 6](#_Toc497825399)

[Servidor 7](#_Toc497825400)

[User Routes 7](#_Toc497825401)

[Product Routes 8](#_Toc497825402)

[Shop List Routes 8](#_Toc497825403)

[Aplicação do Cliente 9](#_Toc497825404)

[Funcionalidades 9](#_Toc497825405)

[Testes Realizados 9](#_Toc497825406)

[Aplicação da Impressora 10](#_Toc497825407)

[Funcionalidades 10](#_Toc497825408)

[Testes Realizados 10](#_Toc497825409)

[Atividades 10](#_Toc497825410)

[MainActivity 10](#_Toc497825411)

[ShopListActivity 10](#_Toc497825412)

[Interações Entre Atividades 10](#_Toc497825413)

[Conclusão 11](#_Toc497825414)

# Introdução

Este projeto foi desenvolvido no âmbito da Unidade Curricular de Computação Móvel, do 5º Ano do curso de Mestrado Integrado em Engenharia Informática e Computação da Faculdade de Engenharia da Universidade do Porto.

O principal objetivo era simular o funcionamento de uma loja, onde um cliente, através de uma aplicação Android, escolhia os produtos que pretendia adquirir através da leitura de código de barras e depois de ajustar a sua lista efetuava a compra. Uma vez confirmada a compra o servidor gerava um identificador único que era transmitido à aplicação do cliente, que convertia esse identificador num QRCode. Depois o cliente deslocava-se a uma impressora (aplicação Android), apresentava o QRCode e era gerado um recibo com os dados da compra.

Estas duas aplicações comunicavam com um serviço REST, localizado no servidor da loja, que posteriormente trocava informação com uma base de dados.

# Principais Funcionalidades

As principais funcionalidades do sistema são:

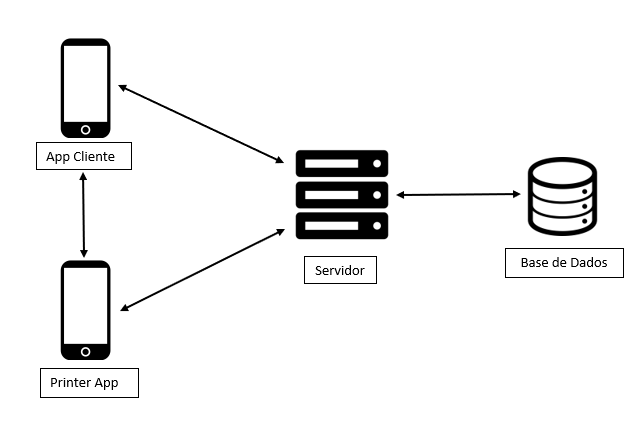
1. O *Log In* de um utilizador registado através do seu e-mail e password;
2. O registo de um novo cliente que terá que fornecer os seguintes dados:
   1. Nome;
   2. Morada;
   3. NIF;
   4. E-mail;
   5. Password;
   6. Tipo Cartão de Crédito;
   7. Número do Cartão;
   8. Validade do Cartão.
3. A leitura de código de barras dos produtos;
4. Consultar as características dos produtos;
5. A adição do produto a uma lista de compras;
6. Efetuar a compra dos produtos que fazem parte da lista de compras;
7. A leitura do *QR Code*, gerado pela aplicação do cliente, por parte da aplicação da impressora;
8. A impressão de um recibo com as principais informações da compra efetuada.
9. A consulta das transações já efetuadas.

# Arquitetura do Sistema

Ambas as aplicações Androidcomunicam com os serviços REST do servidor. Este servidor, por sua vez, comunica também com uma base dados.

As aplicações fazem pedidos ao servidor para registar ou pesquisar dados. O servidor encaminha os pedidos para a base de dados que guarda os novos dados ou retorna os que a aplicação pretende. Com base no resultado da pesquisa à base de dados, o servidor gera uma resposta e envia para a aplicação.

A aplicação da impressora lê o *QR Code* que é apresentado no ecrã da aplicação do cliente, estabelecendo assim um canal de comunicação entre as duas aplicações.



# Esquema de Dados

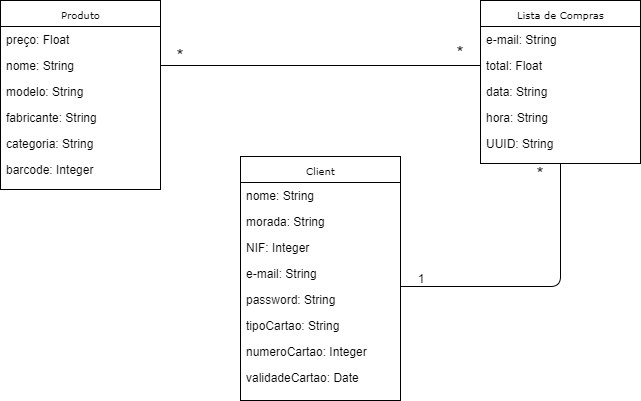
Como já foi referido um cliente é identificado por um email, morada, NIF, nome, password e os dados do cartão de crédito. Possui também uma chave pública que é usada para confirmar uma compra por parte do cliente.

A lista de compras tem o email do cliente, os produtos comprados, a data e a hora da compra e um identificador único (UUID) gerado aquando a confirmação da compra.

Um produto possui um preço, um nome, um modelo, um fabricante, uma categoria e um código de barras.

Só pode existir um produto com um código de barras, uma lista de compras com um UUID e um cliente com um email e NIF.

A base de dados foi implementada em *NoSQL*, mas precisamente em *mongoDB*.



# Servidor

O servidor consiste num conjunto de serviços REST.

Tanto os pedidos feitos ao servidor como as respostas estão formatados em JSON.

O serviço de compra de um conjunto de produtos requer que o pedido seja assinado através de uma chave privada, que posteriormente é verificada pelo servidor que tem guardada a chave pública correspondente, uma vez confirmada a compra o servidor gera um identificador único que envia como resposta para o dispositivo do cliente.

Nas próximas tabelas estão definidos os *endpoints*, bem como os parâmetros e uma breve descrição dos mesmos.

## User Routes

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo** | **Rota** | **Parâmetros** | **Descrição** |
| POST | /api/user | Name  Address  NIF  E-mail  Password  CardType  CardNumber  CardValidty | Regista um novo cliente na base de dados. |
| POST | /api/login | E-mail  Password | Valida login de um cliente já registado. |

## Product Routes

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo** | **Rota** | **Parâmetros** | **Descrição** |
| GET | /api/product/:barcode | barcode | Retorna o produto com o código de barras identificado nos parâmetros. |

## Shop List Routes

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo** | **Rota** | **Parâmetros** | **Descrição** |
| POST | /api/shoplit | E-mail  TotalPrice  Products | Regista a lista de compras na base de dados e efetua a compra. Retorna o UUID gerado pelo servidor. |
| GET | /api/shoplists/:email | E-mail | Retorna todas as listas de compras do utilizador cujo e-mail está identificado nos parâmetros. |
| GET | /api/shoplist/:UUID | UUID | Retorna a lista com o UUID identificado nos parâmetros. |

# Aplicação do Cliente

## Funcionalidades

A aplicação do cliente permite fazer o Log In de um cliente já registado na base de dados; registar um novo cliente; ler códigos de barras; mostrar os detalhes do produto pertencente ao código de barras lido; adicionar um produto à lista de compras; efetuar a compra dos produtos presentes na lista; remover um produto da lista que o cliente já não pretenda adquirir; mostrar informações de transações passadas, tais como a data e hora da aquisição, bem como os detalhes dos produtos comprados.

## Testes Realizados

|  |  |  |
| --- | --- | --- |
| **Atividade** | **Teste** | **Resultado** |
| LogIn | Não preencher os campos pedidos. | Exibição de um Toast a pedir ao cliente que preencha todos os campos. |
| LogIn | Introdução de endereço de  e-mail não válido. | É mostrado um Toast a pedir a correção do endereço de  e-mail. |
| LogIn | Introdução de informação errada. | Exibição de um Toast a dizer que o cliente não existe ou que a password está incorreta. |
| LogIn | Tentar autenticação sem ligação à internet. | É mostrado um Toast a dizer que a aplicação não está ligada ao servidor. |
| LogIn | Introdução dos elementos válidos para a autenticação. | O cliente é direcionado para a MainActivity. |
| SignUp | Não preencher os campos pedidos. | Exibição de um Toast a pedir ao cliente que preencha todos os campos. |
| SignUp |  |  |

# Aplicação da Impressora

## Funcionalidades

A aplicação da impressora permite ler um QRCode gerado pela aplicação do cliente e numa outra atividade mostrar o fabricante, o modelo e o preço de cada produto presente na lista de produtos que o cliente comprou.

Todas as funcionalidades requeridas no enunciado para a aplicação da impressora foram concluídas com sucesso.

## Testes Realizados

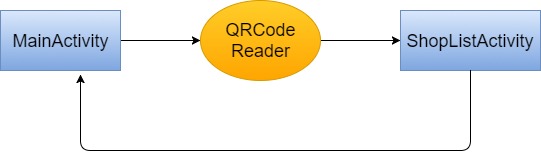
|  |  |  |
| --- | --- | --- |
| **Atividade** | **Teste** | **Resultado** |
| MainActivity |  |  |

## Atividades

### MainActivity

### ShopListActivity

## Interações Entre Atividades



# Conclusão