

Computação Móvel

Mestrado Integrado de Engenharia Informática e Computação



Professor:

António Miguel Pontes Pimenta Monteiro

Autor:

José Miguel Botelho Mendes - up201304828@fe.up.pt

Índice

1 Arquitetura	2
1.1 Base de Dados	2
1.2 Servidor	2
1.3 Rest API (Application Program Interface)	2
1.4 Aplicação Acme-Electronics-Shop	3
1.4.1 Activities	3
1.4.2 Structures	4
1.4.3 API	4
1.5 Aplicação Acme-Electronics-Shop-Printer	4
2 Representação e Armazenamento de Dados	5
2.1 Cliente	5
2.2 Servidor	5
3 Funcionalidades	6
3.1 Registo	6
3.2 Visualização de Encomendas Feitas	6
3.3 Visualização Detalhada de cada Encomenda	7
3.4 Gestão de Carrinho de Compras	7
3.6 Visualização de Lista Completa de Produtos	8
3.7 Visualização Detalhada de cada Produto	8
3.8 Scan de Código de Barras	9
3.9 Scan de QR Codes	10
3.10 Simulação de Impressora	10
3.11 Persistência de Toda a Informação	11
4 Testes	12

1 Arquitetura

A arquitetura da Loja Online ACME assenta sobretudo num servidor, externo, que corre uma Rest API que faz a ponte entre os dados a que o cliente necessita de aceder e o próprio cliente. O cliente acede à aplicação *mobile* desenvolvida em Android nativo, comunica com este servidor e com a aplicação da impressora (também desenvolvida em Android nativo).

1.1 Base de Dados

A base de dados foi desenvolvida em SQLite devido à sua fácil implementação e mobilidade. O seu aspeto mais negativo é que não admite concorrência ao nível da escrita, o que pode ser problemático caso haja mais do que um cliente a tentar escrever na mesma ao mesmo tempo. No entanto, a migração desta base de dados para uma desenvolvida em, por exemplo, MySQL, é de extrema facilidade, pois os scripts de criação e população da base de dados foram guardados. No próximo capítulo são abordados aspetos mais específicos, como que tipo de dados são guardados, de que forma e a segurança dos mesmos.

1.2 Servidor

O servidor em que corre uma boa parte do tratamento e armazenamento de dados foi desenvolvido em NodeJS. Para efeitos de teste, foi instalado o *ngrok* na máquina onde o servidor estava a ser corrido, para permitir o acesso ao mesmo em qualquer lugar. O programa *ngrok* tem como função expôr uma porta do *localhost* da máquina, através da criação de um link específico do *ngrok* e serviços de *tunneling*.

1.3 Rest API (Application Program Interface)

A API que está aberta ao público trata de criar uma ligação entre o servidor e as aplicações *mobile*. De seguida são analisadas todas as rotas disponíveis:

- **/register** - recebe um objeto em *JSON (JavaScript Object Notation)* que contém todas as informações relevantes do utilizador e cria o mesmo na base de dados.
- **/cart/:idUser** - recebe como parâmetro o id do utilizador de forma a enviar de volta os produtos existentes no carrinho do mesmo.
- **/cart/add/:idUser/:idProduct** - recebe como parâmetros o id do utilizador e o id do produto de forma a adicionar o mesmo ao carrinho do utilizador.
- **/cart/remove/:idUser/:idProduct** - recebe como parâmetros o id do utilizador e o id do produto de forma a remover o mesmo do carrinho do utilizador.
- **/cart/add/:idUser/:idProduct/:quantity** - recebe como parâmetros o id do utilizador, o id do produto e a quantidade de forma a adicionar mais produtos do mesmo tipo ao carrinho do utilizador.

- **/cart/remove/:idUser/:idProduct/:quantity** - recebe como parâmetros o id do utilizador, o id do produto e a quantidade de forma a remover uma determinada quantidade do produto em causa do carrinho do utilizador.
- **/order** - recebe um objeto em JSON que contém o id do utilizador em questão e uma string hashed com recurso à chave privada do utilizador (através do algoritmo de assinatura SHA1withRSA). Com a chave pública do utilizador em questão que foi guardada anteriormente no servidor, a string é validada e, de seguida, a encomenda é criada e um *UUID (Universal Unique Identifier)* é enviado para o utilizador.
- **/order/:uuid** - recebe como parâmetro o id de uma encomenda e retorna todos os produtos relativos a essa encomenda, assim como a data em que foi realizada.
- **/order/printer/:uuid** - recebe como parâmetro o id de uma encomenda e retorna literalmente todos os dados relativos à mesma (os dados do utilizador que podem ser mostrados publicamente, os dados da encomenda e dos produtos associados à mesma).
- **/order/previous/:idUser** - recebe como parâmetro o id do utilizador de forma a enviar de volta todas as encomendas que o utilizador já realizou.
- **/product/:idProduct** - recebe como parâmetro o id do produto e retorna toda a informação deste mesmo produto.
- **/product** - retorna todos os produtos existentes na loja.

1.4 Aplicação Acme-Electronics-Shop

A aplicação Acme-Electronics-Shop está dividida, estruturalmente, em três *packages*: Activities, Structures e API. A navegação entre as *activities* é feita através de uma *tab* que está sempre presente em qualquer *activity*. É nesta *tab* também que existe uma opção de *Scan* de código de barras, que lê qualquer código de barras e, caso o código coincida com um id de um produto em questão, o utilizador é levado para os detalhes do mesmo. Este scan é feito recorrendo a uma biblioteca *open-source* externa de nome XZing (<https://github.com/zxing/zxing>).

1.4.1 Activities

Este *package* contém todas as *activities* que a aplicação em questão possui:

- **RegisterActivity**: esta é a *activity* em que a aplicação inicia. Se o utilizador já tiver utilizado a aplicação no mesmo telemóvel e um registo criado, o utilizador é levado para a *activity* FrontPage. Caso contrário, terá apenas acesso a um formulário de registo, com todos os campos necessários e relevantes para a criação de um utilizador. Nenhum campo poderá ficar por preencher e a data de validade do cartão de crédito terá de ter o formato indicado (mm/yy). Caso o registo tenha tido sucesso, será levado também para a *activity* FrontPage. É de notar também que é nesta *activity* que são geradas as chaves privada e pública do utilizador e um *Serializable* com a toda a sua informação (é através deste *Serializable* que a aplicação consegue determinar se o telemóvel em questão já tem um registo e um

utilizador criado. Ou seja, caso a aplicação seja desinstalada e instalada novamente, será necessário criar um novo utilizador, devido à falta de uma *activity* de autenticação).

- **Cart:** nesta *activity* o utilizador gere o seu carrinho quase na íntegra, ou seja, consegue retirar produtos e adicionar e remover quantidades de produtos já existentes no carrinho. É também aqui que é possível criar a encomenda com o carrinho atual.
- **FrontPage:** esta *activity* funciona como uma *landing page* para o utilizador, após o seu registo com sucesso.
- **OrderDetails:** aqui é possível visualizar os detalhes de uma encomenda específica do utilizador, assim como um *QR Code* que é possível de ser lido pela impressora.
- **PastOrders:** nesta *activity* é possível visualizar todas as encomendas do utilizador e escolher qualquer uma de forma a saber mais detalhes sobre a mesma.
- **ProductDetails:** aqui é possível visualizar os detalhes de um produto em específico da loja e adicioná-lo ao carrinho.
- **ProductList:** aqui é possível visualizar todos os produtos existentes em loja.

1.4.2 Structures

Este *package* contém três estruturas que representam uma Encomenda, um Produto e um Utilizador.

1.4.3 API

Este *package* trata de conectar a aplicação ao servidor através da utilização de uma conexão de HTTP e uma *async task*.

1.5 Aplicação Acme-Electronics-Shop-Printer

Esta aplicação permite a “impressão virtual” de uma encomenda, como se de um recibo se tratasse. Esta aplicação inicia com um *scan* aberto e, caso leia um *QR Code* relativo à encomenda através do telemóvel do cliente, irá recuperar toda a informação relativo à mesma, mostrando no seu ecrã este “recibo”. Tem também um botão que permite um novo scan, de forma a ser completamente autónoma.

2 Representação e Armazenamento de Dados

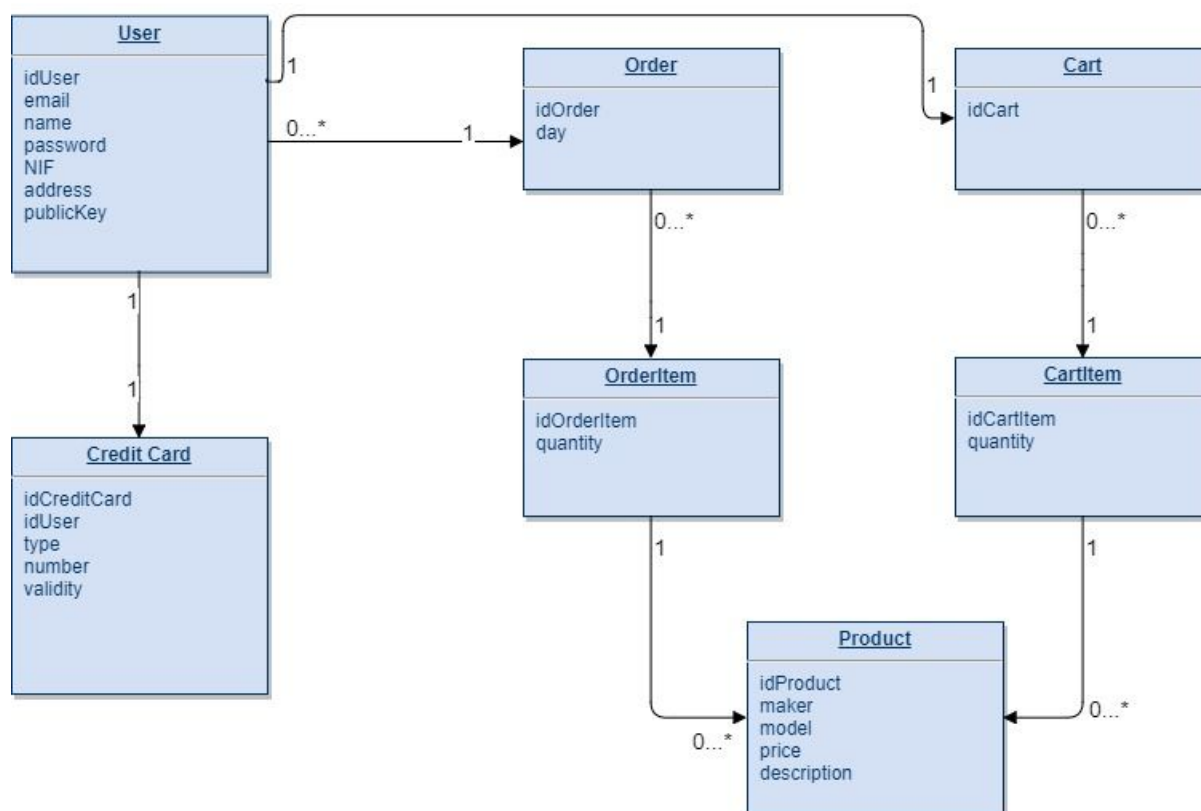
Neste projeto, os dados são representados e armazenados de diferentes formas, tendo em conta se são guardados do lado do cliente ou do lado do servidor.

2.1 Cliente

No que toca ao cliente, apenas os dados relativos ao utilizador são guardados. Este *storage* é feito tomando como recurso um ficheiro *Serializable*, que é guardado na memória interna do dispositivo. É apenas possível de ser guardado uma vez, aquando da criação de um novo utilizador, o registo.

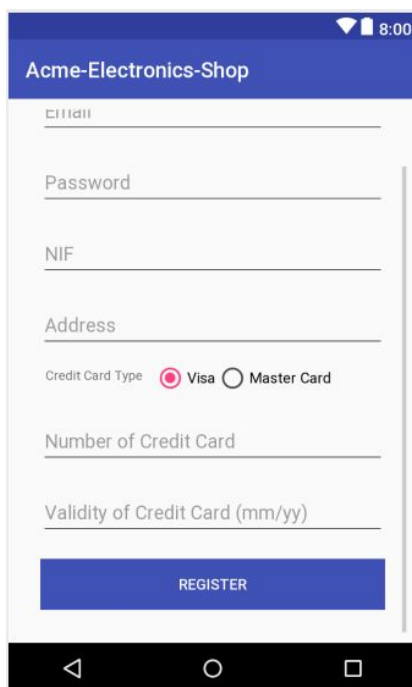
2.2 Servidor

Do lado do servidor, a estrutura é bem mais complexa. De seguida é mostrado um esboço da base de dados:



3 Funcionalidades

3.1 Registo



A screenshot of a mobile application interface for 'Acme-Electronics-Shop'. The screen displays a registration form with the following fields: 'Email', 'Password', 'NIF', 'Address', 'Credit Card Type' (with radio buttons for 'Visa' and 'Master Card'), 'Number of Credit Card', and 'Validity of Credit Card (mm/yy)'. A blue 'REGISTER' button is located at the bottom of the form. The status bar at the top shows the time as 8:00.

3.2 Visualização de Encomendas Feitas

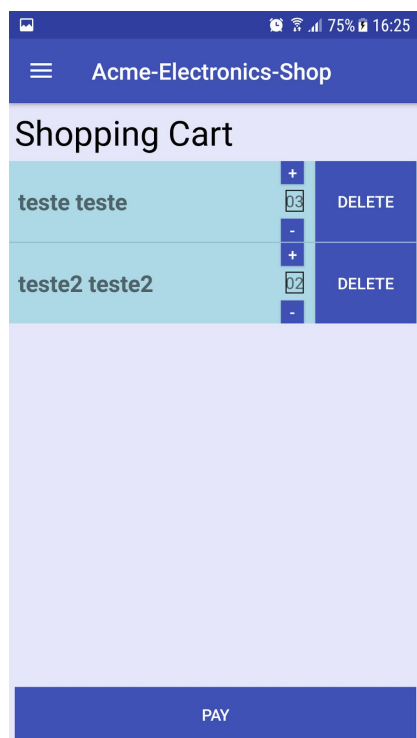


A screenshot of a mobile application interface for 'Acme-Electronics-Shop' showing the 'Past Orders' section. The screen has a blue header with a hamburger menu icon and the app name. Below the header, the title 'Past Orders' is displayed. The list of orders includes: 'Order nº: 3', 'Order nº: 4', and 'Order nº: c55c159c-1918-4e42-9d3c-f9ef05a5e87b'. The bottom of the screen is a solid light blue color. The status bar at the top shows the time as 16:25 and 75% battery.

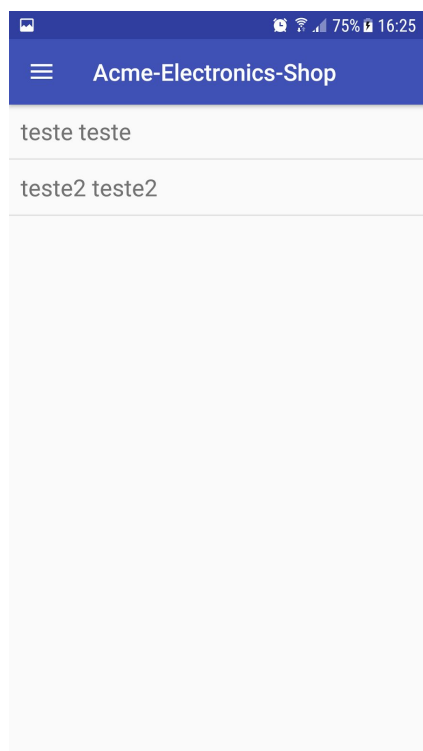
3.3 Visualização Detalhada de cada Encomenda



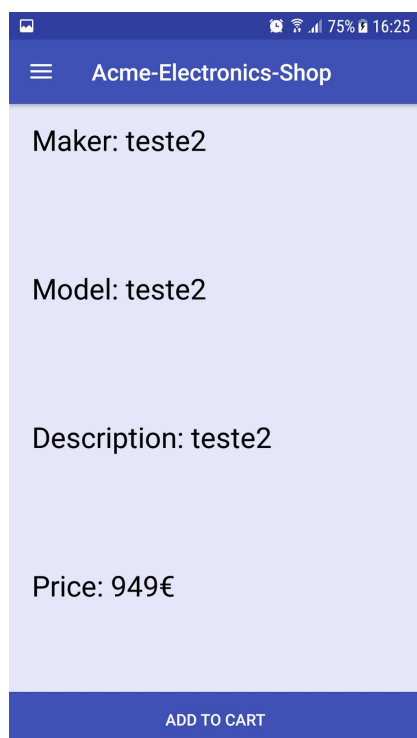
3.4 Gestão de Carrinho de Compras



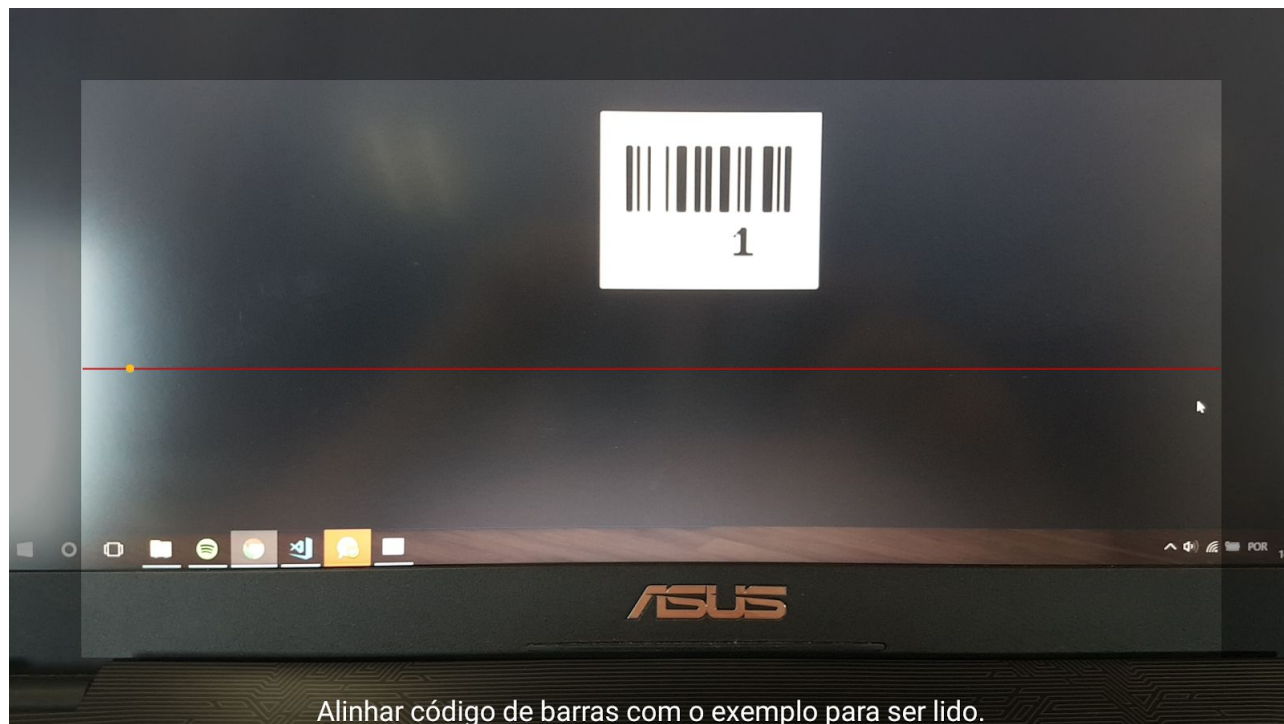
3.6 Visualização de Lista Completa de Produtos



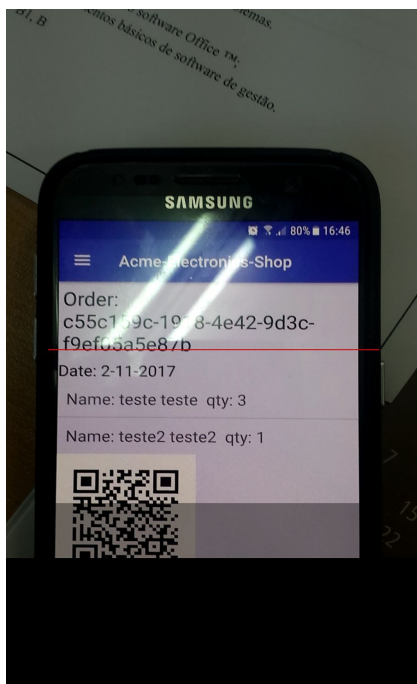
3.7 Visualização Detalhada de cada Produto



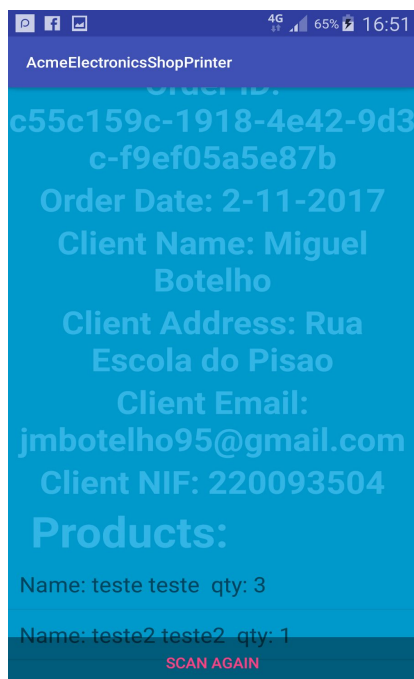
3.8 *Scan* de Código de Barras



3.9 Scan de QR Codes



3.10 Simulação de Impressora



3.11 Persistência de Toda a Informação

Uma das grandes vantagens, se não a maior, é o facto de toda a informação persistir caso o utilizador decida utilizar outro dispositivo para aceder à aplicação (o seu carrinho de compras, as encomendas e os seus dados pessoais).

4 Testes

Toda a aplicação foi testada com valores normais e com um comportamento o mais próximo da realidade possível. Foram testadas simulações de erro, como por exemplo tentar ler um código de barras errado, códigos cuja informação não residia na base de dados. Foi simulado também comportamento errático do utilizador (utilizar a aplicação bastante depressa), o não preenchimento de todos os campos. Todos estes testes resultaram no bom funcionamento da aplicação, fazendo com que a mesma não tivesse qualquer tipo de ocorrência negativa.