



FEUP **FACULDADE DE ENGENHARIA**
UNIVERSIDADE DO PORTO

5.º ano do Mestrado Integrado em Engenharia Informática e
Computação

Computação Móvel
2018/2019

Trabalho Prático 1
Design and Development
Android applications and services for a theater ticketing and payment

Lázaro Costa - up201405342@fe.up.pt
Luís Barbosa - up201405729@fe.up.pt

19 de Novembro de 2018

Índice:

Arquitetura da aplicação	3
Arquitetura geral	3
Arquitetura do servidor	3
Arquitetura da “Customer App”	4
Arquitetura da “Validation App”	4
Arquitetura da “Cafeteria App”	4
Arquitetura da biblioteca “Rest Services”	4
Arquitetura da biblioteca “Shared Library”	4
Estruturação dos dados	5
Modelo da base de dados	5
Modelo de dados do QR code	6
Funcionalidades implementadas	6
Customer App	6
Validation App	7
Cafeteria App	7
Funcionalidades reformuladas pela equipa	7
Testes de aceitação	7
Manual de utilização	8
Customer App	8
Validation App	13
Cafeteria App	14
Referências	15

Arquitetura da aplicação

Arquitetura geral

A aplicação para a aquisição e validação de vales de cafetaria é composta por 4 componentes principais:

- Um servidor remoto (serviço REST) capaz de armazenar toda a informação necessária para o funcionamento da aplicação. Sendo capaz de realizar as operações tais como: registo do cliente, compra de bilhetes, validação dos bilhetes, emissão e validação de vouchers e as transações do cliente
- Uma aplicação Android “Customer App”, destinada aos clientes, sendo responsável pelo registo do utilizador, obtenção da lista de eventos existentes, compra e validação dos bilhetes nas entradas nos eventos, realizar encomendas para a cafetaria podendo seleccionar os vouchers existentes para a obtenção de descontos e por último ver as transações realizadas.
- Uma aplicação Android “Validation App” que é capaz de receber os bilhetes da aplicação do cliente e realizar a sua validação com o servidor.
- Uma aplicação Android “Cafeteria App” que recebe da aplicação do cliente as encomendas dos produtos disponíveis juntamente com possíveis vouchers para a obtenção de descontos. Após a validação dos vouchers, realização do pagamento e a emissão da fatura, o cliente pode levantar os produtos comprados.

As comunicações entre Customer App - servidor, Cafeteria App - servidor e Validation App - servidor são do tipo REST em que, para todos os pedidos, uma resposta é recebida, enquanto que entre Customer App - Validation App e Customer App - Cafeteria App a comunicação é através da geração/leitura de um QR code, daí a comunicação ser unidirecional.

Arquitetura do servidor

O servidor Rest foi desenvolvido no Firebase tendo a seguinte API:

Verbo	Caminho	Descrição
PUT	/register	Registo do utilizador.
POST	/payOrder	Valida e regista a compra de produtos e uso de vouchers na “Cafeteria App”.
POST	/validTickets	Valida os bilhetes do utilizador na “Validation App”.

POST	/buyTickets	Compra os bilhetes para o utilizador.
POST	/listTransactionsUser	Lista todos os bilhetes e vouchers do utilizador não validados assim como os produtos comprados.
GET	/listProducts	Lista os produtos que se pode comprar.

Arquitetura da “Customer App”

A aplicação “Customer App” contém todas actividades necessárias à realização daquilo que foi pedido no enunciado para a aplicação do cliente, sendo que faz uso das duas bibliotecas enunciadas abaixo para a utilização de código partilhado pelas diferentes aplicações.

Arquitetura da “Validation App”

A aplicação “Validation App” contém duas actividades: a actividade que arranca automaticamente e que lê o QR code e a actividade que mostra se a leitura do QR code foi bem sucedida. Esta aplicação corresponde ao terminal de validação de bilhetes, sendo que faz uso das duas bibliotecas enunciadas abaixo para a utilização de código partilhado pelas diferentes aplicações.

Arquitetura da “Cafeteria App”

A aplicação “Cafeteria App” contém duas actividades: a actividade que arranca automaticamente e que lê o QR code e a actividade que mostra os dados do pedido de compra, sendo que faz uso das duas bibliotecas enunciadas abaixo para a utilização de código partilhado pelas diferentes aplicações.

Arquitetura da biblioteca “Rest Services”

A biblioteca “Rest Services” contém duas classes: a classe privada que realiza os pedidos REST e que implementa uma `HTTPAsyncTask` e a classe que expõe a API para fazer os pedidos REST às aplicações. Esta biblioteca pode receber objectos do tipo `JSONObject` e `byte[]` e retorna sempre um `JSONObject`.

Arquitetura da biblioteca “Shared Library”

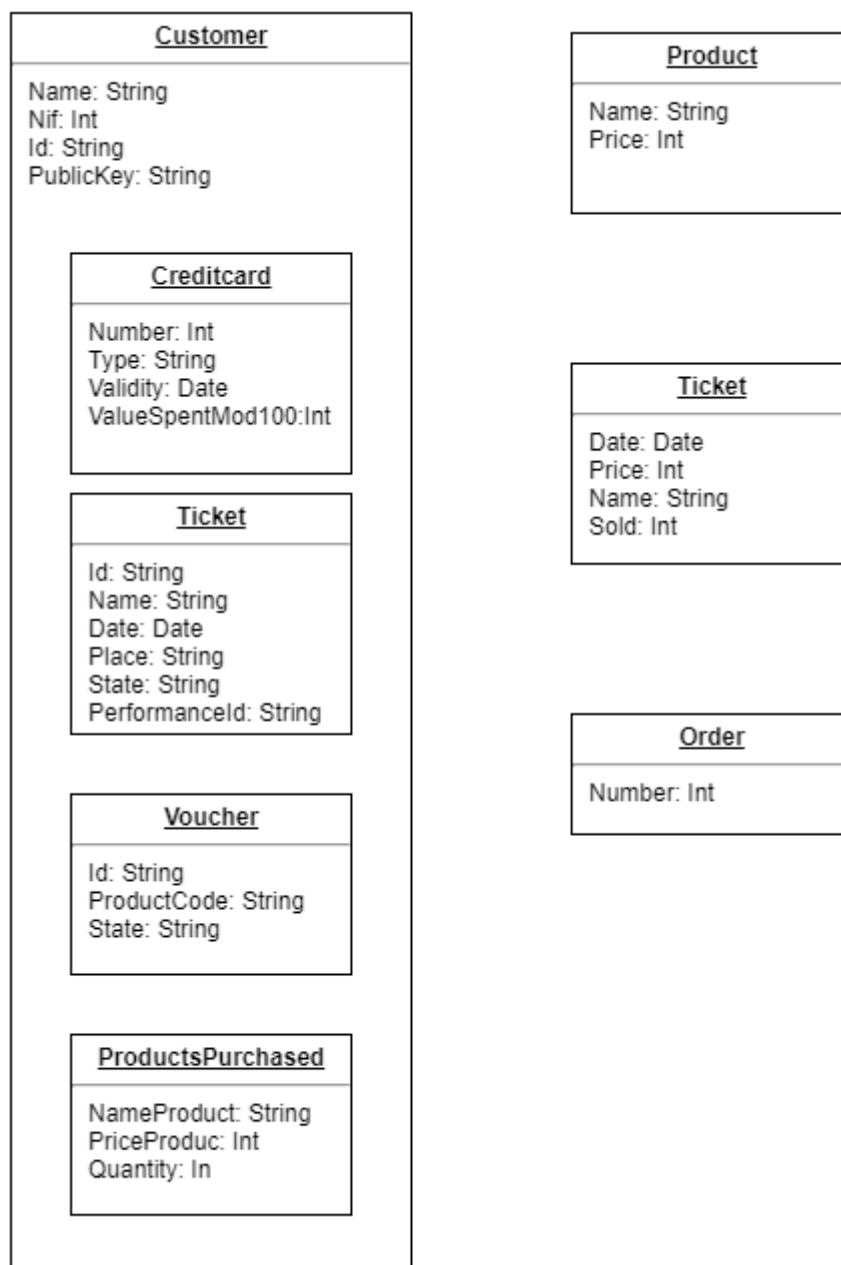
A biblioteca “Shared Library” contém um conjunto de classes que permitem realizar operações que são utilizadas pelas diferentes aplicações. Esta biblioteca implementa uma classe abstracta que faz a leitura do QR code e que será estendida por uma das classes das aplicações que lêem QR codes, contém as classes “Voucher”, “Product” e “MyDate” utilizadas por todas as aplicações e contém a classe “KeyStoreManager” que faz a gestão

das chaves, da assinatura de algumas das mensagens e das transformações que se podem fazer aos bytes assinados, como por exemplo, passar um byte[] para uma string Base64 e passar uma string Base64 para um byte[].

Estruturação dos dados

Modelo da base de dados

Utilizamos uma base de dados não relacional pelo facto de não ter esquema de dados fixo, permitindo uma fácil e eficaz recuperação dos dados de um utilizador, tais como, bilhetes, vouchers, cartão de crédito ou até mesmo os produtos comprados. Esta base de dados foi colocada no Firebase.



Modelo de dados do QR code

Há dois modelos de dados para os QR code. Um que é usado pela/para a “Validation App” e outro que é usado pela/para a “Cafeteria App”.

O modelo usado pela “Validation App” é o seguinte: UserId|TicketId>TicketQty...

Por exemplo: UserId|TicketId1>TicketQty1>TicketId2>TicketQty2

Podem ser colocados 0 ou mais bilhetes e cada bilhete tem sempre uma quantidade associada. O campo UserId é obrigatório. Em termos práticos, a interface da “Customer App” não permite a colocação de zero bilhetes.

Já o modelo usado pela “Cafeteria App” é o seguinte:

UserId|ProductId>ProductQty...|VoucherId>VoucherId...

Por exemplo:

UserId|ProductId1>ProductQty1>ProductId2>ProductQty2>ProductId3>ProductQty3|VoucherId1>VoucherId2

Podem ser colocados 0 ou mais produtos e vouchers e cada produto tem sempre uma quantidade associada. Tal já não acontece com os vouchers. O campo UserId é obrigatório. Em termos práticos, a interface da “Customer App” não permite a colocação de zero produtos, mas permite a colocação de zero vouchers.

Neste caso, na comunicação entre a “Customer App” e a “Cafeteria App”, a mensagem é assinada, sendo que para o exemplo mencionado são realizadas as seguintes transformações:

1. Obter os bytes da string;
2. Concatenar a assinatura dos bytes ao array de bytes;
3. Passar o array de bytes com assinatura para uma string Base64.

Após a obtenção da string Base64, esta é enviada e transmitida pela “Cafeteria App” ao servidor que reverte as transformações e processa os dados.

Funcionalidades implementadas

Foram implementadas todas as funcionalidades pedidas no enunciado para as 3 aplicações Android.

Customer App

- Um utilizador pode realizar o seu registo na plataforma;
- Um utilizador pode visualizar os eventos que se irão realizar;
- Um utilizador pode comprar bilhetes para um evento;
- Um utilizador pode seleccionar quais os bilhetes que quer validar na “Validation App”;
- Um utilizador pode visualizar os bilhetes e vouchers disponíveis;
- Um utilizador pode visualizar quais os produtos disponíveis na cafeteria;
- Um utilizador deve mostrar o QR code gerado para leitura na “Validation App”, o QR code pode conter no máximo 4 bilhetes e todos devem ser do mesmo evento;

- Um utilizador deve mostrar o QR code gerado para leitura na “Cafeteria App” para proceder ao pedido de compra de produtos na cafetaria;
- Um utilizador pode adicionar no máximo 2 vouchers a cada pedido de compra.

Validation App

- Um utilizador pode, após a leitura do QR code gerado pela “Customer App”, visualizar se os seu bilhetes foram validados corretamente ou não.

Cafeteria App

- Um utilizador pode, após a leitura do QR code gerado pela “Customer App”, visualizar o número do pedido, os produtos pedidos, os vouchers aceites pelo servidor, assim como o valor descontado no cartão de crédito tendo em conta os vouchers.

Funcionalidades reformuladas pela equipa

De todas as funcionalidades implementadas pela equipa de desenvolvimento, optamos por reformular a operação de validação dos bilhetes.

Para a validação dos bilhetes, no enunciado pedia que os parâmetros da operação fossem: user id, número de bilhetes, id dos bilhetes e a data do evento. No entanto, não achamos necessário serem mandados tantos campos, pois era informação redundante, passando assim a operação a ter apenas o user id e um conjunto de ids de bilhetes.

Depois, assumimos que podia haver mais que um evento no mesmo dia daí não podermos verificar se um bilhete é do mesmo evento que outro bilhete apenas pela data. Daí, nós no modelo de dados adicionamos um campo redundante ao bilhete “performancelId” que é o id do evento.

Assim o servidor, apenas tinha que verificar se os bilhetes pertenciam ao user em questão e verificar se todos os tickets tinham o mesmo campo “performancelId”.

Desta forma, tornamos a app mais genérica por permitir a realização de vários eventos no mesmo dia.

Testes de aceitação

Para verificar se as funcionalidade implementadas pela equipa estão a funcionar conforme o especificado no enunciado, não foram quaisquer tipo de testes automatizados.

No entanto, foram realizadas várias operações e verificou-se se efetivamente estavam a ocorrer os resultados esperados, quer na base de dados, quer na resposta retornada ao pedido, quer na interface do utilizador.

Foram efetuadas experiências para o caso de o utilizador inserir tipos de dados que não respeitassem os pretendidos assim como o não preenchimento de campos fundamentais para a realização da operação e os resultados foram os esperados.

Manual de utilização

Customer App

O utilizador, ao aceder à aplicação pela primeira vez, é redirecionado para proceder ao registo (Fig. 1). Após o preenchimento dos campos, é necessário clicar no botão do canto superior direito para finalizar a operação, necessitando de ligação à Internet. Após a realização do registo, é apresentado o ecrã Dashboard, com todos os eventos que se irão realizar (Fig. 2), aparecendo as quantidades de cada evento pré-preenchidas a zero.

New user	REGISTER
Name: Luís	
NIF: 123456789	
Credit card data:	
Type: Débito	
Number: 123456789	
Validity: 2018-10-20	

Fig. 1. Registo

Dashboard	BUY
evento4 2019-10-14 11:23	€14.00 Qty: 0
Festival da canção 2019-10-28 11:23	€18.00 Qty: 0
Gala do ano 2019-10-28 11:23	€16.00 Qty: 0
Festival da música 2019-11-11 11:23	€13.00 Qty: 0
Concerto do ano 2020-00-23 11:23	€11.00 Qty: 0
Evento do ano 2020 2020-00-23 11:23	€15.00 Qty: 0
Concerto 2020-00-30 11:23	€10.00 Qty: 0
Concerto 2020-01-12 11:23	€12.00 Qty: 0
Novo evento 2020-02-20 11:23	€17.00 Qty: 0
25 de abril 2020-03-25 10:23	€19.00 Qty: 0

Fig. 2. Dashboard

O utilizador poderá inserir as quantidades de bilhetes que deseja e clicar no botão do canto superior direito (Fig. 3), sendo redirecionado para a confirmação da compra (Fig. 4), onde terá que confirmar a sua compra.

Dashboard		BUY
evento4 2019-10-14 11:23	€14.00	Qty: 0
Festival da canção 2019-10-28 11:23	€18.00	Qty: 1
Gala do ano 2019-10-28 11:23	€16.00	Qty: 0
Festival da música 2019-11-11 11:23	€13.00	Qty: 0
Concerto do ano 2020-00-23 11:23	€11.00	Qty: 0
Evento do ano 2020 2020-00-23 11:23	€15.00	Qty: 0
Concerto 2020-00-30 11:23	€10.00	Qty: 0
Concerto 2020-01-12 11:23	€12.00	Qty: 0
Novo evento 2020-02-20 11:23	€17.00	Qty: 0
25 de abril 2020-03-25 10:23	€19.00	Qty: 0

Fig. 3. Lista de desejos

Checkout		BUY
		Total price: €18.00
Festival da canção 2019-10-28 11:23	€18.00	Qty: 1

Fig. 4. Checkout

Para ser possível aceder ao menu de navegação, o utilizador deve deslizar o dedo do lado esquerdo para o lado direito do ecrã (Fig. 5).

Dashboard	BUY
Select tickets	Qty: 0
Transactions	Qty: 0
Select products	Qty: 0
	Qty: 0
	Qty: 0
	Qty: 0
	Qty: 0
	Qty: 0
	Qty: 0
	Qty: 0

Fig. 5. Menu de navegação

O utilizador pode visualizar as suas transações clicando no elemento “Transactions” do menu, mostrando os bilhetes comprados não usados e os vouchers não usado (Fig. 6), pois neste caso ainda efetuou a compra de produtos,

Transactions		
Ticket		Place
Festival da canção		0
2019-10-28 11:23		
Vouchers		
popcorn		
Product	Price	Quantity

Fig. 6. Transações

Após deslizar novamente ecrã para a direita, acede ao menu e escolhe o elemento “Select tickets”, onde pode visualizar quais os bilhetes que tem disponíveis para validar na Validations App (Fig. 7), assim como escolher quais os que quer validar através do preenchimento da check box do bilhete pretendido (Fig. 8).

Select tickets		FINISH
Festival da canção		Room place:0 <input type="checkbox"/>
2019-10-28 11:23		

Fig. 7. Bilhetes não validados

Select tickets		FINISH
Festival da canção		Room place:0 <input checked="" type="checkbox"/>
2019-10-28 11:23		

Fig. 8. Bilhetes a validar

Clicando no botão do canto superior direito é mostrado o QR code que deverá ser lido pela Validation App, finalizando a validação (Fig. 9).

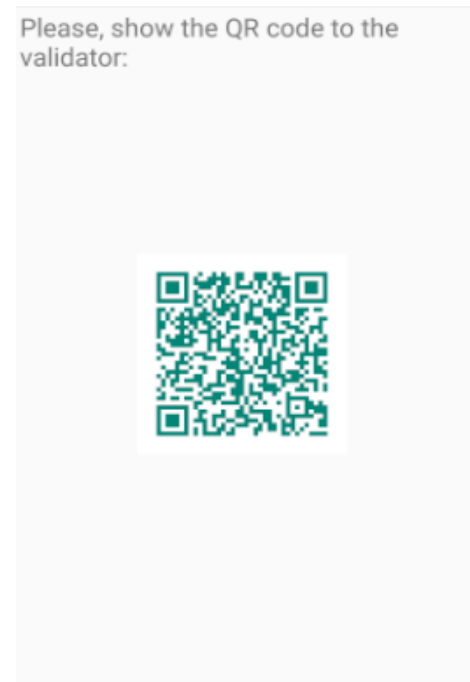


Fig. 9. QR code para validação na Validation App

Mais uma vez, após o utilizador deslizar o menu para a lado direito e escolher o elemento “Select Products”, o utilizador poderá escolher quais os produtos que deseja comprar na Cafeteria App (Fig. 10), procedendo ao preenchimento das quantidades pretendidas (Fig. 11) e, por fim, clicar no botão do canto superior direito.

Select products		BUY
Coffee	€1.00	Qty: <input type="text" value="0"/>
Popcorn	€2.00	Qty: <input type="text" value="0"/>
Sandwich	€5.00	Qty: <input type="text" value="0"/>
Soda drink	€4.00	Qty: <input type="text" value="0"/>

Select products		BUY
Coffee	€1.00	Qty: <input type="text" value="1"/>
Popcorn	€2.00	Qty: <input type="text" value="0"/>
Sandwich	€5.00	Qty: <input type="text" value="0"/>
Soda drink	€4.00	Qty: <input type="text" value="0"/>

Fig. 10. Produtos disponíveis a comprar

Fig. 11. Produtos a comprar

Nesta etapa, poderá visualizar quais os vouchers que tem disponíveis (Fig. 12).

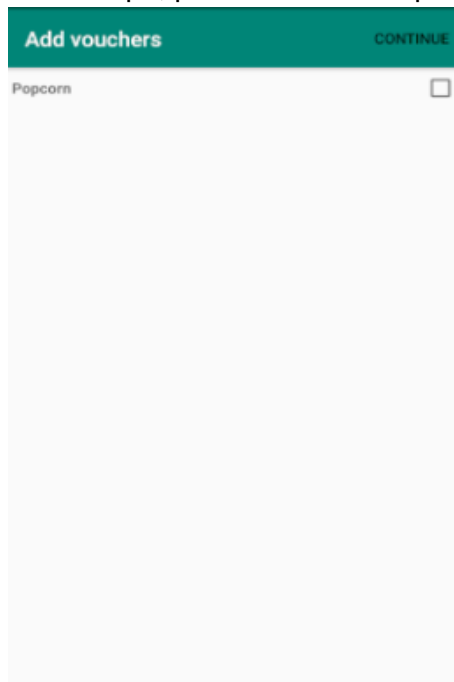


Fig. 12. Lista de vouchers do utilizador

O utilizador poderá escolher no máximo 2 vouchers a adicionar à compra de produtos para obter descontos caso se apliquem e por fim clicar no botão do canto superior direito (Fig. 13).

Será mostrado novamente um QR code que deve ser apresentado à Cafeteria App para que seja feito o processamento do pedido.(Fig. 14).

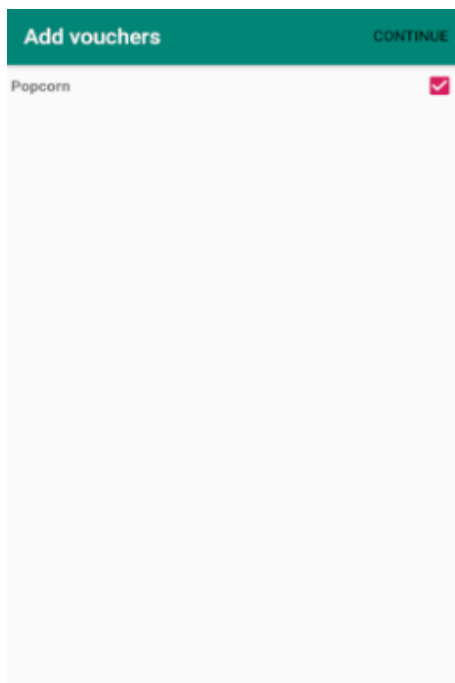


Fig. 13. Adicionar vouchers à compra

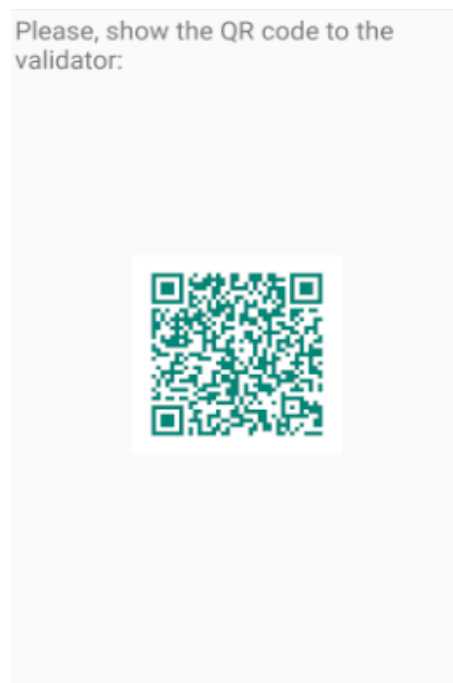


Fig. 14. QR code para processamento na Cafeteria App

Após a compra dos produtos na “Cafeteria App” o utilizador poderá visualizar as suas transacções realizadas (Fig 15).

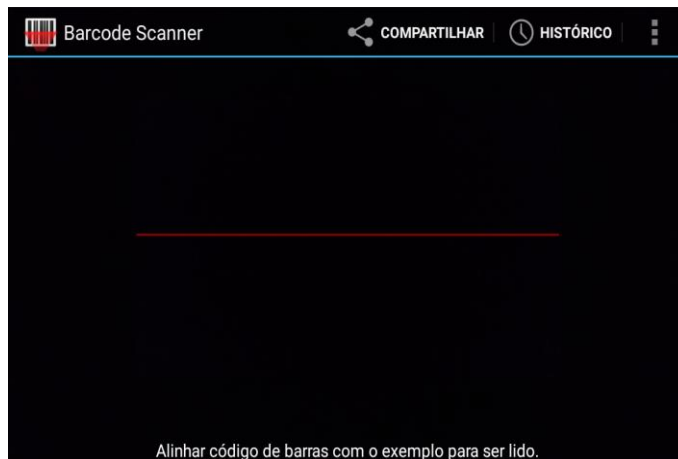
Transactions		
Ticket		Place
Vouchers		
popcorn		
Product	Price	Quantity
coffee	€1.00	1

Fig. 15 Lista de transacções

Validation App

Ao iniciar a aplicação da “Cafeteria App”, irá aparecer um ecrã para a leitura do QR code gerado pelo utilizador (Fig 16).

Após a leitura e conexão com o servidor mostra os dados relativos à validação dos bilhetes do utilizador (Fig 17).



16 Ecrã inicial da “Validation App”



Fig. 17 Ecrã de validação dos bilhetes

Cafeteria App

Ao iniciar a aplicação da “Cafeteria App”, irá aparecer um ecrã para a leitura do QR code gerado pelo utilizador (Fig 18).

Após a leitura e conexão com o servidor mostra os dados relativos da compra do utilizador (Fig 19).

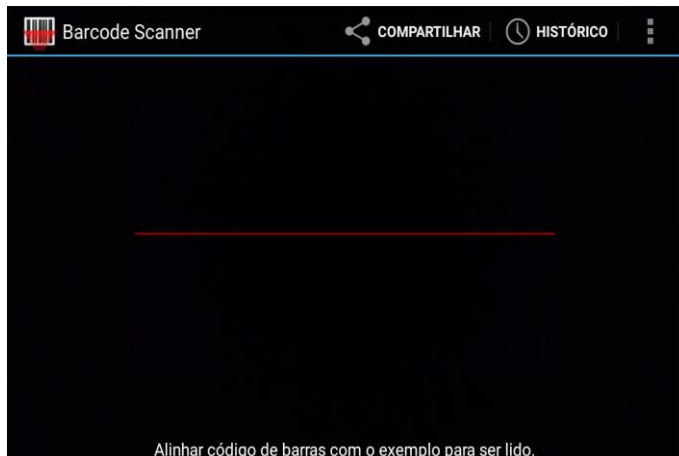


Fig.18 Ecrã inicial da “Cafeteria App”

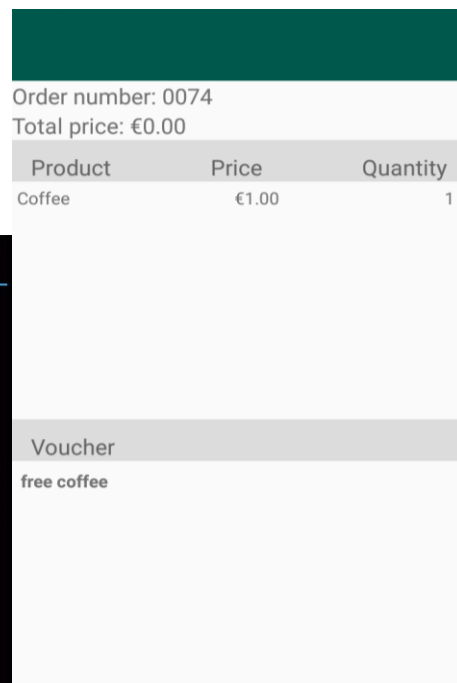


Fig. 19 Ecrã após compra dos produtos

Referências

1. Miguel Pimenta Monteiro. Mobile computing tutorials & demos. Acedido em outubro de 2018. <https://paginas.fe.up.pt/~apm/CM/>
2. Google. Android Developers. Acedido em outubro de 2018. <https://developer.android.com/>
3. <https://www.androidauthority.com/use-android-keystore-store-passwords-sensitive-information-623779/> Acedido em novembro de 2018.
4. <https://codedump.io/share/9UiToGn1kp26/1/crash-casting-androidkeystoreprivatekey-to-rsaprivatekey> Acedido em novembro de 2018.
5. <https://paginas.fe.up.pt/~apm/CM/docs/MsgSignNFCandQR.zip> Acedido em novembro de 2018.