

# 1º Trabalho Sistemas Operativos 2019/2020

Ricardo Mochila - 37762

Inês Veríssimo - 40102

Abril 2020

## Introdução

De modo a criar uma solução para um cenário de escassez alimentar, desenvolveram-se duas aplicações, cliente e servidor. Neste relatório irá ser explicado cada ficheiro java que compõe o trabalho, bem como as operações que podem ser realizadas pelo cliente.

Ao longo da implementação deste trabalho, decidiu-se que não seria possível adicionar necessidades enquanto não estiver estabelecida a ligação ao servidor, pois os produtos podem existir em alguma loja e não é possível verificar-se enquanto se estiver offline.

## Tecnologias

Neste trabalho foram usadas as tecnologias java rmi, para estabelecer a relação cliente servidor, Postgres SQL para a base de dados do servidor e hyperSQL para a base de dados embebida.

## Ficheiros

### Client

Em primeiro lugar, são definidos Host e Porto para conexão com o servidor. O cliente tem acesso a um menu que corre até que seja terminada a sessão. Este menu dispõe de várias opções que podem ser realizadas pelo utilizador, como consultar um produto, registar um produto numa loja, informar a necessidade de um produto. Se o cliente declarar uma necessidade, o servidor atribui-lhe um id para que, quando existirem atualizações sobre o produto pretendido, saiba que cliente informar. No caso de não se conseguir a ligação ao servidor, a o registo da necessidade é encaminhado para uma base de dados local.

## Server

É estabelecida a ligação com o cliente. Através do ficheiro *Invocations*, o servidor mostra os pedidos que o cliente vai fazendo ao longo do tempo, bem como o que acontece ao nível do sistema e das bases de dados.

## Invocations

Este ficheiro contém os métodos que tratam dos pedidos, assim como da interação com as bases de dados. Os métodos *registarProduto*, *consultarProduto*, *necessidadeProduto* e *necessidadeGeral* interagem base de dados.

Sempre que é feito um **pedido de registo** de um produto, se ainda não existir uma tabela de produtos, esta irá ser criada e serão inseridos o produto e a loja correspondente. No caso de o não existir ligação ao servidor, o registo do produto é guardado, temporariamente, numa base de dados local até que se consiga ter acesso ao servidor. Quando isso acontece, a base de dados fica vazia e os produtos são registados na base de dados dos produtos.

Ao **consultar um produto**, é feita uma pesquisa na base de dados e devolvida, caso exista, a loja que lhe corresponde. Neste caso, se não existir conexão ao servidor informa-se o cliente de que o seu pedido não poderá ser realizado pois não é possível saber quais os produtos existentes na base de dados.

As **necessidades**, por sua vez, separam-se em dois tipos. Necessidade de um produto numa loja e necessidade geral, em que se regista uma necessidade de um produto que não existe em nenhuma loja da comunidade. Se não existir em nenhuma loja, os pedidos ficam guardados numa tabela com o nome *necessidades* que é atualizada, sempre que se faz um registo que corresponda à informação que contém. Se o servidor não estiver ativo, não se regista uma necessidade, pelo mesmo motivo que é referido na consulta de um produto.

O método **check** é utilizado para atribuir um id ao user quando existe uma necessidade geral.

O **checkUpdates** é um método responsável por informar o utilizador sobre atualizações ao seus registos de necessidade. É executado sempre antes de o cliente fazer um novo pedido.

## **FileSaving**

Este ficheiro contém os métodos `save` e `load` que atribuem um id ao user. O ficheiro que é criado aqui, não pode depois ser alterado pelo cliente, por motivos de segurança.

## **LocalDatabase**

Este ficheiro contém os métodos que gerem a base de dados local, como `save`, para guardar o produto e a respetiva loja, `delete` que apaga as informações que a tabela contém, `isEmpty` para verificar se está vazia e `convertData` converter as informações para a base de dados dos produtos.

## **PostgresConnector**

É o ficheiro que trata da conexão com a base de dados.

## **Conclusão**

Neste trabalho foi desenvolvida uma aplicação que permite a consulta de produtos em lojas de uma comunidade, num cenário de necessidade. Foram implementadas funcionalidades que vão de encontro ao proposto no enunciado e que supotam falhas na comunicação com o servidor. Utilizaram-se tecnologias lecionadas e apropriadas.