

## **Gestão Inteligente de Produtos**

42142 Ana Santos, 42142@alunos.isel.ipl.pt, 967064568  
42162 Inês Soares, 42162@alunos.isel.ipl.pt, 914182857  
42181 Nuno Veloso, 42181@alunos.isel.ipl.pt, 910364327

Orientadores Nuno Datia, datia@isel.ipl.pt  
Matilde Pato, mpato@deetc.isel.pt

Proposta de Projeto realizado no âmbito de Projeto e Seminário,  
do curso de licenciatura em Engenharia Informática e de Computadores  
Semestre de Verão 2017/2018

Março de 2018

# Introdução

Com o *boom* da *IoT* nos dias de hoje, o homem ainda realiza tarefas no seu dia-a-dia que poderiam ser substituídas por recursos mais inteligentes. Libertando-o para outras atividades como lazer. Assim, o nosso trabalho vai no sentido de dar respostas a ”como evitar transtornos causados na altura de reabastecer a nossa despensa? controlo de stock de alimentos e outros produtos? artigos fora de prazo?”. Se entendermos que a nossa casa funciona como uma empresa, onde existem pessoas que podem realizar as mesmas tarefas, e.g. ir às compras, tornar o sistema de controlo de stock mais eficiente nas nossas casas é como planear e montar uma infraestrutura de ligação, processamento e armazenamento na gestão de frotas numa transportadora.

De forma a responder às perguntas levantadas anteriormente, pretendemos desenhar duas aplicações, uma móvel e uma web. Aplicações estas que interagem diretamente com uma Web API, que está relacionada com uma Base de Dados, através de uma Camada de Acesso a Dados (DAL) e de uma com a Lógica de Negócio (BLL). A recolha de dados, i.e., informação dos produtos existentes, é lida por um leitor de *tags* (NFC ou RFID) e transmitida para a Web API, para ser armazenada. Os locais de armazenamento de produtos devem dispor de dispositivos de hardware, equipados com scanners capazes de ler os dois tipos de *tags* e sensores de movimento. A adoção destas peças é a chave na monitorização dos stocks, é de realçar a dependência do projeto nelas para a distinção do tipo de movimento, de entrada ou saída.

No âmbito do nosso projeto assumimos a existência de dois estados para os produtos, avulsos e embalados. Os primeiros são conservados em sistemas de arrumação (caixas, sacos, etc.), que contém *tags* NFC programáveis por *smartphones*. Os detalhes dos produtos são especificados pelo utilizador e carregados para a *tag*. Enquanto que para os produtos embalados, admitimos que os produtores utilizam *tags*, NFC ou RFID, para guardar os rótulos em formato standard (CSV).

# Análise

Para a realização deste projeto encontrámos algumas dificuldades que se prendem com o facto de os produtos não terem rótulos digitais. Isto é um problema para a concretização do projeto, na medida em que se torna menos eficiente a obtenção dos dados presentes nos produtos. Contudo, assumindo que este dilema é resolvido fora do âmbito do projeto, apenas é preciso definir um formato standard de como os dados devem ser armazenados nas *tags*, que podem ser NFC ou RFID. Formato este que deve ser respeitado por todos os produtores, assim, os produtos em vez de terem um código de barras, têm uma *tag* RFID ou uma *tag* NFC, com a informação necessária. E aplicando a responsabilidade da leitura das *tags* nos dispositivos de hardware, e não fazendo este parte do campo de ação do nosso projeto, apenas necessitamos de gerir a informação transmitida pelos mesmos.

Como já deveríamos saber a segurança da informação nos dias que correm é um assunto com que todos nos devemos preocupar. E os dispositivos *IoT* não fogem à regra. Desta forma temos de nos preocupar com a segurança dos dados pois se existir uma vulnerabilidade estamos a comprometer a casa e a segurança dos nossos utilizadores. Por estas razões óbvias temos de adotar funcionalidades de segurança ao nosso projeto.

O projeto será dividido em 5 camadas principais, inter relacionadas entre si. A camada mais acima é a das aplicações, uma mobile, para a plataforma *Android* e usando a linguagem *Kotlin*, e uma web, utilizando a linguagem *JavaScript*, com a *framework Express*. Abaixo encontra-se a Web API, para a construção desta faremos uso de uma *framework* da *Spring*, chamada de *Spring Boot*. Esta ferramenta nunca foi utilizada, porém está a ser lecionada numa cadeira assistida por todos. A terceira camada, Camada da Lógica de Negócio (BLL), cuja separação da camada que a antecede é uma linha muito ténue. A Camada de Acesso a Dados, a segunda camada, será realizada com a linguagem de programação JAVA, com JDBC API, e é responsável por aceder à base dados com uma maior abstração, podendo-se escrever instruções SQL e obter os dados através de uma interface. Por último temos a Base de Dados, recorrendo ao SGBD *PostgreSQL*, também ainda não usado.

No caso de existir uma desistência de um elemento do grupo, o trabalho fica comprometido pois não será completado na sua totalidade. Se um elemento eventualmente desistir, cuja a responsabilidade seja a de realizar uma camada imprescindível para a realização do resto do trabalho, um outro elemento terá de prescindir das suas tarefas para realizar o trabalho do

elemento que desistiu, caso contrário, o projeto não se desenvolveria. O risco seria as camadas superiores não ficarem realizadas na sua totalidade ou da forma mais correta.



# Planeamento

