

UNIVERSIDADE AUTÓNOMA DE LISBOA LUÍS DE CAMÕES
Departamento de Ciências e Tecnologias
Sistemas Distribuídos e Paralelos 2020/2021
Projeto

Datas

- Enunciado: 29/11/2020
- Entrega no *e-learning*: 17/01/2021 até às 23:59

Objetivos

Pretende-se a construção de uma aplicação para gestão de transporte de itens de uma empresa. A empresa deposita itens num armazém central, do qual originam várias entregas. Cada entrega é composta por itens que estão no armazém central, em quantidade suficiente. O sistema deverá permitir:

- O registo de um novo item, com a indicação do seu nome;
- O registo do depósito de um item, com a indicação da quantidade;
 - O item tem que estar previamente registado.
- O registo de uma entrega de itens, com a indicação da quantidade de cada item, e a descrição do local de entrega;
- A consulta da coleção de itens registados;
 - Para cada item mostra: nome, descrição, e quantidade em armazém.
- A consulta da coleção de itens em stock no armazém central;
 - Para cada item mostra: nome, descrição, e quantidade em armazém;
 - Um item está em stock se a sua quantidade em armazém for superior a zero.
- A consulta das entregas registadas;
- A alteração de descrição de itens;
- A alteração do local de destino de uma entrega;
- A eliminação de um item, caso nunca tenha sido registado um depósito ou entrega com este.

Tarefas

O projeto deverá modelar e implementar um sistema distribuído com os seguintes nós:

1. (4 valores) Base de dados para persistência de informação;
 - Responsável por guardar toda a informação relevante no sistema;
 - Pode ser utilizado qualquer SGDB, desde que seja possível recorrer a *containers* Docker Linux. Exemplos são PostgreSQL ou MariaDB;
 - Só será assegurado apoio para PostgreSQL.
 - Só existirá um nó deste tipo no sistema.
2. (12 valores) API REST para disponibilização da funcionalidade;
 - Toda a informação trocada com a API deve ser em JSON;
 - Inclui *request* e *response*.
 - Deve permitir *criar*, *alterar*, *consultar*, e *eliminar* a informação de negócio;
 - Deve ser o único ponto de acesso à base de dados, através de JDBC, quer diretamente, quer apoiado por um ORM (e.g., Hibernate);
 - Pode ser utilizado qualquer servidor de aplicações Java EE, desde que seja possível recorrer a *containers* Docker Linux. Exemplos são WildFly ou Payara;
 - Só será assegurado apoio para WildFly.
 - Deve ser possível replicar arbitrariamente este nó no sistema.
3. (4 valores) Aplicação web, i.e., interface HTML, para interação com o utilizador.
 - Deve permitir *consultar* a informação de negócio;
 - A aplicação web que serve conteúdo HTML não deve ter acesso à base de dados. No servidor da aplicação HTML, o acesso à informação deve ser feito através da API REST;
 - Não existem restrições quando ao estilo aplicado no HTML, desde que esteja assegurada a leitura da informação;
 - Deve ser possível replicar arbitrariamente este nó no sistema.

As tarefas 1 e 2 são necessárias para obter avaliação.

Regras

O trabalho deve ser realizado por grupos de 4 elementos, previamente registados no *e-learning*.

A entrega do trabalho é feita no *e-learning*. Devem ser entregues dois ficheiros, com o seguinte conteúdo:

ZIP com todo o código fonte, incluindo:

- Dockerfiles;
- Esquemas de criação de base de dados, i.e., código SQL.

Pode ser entregue o diretório com todo o projeto IntelliJ;

PDF com relatório, contendo:

- Identificação do grupo de trabalho,
- Decisões arquiteturais tomadas:
- Listagem de operações suportadas pela API REST onde, para cada operação, deve surgir:
 - Verbo;
 - Esquema de URL;
 - Descrição sumária da operação;
 - Códigos de resposta HTTP aplicáveis, em caso de sucesso e insucesso.
- Eventuais diagramas;
- Outra documentação considerada relevante.

A entrega não será válida se for utilizado um formato de compressão além do zip para o código fonte, ou o formato do relatório além do pdf, ou se não for possível abrir algum dos ficheiros.