deti universidade de aveiro departamento de eletrónica, telecomunicações e informática

Projeto individual

Updated: 2023-06-21.

Âmbito funcional da aplicação a desenvolver

Módulo identificado como "eStore" referido nos termos do projeto de grupo, ou seja, uma interpretação (simplificada) de uma loja online de *eCommerce* para um dado tema concreto (e.g.: loja online de flores,…).
Naturalmente, não se espera uma solução completa, admitindo-se várias simplificações (pode não ter autenticação e segurança nas sessões). Deve pelo menos permitir listar produtos, formar um carrinho de compras, realizar compras/encomendas, seguir posteriormente as encomendas anteriores.

Requisitos técnicos da implementação

A solução deve seguir uma arquitetura de camadas, usando o *framework* Spring Boot, com um *frontend* e um *backend*:

- O armazenamento (catálogo de produtos, encomendas) deve ser feito numa **base de dados** persistente.
- Deve haver um *backend* com uma **API (REST)**, permitindo a **gestão completa** da eStore através de um cliente REST externo (e.g.: ações via Postman).
- O *frontend, na* **Web,** não precisa de ser feito necessariamente com a tecnologia Java/Spring Boot, podendo ser usado outro tipo de *framework* para implementação do *frontend*.

Práticas necessárias:

- Gestão de versões do código num repositório Git, com *feature-branching.*
- Devem existir (alguns) *pull-request* de exemplo (simulando um segundo programador associado ao projeto)
- Ambiente de execução baseado no conceito de "*infrastructure as code*" (e.g.: containers Docker). Incluir no repositório configurações/scripts necessários.
- Valorização: instalação dos serviços em ambiente de servidor remoto (e.g.: infraestrutura cloud).

Elementos da estratégia de QA:
- "*clean code*";
- testes relevantes, para diferentes aspetos/camadas do sistema;
- *workflow* de CI, integrando a verificação dos testes, análise estática de código e *branches* protegidos por *quality gates* adequados.
- *Workflow* de CD para *branches* designados, usando *containers*.
- monitorização das operações (*observability* dos sistemas em produção)
- deve testar a integração com uma implementação viável (real ou sintetizada) da API da *Picky Platform*, ou seja, deve simular a utilização da API da plataforma externa de gestão dos *pick-up points,* mas não é preciso construí-la (e.g.: *mocking* nos testes, recurso a WireMock, etc).