

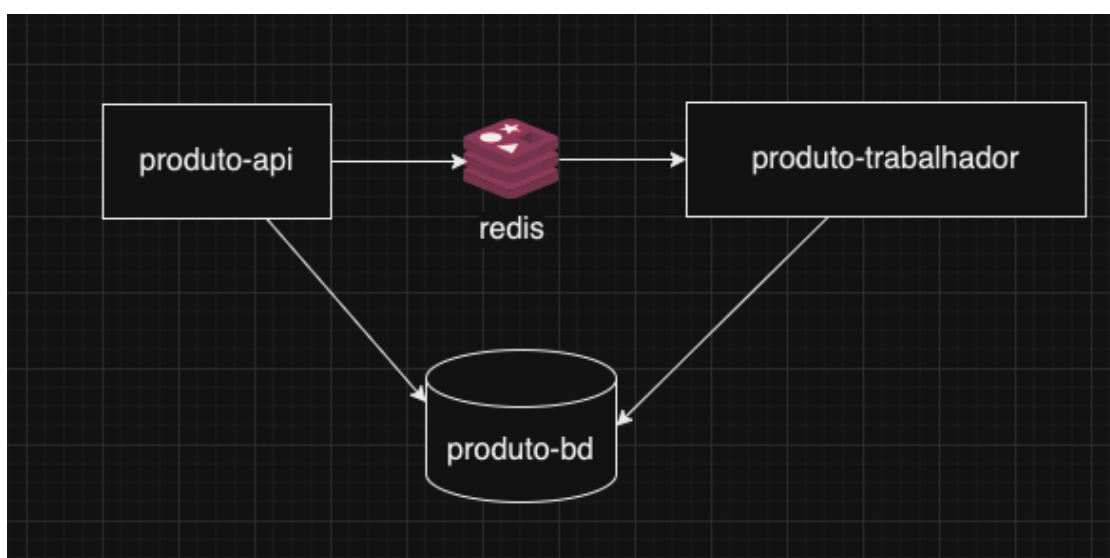
# Desafio técnico de backends 01/2024

Neste desafio, você irá desenvolver um projeto para o cadastro de produtos.

Cada produto possui as seguintes informações:

- Código: Código ou chave inteira única do produto.
- Título: Título do produto.
- Preço: Preço do produto.

A *arquitetura* de nosso projeto é a seguinte:



O **produto-api** é uma aplicação que trabalha sob o padrão API Rest. Teremos as seguintes APIs publicadas:

- Cadastrar produto: Cadastro do produto diretamente no banco de dados.
  - Cada produto possui um código único.
  - Cada produto deve ter um título e um preço maior que zero.
- Pesquisar o produto pelo código. Pesquisa por um produto que possua o código informado.
- Pesquisar produtos pelo título. Esta pesquisa irá procurar pelos produtos que contém parte do texto informado na pesquisa. Esta pesquisa será paginada.
- Atualizar preço do produto. Atualiza o preço de um produto, conforme o código informado.
- Cadastrar produto de arquivo CSV. Neste cadastro, enviamos pela API um arquivo CSV com os dados do produto. A API irá salvar o arquivo para que a aplicação produto-trabalhador faça o cadastro dos itens do arquivo CSV de forma *assíncrona*. Logo, a API notifica o trabalhador de que há um arquivo para ser processado.
  - Será que precisamos de uma outra API para sabermos se o arquivo já foi processado? Como seria? Projete para nós.

O **redis** será nossa fila (<https://redis.com/glossary/redis-queue/>). Logo, o **produto-api** envia as notificações para o **produto-trabalhador** através de um Redis.

O **produto-trabalhador** é responsável por analisar e inserir os registros de produtos de um arquivo CSV. Em caso de erros, quer do arquivo quer dos dados do arquivo, nós iremos descartar toda a *carga* deste arquivo.

O **produto-bd** é nosso banco de dados. Você pode escolher entre o PostgreSQL e o MongoDB.

São requisitos deste projeto:

- Tanto a API como o trabalhador podem ser em Python ou NodeJs.
  - O projeto pode ser *mono-repo*, em um único repositório.
- Todo o código **deve** estar em português sem acentuação
  - Comentários, funções, métodos, classes, tabelas... tudo em português.
  - Exemplo de função no Python: registrar(p: Produto).
- As configurações de sistema devem estar em variáveis de ambiente.
  - Exemplo: URI de conexão com o banco de dados.
- Criar um documento .md para descrever como instalar e executar e testar a aplicação.

São diferenciais para o projeto:

- Trabalhar e configurar com Docker e/ou docker-compose.
- Utilizar algum padrão de projeto ou técnica de codificação *limpa*.
- Estrutura e organização do código.
- Testes unitários, ... , de integração.
  - Cobertura de código.
- Simplicidade e clareza do código.