

## PROVA PARA DESENVOLVEDOR BACKEND JAVA - ERP

- Essa prova tem como objetivo avaliar três possíveis níveis de conhecimento usando a mesma problemática, sendo cada nível mais complexo do que o outro.
- Você pode escolher o nível de dificuldade que está mais de acordo com o seu grau de conhecimento para resolver a prova.
- Cada nível incrementa as tecnologias mínimas que devem ser utilizadas, os requisitos que devem ser implementados e os critérios de aceitação.
- Não é necessário fazer a parte do front-end desse desafio, apenas back-end; se você for um programador full stack você receberá uma prova diferente para front-end.
- O resultado da prova deve ser publicado no GitHub (fornecer a URL) ou enviado em um arquivo ZIP.

## PROVA NÍVEL I

Tecnologias mínimas que devem ser utilizadas:

- Banco de dados PostgreSQL
- Java 8+
- JPA
- REST com JSON

Requisitos da prova:

- Deverá ser desenvolvido um cadastro (Create/Read/Update/Delete/List) para as seguintes entidades: produto/serviço, pedido e itens de pedido.
- Todas as entidades deverão ter um ID único do tipo UUID
- No cadastro de produto/serviço deverá ter uma indicação para diferenciar um produto de um serviço
- Deverá ser possível aplicar um percentual de desconto no pedido, porém apenas para os itens que sejam produto (não serviço); o desconto será sobre o valor total dos produtos

Critérios de aceitação

- A prova deverá ser entregue completa (todos os itens resolvidos)
- O código não poderá ter erros de compilação
- Deverá haver uma documentação mínima de como executar o projeto e suas funcionalidades

## PROVA NIVEL II

Tecnologias mínimas que devem ser utilizadas:

- Banco de dados PostgreSQL
- Java 8+
- Maven
- Spring
- JPA
- REST com JSON

Requisitos da prova:

- Deverá ser desenvolvido um cadastro (Create/Read/Update/Delete/List com paginação) para as seguintes entidades: produto/serviço, pedido e itens de pedido.
- Todas as entidades deverão ter um ID único do tipo UUID gerado automaticamente
- No cadastro de produto/serviço deverá ter uma indicação para diferenciar um produto de um serviço
- Deverá ser possível aplicar um percentual de desconto no pedido, porém apenas para os itens que sejam produto (não serviço); o desconto será sobre o valor total dos produtos
- Somente será possível aplicar desconto no pedido se ele estiver na situação Aberto (Fechado bloqueia)
- Não deve ser possível excluir um produto/serviço se ele estiver associado a algum pedido
- Não deve ser possível adicionar um produto desativado em um pedido

Critérios de aceitação

- A prova deverá ser entregue completa (todos os itens resolvidos)
- Deverão ser criados testes automatizados
- O código não poderá ter erros de compilação
- Deverá haver uma documentação mínima de como executar o projeto e suas funcionalidades

## PROVA NÍVEL III

Tecnologias mínimas que devem ser utilizadas:

- Banco de dados PostgreSQL
- Java 8+
- Maven
- Spring
- JPA
- Bean Validation
- QueryDSL
- REST com JSON

Requisitos da prova:

- Deverá ser desenvolvido um cadastro (Create/Read/Update/Delete/List com paginação) para as seguintes entidades: produto/serviço, pedido e itens de pedido.
- Deverá ser possível aplicar filtros na listagem
- As entidades deverão utilizar Bean Validation
- Deverá ser implementado um ControllerAdvice para customizar os HTTP Response das requisições (mínimo BAD REQUEST)
- Todas as entidades deverão ter um ID único do tipo UUID gerado automaticamente
- No cadastro de produto/serviço deverá ter uma indicação para diferenciar um produto de um serviço
- Deverá ser possível aplicar um percentual de desconto no pedido, porém apenas para os itens que sejam produto (não serviço); o desconto será sobre o valor total dos produtos
- Somente será possível aplicar desconto no pedido se ele estiver na situação Aberto (Fechado bloqueia)
- Não deve ser possível excluir um produto/serviço se ele estiver associado a algum pedido
- Não deve ser possível adicionar um produto desativado em um pedido

Critérios de aceitação

- A prova deverá ser entregue completa (todos os itens resolvidos)
- Deverão ser criados testes automatizados
- O código não poderá ter erros de compilação
- Deverá haver uma documentação mínima de como executar o projeto e suas funcionalidades