

Desafio Técnico

Desenvolvedor Jr./Analista de Operações

Este desafio tem como objetivo avaliar o candidato diante de situações que ocorrem no dia a dia de um Desenvolvedor Júnior/Analista de Operações com foco em suas habilidades técnicas.

O teste será dividido em fases, onde cada uma será utilizada para avaliar algumas habilidades mais específicas direcionadas ao desenvolvimento de software, principalmente voltado para ambientes WEB.

Para o desenvolvimento de todas as fases, sugerimos a utilização de algumas linguagens voltadas para WEB: C# - ASP.NET MVC ou .NET Core (**preferência**), PHP, Java, etc.

Não é **obrigatória** a conclusão de todas as fases, mas a conclusão delas será um diferencial.

Fase 1 - Front End

O **front end** pode ser classificado, resumidamente, como a parte visual de um software. Ele é responsável por apresentar ao usuário, de forma gráfica, todos os recursos que um software tem disponível.

Sabendo disso, esta fase deve ser cumprida da seguinte forma:

- Uma tela de login deve ser criada utilizando HTML5 e CSS3.
- Este formulário deve conter apenas um formulário com dois campos (usuário e senha) e um botão para realizar a autenticação.
- O design fica por conta do candidato, ou seja, de livre escolha.
- Sugerimos utilizar alguma biblioteca para auxiliar na montagem do layout. Por exemplo, o Bootstrap.

Ao concluir esta fase, o candidato deve armazenar todos os scripts criados e enviar aos avaliadores.

Fase 2 - Back End

O **back end** é o local de uma aplicação onde os dados são processados sem que o usuário visualize, como um carregamento ou registro de informações em banco de dados, uma leitura de algum arquivo de configuração, criação e leitura de cookies, etc.

Com essas informações, esta fase desenvolvida da seguinte forma:

- O formulário criado deve realizar uma requisição para uma URL que deve validar o usuário e a senha informados. Para que o login seja realizado com sucesso, o usuário deve ser **magalu** e a senha **m@galu123**.
- Caso as informações estejam corretas, o usuário deve ser redirecionado para uma tela com um texto bem simples escrito “Bem vindo ao Magalu.”
- Caso contrário, o formulário de login deve ser apresentado novamente com uma mensagem informando que os dados informados estão incorretos.

Ao concluir esta fase, o candidato deve armazenar todos os scripts criados e enviar aos avaliadores.

Diferencial

- Armazenamento de informações de login em cookie para que, ao entrar na tela de login novamente, o usuário não precise realizá-lo novamente e seja direcionado para a tela de boas vindas.

Fase 3 - Banco de dados

Os **banco de dados** tem como objetivo registrar diversos tipos de informações visando diversas finalidades.

Esta fase consiste em avaliar alguns conceitos e comandos básicos para a gestão de um banco de dados.

O candidato pode escolher entre três ambientes para esta fase, sendo dois **relacionais** (SQL Server ou MySQL) e um **não relacional** (MongoDB).

Ao concluir esta fase, o candidato deve armazenar todos os scripts criados e enviar aos avaliadores.

Para a escolha de um banco **relacional**:

- Criar um banco de dados chamado *db_magalu*.
- Neste banco de dados devem ser criadas três tabelas: `tb_product`, `tb_category`, `tb_product_category`.
 - `tb_product`: deve conter um campo **id** (inteiro, chave primária), um campo **name** (texto) e um campo **price** (decimal).
 - `tb_category`: deve conter um campo **id** (inteiro, chave primária) e um campo **name** (texto).
 - `tb_product_category`: deve conter um campo **id** (inteiro, chave primária), um campo **productid** (inteiro, criando uma chave estrangeira e relacionando à tabela `tb_product`) e um campo **categoryid** (inteiro, criando uma chave estrangeira e relacionando à tabela `tb_category`).
- Após a criação do banco de dados, alguns scripts/comandos devem ser criados:
 - Três comandos **INSERT** para a `tb_product`.
 - Três comandos **INSERT** para a `tb_category`.
 - Três comandos **INSERT** para a `tb_product_category`, assim criando um relacionamento entre produtos e categorias.
 - Um comando **SELECT** na `tb_product` com o objetivo de carregar produtos com preços superiores a R\$ 100,00.
 - Um comando **UPDATE** com o objetivo de atualizar o nome de um dos três produtos, realizando a busca pelo id.
 - Um comando **DELETE** com o objetivo de apagar um dos três produtos, realizando a busca pelo id.

Para a escolha de um banco **não-relacional**:

- Criar um banco de dados chamado *db_magalu*.
- Neste banco de dados devem ser criadas duas coleções: product e category.
- Estas coleções devem possuir as estruturas abaixo, portanto para obtê-las é necessário que alguns registros já sejam inseridos logo após a criação do banco.
 - product: deve conter um campo **id** (ObjectId, chave primária), um campo **name** (texto), um campo **price** (decimal) e um campo **categoryIds** (Array de ObjectId).
 - category: deve conter um campo **id** (ObjectId, chave primária) e um campo **name** (texto).
- Após a criação do banco de dados e das coleções, alguns scripts/comandos devem ser criados:
 - Três comandos **INSERT** para a category.
 - Dois comandos **INSERT** para a product.
 - Um comando **FIND** na product com o objetivo de carregar produtos com preços superiores a R\$ 100,00.
 - Um comando **UPDATE** com o objetivo de atualizar o nome de um dos dois produtos, realizando a busca pelo id.
 - Um comando **REMOVE** com o objetivo de apagar um dos dois produtos, realizando a busca pelo id.

Diferencial

- Utilizar o banco de dados para encontrar o usuário e realizar o login das fases 1 e 2. Para isso será necessário configurar uma conexão com banco de dados e criar uma tabela para usuários.

Conclusão do desafio

Ao concluir o desafio, o candidato deve compactar todos os arquivos gerados (projetos, scripts, etc) e enviar aos avaliadores.