

# Desafio técnico SMN

## Instruções

- Esse documento contém 3 desafios que serão descritos abaixo, todos deverão ser entregues;
- Utilize *Node.js* ou *ASP .NET WebAPI* para o desafio de back-end;
- Utilize *Angular* ou *ASP .NET MVC* para o desafio de front-end;
- Utilize SQL Server ou PostgreSQL para o desafio de banco de dados;
- Utilize o *git* através do GitHub para controle de versão;
- Faça um README com a explicação do seu projeto e a solução que propôs, como fazer o setup, e o mínimo para outro desenvolvedor entender o seu código;
- A entrega deverá ser efetuada com o envio da URL do repositório git.

## Desafio 01

Existem dois arquivos, um arquivo chamado “Produtos.txt” e outro chamado “Vendas.txt” Que contém informações de produtos e vendas de uma loja.

Construa uma API RESTful (JSON) que disponibilize as seguintes funcionalidades com base nos arquivos txt:

1. Ranking de produtos mais vendidos por período.
  - a. **Filtros:** Data inicial(obrigatório) e data final(obrigatório);
  - b. **Retorno:** Lista contendo: nome do produto, quantidade de vendas, valor total das vendas;
  - c. **Observações:** deverá ser possível ordenar os resultados por quantidade de vendas;
2. Lista de produtos vendidos por período.
  - a. **Filtros:** Data inicial(obrigatório) e data final(obrigatório);
  - b. **Retorno:** Lista contendo: nome do produto, data da venda, valor do produto;
  - c. **Observações:** deverá ser possível ordenar os resultados por nome do produto e data da venda. Não deverão aparecer produtos sem venda.
3. Ranking de vendas por mês.
  - a. **Filtros:** ano (não obrigatório).
  - b. **Retorno:** Lista contendo: nome do mês, quantidade de vendas, valor total de vendas
  - c. **Observações:** deverá ser possível ordenar os resultados por nome do produto e data da venda. Nos meses sem vendas de anos que possuem vendas, os valores deverão vir zerados.
4. Estoque inicial dos produtos.
  - a. **Filtros:** ano (não obrigatório).
  - b. **Retorno:** Lista contendo: nome do produto, estoque inicial.
  - c. **Observações:** Considerando o estoque atual dos produtos, e as vendas efetuadas, a API deve retornar o estoque inicial dos produtos.

5. Ranking de clientes.

- a. **Filtros:** CNPJ do cliente (não obrigatório).
- b. **Retorno:** Lista contendo: nome do cliente, produtos que o cliente comprou [nome do produto, valor do produto], quantidade total de produtos comprados, valor total de produtos comprados.
- c. **Observações:** deverá ser possível ordenar os resultados por nome do cliente, valor total de produtos e quantidade total de produtos. O nome do cliente deverá ser recuperado pelo CNPJ utilizando a seguinte API:  
<https://www.receitaws.com.br/v1/cnpj/{cnpj}>  
\*considerar o nome fantasia, se não houver, a razão social.

Desafio 02

Construir uma aplicação (website) para o controle de lista de tarefas. A aplicação tem alguns requisitos mínimos, que são:

1. Cadastrar uma tarefa
2. Excluir uma tarefa
3. Completar uma tarefa
4. Editar uma tarefa
5. Adicionar uma imagem a uma tarefa (utilizando upload ou link)
6. Uma tarefa deve conter no mínimo um nome.

Desafio 03

Construa a modelagem de dados referente ao “Desafio 01”.

Após construir a modelagem escreva as queries referentes aos itens 1, 2, 3, 4 e 5 do “Desafio 01”