



Desafio desenvolvedor Full Stack

Construir um portal de acompanhamento de vendas, utilizando as tecnologias:

- PostgreSQL
- Vue.js (projeto configurado via vue-cli), utilizando [Single File Components](#)
- BootstrapVue
- Versão LTS mais recente do Node JS
- [Fastify](#) (incluindo a utilização de JSON Schema para validar as requisições)
- AWS (EC2 e RDS)
- GitHub

O layout deve ser baseado (réplica ou aproximação) no arquivo layout-desafio.png anexo.

Especificações:

1. Seguem anexas três planilhas de Excel contendo os dados de exemplo que devem ser utilizados para popular o banco de dados.
2. Os dados das planilhas de excel devem compor 3 tabelas diferentes (os nomes das tabelas devem ser iguais aos nomes de cada arquivo) no banco de dados PostgreSQL (RDS da AWS).
3. O esquema do projeto nada mais é que uma construção simples de uma página web. Front-end (Vue.js), Back-end (Fastify) e banco de dados (RDS Postgres).
4. EC2 são os servidores da AWS. O acesso a eles é através de SSH. Essencial subir servidores **Linux**.
5. RDS é o banco de dados gerenciado da AWS. Utilize o PostgreSQL. O acesso para o conteúdo do banco e tabelas pode ser através do pgAdmin4.

Para calcular os valores contidos na tabela do layout em anexo:

1. Para calcular os valores da coluna venda: na tabela **vendas**, calcular a soma da coluna valor_efetivo filtrando por vendas não canceladas e por loja.
2. Para os valores da coluna meta: na tabela **metas**, pegar o valor da coluna meta_valor filtrando por loja.
3. Para calcular o valor da coluna atingimento da meta: calcular o percentual de atingimento da meta com relação a venda (valores já gerados pelos itens 1 e 2).
4. Para calcular a quantidade de produtos/cliente: na tabela **venda_produtos**, calcular a quantidade total de produtos filtrando por loja e por produtos que NÃO sejam do grupo "RETIRADA/SUBST" dividido pela quantidade total de produtos do grupo BURGER filtrando por loja.

Exemplo:

Quantidade total de produtos que não sejam do grupo retirada/subst.: 16

Quantidade total de produtos do grupo burger: 5

Quantidade de produtos/cliente: $16/5 = 3,2$

Para a seta ao lado do número:

- Ela deve ser verde e apontada para cima quando a quantidade calculada for maior que o número da coluna meta_prod_clt contido na tabela **metas**.
- Ela deve ser vermelha e apontada para baixo quando a quantidade calcular for menor que o número da coluna meta_prod_clt contido na tabela **metas**.

5. Para calcular o ticket médio/cliente: usar o valor da venda já calculado no item 1 dividido pela quantidade total de produtos do grupo burger (já calculado em parte do item 4).

Você deverá criar uma conta na AWS para utilizar os recursos que especificamos. A AWS fornece uma taxa para uso grátis dos serviços de EC2 e RDS [gratuitamente por 12 meses](#). Portanto não se preocupe com custos. Tudo que pedimos para fazer está dentro dessa taxa grátis permitida para novos usuários. Qualquer problema que tenha com isso, avise.

O resultado esperado é:

- um (ou mais) repositórios no GitHub contendo o código, junto com instruções simplificadas para executá-lo
- endereço (URL) para acessar o projeto em produção (live)

Para colocar live, não precisa ter DNS. Pode acessar apenas através do IP do servidor na AWS.

OBS: Será necessário apresentar o conteúdo do desafio presencialmente.

O prazo para o desafio é de **uma semana**.

MUITO IMPORTANTE: Caso não seja possível completar o desafio integralmente, entregue o projeto no estado em que se encontra. Alguma coisa é melhor que nada. Isso não é uma prova, é um desafio. Faça o melhor que conseguir!!! Divirta-se!

Qualquer dúvida entre em contato: patrick@ttburger.com.br, mayli@ttburger.com.br ou (21)99601-1161 (whatsapp).