
Desafio – VirteX - Desenvolvimento

Pequeno desafio para processo de seleção do time de Desenvolvimento da VirteX, ele aborda tudo o que utilizamos um pouco aqui:

- Interpretação de Texto para Dados (obtenção de dados de equipamentos);
 - Desenvolvimento Backend, para criar um Web Server que entregue os dados da etapa anterior, mas consultados dentro de um banco de dados;
 - Criação de um banco de dados simples (subindo a infraestrutura e modelagem);
 - Um simples FrontEnd que exige os dados deste Backend.
-

O que você precisa saber antes de começar?

- **Benefícios**
 - Oportunidade de Crescimento;
 - Vale alimentação;
 - Plano de saúde e odontológico;
 - Internet na faixa
 - **Requisitos Obrigatórios**
 - Chegar no Final do Desafio (ou entregar o máximo de passos);
 - Ir bem na entrevista;
 - Possuir disponibilidade;
 - Ser bom comunicador;
 - Ser proativo.
 - **Objetivo do Desafio:**
 - Saber o conhecimento do candidato a respeito das estruturas que utilizamos dentro da VirteX (Banco de Dados, Backend e FrontEnd);
 - Caso o candidato não souber o que está sendo proposto, mas avaliar até onde ele estava disposto para procurar saber;
 - Avaliar o “desenrolometro” do candidato.
-

- **Observações:**

- O desafio pode ser realizado em qualquer banco SQL (Postgres ou MySQL), então fica a seu critério.
- A linguagem de programação fica a critério também. Entretanto, internamente na VirteX, a grande maioria dos projetos estão escritos em NodeJS (no BackEnd) e em ReactJS (no FrontEnd).
- A sugestão de utilização do **Docker** é para trazer facilidade na hora de subir a infraestrutura da aplicação.

Etapas do Desafio

O desafio está sendo dividido em 5 etapas, são elas:

1. Transformar uma saída de texto provindo de um equipamento de rede, em dados. Como trabalhamos com Redes, então as vezes, será necessário realizar esse tipo de tarefa.
2. Subir um Banco de Dados, realizar uma pequena modelagem para armazenar os dados anteriores. Recomendamos subir com [Docker e docker-compose](#), pois assim, conseguiremos reproduzir a aplicação em qualquer máquina. Recomendamos também realizar a utilização de volumes.
3. Criar um Web Server, que possua uma rota que realize a leitura dos arquivos (estaticamente) e realize o armazenamento dos dados obtidos no banco de dados. E uma rota que realize a obtenção destes dados armazenados.
4. Criar um pequeno FrontEnd que consuma estes dados.
5. Subir o projeto em um repositório público. No README.md do projeto, precisa constar o passo a passo para que possamos executar o projeto em nossas máquinas, assim como testar o funcionamento dele.

OBS: não há problema se você não conseguir todas as etapas, o que queremos saber de você, é como você se desenrola neste processo.

Cenário Geral – O que é uma Rede GPON?

A rede GPON é uma rede óptica passiva (Passive Optical Network – PON). Ela permite a adesão de uma rede multiponto, garantido a possibilidade de uma geração de rede Fiber to the Home (FTTH, ou “Fibra para casa”) e Fiber to the Building (FTTB, ou “Fibra para o prédio”). Os equipamentos utilizados não usam energia elétrica para seu funcionamento, de forma que trazem economia.

Redes passivas são aquelas nas quais não há elementos energizados entre a Optical Line Terminal (OLT) e a Optical Network Unit (ONU), baseada em óptica e transmissão de luz.

Fonte: <https://blog.aloo.com.br/rede-gpon/>

Para uma rede GPON, é necessário existir uma OLT (que é o equipamento que transmite luz para os dispositivos na ponta da rede (as ONUs). Cada porta desse equipamento, no caso das redes GPON, podem ser atendidos 128 clientes. Na imagem abaixo, um exemplo de OLT (uma que está em produção na nossa rede), e cada fio que sai do equipamento, é o que suporta os 128 clientes, e chamamos de ramal.



Essa OLT descrita na imagem acima, é apenas um modelo de OLT. Sendo que existe diversas fabricantes, cada uma com uma CLI própria, e saídas próprias de comandos. E será nisso que irá consistir o nosso desafio.

Etapa 01 – Leitura dos arquivos

Para que possamos ter um cenário um pouco real, acabamos pegando a saída do comando nas OLTs que nos listam as informações as ONUs autorizadas, e qual está online ou off-line no momento. O modo de identificação da ONU, além de ser o endereço slot (card)/port (porta do card)/ont_id (identificador a ONU no endereço), também temos o SN do dispositivo (que é uma chave única que também utilizamos para documentar no sistema, pois isso representa um equipamento único dentro do universo de ONUs).

Na pasta de **Inputs**, possuímos 3 arquivos txt.

- OntInfo – Huawei: saída do comando, onde em um único comando, é possível pegar: slot, port, ont_id, sn e status (online e off-line).
- “OntInfo – ZTE – SNs” e “OntInfo – ZTE – SNs – State”: saída de dois comandos, onde em um eu possuo a informação de slot, port, ont_id e sn. E no outro comando, possuo a saída de slot, port, ont_id e status (online e off-line).

Sendo assim, o objetivo desta etapa, é criar um parse para cada informação, onde possuímos as informações padronizadas de: **slot, port, ont_id, sn e state**.

Etapa 02 – Banco de Dados

Com os parsers da etapa 01, vamos precisar guardar as informações obtidas no banco de dados. Além das informações de **slot, port, ont_id, sn e state**. Vamos precisar puxar no FrontEnd algo que diferencie de qual OLT é tal **sn**, se veio da Huawei ou da ZTE.

Nessa etapa, recomendamos utilizar uma imagem **Docker**, para que na hora de executarmos o projeto internamente, possamos conseguir subir o banco de dados sem dificuldades.

Etapa 03 – Web Server/BackEnd

Com o banco de dados criados, e os parsers dos comandos tudo ok, agora é hora de criar um Web Server no qual possuímos duas rotas:

1. Ler os arquivos (estaticamente mesmo) e realizar o registro no banco de dados.
2. Obter os dados gravados no banco.

Etapa 04 – FrontEnd

Com todas as etapas anteriores “no jeito”, agora, só precisamos mostrar os dados em tela, e ter um botão para que possamos fazer a inserção dos dados estáticos no banco de dados, e tal comando ter sido feito pelo o usuário.

Etapa 05 – Documentação

Para a entrega do projeto, preferimos que ele seja entregue em um repositório público do Git, no qual, possua um README.md descrevendo o passo a passo para que possamos executar o projeto na nossa máquina local, além de descrever as dependências necessárias para a execução do projeto.