

Teste Prático: Programador NodeJS + Elasticsearch + SQL

Descrição do Desafio

Você deve criar uma aplicação em **Node.js** capaz de realizar as seguintes tarefas:

1. **Web Crawler:** Extraia informações de imóveis de um portal de imóveis (escolha um portal público).
2. **Elasticsearch:** Normalize e grave as informações no Elasticsearch em um índice específico.
3. **SQL:** Utilize um banco PostgreSQL local para armazenar informações sobre os portais e parâmetros de captura.

O objetivo é criar um script que, ao ser executado, realize todo o processo de extração, transformação e carregamento dos dados.

Especificações

1. Web Crawler

- Escolha qualquer portal público de imóveis para capturar os dados.
- Extraia no mínimo **100 imóveis**, com os seguintes campos:
 - **Título/Descrição do imóvel.**
 - **Portal, URL e tipo de negócio.**
 - **Endereço.**
 - **Preço.**
 - **Número de quartos, banheiros e vagas de garagem.**
 - **Área útil.**
- O web crawler deve ser eficiente e evitar sobrecarregar o portal.
- Os dados capturados devem ser compatíveis com os filtros configurados no banco PostgreSQL (detalhados abaixo).

2. Elasticsearch

Grave os dados capturados em um índice chamado **imoveis**, no seguinte formato JSON:

```
{  
  "id": "identificador_unico",  
  "titulo": "Título do Imóvel",  
  "descricao": "Descrição do Imóvel",  
  "portal": "Nome do Portal",  
  "url": "URL do Imóvel",
```

```
"tipoNegocio": "Venda ou Aluguel",

"endereco": "Endereço",
"preco": 123456.78,
"quartos": 3,
"banheiros": 2,
"vagas_garagem": 1,
"area_util": 80.0,
"capturado_em": "2024-11-29 10:00:00",
"atualizado_em": "2024-11-29 10:00:00"
}
```

3. Banco SQL

- Utilize **PostgreSQL** para armazenar e consultar os parâmetros de captura.
 - Tabela **portal**: id, nome, url, observações
 - Tabela **captura**: id, id do portal, filtros (pode ser texto ou jsonb), status, data e hora do início da última captura, data e hora do fim da última captura
 - Antes de executar o web crawler, o script deve:
 1. Buscar no banco as informações do portal e os parâmetros de captura (se definidos).
 2. Aplicar os filtros informados caso tenha sido configurado algum.
-

Execução

1. **Configurar o ambiente:**
 - Fornecer as instruções para criar as tabelas SQL (os comandos devem estar no README ou como parte do script).
2. **Executar a captura:**
 - Rodar o comando `node teste-web-crawler.js` para iniciar o processo.
3. **Validar os resultados:**
 - Inserir os dados no Elasticsearch.

- Atualizar no banco PostgreSQL as colunas **status** (rodando, erro ou concluído), **data_hora_inicio_ultima_captura** e **data_hora_fim_ultima_captura** da tabela **captura**.
-

Entrega

O candidato deve fornecer:

- O código-fonte do projeto, com as instruções para:
 1. Criar as tabelas no banco PostgreSQL.
 2. Configurar o Elasticsearch (somente o índice **imoveis**) - apenas o nome do índice basta para executar o código.
 3. Rodar o script de captura.
 - Um arquivo README detalhado com os passos para executar o teste na máquina avaliadora.
-

Critérios de Avaliação que serão avaliados

1. **Funcionalidade:** O script executa todas as etapas corretamente?
 2. **Código:** O código é legível, bem estruturado e segue boas práticas?
 3. **Eficiência:** O web crawler respeita limites e evita sobrecarregar o site?
 4. **Elasticsearch:** Os dados são gravados corretamente no índice.
 5. **Banco de Dados:** A integração com o PostgreSQL funciona?
 6. **Instruções:** As instruções fornecidas são claras e permitem executar o teste sem dificuldades?
-

Exemplo de Comando para Rodar

```
node teste-web-crawler.js
```