

# Documentação do Desafio Técnico para Time de Low Code

## Objetivo

Avaliar a capacidade dos candidatos de resolver problemas lógicos, manipular dados com SQL e desenvolver soluções práticas que integrem requisições web e persistência em banco de dados. O desafio busca identificar competências em:

- **Lógica de Programação:** Resolução de problemas por meio de testes online.
- **Banco de Dados (SQL):** Consultas e manipulação de dados utilizando o ambiente disponibilizado.
- **Integração de Sistemas:** Criação de um algoritmo prático que consuma uma API.
- **Documentação e Planejamento:** Clareza na apresentação da solução e racionalidade na escolha de ferramentas e abordagens.

## Desafios

Serão duas etapas de desafio:

- **Desafio 1:** Primeiro será realizar a avaliação da criação de scripts utilizando javascript para consumir e processar esses dados.
- **Desafio 2:** Manipulação de banco de dados para cadastro de dados.

### Desafio 1

O primeiro desafio consiste na manipulação e utilização de APIs com javascript.

#### Acesso a API

Para a realização da primeira parte, deve-se utilizar a API do TMDb, para isso é necessário criar uma conta no [Getting Started](#) e criar uma chave de API.

## Script 1

**Objetivo:** Realizar chamadas à API para obter filmes populares e seus trailers.

- **Chamada à API:**

- Utilize a API <https://api.themoviedb.org/3/movie/popular> para listar os filmes mais populares.
- Query Params obrigatórios:
  - `api_key`: "SUA CHAVE API".

- **Extração de Dados:**

- Para cada filme, exibir:
  - `id`
  - `título`
  - `data de lançamento`
  - `descrição`

- **Tratamento de Erros:**

- Verificar se a resposta da API retornou status `200`.
- Exibir mensagem de erro caso a requisição falhe.

- **Obtenção de Trailers:**

- Identificar os **5 filmes mais recentes**.
- Para cada um, buscar o trailer usando a API <https://api.themoviedb.org/3/movie/{idFilme}/videos>.
- Retornar a chave (`key`) do trailer.
- Formatar a URL do YouTube: <https://www.youtube.com/watch?v={key}>.

- **Boas Práticas:**

- Adicionar comentários explicativos no código.
- Garantir a clareza e legibilidade dos dados extraídos.

---

## Script 2

**Objetivo:** Listar filmes de gêneros específicos.

### 1. Obtenção de Gêneros:

- Utilize a API <https://api.themoviedb.org/3/genre/movie/list> para listar os gêneros.
- Filtrar apenas os gêneros cujo nome comece com a letra "A".
- Extraia o `genre_id` correspondente.

### 2. Listagem de Filmes por Gênero:

- Utilizar a API <https://api.themoviedb.org/3/discover/movie>.
- Query Params:
  - `api_key`: "SUA CHAVE API".
  - `with_genres`: [genre\_id] (substituir pelo ID filtrado anteriormente).

### 3. Extração de Dados:

- Para cada filme, exibir:
  - `id`
  - `título`
  - `url` do pôster (concatene com <https://image.tmdb.org/t/p/w500>).

### 4. Tratamento de Erros:

- Verificar status `200` e exibir mensagens de erro apropriadas.

### 5. Boas Práticas:

- Adicionar comentários explicativos no código.
- Garantir clareza na apresentação dos dados.

---

## Script 3

**Objetivo:** Trabalhar com filmes em alta e contagem de votos.

### 1. Obtenção de Filmes em Alta:

- Utilize a API [https://api.themoviedb.org/3/movie/now\\_playing](https://api.themoviedb.org/3/movie/now_playing).
- Query Params:
  - `api_key`: "SUA CHAVE API".

### 2. Extração de Dados:

- Para cada filme, exibir:
  - `id`
  - `título`
  - `data de lançamento`
  - `média de votos`
  - `descrição`

### 3. Busca pela Contagem de Votos:

- Escolher um filme da lista anterior.
- Fazer uma chamada à API <https://api.themoviedb.org/3/movie/{idFilme}>.
- Retornar **apenas** a contagem de votos do filme.

### 4. Tratamento de Erros:

- Verificar status `200` e exibir mensagens de erro apropriadas.

### 5. Desafio Extra:

- Implementar um **menu interativo** que permita ao usuário escolher um filme pelo título e obter a contagem de votos.

## 6. Boas Práticas:

- Adicionar comentários explicativos no código.
- Utilizar formatação adequada para melhor legibilidade dos dados.

## Desafio 2

Você foi designado para gerenciar e atualizar os dados da plataforma de solicitação de empréstimos do banco Tech4Bank. O objetivo é realizar diversas operações de inserção, atualização e exclusão de dados, utilizando a estrutura fornecida, garantindo a integridade e a correta associação entre as informações.

### Descrição da Atividade

Execute as seguintes tarefas de manipulação de dados, refletindo cenários comuns de administração e evolução de um sistema de cadastros. Utilize as operações SQL **INSERT**, **UPDATE** e **DELETE** conforme necessário.

### Lista de Tarefas:

1. **Inserir Novo Cliente:** Cadastre um novo cliente chamado "Tech4Bank" no sistema (tabela **CLIENT**).
2. **Adicionar Serviço ao Cliente:** Associe um novo serviço chamado "Consultoria Financeira Personalizada" à cliente "Tech4Bank" (tabela **CLIENT\_SERVICE**).
3. **Criar Novo Formulário:** Crie um novo formulário (**FORMS**) com o nome "Declaração de Bens" e descrição "Formulário para declaração de patrimônio do solicitante".
4. **Adicionar Campos ao Formulário:**
  - Ao formulário "Declaração de Bens", adicione os seguintes campos (**FORM\_FIELD**):
    - **label:** "Descrição do Bem", **type:** "text", **required:** true

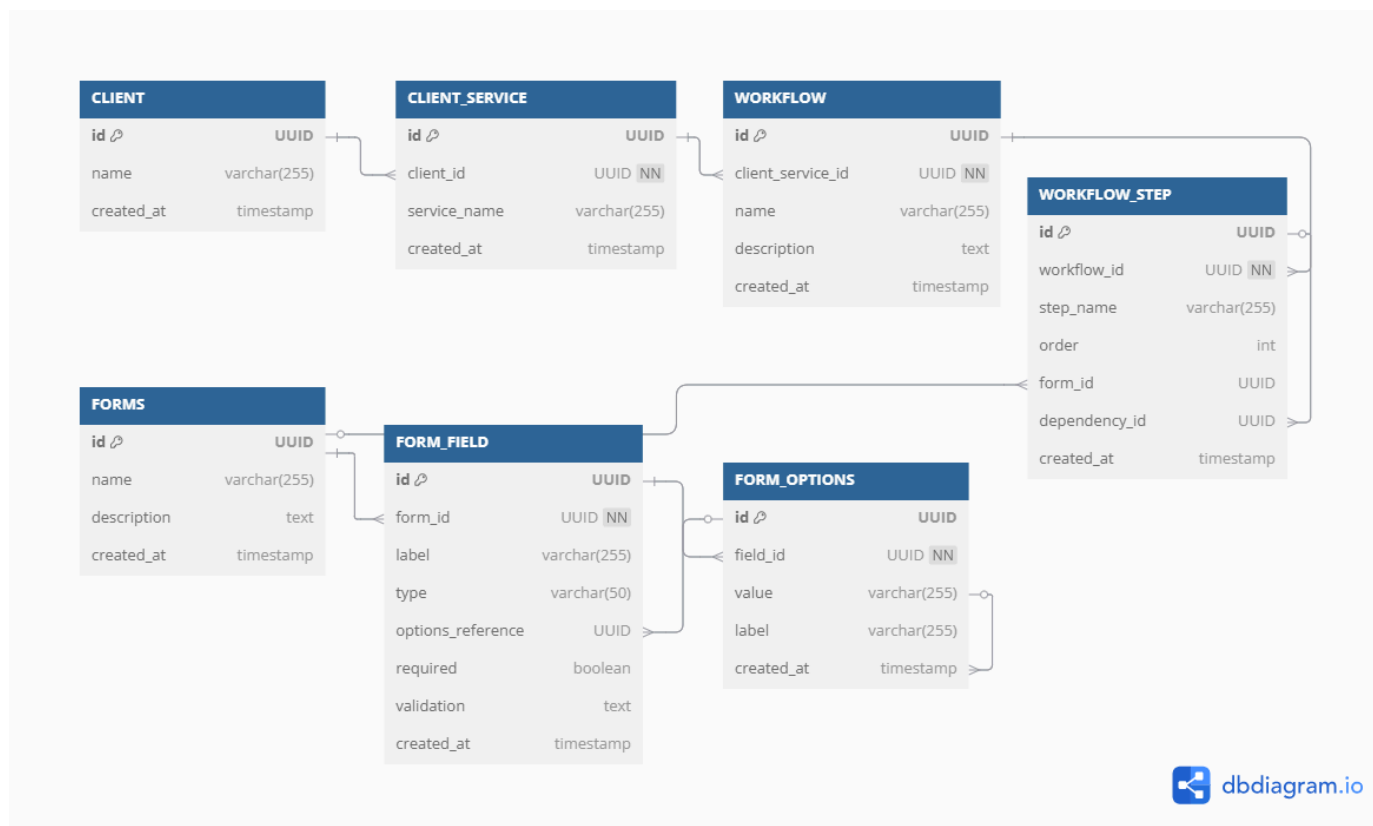
- **label:** "Valor Estimado do Bem", **type:** "number", **required:** true
  - **label:** "Possui Alienação?", **type:** "boolean", **required:** false
5. **Atualizar Dados do Cliente:** O cliente com `_id` (um ID fictício que você pode definir ou buscar, ex: 15) mudou de nome para "Tech4BankEditado". Atualize o registro na tabela `CLIENT`.
  6. **Modificar Campo de Formulário:** No formulário "Dados Pessoais", altere o campo "CPF" (**label:** "CPF") para que sua validação (**validation**) agora exija um formato específico (ex: **regex** para `###.###.###-##`). Se a coluna **validation** não existir ou precisar ser adaptada, considere como faria isso.
  7. **Atualizar Ordem de Passo em Workflow:** Em um workflow existente (ex: workflow com `_id` 20), altere a ordem (**order**) do `WORKFLOW_STEP` "Formulário de Dados de Endereço" de 2 para 3, e ajuste o passo que anteriormente era 3 para 2.
  8. **Remover Opção de Campo:** De um campo do tipo **select** ou **radio** (ex: campo "Estado Civil" no formulário "Dados Pessoais"), remova uma opção específica (ex: "Viúvo(a)") da tabela `FORM_OPTIONS`.
  9. **Excluir um Serviço de Cliente:** Remova o serviço "Seguro Residencial" de um cliente específico (identificado pelo `client_id` e `service_name` na tabela `CLIENT_SERVICE`).
  10. **Desativar/Excluir Workflow:** Um workflow chamado "Análise de Crédito Simplificada" tornou-se obsoleto.
    - Primeiro, delete todos os seus `WORKFLOW_STEP` associados.
    - Em seguida, delete o próprio `WORKFLOW`.

### Requisitos Técnicos e de Documentação:

- Utilize as tabelas `CLIENT`, `CLIENT_SERVICE`, `WORKFLOW`, `WORKFLOW_STEP`, `FORMS`, `FORM_FIELD` e `FORM_OPTIONS` conforme a estrutura fornecida.

- Garanta que todas as operações preservem a integridade referencial do banco de dados. Por exemplo, ao deletar um **FORM\_FIELD**, pense nas **FORM\_OPTIONS** associadas (se houver).
- Documente os comandos SQL (ou pseudocódigo da lógica de programação) utilizados para cada uma das 10 tarefas.
- Para operações de **UPDATE** e **DELETE**, certifique-se de que está utilizando cláusulas **WHERE** apropriadas para evitar modificações ou exclusões indesejadas em massa.

## MER



## Acesso ao banco de dados

Nome: **techlab**

Senha: **techlab\_55afb0c9**

Host: **webapp-cand-test.postgres.database.azure.com**

Porta: **5432**

Banco de Dados: **techlab**

## Requisitos do Desafio

- **Integração de API:** O candidato deve utilizar bibliotecas/recursos que possibilitem a requisição HTTP e o consumo de uma API REST.
- **Tratamento de Erros:** Implementação de lógica para capturar e tratar exceções, tanto na requisição web quanto na comunicação com o banco de dados.
- **Documentação do Código:** Comentários e uma breve explicação sobre as decisões tomadas e a estrutura da solução.
- **Utilização de Ferramentas AI:** Se o candidato optar por empregar ferramentas de inteligência artificial (como ChatGPT ou Bard) para auxiliar na elaboração da solução ou na criação dos testes, deverá documentar como e onde essas ferramentas foram aplicadas, destacando os benefícios e eventuais limitações encontradas.

Okay, com base na descrição detalhada dos desafios e nos objetivos do processo seletivo, aqui está uma versão atualizada e mais alinhada dos Critérios de Avaliação:

## Critérios de Avaliação

A avaliação será baseada na performance do candidato nos dois desafios propostos, na qualidade da documentação e nas boas práticas gerais de desenvolvimento, com a possibilidade de uma etapa opcional de apresentação.

### 1. Desafio 1: Integração API com Javascript (Peso: 40%)

- **Funcionalidade e Correção:**
  - Sucesso na realização das chamadas às APIs especificadas (popular, videos, genres, discover, now\_playing).
  - Extração correta e completa dos dados solicitados para cada nível (id, título, data, descrição, chave do trailer, URL do pôster, média de votos, contagem de votos).
  - Implementação correta da lógica de filtragem (5 filmes mais recentes, gêneros começando com "A").
  - Formatação adequada de URLs (YouTube, pôsteres).



- Funcionamento do menu interativo (Desafio Extra), se implementado.
- **Qualidade do Código Javascript:**
  - Clareza, legibilidade e organização do código.
  - Uso adequado de comentários explicativos para lógica complexa ou decisões importantes.
  - Boas práticas de Javascript (nomenclatura, estrutura).
- **Tratamento de Erros:**
  - Verificação consistente do status da resposta da API (status 200).
  - Apresentação de mensagens de erro claras e informativas em caso de falha na requisição.

## **2. Desafio 2: Manipulação de Banco de Dados SQL (Peso: 30%)**

- **Correção das Operações SQL:**
  - Execução bem-sucedida de todas as 10 tarefas de manipulação de dados (INSERT, UPDATE, DELETE).
  - Sintaxe SQL correta e apropriada para cada operação.
- **Precisão e Integridade dos Dados:**
  - Uso correto de cláusulas **WHERE** para garantir que apenas os registros desejados sejam afetados (evitando modificações/exclusões em massa acidentais).
  - Preservação da integridade referencial do banco de dados (considerando chaves estrangeiras e dependências ao atualizar ou excluir dados, como nos passos de workflow ou opções de formulário).
- **Aderência aos Requisitos:**
  - Realização das operações conforme especificado em cada uma das 10 tarefas listadas.

## **3. Documentação, Planejamento e Boas Práticas Gerais (Peso: 20%)**

- **Documentação Explicativa (Arquivo MD/PDF):**
  - Clareza na descrição da compreensão dos requisitos e do planejamento inicial.

- Explicação da abordagem utilizada para cada desafio (Desafio 1 e Desafio 2).
- Descrição do fluxo de execução da solução (especialmente para o Desafio 1).
- Detalhamento das estratégias de tratamento de erros implementadas.
- Documentação clara dos comandos SQL utilizados no Desafio 2 (se não comentados diretamente no script).
- **Documentação do Uso de Ferramentas de IA (quando aplicável):**
  - Descrição transparente de como e onde ferramentas de IA foram utilizadas.
  - Avaliação dos benefícios e limitações percebidos no uso dessas ferramentas.
- **Qualidade Geral:**
  - Organização do repositório de código (se aplicável).
  - Clareza geral e profissionalismo da entrega.

#### 4. Apresentação da Solução (Peso: 10%)

- **Clareza na Comunicação:** Capacidade de explicar a solução de forma lógica, concisa e tecnicamente precisa.
- **Domínio Técnico:** Demonstração de compreensão dos conceitos envolvidos (APIs REST, SQL, manipulação de dados, tratamento de erros).
- **Justificativa das Decisões:** Habilidade para justificar as escolhas de implementação, ferramentas e abordagens.
- **Interação e Resposta a Perguntas:** Capacidade de responder a questionamentos de forma eficaz.

#### Processo de Entrega

- **Submissão:**
  - Repositório Git contendo o código Javascript do Desafio 1.
  - Script(s) SQL ou documento contendo os comandos SQL utilizados para o Desafio 2.
  - Documento explicativo (Markdown ou PDF) conforme detalhado no critério 3.