

## Teste de aptidão em análise e desenvolvimento em Go

**Data Source (DS):** <https://ed-public-download.app.cloud.gov/downloads/Most-Recent-Cohorts-Scorecard-Elements.csv>

Utilizando os dados contidos no **DS**, desenvolva uma aplicação utilizando a linguagem Go ([www.golang.org](http://www.golang.org)) que atende os seguintes requisitos:

1. Ser possível consultar registros utilizando os campos **UNITID**, **INSTNM**, **CITY** ou **STABBR**;
2. O resultado deve ser ordenado de forma crescente em relação a **STABBR**, **CITY**, **INSTNM**;
3. O resultado deve conter, no máximo, 100 registros.

Critérios de pontuação (cada item pode ser parcialmente pontuado dependendo da implementação):

- O **DS** pode ser armazenado das seguintes formas:
  - a. Fixo no código (hard-coded): 0 pontos;
  - b. Lido do arquivo CSV durante a inicialização da aplicação: 300 pontos;
  - c. Em base de dados relacional (apenas uma das opções):
    - i. SQLite: 200 pontos;
    - ii. PostgreSQL: 400 pontos.
- Apenas os 6 primeiros campos possuem mapeamento obrigatório (0 pontos), contudo cada campo adicional mapeado corretamente aumenta a pontuação em 2;
- A aplicação pode ser desenvolvida nos seguintes modelos:
  - a. Standalone: o programa inicia, executa a busca localmente e encerra. Os parâmetros de consulta são definidos por:
    - i. Fixos no código: 0 pontos;
    - ii. Linha de comando e/ou arquivo de configuração: 100 pontos.
  - b. Client-server: o processo servidor está em execução. O processo cliente inicia, executa a consulta de busca no servidor e encerra. O processo servidor continua em execução. A pontuação de parâmetros de consulta utilizados pelo cliente segue a mesma regra do item anterior. A comunicação deve ser feita por um dos seguintes protocolos:
    - i. HTTP: 200 pontos;
    - ii. GRPC: 600 pontos. Observe que neste caso a curva de aprendizado é **substancialmente maior** para quem não possui experiência com a linguagem e tecnologias envolvidas.
- Documentação no código: 100 pontos;
- Teste unitário da função (ou funções) de consulta aos registros: 200 pontos;
- Celeridade: cada candidato inicia com 300 pontos neste quesito, que é diminuído de 100 pontos a cada 24 horas do prazo de início do teste. Dessa forma a pontuação recebida será:
  - a. 300 pontos se entregue até 24 horas;
  - b. 200 pontos se entregue entre 24 e 48 horas;

- c. 100 pontos se entregue entre 48 e 72 horas;
- d. 0 pontos após isso.
- Não há limite na quantidade de versões enviadas, pois é desejável que o candidato tenha algo “funcionando” antes e garanta uma pontuação maior por celeridade. Ao final será considerada sua versão com maior pontuação. No caso de utilizar repositório GIT para publicação separar as versões em branches de forma que no master esteja a versão mais recente.

Notas:

- No caso de armazenamento em base de dados deve ser fornecido também o código de criação de suas estruturas e importação de registros;
- No caso de aplicação no modelo “cliente-servidor” não é necessário que ela atenda ambos os protocolos HTTP e GRPC simultaneamente, contudo se for implementado as pontuações serão somadas desde que o servidor seja executado a partir de um único binário;
- Itens que não fazem parte dos critérios acima podem ser pontuados, no entanto serão relevantes apenas em situações de desempate;
- Deve ser entregue todo o código-fonte utilizado pela aplicação, incluindo dependências externas (vendor), em formato ZIP ou em repositório GIT publicamente acessível (e.g. github.com).

Links úteis:

- <https://tour.golang.org/>
- <https://marketplace.visualstudio.com/items?itemName=lukehoban.Go>
- <https://golang.org/doc/articles/wiki/>
- <https://grpc.io/>
- <http://go-database-sql.org/>