

## **Globo.com: coding challenge**

---

### **Considerações Gerais**

Registre tudo: testes que forem executados, idéias que gostaria de implementar se tivesse tempo (explique como você as resolveria, se houvesse tempo), decisões que forem tomadas e seus porquês, arquiteturas que forem testadas, os motivos de terem sido modificadas ou abandonadas, instruções de deploy e instalação, etc. Crie um único arquivo COMMENTS.md ou HISTORY.md no repositório para isso.

---

### **O Problema**

O problema que você deve resolver é o problema da votação do paredão do BBB usando a linguagem de programação <https://go.dev/> e outras ferramentas open-source da sua preferência.

O paredão do BBB consiste em uma votação que confronta dois ou mais integrantes do programa BBB, simulando o que acontece na realidade durante uma temporada do BBB. A votação é apresentada em uma interface acessível pela WEB onde os usuários optam por votar em uma das opções apresentadas. Eles não precisam estar logados para conseguirem participar. Uma vez realizado o voto, o usuário recebe uma tela com o comprovante do sucesso e um panorama percentual dos votos por candidato até aquele momento.

---

### **Regras de negócio**

Os usuários podem votar quantas vezes quiserem, independente da opção escolhida. Entretanto, a produção do programa não quer receber votos oriundos de uma máquina, apenas votos de pessoas.

A votação é chamada na TV em horário nobre, com isso, é esperado um enorme volume de votos concentrados em um curto espaço de tempo. Esperamos ter um teste disso, e por razões práticas, podemos considerar 1000 votos/seg como baseline de performance deste teste.

A produção do programa gostaria de consultar as seguintes informações: o total geral de votos, o total por participante e o total de votos por hora de cada paredão.

---

### **O que será avaliado na sua solução?**

Seu código será observado por uma equipe de desenvolvedores que avaliarão a implementação do código, simplicidade e clareza da solução, a arquitetura, estilo de código, testes unitários, testes funcionais, nível de automação dos testes e documentação.

A automação da infra-estrutura também é importante. Imagine que você precisará fazer deploy do seu código em múltiplos servidores, então não é interessante ter que ficar entrando máquina por máquina para fazer o deploy da aplicação.

---

### **Dicas**

Use ferramentas e bibliotecas open-source, mas documente as decisões e porquês; Automatize o máximo possível; Em caso de dúvidas, pergunte.