

## Desafio Técnico: Comunicação com WebSockets

### Visão Geral

Tradicionalmente, a comunicação via HTTP é **sem estado** e **sem conexão persistente**: cada requisição e resposta é independente, e o servidor pode tratar chamadas em instâncias distintas sem manter qualquer tipo de conexão contínua com o cliente. Com isso, só é possível uma comunicação simples (o cliente inicia a comunicação e o servidor apenas responde).

**WebSockets** mudam esse paradigma, permitindo uma comunicação **bidirecional e persistente**. Após uma conexão inicial via HTTP, o cliente e o servidor podem enviar mensagens entre si livremente, mantendo uma única conexão TCP ativa. Isso torna o servidor **com estado**, pois mantém a referência da conexão de cada cliente conectado.

Durante o processo de handshake, o cliente envia uma requisição HTTP com cabeçalhos específicos (veja a documentação do protocolo WebSocket). O servidor, ao aceitar a conexão, responde com o código **101 Switching Protocols**, mantendo o canal de comunicação aberto para trocas futuras de mensagens.

---

### O Que Você Deve Fazer

Implemente uma solução básica que utilize WebSockets, contendo:

#### Backend (Servidor)

- Deve expor **pelo menos um endpoint HTTP** que aceite a criação de uma conexão WebSocket.
- Deve manter um **pool de conexões WebSocket ativas**.
- Cada conexão deve ser monitorada tanto para **mensagens recebidas** quanto para **envio de mensagens**.
- A abordagem mais simples (e aceita para este desafio) é o **broadcast**: cada mensagem recebida de um cliente deve ser enviada para todos os outros clientes conectados.

#### Frontend (Cliente)

- Deve ser capaz de **estabelecer uma conexão WebSocket com o servidor** via HTTP.
- Deve haver uma forma de **enviar mensagens para o servidor**.

- Deve haver uma forma de **observar mensagens recebidas** e exibi-las para o usuário.
- 

## Requisitos de Qualidade

- O código deve ser **limpo, bem organizado e de fácil leitura**.
  - Caso ache necessário, **adicione comentários explicativos** no código para esclarecer trechos importantes ou decisões técnicas.
  - Utilize boas práticas de programação.
  - Utilize controle de versão (ex: Git).
- 

## Entrega

- Submeta o projeto em um repositório público (como GitHub ou GitLab).
  - Testes automatizados poderão ser desenvolvidos mas não serão contados como requisitos obrigatórios.
  - Inclua um arquivo **README.md** com:
    - Instruções claras de **instalação e execução** do projeto.
    - Tecnologias utilizadas.
    - Comentários e explicações que achar pertinentes.
- 

## Sugestão de Tecnologias

Você pode usar a linguagem de sua preferência. Por exemplo, com **Python**. Você poderá usar bibliotecas que apenas estão no pacote da linguagem e versão utilizada ou bibliotecas externas que apenas façam a abstração da criação do servidor com a web. Todo o resto do código deve ser produzido pelo candidato.

---

Boa sorte! Estamos avaliando sua capacidade de desenvolver uma solução funcional e com boa qualidade de código, e não apenas um sistema completo ou com interface refinada.

