

Para começar a escrever sobre este exercício, diria que foi uma experiência bastante difícil. Provavelmente por não ter estado presente na aula, por motivos de saúde, e por não ter experiência com este tipo de ferramenta, senti que foi complicado perceber o que era necessário fazer. Uma das maiores dúvidas foi perceber se era mesmo obrigatório colocar o vídeo chat na minha webpage. Mesmo assim, experimentei duas abordagens diferentes. A primeira tentativa, que acabei por colocar na minha webpage, não utilizava o Node.js (sem ser dependente de servidores). Segui tutoriais para conseguir implementar o acesso à câmara e o vídeo local.

Porém percebi que necessitava do WebRTC com um sistema de sinalização, através de WebSocket para trocar as chamadas de oferta (offer), resposta (answer) e os candidatos ICE. O WebRTC exige uma sincronização muito cuidadosa entre os separadores, o que se tornou numa dificuldade. Por isso é que o código que fiz, separado (WebRTC-projeto- feito por mim) do que está inserido na minha webpage não consegue executar as duas câmaras em simultâneo.

Com a Inteligência Artificial consegui fazer com que funcionasse da forma correta e como queria, com um uso de WebSocket simples. No entanto, a IA não explicou em detalhe como funcionava a parte da sinalização no WebRTC.

Ao integrar o código para o chat de vídeo (mini-chat.html) na minha webpage, verifiquei que a câmara não é ativada, mesmo quando a câmara não está a ser utilizada pelos filtros. Contudo, o código funciona corretamente quando executado de forma independente.

Utilizei dois hosts diferentes para organizar o projeto:

- WebRTC-projeto-feito por mim: **8080**
- WebRTC-projeto-eu+ia: **9090**

What new insights did you gain by comparing your implementation with that of the AI?

- Ao comparar o meu código com o da IA, percebi melhor como funciona o processo de ligação do WebRTC, especialmente a parte da sinalização entre os separadores (offer, answer e ICE). A minha versão não funcionava direito e era bem mais simples, ainda não percebia tão bem como o Node.js e o WebSocket funcionam como ponte entre os utilizadores.

What were the biggest difficulties?

- A maior dificuldade foi conseguir que as duas câmaras aparecessem ao mesmo tempo e entender a sincronização necessária para que a comunicação entre os separadores funcionasse. Além disso, o Node.js e HTTPS, por vezes complicaram o teste local.

Where did the AI help and where did it hinder understanding?

- A IA ajudou-me bastante neste exercício, comparado com os outros. Percebi, depois, a estrutura do WebRTC e a importância da troca de sinalização. Gerou códigos avançados (dependentes de servidores).

Bárbara Ferreira

UP202401520