

Nome: João Vítor Santana Depollo  
Matrícula: **2021039751**

Item	Status	Solução
Abertura conexão em IPV4 e IPV6	Concluído	Ao receber a requisição de abertura, gera uma porta aleatória para o outro peer dar um listen e o fechamento limpa os dados do outro peer na memória
Recebimento de argumentos e interpretação dos retornos de sucesso e erros	Concluído	Foi criado um arquivo "constants.h" com todos os códigos. Dessa forma, o common utiliza estes códigos para interpretar se o payload é de sucesso ou erro e também interpretar os resultados
Ações de persistência do Usuário	Em andamento	Array que assina o usuário para novo para uma nova posição do array
Ações de localização	Em andamento	
Fechamento de conexão cliente	Não Iniciado	

## Facilidades

- Após a implementação base do p2p, adicionar o ipv4 e ipv6 de forma simultânea foi possível de forma tranquila, bastava utilizar o *setsockopt*, desabilitando, o *ipv6\_only*.
- A Gerencia de usuários está sendo feita toda in memory, com um array e um struct de usuários, o que foi trivial de ser implementado.

## Dificuldades

- Problemas para entender exatamente como a comunicação p2p deveria funcionar. Na minha solução atual, a porta está sendo determinada com a função *rand()*. Pretendo melhorar este ponto após tirar algumas dúvidas na monitoria.
- *Seg Faults*: Durante o trabalho, recebi muito erro de seg faults por não estar alocando devidamente o array do buffer em que eu estava recebendo e enviando as requisições.
- Utilizar a função *select()* foi desafiador, foi necessário buscar na internet alguma implementação prática dela

## Próximos passos:

- Implementar o servidor de localização e demais funções de usuários

- Melhorar a comunicação p2p