



# Apresentação do projeto final

Computação Distribuída  
2020/2021

João Reis, 98474  
Artur Romão, 98470  
P2



# Arquitetura Peer-to-peer

O nosso sistema assenta numa arquitetura P2P.

Não é centralizado, uma vez que não existe um servidor e o objetivo é todos os slaves comunicarem entre si de maneira a distribuir o “trabalho” por todos e, assim que um deles descubra a password, envia uma mensagem para os restantes pararem de procurar.



# Protocolo

Elaborámos o nosso protocolo no ficheiro `protocol.py` com todos os métodos necessário às mensagens:

- **Método Join:** Mensagem enviada por multicast pelos slaves para informarem que se conectaram.
- **Método Distribute Work:** Mensagem enviada por multicast para distribuir o trabalho pelos vários slaves
- **Método Found:** Mensagem para notificar todos os slaves que a password foi encontrada

```
class Protocol:
    """Computação Distribuida Protocol."""

    @classmethod
    def join(cls, addr):
        dic = {"command" : "join"}
        dic["address"] = addr
        return json.dumps(dic).encode("utf-8")

    @classmethod
    def distribute_work(cls, work):
        dic = {"command" : "distribute"}
        dic["work"] = work
        return json.dumps(dic).encode("utf-8")

    @classmethod
    def found(cls, work, password):
        dic = {"command" : "found"}
        dic["work"] = work
        dic["password"] = password
        return json.dumps(dic).encode("utf-8")
```



## Resultados Obtidos

- Cada slave consegue descobrir uma password, porém não em conjunto, síncronos e coordenados
- Uma password com tamanho maior que 2 tem um tempo de descoberta exponencial
- Vários slaves conhecem-se por multicast, através das mensagens Join
- Lista de todos os slaves conectados



## O que falta implementar...

- Organização e distribuição de “trabalho” pelos slaves conectados
  - O primeiro a chegar fica encarregue das passwords com tamanho 1, o segundo de tamanho 2, e por aí em diante (quando o primeiro termina as possibilidades de tamanho 1, passa para as de tamanho 4, caso exista)
- Tolerância a falhas