

# Computação Distribuída- Projeto Final

Filipe André Seabra Gonçalves - 98083  
João Pedro Saraiva Borges - 98155





# Algoritmo

Dada a password inicial, geramos a password seguinte;

Dada a password da combinação  $i$ , gera a password da combinação exatamente a seguir,  $i + 1$

```
import string

def getNext(charlist, increment, ID, current):
    count = 0
    nxt = ""
    if(current == ""):
        return charlist[increment-1+ID]
    else:
        for i in range(len(current)-1, -1, -1):
            ind = charlist.index(current[i])
            if count == 0:
                if ind+increment >= len(charlist):
                    nxt = charlist[(ind+increment)%len(charlist)] + nxt
                    increment = 1
                else:
                    count = 1
                    nxt = charlist[ind+increment] + nxt
                    increment = 1
            else:
                nxt = current[i] + nxt
        if count == 0:
            nxt = charlist[0] + nxt
        return nxt
```



# Protocolo

Connection- Conecta-se com o multicast, identificando-se com o id que lhe pertence e o seu ip.

Coordinator-Define-se como coordenador das slaves.

Scanning Troca de mensagens entre slaves para decidir qual das slaves será o coordenador, sendo essa a que tiver maior id (bullying).

Fail- Indica que a palavra passe falhou

Success-Indica qual password resultou

Working- indica a palavra passe a ser testada

PROTOCOL				
CONNECTION	id	ip/port		
COORDENATOR	id	ip/port		
SCANNING	id	ip/port		
FAIL	id	pwd	idx	
SUCCESS	id	pwd		
WORKING	id	pwd	count	



## Método

- Eleição por flooding + bullying
- Distribuição de trabalho pelos restantes slaves
- Geração da password através do algoritmo



# Resultados

- Size 1: 17.07 segundos
- Size 2: 520.61 segundos