

# PROJETO FINAL DE CD

*Distributed Photo Organizer*

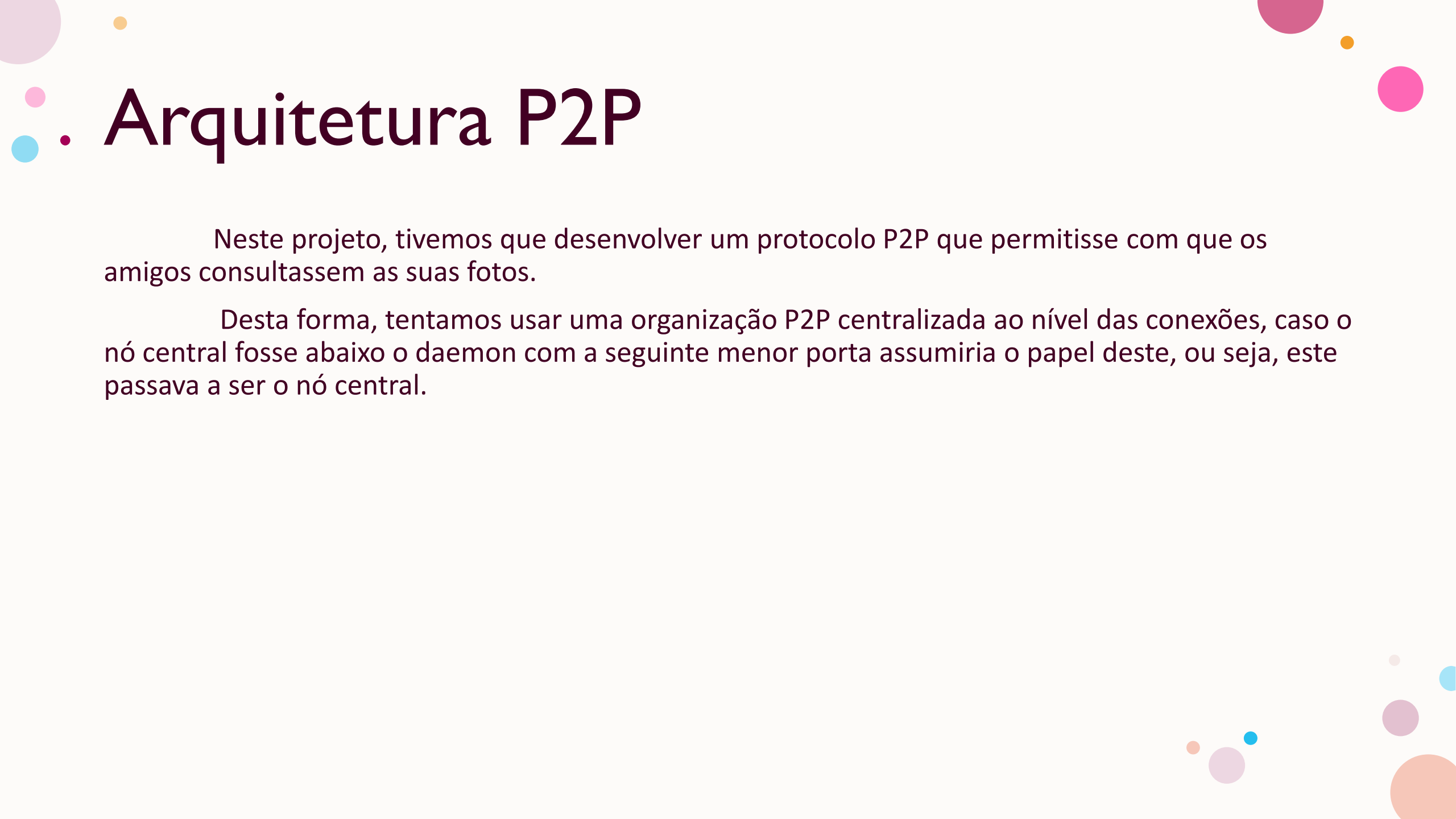


universidade  
de aveiro

2021/2022

Catarina Costa – 103696

Diogo Magalhães - 102470



# Arquitetura P2P

Neste projeto, tivemos que desenvolver um protocolo P2P que permitisse com que os amigos consultassem as suas fotos.

Desta forma, tentamos usar uma organização P2P centralizada ao nível das conexões, caso o nó central fosse abaixo o daemon com a seguinte menor porta assumiria o papel deste, ou seja, este passava a ser o nó central.

# Protocolo

Neste trabalho usamos o protocolo TCP.

Aqui conectamos com o 5000, porta central, e de seguida conectamo-nos com as restantes existentes.

```
def __init__(self, path, timeout=3):
    self.addr = "localhost" #Em principio vai ser o localhost
    self.port = 5000 #A porta inicial será a porta 5000
    self.canceled = False
    self.path = str(os.getcwd()) + ("/") + str(path)
    self.clients = []
    self.con_send = {}
    self.con_rcv = {}
    self.porta_central = 5000
    self.stop_thread = False

    self.all_images = {} #   hashCode: (size, port)

    self.s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    self.s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

    try:
        self.s.bind((self.addr, self.porta_central))
        print("Sou o 5000")
    except:
        porta = 5001
        while True:
            try:
                self.s.bind((self.addr, porta))
                self.port = porta
                print("Sou o " + str(porta))
                break
            except:
                porta += 1
```

```
if(self.port != self.porta_central):
    print("Connecting to 5000")
    j = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    j.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    j.connect((self.addr, self.porta_central))
    self.con_send[self.porta_central] = j
    message = pickle.dumps({"command": "daemon_join", "port": self.port})
    size = len(message).to_bytes(2, "big")
    self.con_send[self.porta_central].send(size + message)

    self.thread = threading.Thread(target=tempo)
    self.thread.start()
```



# Resultados

Conseguimos assim estabelecer uma rede P2P entre os *daemons*.

Ligar cada *client* ao seu *daemon*.

Tentamos fazer a partilha de imagens pela rede.

