



Prática Sockets Processing

Eric Vinícius - Monitor Disciplina de Redes de
Computadores



O que veremos hoje?

- Relembrar conceitos aprendidos na aula passada sobre Sockets
- Explorar esse assunto utilizando agora o Processing
- **Prática** Cliente - Servidor utilizando TCP (Envio pelo sigaa)
- Explorar algumas situações específicas.
- Biblioteca .Network - Processing



Network Lib

- A biblioteca de rede possibilita ler e gravar dados em máquinas na Internet.
- Permite a criação de clientes e servidores utilizando TCP.

<https://processing.org/reference/libraries/net/index.html>



Classes

- Cliente
- Servidor



Classe Client

Um cliente se conecta a um servidor e envia dados de um lado para outro. Se algo der errado com a conexão, por exemplo, o host não está presente ou está ouvindo em uma porta diferente, uma exceção é lançada.

```
Client myClient = new Client(this, "127.0.0.1", 5204);
```

IP-Passa como String

Porta - Passa como inteiro

parent - Passa um this(?Mistério)



Classe Server

Um servidor envia e recebe dados para e de seus clientes associados (outros programas conectados a ele).

Quando um servidor é iniciado, ele começa a escutar conexões na porta especificada pelo parâmetro da porta.

```
Server myServer = new Server(this, 5204);
```

Porta - Passa como inteiro

parent - Passa um this(?Mistério)



Atenção!

Os computadores têm muitas portas para a transferência de dados e alguns são normalmente usados, portanto, não selecione um deles.

Por exemplo, os servidores da web geralmente usam a porta 80 e o email POP usa a porta 110.



Atenção!

- Não nomeiem seus arquivos com a nomenclatura Server e Client
 - Coloquem qualquer outro nome:
 - Server1,ServerBolado,MeuServer...
 - Client1,ClientTopzera,MeuCliente...

Esses nomes(Client e Server) já estão destinados à Biblioteca



Funções_Importantes()

- Classe Cliente
 - stop() Desconecta do servidor
 - active() Retorna true se este cliente ainda estiver ativo
 - ip() Retorna o endereço IP da máquina como uma string
 - available() Retorna o número de bytes no buffer esperando para ser lido
 - read() Retorna um valor do buffer
 - write() Escreve bytes, chars, ints, bytes[], Strings



Funções_Importantes()

- Classe Server
 - `disconnect()` : se Desconecta de um cliente em especifico passado como parâmetro.
 - `active()`: Retorna true se o server ainda ta ativo(Serve pra testar).
 - `available()`: Returns the next client in line with a new message.
 - `stop()` : Desconecta todos os clients e para o funcionamento do server.
 - `write()`: Escreve dados para todos os clientes conectados



Preparação

- Verifiquem se suas máquinas possuem o Processing
- O que faremos agora?
 - Classe Servidor
 - Classe Cliente
 - Documento com umas questões que vocês enviarão pelo sigaa