

INSTITUTO FEDERAL
SANTA CATARINA



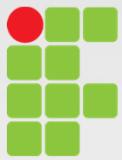
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Programação Concorrente e Distribuída

Professor:

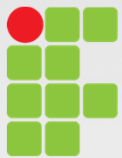
Felipe Schneider Costa

felipe.costa@ifsc.edu.br



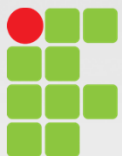
Tópicos

- Apresentação da disciplina
- Introdução a Programação Concorrente e Distribuída
- Atividade



Informações Gerais

- Contato:
 - ✓ Por e-mail: felipe.costa@ifsc.edu.br
 - ✓ Discord:
- Conteúdo da disciplina:
 - ✓ SIGAA
- Encontros presencias na sala:
 - ✓ A07 - Laboratório de informática



INSTITUTO FEDERAL
SANTA CATARINA



Plano de Ensino



Informações Gerais

- **Bases tecnológicas**
 - ✓ Conceitos de sistemas paralelos e concorrência.
 - ✓ Conceitos: processos, threads, interrupções, escalonamento.
 - ✓ Problemas de programação concorrente: deadlock, alocação de recursos, leitura e escrita concorrente, exclusão mútua, consenso.
 - ✓ Memória distribuída.



Informações Gerais

- **Objetivos**

- ✓ Compreender conceitos relacionados à programação concorrente e distribuída.
- ✓ Desenvolver programas de computador para aplicar os conceitos básicos da PCD.
- ✓ Utilizar ferramentas e frameworks para programação concorrente e distribuída.



Informações Gerais

- Avaliações:
 - ✓ Quatro avaliações (AV1, AV2, TG1, TG2)
 - ✓ **Média final** = $(AV1 + AV2 + TG1 + TG2) / 4$
- Como serão as aulas?
 - ✓ Sala de aula, atividades extraclasse e práticas em laboratório.
 - ✓ Metade teoria e metade prática.



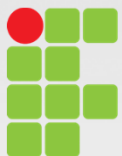
Informações Gerais

- **Recuperação Paralela:**

- ✓ A recuperação de conteúdos acontecerá através da correção coletiva das avaliações, de atividades extraclasse e atendimento ao aluno.
- ✓ Terá direito a recuperação de conceito o aluno que realizar a atividade avaliativa e obtiver nota inferior a 6.
- ✓ O conceito final será calculado através da média aritmética entre a avaliação e a recuperação.

- **Solicitação de 2ª chance para prova**

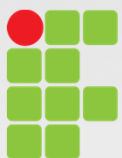
- ✓ Conforme RDP do IFSC (Resolução nº 41, de 20 de novembro de 2014).



Informações Gerais

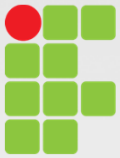
- **Sobre a utilização de meios ilícitos em atividades de avaliação ou recuperação**

- ✓ Não será tolerada a utilização de meios ilícitos (cola, plágio e afins) durante a realização de atividades avaliativas.
- ✓ Caso seja constatado durante ou após a realização da avaliação, esta será tornada sem efeito, desta forma, não haverá direito a recuperação.



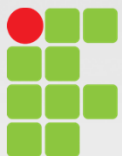
O que vamos trabalhar

Unidades	Temas	CH
Unidade 1	1.1 Programação Paralela 1.2 Arquiteturas 1.3 Processos 1.4 Comunicação e Nomeação	30
Unidade 2	2.1 Sincronização 2.2 Consistência e Replicação 2.3 Tolerância a falhas	20
Unidade 3	3.1 Conceitos básicos Internet das Coisas (IoT) e Histórico 3.2 IoT x M2M 3.3 IoT nos diferentes cenários da matriz de produção 3.3.1 Casas e cidades inteligentes 3.3.2 Indústria 3.3.3 Transportes e logística 3.3.4 Saúde e entretenimento 3.3.5 Agronegócio 3.4 Bases tecnológicas da IoT 3.4.1 Conectividade 3.4.2 Arduino básico, Shields e NodeMCU 3.4.3 Aplicações IoT em Arduino e NodeMCU 3.5 Processamento de Dados IoT 3.6 Sistemas de arquivos distribuídos	30



INSTITUTO FEDERAL
SANTA CATARINA

Introdução a Programação Concorrente e Distribuída (PCD)



INSTITUTO FEDERAL
SANTA CATARINA

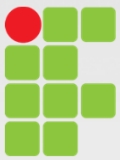
PCD: Conceitos básicos



PCD: Conceitos básicos

Panorama atual

- Poder de processamento das máquinas vem crescendo rapidamente.
- Grande parte das máquinas interligada por redes de computadores.
- Sistemas e aplicações estão cada vez mais complexos:
 - ✓ Funcionalidade, interfaceamento gráfico, comunicação, ...
 - ✓ Maior carga, maior número de usuários, ...
 - ✓ Melhor tempo de resposta, maior confiabilidade ...



PCD: Conceitos básicos

Programação paralela

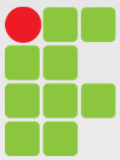
- Consiste em executar simultaneamente várias partes de uma mesma aplicação.
- Aplicações são executadas paralelamente:
 - ✓ Em um mesmo processador.
 - ✓ Em uma máquina multiprocessada.
 - ✓ Em um grupo de máquinas interligadas que se comporta como uma só máquina.



PCD: Conceitos básicos

Programação distribuída

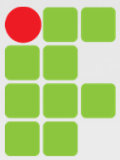
- Consiste em executar aplicações cooperantes em máquinas diferentes.
- Tornou-se possível a partir da popularização das redes de computadores.
- Aplicações são executadas em máquinas diferentes interligadas por uma rede.



PCD: Conceitos básicos

Diferenças entre computação paralela e distribuída

- Acoplamento
 - ✓ Sistemas paralelos são fortemente acoplados:
 - compartilham hardware ou se comunicam através de um barramento de alta velocidade.
 - ✓ Sistemas distribuídos são fracamente acoplados.
- Previsibilidade
 - ✓ O comportamento de sistemas paralelos é mais previsível; já os sistemas distribuídos são mais imprevisíveis devido ao uso da rede e a falhas.



PCD: Conceitos básicos

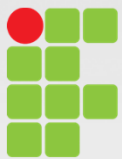
Diferenças entre computação paralela e distribuída

- Influência do Tempo
 - ✓ Sistemas distribuídos são bastante influenciados pelo tempo de comunicação pela rede; em geral não há uma referência de tempo global.
 - ✓ Em sistemas paralelos o tempo de troca de mensagens pode ser desconsiderado.
- Controle
 - ✓ Em geral em sistemas paralelos se tem o controle de todos os recursos computacionais; já os sistemas distribuídos tendem a empregar também recursos de terceiros.

PCD: Conceitos básicos

Vantagens

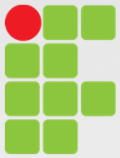
- Usam melhor o poder de processamento
- Apresentam um melhor desempenho
- Permitem compartilhar dados e recursos
- Permitem reutilizar serviços já disponíveis
- Atendem um maior número de usuários



PCD: Conceitos básicos

Dificuldades

- Desenvolver, gerenciar e manter o sistema.
- Controlar o acesso concorrente a dados e a recursos compartilhados.
- Evitar que falhas de máquinas ou da rede comprometam o funcionamento do sistema.
- Garantir a segurança do sistema e o sigilo dos dados trocados entre máquinas.
- Lidar com a heterogeneidade do ambiente.



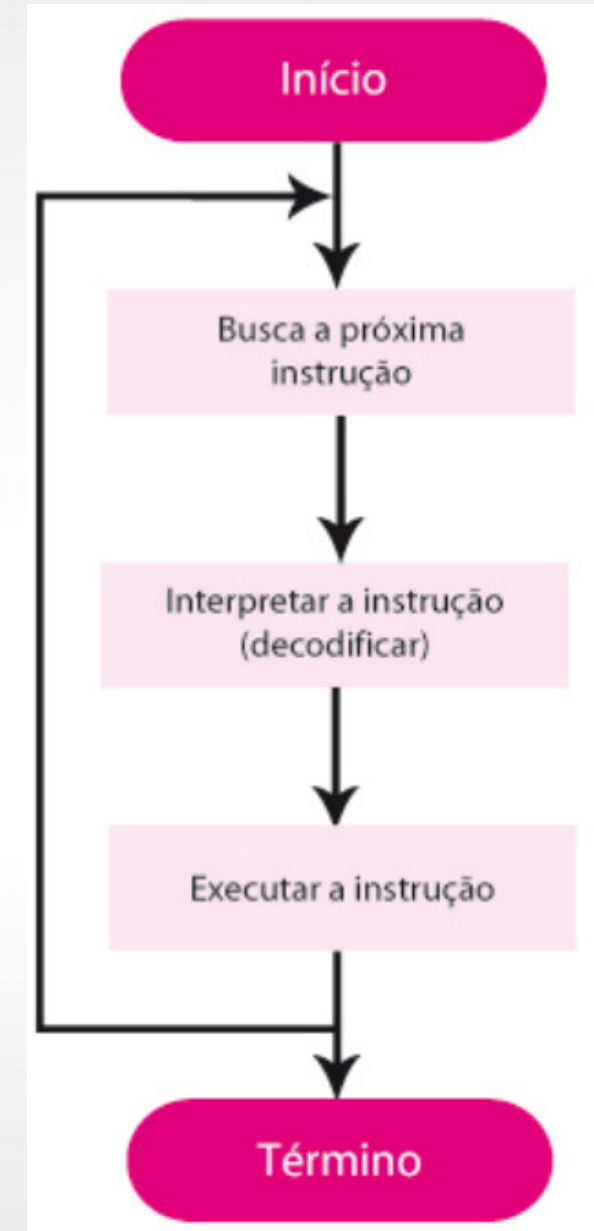
PCD: Conceitos básicos

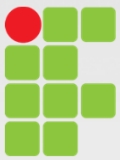
Todo computador é um computador paralelo



PCD: Conceitos básicos

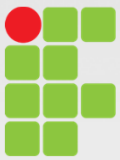
- Atualmente, todos os computadores são essencialmente paralelos.
- O paralelismo está presente profundamente na microarquitetura do processador.
- No passado, os processadores executavam programas repetindo o chamado ciclo de instrução, uma sequência de quatro etapas:
 1. leitura da próxima instrução;
 2. interpretar a instrução (decodificar);
 3. executar a instrução;
 4. escrever o resultado.





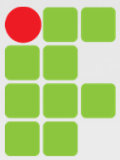
PCD: Conceitos básicos

- Ao longo dos anos a pesquisa se concentrou no projeto de um processador capaz de executar várias instruções simultaneamente.
- Esses processadores permitiram a exploração do paralelismo inerente à execução de instruções com velocidades de execução de programas ainda maiores.



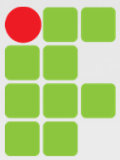
PCD: Conceitos básicos

- Qualquer computador, tablet e smartphone contém um processador com vários núcleos.
- Se os fluxos forem projetados para que os núcleos colaborem na execução de um aplicativo, o aplicativo será executado em paralelo e poderá ser consideravelmente acelerado.
- Muitos servidores contêm vários processadores ***multicore***.
- Muitos servidores contêm vários processadores ***multithread***.
- Tais servidores são capazes de executar um serviço em paralelo, e também vários serviços em paralelo.
- Mesmo os computadores de nível de consumidor contêm processadores gráficos capazes de executar centenas ou mesmo milhares de threads em paralelo.



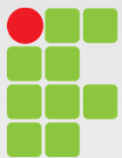
PCD: Conceitos básicos

- Razões para tornar os computadores modernos paralelos:
 - ✓ Não é possível aumentar indefinidamente as frequências de processador e memória com a atual tecnologia baseada em silício.
 - ✓ Portanto, para aumentar o poder computacional dos computadores, são necessários novos conceitos arquitetônicos e organizacionais.



PCD: Conceitos básicos

- Razões para tornar os computadores modernos paralelos (**cont.**):
 - ✓ O consumo de energia aumenta com a frequência do processador enquanto a eficiência energética diminui.
 - ✓ O paralelismo tornou-se parte de qualquer computador e provavelmente permanecerá inalterado devido à simples inércia: **o paralelismo pode ser feito!**



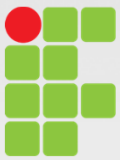
PCD: Conceitos básicos

Tipos predominantes de paralelismo



PCD: Conceitos básicos

- Diversos sistemas de computação paralela surgiram no mercado como supercomputadores dedicados a resolver problemas científicos específicos.
- As soluções evoluíram para sistemas paralelos modernos que exibem pelo menos um dos três tipos predominantes de paralelismo:
 - ✓ **Sistemas de memória compartilhada:** várias UCP e uma única memória.
 - ✓ **Sistemas distribuídos:** vários computadores, cada uma com sua própria UCP e memória, conectadas por redes. Exemplos:
 - *Cluster Computing*
 - *Grid Computing (Cloud Computing)*
 - ✓ **GPUs:** usadas como coprocessadores para resolver problemas numericamente intensivos.



PCD: Conceitos básicos

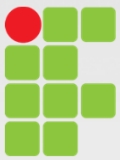
- Supercomputadores extremamente poderosos continuam a dominar as conquistas da computação paralela.
- Podem ser encontrados na lista Top 500 dos sistemas de computador mais rápidos já construídos (<https://www.top500.org/lists/top500/list/2022/11/>).
- Mesmos princípios da computação paralela e evolução das técnicas de programação.
- **O projeto de algoritmos paralelos e programação paralela ainda são considerados uma ordem de grandeza mais difícil do que o projeto de algoritmos sequenciais e desenvolvimento de programas sequenciais.**



PCD: Conceitos básicos

Plataformas de execução

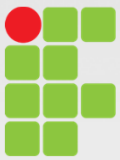
- Um S.O. multitarefa permite simular o paralelismo em um único processador, alternando a execução de processos
- Um processador com núcleo múltiplo permite paralelismo real entre processos, executando múltiplas instruções por ciclo.
- Uma placa mãe multiprocessada permite que cada processador execute um processo.
- Um Cluster é uma solução de baixo custo para processamento de alto desempenho.
- Computação Distribuída é possível em redes, como numa Intranet e na Internet.



PCD: Conceitos básicos

Suporte Computacional

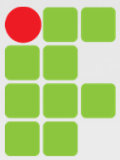
- Suportes para Computação Paralela e Distribuída devem fornecer:
 - ✓ Mecanismos para execução paralela ou distribuída de programas
 - ✓ Mecanismos para controle de concorrência
 - ✓ Mecanismos para comunicação entre processos / threads paralelos / distribuídos
 - ✓ Ferramentas e mecanismos para desenvolvimento, testes, gerenciamento, controle, segurança, tolerância a faltas, etc.



PCD: Conceitos básicos

Suporte Computacional

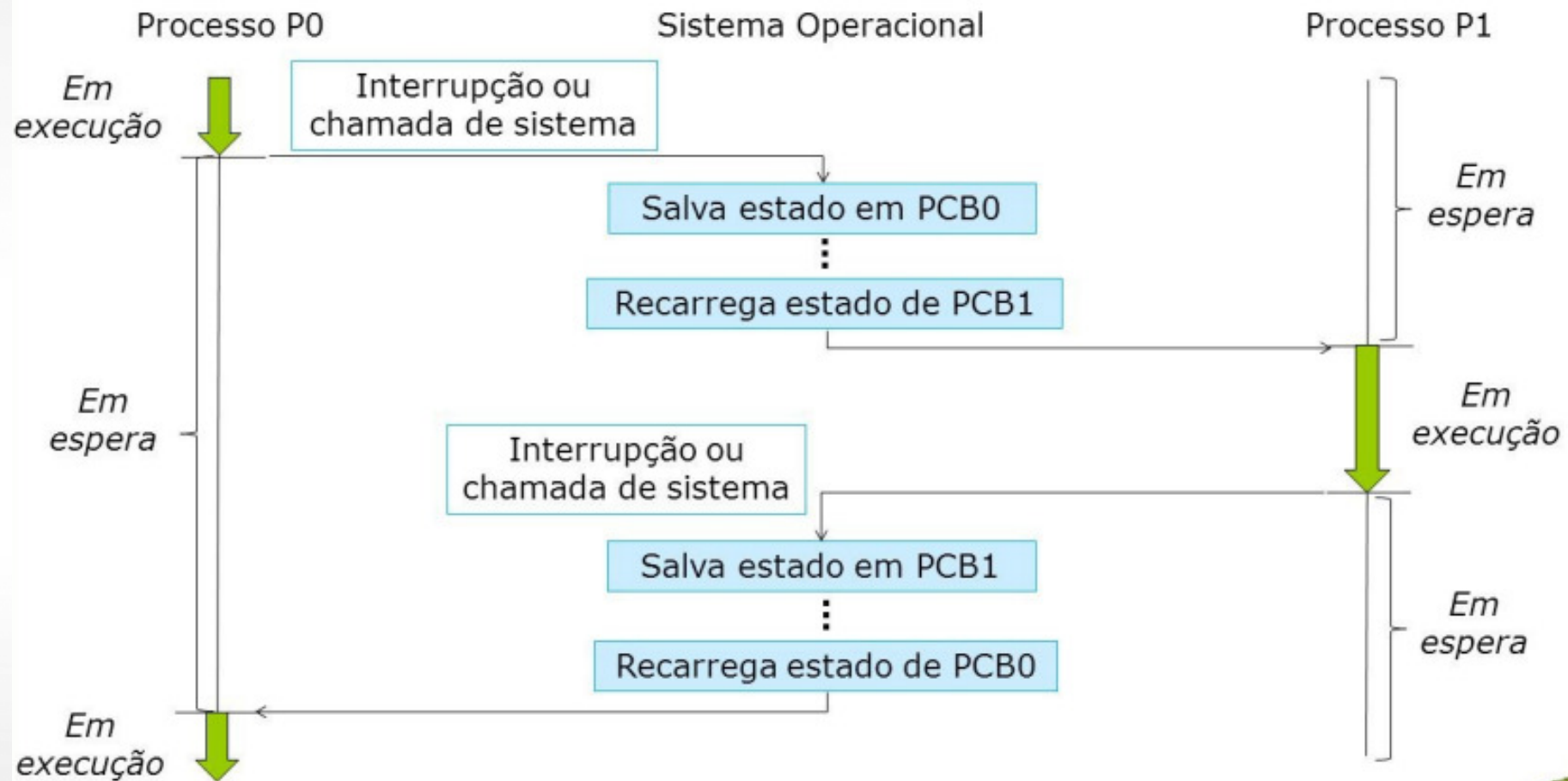
- Suporte para Computação Paralela:
 - ✓ Sistemas Operacionais multitarefa: permitem a troca de contexto entre processos / threads. Ex.: Windows, Linux, Solaris, HP-UX, AIX, etc.
 - ✓ Linguagens multitarefa : permitem escrever programas paralelos. Ex.: Java.
 - ✓ Sistemas Operacionais Paralelos: permitem usar vários processadores em uma máquina. Ex.: Linux, Solaris, Windows, etc.
 - ✓ Suportes para Programação Paralela permitem criar uma máquina paralela virtual. Ex.: PVM.

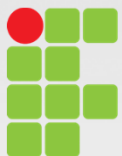


PCD: Conceitos básicos



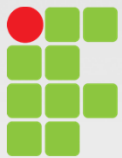
Troca de contexto





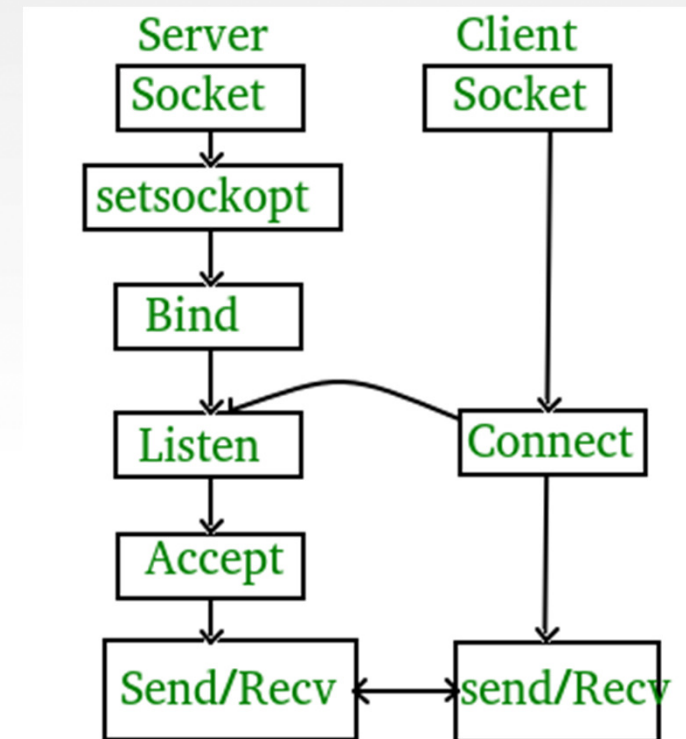
Atividade

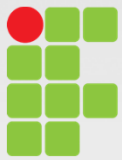
- Comunicação entre processos cliente e servidor via TCP/IP:
 - ✓ TCP é adequado para aplicativos que exigem alta confiabilidade e o tempo de transmissão é relativamente menos crítico.
 - ✓ É usado por outros protocolos como HTTP, HTTPs, FTP, SMTP, Telnet.
 - ✓ Há garantia absoluta de que os dados transferidos permanecem intactos e chegam na mesma ordem em que foram enviados.
 - ✓ Ele também faz verificação de erros e recuperação de erros (pacotes errados são retransmitidos da origem para o destino).



Atividade

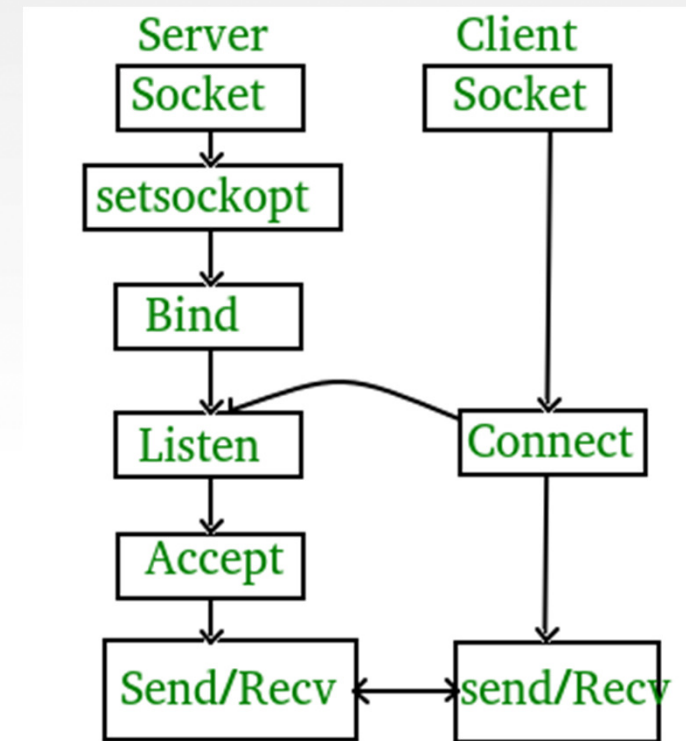
- Todo o processo pode ser dividido nas seguintes etapas:
 - Servidor TCP
 - i. create(): cria socket TCP.
 - ii. bind(): vincula o soquete ao endereço do servidor.
 - iii. listen(): coloca o socket do servidor em modo passivo, onde ele espera que o cliente se aproxime do servidor para fazer uma conexão
 - iv. accept(): neste ponto, a conexão é estabelecida entre cliente e servidor, e eles estão prontos para transferir dados.
 - v. Volta para o passo *iv*.

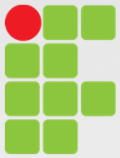




Atividade

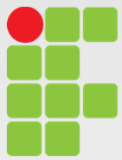
- Todo o processo pode ser dividido nas seguintes etapas:
 - Cliente TCP
 - i. Cria um soquete TCP.
 - ii. Conecta o soquete recém-criado ao socket do servidor.





Cliente e Servidor TCP

<https://www.geeksforgeeks.org/tcp-server-client-implementation-in-c/>



Referências

O conteúdo dos slides foi retirado do(s) livro(s):

*Roman Trobec, Boštjan Slivnik, Patricio Bulić, Borut Robič.
**Introduction to Parallel Computing - From Algorithms
to Programming on State-of-the-Art Platforms.** Suíça:
Springer Cham, 2018.*