Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

ARQUITETURAS DE SISTEMAS DISTRIBUÍDOS

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

INTRODUÇÃO

- A organização dos SDs trata em grande parte dos componentes de software que constituem o sistema
- Os componentes estão espalhados por várias máquinas
- Os sistemas devem ser organizados adequadamente
  - Estilo Arquitetônico => Organização lógica do conjunto de componentes
  - Arquitetura de software => Nos diz como vários componentes de software devem ser organizados e como devem interagir
  - Arquitetura de sistema => A organização física dos componentes

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

ESTILOS ARQUITETÔNICOS

O estilo da arquitetura é formulado em termo dos componentes

- Como os componentes estão conectados uns aos outros
- Como os dados são trocados entre os componentes
- Como os componentes são configurados em conjunto para formar um sistema

Componente: é uma unidade modular com interfaces requeridas e fornecidas bem definidas que é substituível dentro de seu ambiente.

Conector: é um mecanismo que serve como mediador da comunicação ou da cooperação entre componentes.

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

ESTILOS ARQUITETÔNICOS

Usando componentes e conectores, podemos chegar a várias configurações, que foram classificadas em estilos arquitetônicos.

- Arquitetura em camadas
- Arquitetura baseada em objetos e orientada a serviços
- Arquitetura centrada em recursos
- Arquitetura baseada em eventos

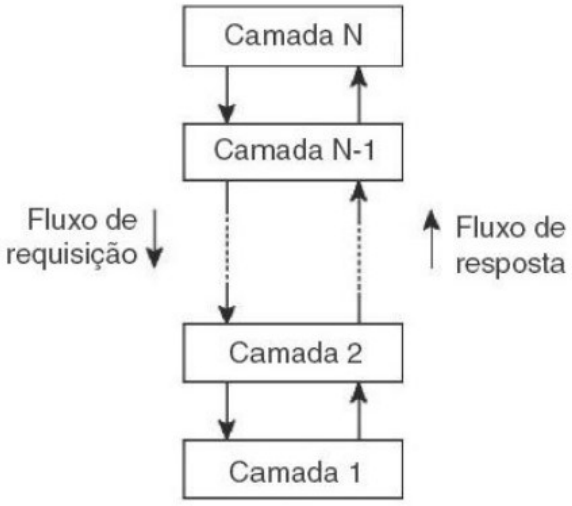
Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

ARQUITETURA EM CAMADAS

- Os componentes são organizados em camadas
- Cada componente da camada Li pode chamar um componente da camada Li-1, mas não o contrário
- Modelo amplamente adotado pela comunidade de redes (OSI, TCP, ...)
- O controle flui de camada para camada
  - As requisições descem pela hierarquia
  - Os resultados fluem para cima

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

ARQUITETURA EM CAMADAS



Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

ARQUITETURA EM CAMADAS

Exemplo

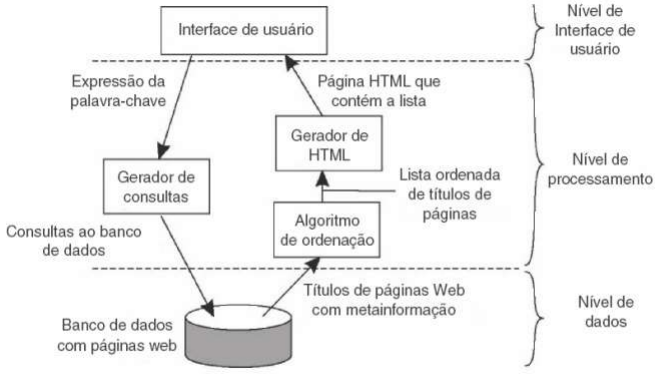
Visão tradicional em 3 camadas:

- **Nível de interface**: contém componentes para fazer a interface da aplicação com usuários ou aplicações externas
- **Nível de processamento**: contém as funções da aplicação, sem dados específicos
- **Nível de dados**: contém os dados que o cliente deseja manipular através da aplicação

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

ARQUITETURA EM CAMADAS

Exemplo



Organização simplificada de um mecanismo de busca da Internet em três camadas diferentes.

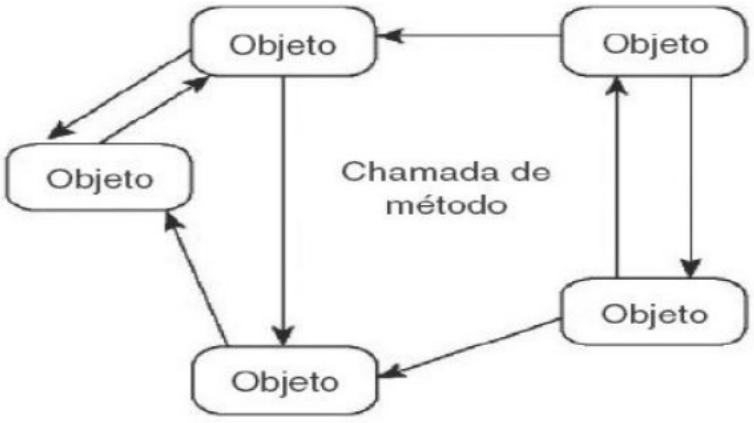
Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

ARQUITETURA BASEADA EM OBJETOS E ORIENTADA A SERVIÇOS

- Arquiteturas baseadas em objetos e orientadas a serviços
  - Objetos correspondem às definições de componentes
  - Objetos (componentes) são conectados por meio de chamadas de procedimento remotas
  - Um tipo de organização mais solta
- Arquiteturas em Camadas e Baseadas em Objetos são os estilos mais importantes para sistemas de SW de grande porte

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

ARQUITETURA BASEADA EM OBJETOS E ORIENTADA A SERVIÇOS

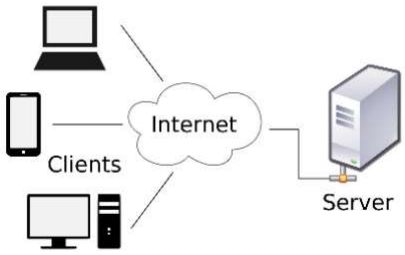


Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

ARQUITETURA BASEADA EM OBJETOS E ORIENTADA A SERVIÇOS

Exemplo

Arquitetura Cliente/Servidor se ajusta a esse modelo



Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

ARQUITETURAS CENTRADAS EM RECURSOS

Arquiteturas centradas em recursos (RESTful)
O sistema distribuído é visto como uma coleção de recursos gerenciados individualmente pelos componentes. Os recursos podem ser adicionados, removidos, recuperados e modificados por aplicações (remotas)

- Os recursos são identificados por um único esquema de nomeação
- Todos os serviços oferecem a mesma interface
- As mensagens enviadas para um serviço ou recebidas do serviço são totalmente auto-contidas
- Após a execução de uma operação em um serviço, o componente esquece tudo sobre o chamador

ARQUITETURAS CENTRADAS EM RECURSOS

Exemplo

Sistema de armazenamento da Amazon

Objetos (arquivos) são colocados em buckets (diretórios). Buckets não podem ser colocados em buckets. Operações em ObjectName no bucket BucketName requerem o seguinte identificador (URI):

http://BucketName.s3.amazonaws.com/ObjectName

- Todas as operações são executadas através de pedidos HTTP:
- Cria um bucket/objeto: POST URI
- Lista objetos: GET no nome do diretório
- Lê um objeto: GET URI

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

ARQUITETURAS CENTRADAS EM RECURSOS

Exemplo

Muitos usuários gostam dessa abordagem RESTful porque a interface para o serviço é simples. O problema é que precisa muito trabalho na identificação dos parâmetros. Amazon provê uma interface baseada em serviços.

Amazon S3 SOAP interface

Bucket operations	Object operations
ListAllMyBuckets	PutObjectInline
CreateBucket	PutObject
DeleteBucket	CopyObject
ListBucket	GetObject
GetBucketAccessControlPolicy	GetObjectExtended
SetBucketAccessControlPolicy	DeleteObject
GetBucketLoggingStatus	GetObjectAccessControlPolicy
SetBucketLoggingStatus	SetObjectAccessControlPolicy

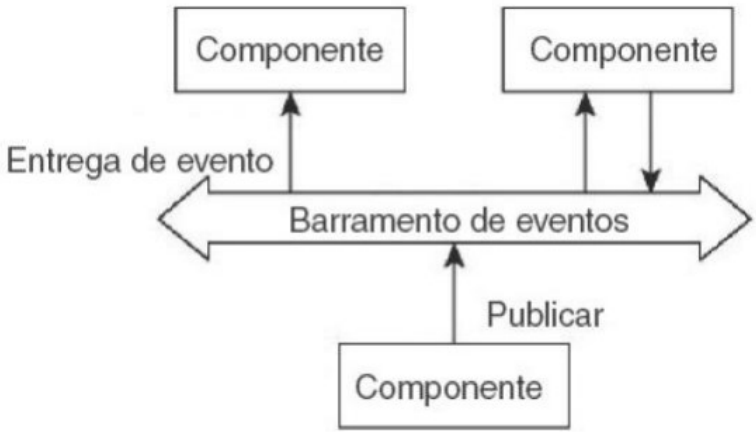
Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

ARQUITETURA BASEADA EM EVENTOS

- Processos se comunicam por meio de propagação de eventos
  - Opcionalmente estes eventos podem transportar dados
- Sistemas Publisher/Subscriber
  - Processos publicam eventos
  - O middleware assegura que somente os processos assinantes irão receber esses eventos.
- Os processos são fracamente acoplados
  - Processos não precisam se referir especificamente uns aos outros

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

ARQUITETURA BASEADA EM EVENTOS



Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

ARQUITETURA BASEADA EM EVENTOS

Exemplo

Apache Camel, Spring Integration ou Mule ESB

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

ARQUITETURA DE SISTEMAS

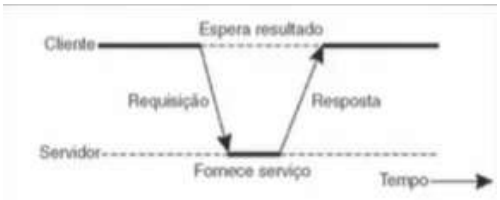
- Arquiteturas Centralizadas Tradicionais
- Arquiteturas Descentralizadas
- Arquiteturas Híbridas

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

ARQUITETURA CENTRALIZADAS

Modelo Cliente/Servidor

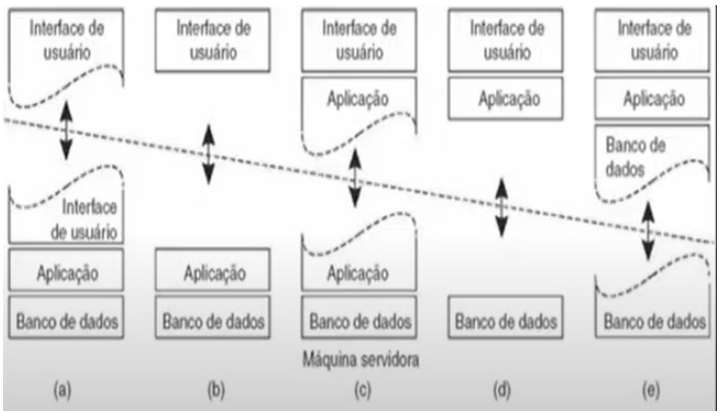
- Processos são divididos em 2 grupos
  - Servidor: processo que implementa um serviço específico
  - Cliente: processo que solicita um serviço ao servidor
- Forma de Interação: Requisição-Resposta



Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

ARQUITETURA CENTRALIZADAS

Modelo Cliente/Servidor
(Arquitetura centralizada com arquiteturas multidivididas)



Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

ARQUITETURA DESCENTRALIZADAS

Peer-to-Peer

- Devido ao comportamento simétrico arquiteturas P2P se dividem pela forma de organizar os processos em uma rede de sobreposição.

Rede de sobreposição: rede na qual os nós são os processos e os enlaces representam os canais de comunicação possíveis.

- Processos não se comunicam diretamente e sim por meio dos canais de comunicação disponíveis.
- Existem 2 tipos de redes de sobreposição: Estruturadas e Não estruturadas

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

ARQUITETURA HÍBRIDAS

Sistemas distribuídos onde soluções cliente-servidor são combinadas com arquiteturas descentralizadas.

Uma ideia é que o modelo cliente-servidor é utilizado para os nós se conectarem ao sistema, depois utilizam esquema descentralizado.

Exemplos:
- WhatsApp
- Skype
- BitTorrent

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

VÍDEOS

Arquitetura de Sistemas Distribuídos
https://www.youtube.com/watch?v=ZCW6NgCu3iE&t=634s

Tipos de arquitetura
https://www.youtube.com/watch?v=HhT7-GmJ1ls

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

INVESTIGAÇÃO

Pesquisar na internet e montar um texto de 2 páginas sobre exemplos práticos de arquiteturas de sistemas (centralizadas, descentralizadas e híbridas).

Enviar via Classroom até o final da aula de hoje.