Estácio

# ARQUITETURA WEB

# OBJETIVOS

- Arquitetura Web
- Protocolo HTTP
- Linguagem Java
- *Web Containers*

# ARQUITETURA WEB

- **Arquitetura Web**
  - Onde está a internet?
  - Onde estão as páginas?
  - Onde está o Google, Facebook, GMail...?
  - O que são esses sistemas?

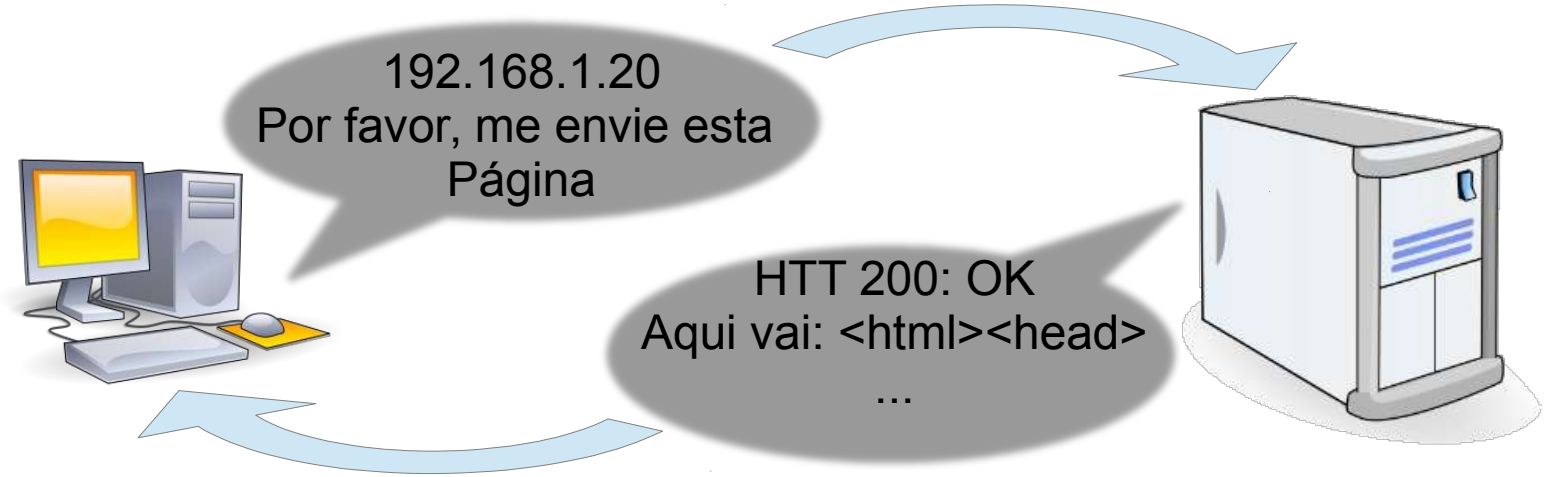
# ARQUITETURA WEB

- **Arquitetura Web**
  - Estes serviços estão em um computador como o seu
    - Com hardware com maior confiabilidde e disponibilidade.

  - Por fornecer serviços, esse equipamento é chamado Servidor, estes serviços podem ser diversos:
    - SMTP
    - FTP
    - HTTP
    - … ( Aplicações diversas )

# ARQUITETURA WEB

- **Arquitetura Web**
  - O software, instalado no seu computador, que usa estes serviços é chamado de Cliente
    - Ex.: **navegadores** são clientes HTTP
  - Um cliente Requisita um serviço e obtém uma Resposta do Servidor.

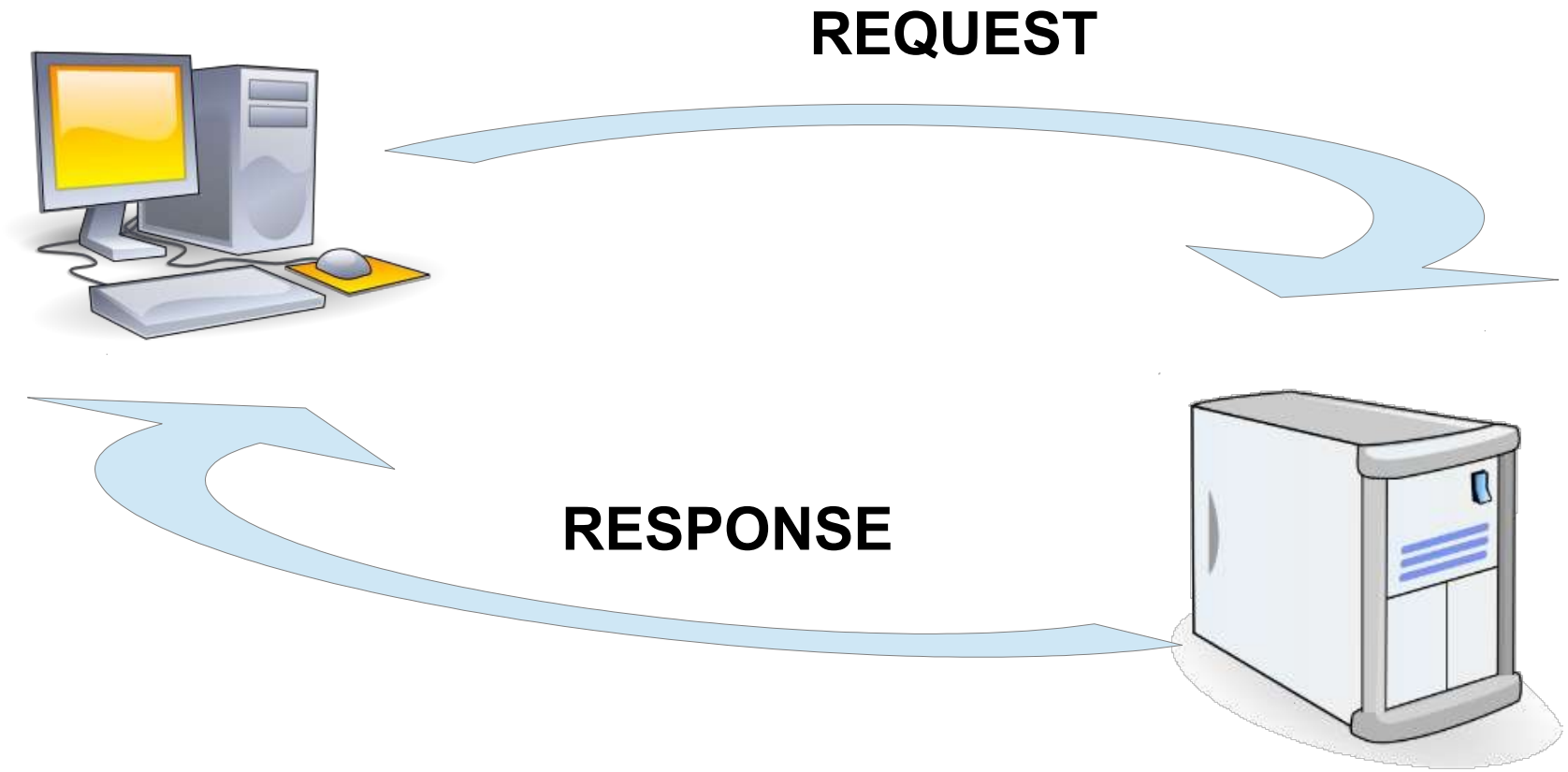


# ARQUITETURA WEB

- **Arquitetura Web**
  - Na prática
    - Pacotes transmitidos pela rede ( Internet ou Intranet )
    - Não nos interessa, nesta disciplina, protocolos da camada de rede

# ARQUITETURA WEB

- **Arquitetura Web**
  - Simplificadamente ...



# ARQUITETURA WEB

- Baseada no Protocolo HTTP (Hypertext Transfer Protocol)
  - Definido pela RFC 7230 (HTTPS → RFC 2818)
  - Protocolo simples de transferência de arquivos
  - Sem estado (não mantém sessão aberta)
  - Métodos principais: GET e POST

# Cliente e Servidor HTTP

- **Servidor HTTP**
  - Gerencia sistema virtual de arquivos e diretórios
  - Mapeia pastas do sistema de arquivos local (ex: c:\htdocs) a diretórios virtuais (ex: /) acessíveis remotamente (notação de URI)
  - Gerencia conexões
    - Interpretar requisições HTTP do cliente (métodos GET, POST, ...)
    - Devolver resposta HTTP ao cliente (código de resposta 200, 404, etc., cabeçalho e dados )
- **Cliente HTTP**
  - Enviar requisições HTTP (GET, POST, HEAD, ...) a um servidor.
  - Requisições contém URI do recurso remoto, cabeçalhos e, opcionalmente, dados
  - Processar respostas HTTP recebidas e, se for o caso, formatá-las para exibição ao usuário.

# HTTP – Principais Métodos

- **GET**
  - Utilizado para pedir ao servidor um arquivo, informando o seu caminho relativo à raiz do servidor
    - GET <uri> <protocolo>/<versão>
  - GET pode enviar dados através da URI (tamanho limitado)
    - GET <uri>?dados  <protocolo>/<versao>

- **POST**
  - Envia dados ao servidor (como fluxo de bytes)
    - POST <uri> <protocolo>/<versão>
  - Dados são enviados logo após o cabeçalho HTTP

- **HEAD**
  - Idêntico ao GET mas servidor não devolve página, devolve apenas o cabeçalho

# LINGUAGEM DO CURSO: JAVA

- Qual a melhor linguagem para isso?
  - Certamente não é C/C++
    - Linguagem de baixo nível
  - Existem várias linguagens no mercado!
    - ASP, PHP, Python, Ruby...
  - Neste curso usaremos **JAVA!**

Java

ORACLE

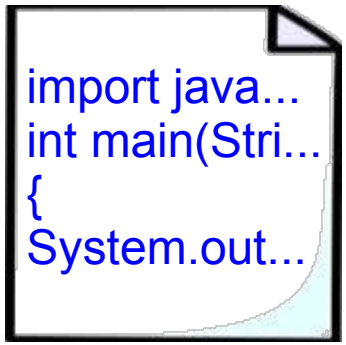
# LINGUAGEM DO CURSO: JAVA

- Por que JAVA
  - Linguagem similar ao C/C++
  - Vasta biblioteca
  - Vários Frameworks para desenvolvimento Web
  - Linguagem bem difundida no mercado de aplicações servidor
  - Alta empregabilidade
  - Robustez e Segurança

# LINGUAGEM DO CURSO: JAVA

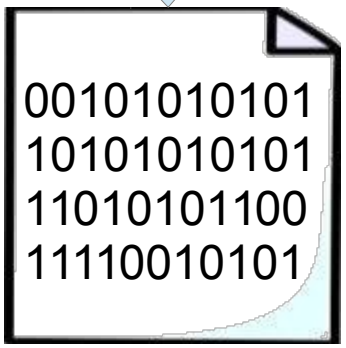
- JAVA – Linguagem Híbrida ( Compilada e Interpretada )

Programador




Código Fonte

Compilador




Byte Code Java

Sistema Operacional “A”
com Interpretador Java

Sistema Operacional “B”
com Interpretador Java

# LINGUAGEM DO CURSO: JAVA

- Um pouco mais sobre JAVA:
  - O nome oficial do Interpretador Java é “Java Virtual Machine”, carinhosamente apelidado de JVM.
  - Para executar aplicações Java, é necessário instalar a JVM.
  - A JVM sozinha, porém, não contém as bibliotecas necessárias para executar tudo que um programa Java precisa.
  - Assim, a Oracle distribui um pacote chamado Java Runtime Environment (JRE) contendo a JVM e as bibliotecas oficiais do Java.

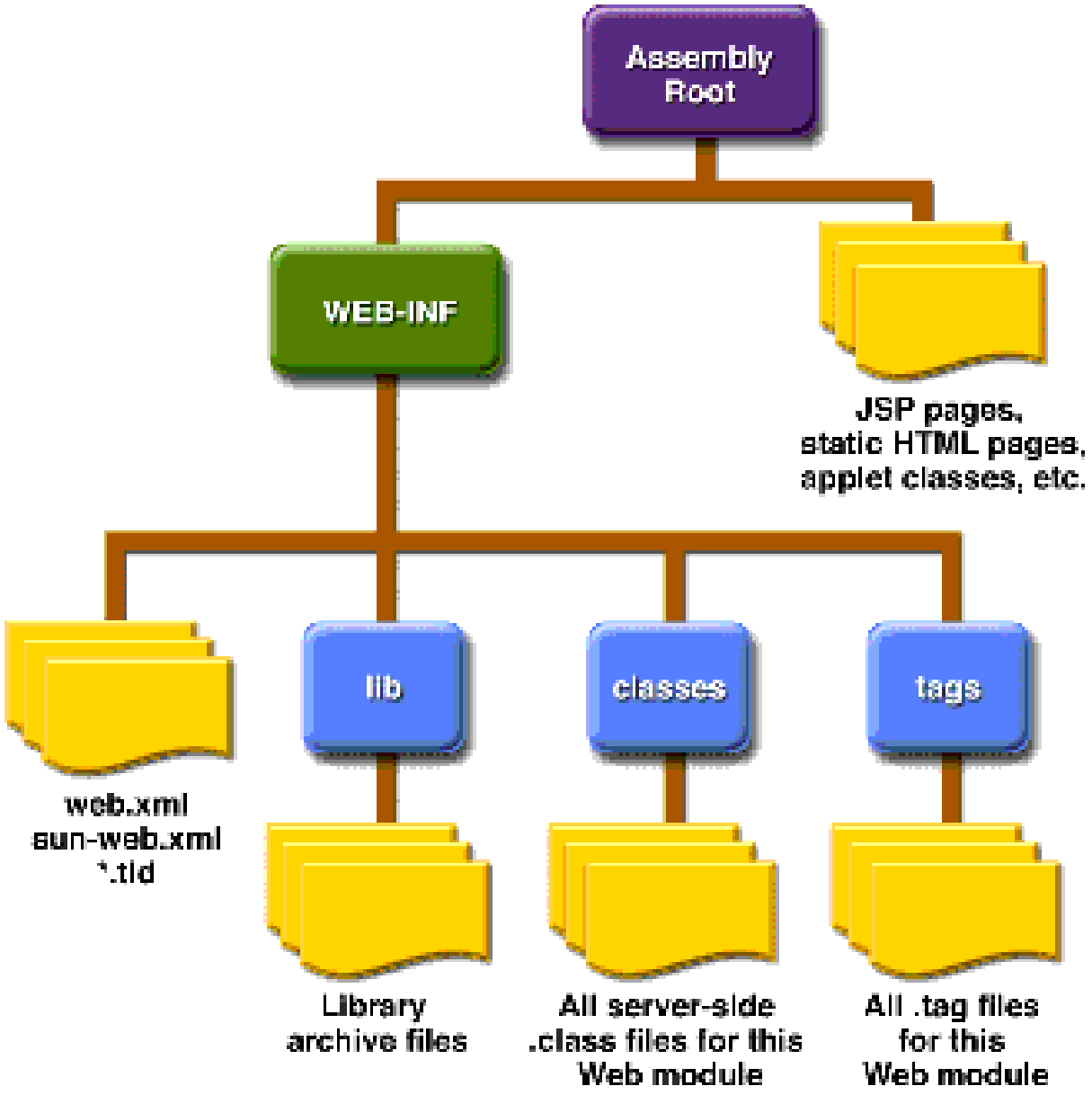
# LINGUAGEM DO CURSO: JAVA

- Um pouco mais sobre JAVA:
  - O JRE inclui apenas os pacotes necessários à execução de programas Java.
  - Para poder gerar programas Java, é necessário baixar um pacote mais completo, chamado Java Development Kit (JDK).
  - O JDK inclui, além dos elementos do JRE, também as ferramentas de desenvolvimento do ambiente Java.

# LINGUAGEM DO CURSO: JAVA

- Um pouco mais sobre JAVA:
  - Como o JRE e o JDK se tornaram muito grandes, a Oracle organizou três pacotes diferentes, de acordo com as necessidades de cada programador:
    - Java Micro Edition (Java ME)
      - para aplicações portáteis, isto é, aquelas que executam no celular do usuário.
    - Java Standard Edition (Java SE)
      - para aplicações desktop, isto é, aquelas que executam no computador do usuário.
    - Java Enterprise Edition (Java EE):
      - para aplicações que executam em um servidor, isto é, respondendo requisições.
      - **Neste curso usaremos o Java EE**

# JEE - ESTRUTURA DE DIRETÓRIOS




Fonte: [http://docs.oracle.com]

# CONTAINER WEB

- **Servlets e JavaServer Pages (JSP)** são as soluções Java para aplicações Web (JEE)
  - Suportam os métodos de requisição HTTP padrão
  - Geram respostas compatíveis com HTTP
  - Interagem com Cookies

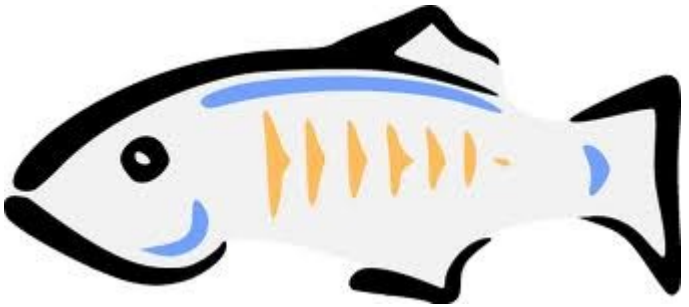
- Além dessas tarefas básicas, também
  - Suportam filtros, que podem ser chamados em cascata para tratamento de dados durante a requisição
  - Suportam controle de sessão transparentemente através de cookies ou rescrita de URLs (automática)

- **Para isso funcionar é preciso usar um servidor que suporte as especificações de Servlets e JSP: Web Container**

# CONTAINER WEB

- Existem vários Contentores Java
  - GlassFish
  - TomCat
  - JBOSS
  - Gerônimo
  - Jetty
  - Dentre outros...

- Neste curso usaremos o **GlassFish**
  - É o mais completo
  - É o padrão sugerido pela Oracle



Até a Próxima Aula!