

IOS – Instituto de
Oportunidade Social

Aula 01 - Introdução ao Java



- > Sobre o Java
- > Configuração do Java
- > JDK vs JRE vs JVM
 - > JDK > Porque usar o JDK?
 - > JRE > Porque usar o JRE?
 - > JVM > Porque usar o JVM?
 - > Como o JDK funciona?
 - > Como o JRE funciona?
 - > Como o JVM funciona?
- > Compilação e execução do programa Java
- > Introdução à logica
- > Algoritmo
- > Etapas de um Algoritmo

IOS – Instituto de
Oportunidade Social

Sobre o Java





- linguagem de programação orientada a objetos, de alto nível e baseada em classes
- programem o código uma vez e possam executar em qualquer lugar
- código Java compilado pode ser executado em todas as plataformas que suportam Java sem a necessidade de recompilação

- James Gosling e sua equipe da Sun Microsystems
- 1995
- Ela é baseada nas linguagens C e C++
- GPL-2.0 (General Public License, version 2)
- IEEE Spectrum



IOS – Instituto de
Oportunidade Social

Configuração do Java



Antes de começar a implementar seus programas e começar a desenvolver aplicações em Java, você precisa realizar algumas configurações no seu computador para que tudo execute perfeitamente como desejado. A primeira configuração é a instalação do JDK, mas antes vamos entender a diferença entre JDK, JRE e JVM

IOS – Instituto de
Oportunidade Social

JDK vs JRE vs JVM



> JDK

JDK (Java Development Kit) é um kit de desenvolvimento Java usado para implementar aplicativos e aplicações em Java. Ele está disponível para os diversos sistemas operacionais, tais como: Windows, macOS, Solaris e Linux. O JDK auxilia os desenvolvedores Java a codificar e executar programas Java. Você pode instalar mais de uma versão do JDK no seu computador.

Porque usar o JDK?



- *JDK têm ferramentas necessárias para escrever programas Java e JRE para executá-los.*
- *Inclui um compilador, Java application launcher, Appletviewer, etc.*
- *O compilador converte o código escrito em Java em código de bytes.*
- *O Java application launcher abre um JRE, carrega as classes necessárias e executa seu método principal.*

> JRE

O JRE (Java Runtime Environment) é ambiente de execução Java, que contém a JVM, a biblioteca padrão do Java, a Classloader, ferramentas de configuração e plug-in para o navegador. Podemos pensar no JRE como um pedaço de software desenvolvido para executar outro software. Se você pode nunca ter programado em Java, mas já executou aplicações em

Porque usar o JRE?



- JRE contém a biblioteca de classes, a JVM e outros arquivos de suporte para executar uma aplicação desenvolvida em Java. Ele não contém as ferramentas de desenvolvimento Java, tais como: debugger, compilador, etc.
- Ele usa importante módulos de classe, tais como: math, swingetc, util, lang, awt e biblioteca de execução.
- Se você já executou alguma aplicação Java, então o JRE está instalado no seu sistema.

> JVM

As aplicações implementadas em Java são compiladas em bytecode, que podem ser executados em qualquer máquina virtual Java (JVM), independente da arquitetura. Então, podemos entender que a JVM é a engine que fornece o ambiente de execução (runtime) para código em Java ou aplicações desenvolvidas em Java. Ele converte o Java bytecode em linguagem de máquina para ser executada no computador. A JVM é parte do JRE e não é instalada separadamente

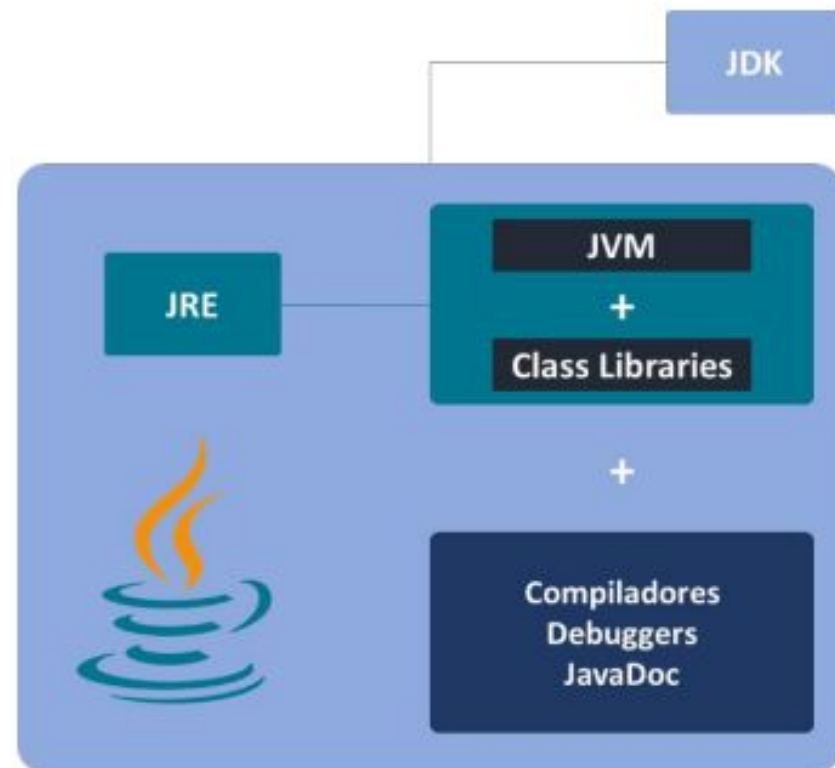
Porque usar o JVM?



- JVM permite executar uma aplicação Java sem depender da plataforma (hardware).
- Ela tem inúmeras bibliotecas, ferramentas e frameworks.
- Uma vez que você executa um programa Java, você pode executá-lo em qualquer plataforma muitas vezes.
- JVM vem com o compilador JIT (Just-in-Time), que converte o código fonte Java em linguagem de baixo nível (linguagem de máquina).

> Como o JDK funciona?

Como foi visto, o JDK tem as ferramentas necessárias para implementar e executar programas em Java. Veja um diagrama de blocos com alguns recursos do JDK:



Funcionalidade do JDK.

JDK e JRE: O JDK permite programadores criarem programas em Java, que são executados pelo JRE, que inclui a JVM e uma biblioteca de classes.

Class Libraries: é um grupo de biblioteca carregadas dinamicamente, que o programa do Java chama no tempo de execução.

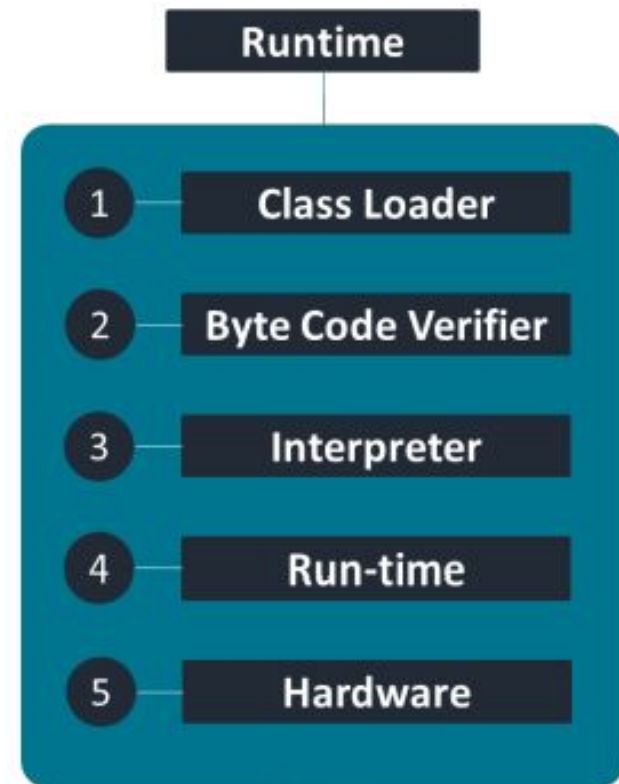
Compiladores: é um programa Java que aceita arquivo de texto (código-fonte) desenvolvidos pelos programadores e compila em um arquivo Java, que geralmente é um arquivo em bytecode.

Debugger: é um programa Java que permite desenvolvedores testar e debuggar (encontrar erros) no código programado.

JavaDoc: é uma documentação criada pela Sun Microsystems pra o Java.

> Como o JRE funciona?

Como citado, o JRE é responsável por garantir a execução de uma aplicação Java no seu computador. Veja os componentes importantes do JRE:



Funcionamento do JRE

> Como o JRE funciona?

Class loaders: Class loader carrega várias classes que são necessárias para executar um programa Java. A JVM utiliza três tipos de

class loaders: bootstrap class loader, extensions class loader e system class loader.

Byte code verifier: verifica o bytecode (programa compilado), que queremos executar. Interpreter (Interpretador): lê linha por linha para executar o programa.

> Como o JRE funciona?

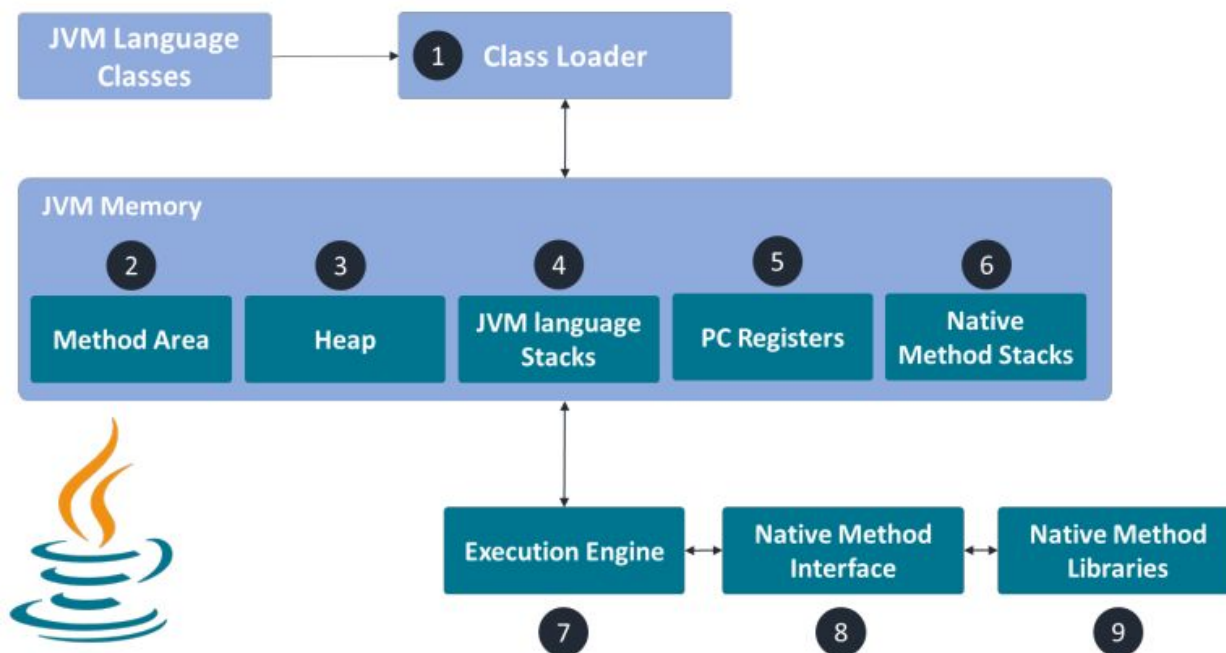
Byte code verifier: verifica o bytecode (programa compilado), que queremos executar. Interpreter (Interpretador): lê linha por linha para executar o programa.

Run-time: é um sistema usado em programação para descrever o período de tempo o qual o programa está executando.

Hardware: é a plataforma para a qual o bytecode foi interpretado, ou seja, é onde o programa irá executar.

> Como o JVM funciona?

A imagem a seguir mostra os componentes importantes da JVM:



Funcionamento do JVM.

Class Loader: é um subsistema usado para carregar os arquivos de classes. 2.

Method Area: JVM Method Area armazena a estrutura das classes, tais como: metadados, o código para os métodos Java e constant pool. 3.

Heap: todo array de objetos e instância de variáveis são armazenadas na Heap, que é uma parte na memória compartilhada por múltiplas threads. 4.

JVM language Stacks: armazena variáveis locais e resultados parciais. Cada thread tem sua própria JVM language Stack, que foi criada no momento que a thread é criada. 5.

PC Registers: armazenam o endereço da instrução, que está sendo executada. 6.

Native Method Stacks: retem a instrução do código nativo.

7. Execution Engine: é um tipo de, que é usada para testar o software, o hardware ou o sistema completo.

8. Native Method Interface: é um framework de programação, que permite o código Java chamar bibliotecas e aplicações nativas, enquanto é executado.

9. Native Method Libraries: é uma coleção de bibliotecas nativas (C, C++), que são necessárias para a execução da engine

IOS – Instituto de
Oportunidade Social

Protocolos da Internet

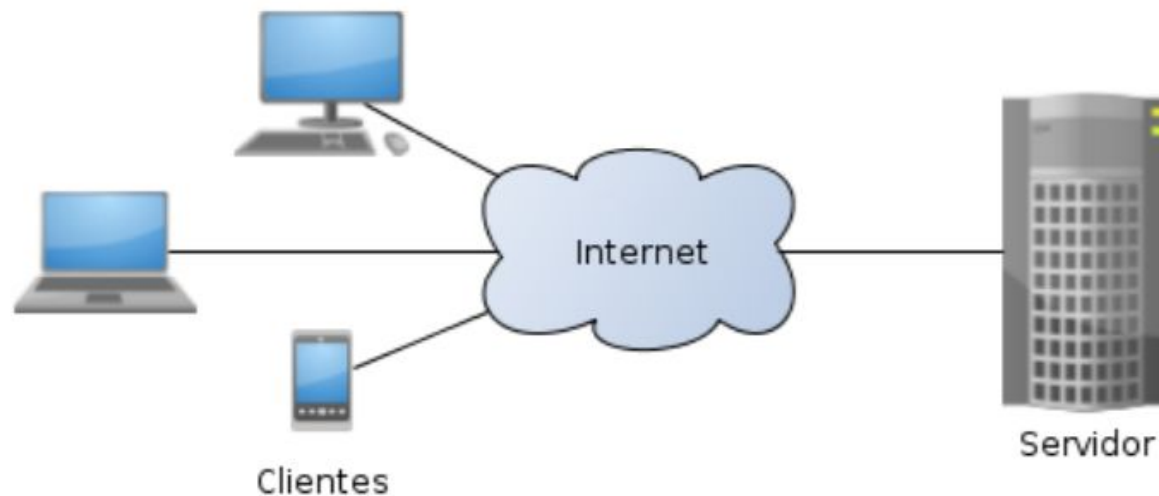


IOS – Instituto de
Oportunidade Social

Protocolos da Internet



O que são os protocolos e para que serve ?



Protocolo de Transferência de Arquivos (FTP)

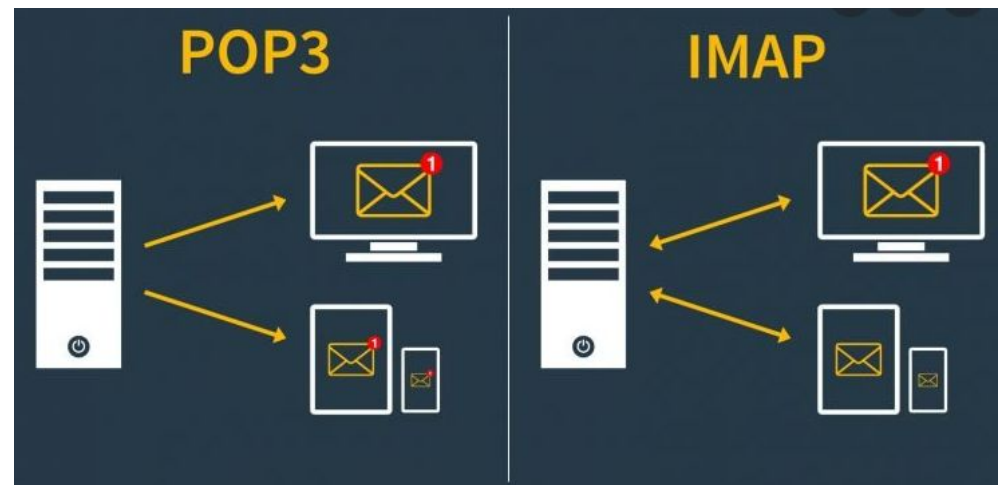


(File Transfer Protocol-FTP)

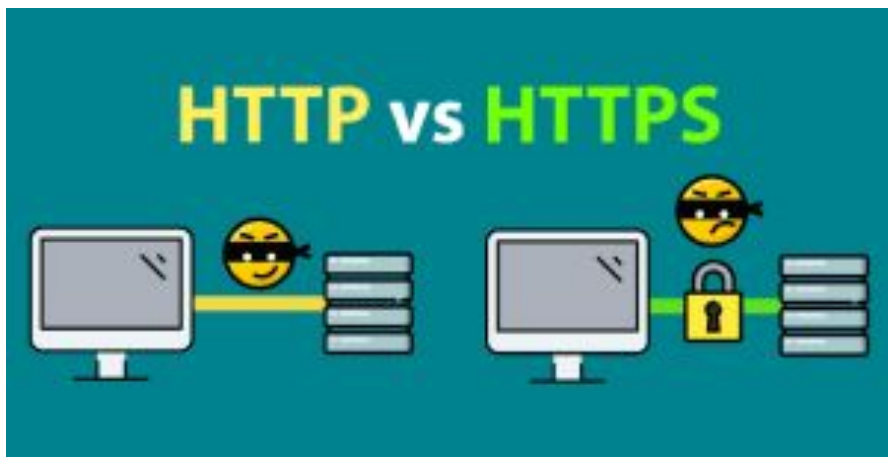
Protocolos de E-mail



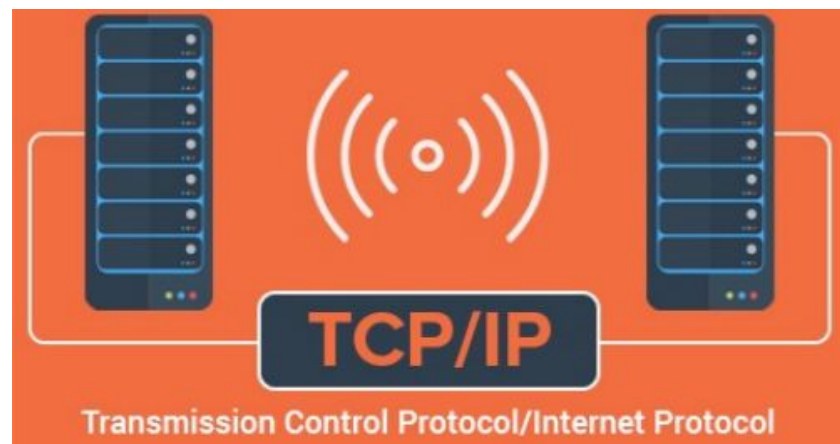
Diferença de POP3 e IMAP



Protocolo de Transferência de Hipertexto (HTTP)



Transmission Control Protocol/Internet Protocol (TCP/IP)



relatório da Statcounter

Mobile

- **Chrome - 61,7%**
- **Safari - 22,84%**
- **Samsung Internet - 6,4%**
- **UC Browser - 4,79%**
- **Opera - 2,14%**

Desktop

- **Chrome - 68,78%**
- **Mozilla Firefox - 9,87%**
- **Safari - 8,64%**
- **Edge - 4,7%**
- **Internet Explorer - 3,7%**

URL (Uniform Resource Locator)

<http://www.ios.org.br/login/index.php>

Protocolo
HTTP

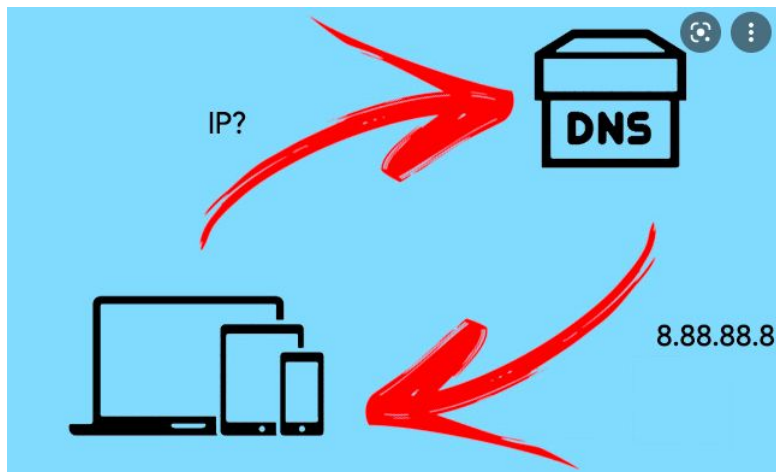
Subdomínio ou nome
do servidor web

Nome do
domínio

Nome do
diretório

Página WEB
Nome do arquivo

Nomes de Domínio (Domain Name System-DNS)



```
Prompt de Comando

Microsoft Windows [versão 10.0.19042.1237]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\Heloisa Oliveira>nslookup www.facebook.com.br
Servidor:  mgwad03.magna
Address:  192.168.4.3

Não é resposta autoritativa:
Nome:      star-mini.c10r.facebook.com
Addresses: 2a03:2880:f148:82:face:b00c:0:25de
           157.240.222.35
Aliases:   www.facebook.com.br
           www.facebook.com

C:\Users\Heloisa Oliveira>
```


IOS – Instituto de
Oportunidade Social

Usos populares para a Internet



Usos populares para a Internet

E-Commerce



Acesso móvel



Blogs



Wikis



Usos populares para a Internet

Redes sociais



Computação na Nuvem



Usos populares para a Internet

Podcasts



Serviços de Streaming de mídia



IOS – Instituto de
Oportunidade Social

HTML



- Tim Berners-Lee - 1980 (Início)
- Lançado no início dos anos 90 com 18 elementos(Tags)
- Atualmente em média tem 112 elementos(Tags).
- HyperText Transfer Protocol (HTTP):
- World Wide Web (www):
- Linguagem de marcação



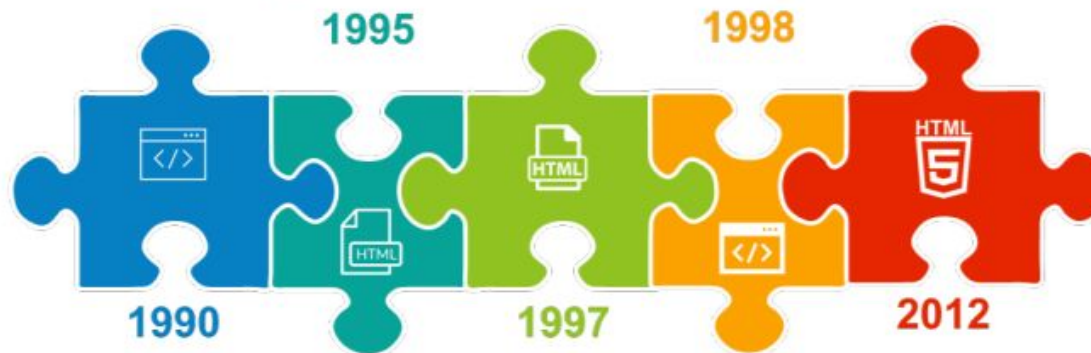
HTML

HTML2

Surgimento do HTML2, que foi desenvolvido em parceria com as empresas de navegadores, onde cada uma delas criou tags específicas para usar em seus produtos.

HTML 4

A versão 4 do HTML é lançada. Após seis anos, a W3C tinha planos de substituir o HTML por uma nova linguagem chamada XHTML.



HTML

Versão 1 do HTML (HyperText Markup Language), que foi criada por Tim Berners-Lee.

HTML 3

Foi a primeira versão desenvolvida e padronizada exclusivamente pelo W3C, com o intuito de padronizar as tags.

HTML 5

Esta nova versão traz consigo importantes mudanças quanto ao papel do HTML no mundo da Web, através de novas funcionalidades como semântica e acessibilidade.

Linha de Tempo da evolução da Linguagem HTML.

IOS – Instituto de
Oportunidade Social

Conhecendo as IDEs ou Editor de Código



IDEs



Sublime Text



Visual Studio Code