**Digital Innovation One**

**Desenvolvendo um sistema de gerenciamento de pessoas em API REST com Spring Boot**

Java 11

Maven 3.6.3

Spring Boot (última versão estável lançada)

Git/GitHub

Heroku (para deploy na nuvem)

**Padrão arquitetural REST**

Facilita integração entre múltiplos sistemas através do próprio protocolo HTTP.

Operações: *POST, GET, PUT, DELETE*, PATCH. Tem 8: OPTIONS, TRACE, READ.

REST x RESTful

REST: modelo arquitetural

RESTful: quão API é aderente aos padrões REST. Qual então é padrão?

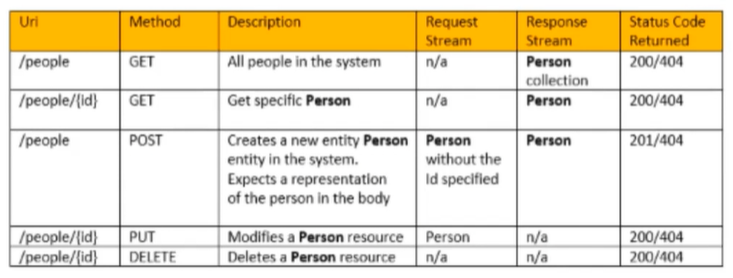
Módulo de maturidade de Richardson

Nível 0: API criada anteriormente, tem como base, sem nomenclatura de operações, verbo.

Nível 1: define o nome de recursos (URI 🡪 /people/{id})

Nível 2: “http verbs” definição dos métodos HTTP (post, get...)

Nível 3: “hypermedia controls” 🡪 controle de hypermedia diz várias outras operações que podem ser feitas. Neste curso não chegaremos aqui.



Java Developer

**Java**: Linguagem de programação e plataforma computacional lançada em 1995 pela Sun Microsystems (James Gosling). Anos depois adquirida pela Oracle.

Diferente de outras linguagens (compiladas para código nativo), o Java é compilado para um bytecode que é interpretado por uma máquina virtual.

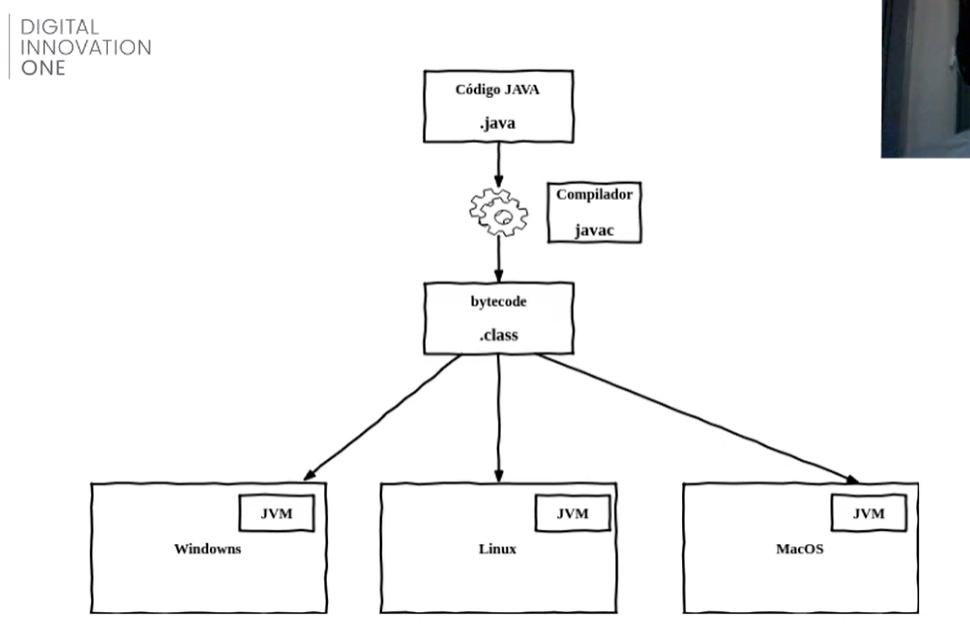
**Compilador**: programa, que a partir de código fonte, cria um programa semanticamente equivalente, porém escrito em outra linguagem (código objeto).

Utilizado principalmente para os programas que traduzem o código fonte de uma linguagem de programação de alto nível para baixo nível (por exemplo Assembly ou código de máquina).

**Bytecode**: código originado da compilação do Java. Ele será interpretado e executado pela Máquina Virtual Java (JVM).

Máquina Virtual (VM): simula uma máquina física (parte do hardware mesmo).

JVM: executa os programas em Java, lendo o bytecode. Não precisa compilar para cada sistema operacional, desta forma é uma vantagem do Java.



JRE: Java Runtime Environment: roda máquina virtual do java, bibliotecas e APIs. Parte responsável pela execucação do software.

JDK: Kit de Desenvolvimento Java. Conjunto de vários utilitários que permitem criar software para a plataforma. Composto do compilador Java, bibliotecas da linguagem, JRE e demais ferramentas.

Java SE (Standard Edition): distribuição mínima da plataforma de desenvolvimento de aplicações JAVA.

Java EE (Enterprise Edition): extensão do Java SE. Parte corporativa. Diversas especificações de partes da infraestrutura de aplicações, acesso a banco de dados, mensageria, serviços web, etc.

Jakarta EE: Oracle doou código, implementações e especificações do JAVA EE para Eclipse Foundation. Assim, evolução das especificações e padrões JAVA EE terão nome de Jakarta EE.

1. Características da linguagem

1.1. Classes

Letra maiúscula, PascalCase

Package

*Public class* NomeClasse: *modificador de acesso*

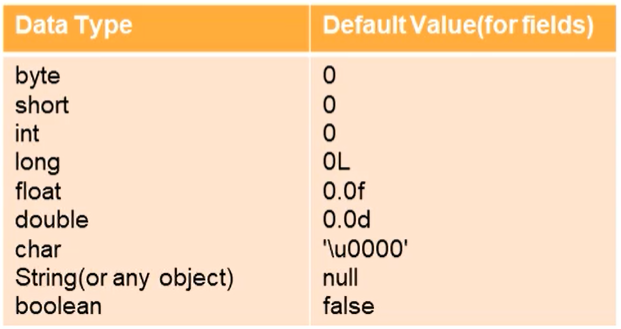
Public static void main (String[] args) {

System.out.printLn(“Hello World”)

}

2. Tipos de dados

2.1. Primitivos: não pode ser nulo, possuem valores default.



2.2. Wrappers: são objetos que representam os primitivos

Auto-boxing: atribui valor direto (autoinicialização), passa valor puro já pro objeto

Unboxing: quando tem o new. Quando constrói um objeto mas atribui ao tipo primito.

2.3. Não Primitivos

String, Number, Object, Qualquer outros objetos

2.4. Tipagem forte e estática

TipagemEstatica: não consegue converter para outro tipo de dado

**Modificadores de acesso**

- public: pode ser acessada de qualquer lugar por qualquer entidade que possa visualizar a classe a que ela pertence.

- private: os métodos e atributos da classe definidos como privados não podem ser acessados ou usados por nenhuma outra classe.

- protected: torna o membro acessível às classes do mesmo pacote ou através da herança.

- default: classe e/ou membros são acessíveis por classes do mesmo pacote. Na sua declaração não tem modificador, sendo este identificado pelo compilador.

Resumo feito por Vivian Nakano.