



DIGITAL  
INNOVATION  
ONE

# Development Environment

---

Daniel Hatanaka  
Especialista de Software

# Mais sobre mim

- Técnico em informática – IFPA
- Bacharel em Ciências de Computação – ICMC – USP – São Carlos
- Engenheiro de Software Especialista – PagSeguro PagBank
- Amante da tecnologia
- Tucuruí - Pará





DIGITAL  
INNOVATION  
ONE

# Mais sobre mim



[facebook.com/hatanakadaniel](https://facebook.com/hatanakadaniel)



[github.com/hatanakadaniel](https://github.com/hatanakadaniel)



[instagram.com/hatanakadaniel](https://instagram.com/hatanakadaniel)



[linkedin.com/in/hatanakadaniel](https://linkedin.com/in/hatanakadaniel)

# Percurso

## Aula 1

Instalando e executando Java

## Aula 2

Utilizando Java 11

## Aula 3

Usando IntelliJ para desenvolvimento Java

## Aula 4

Empacotando aplicações Java

# Requisitos

- ✓ Distribuição Linux (Ubuntu, Mint, etc)
- ✓ Conhecimentos comandos básicos terminal Linux
- ✓ Conhecimentos em lógica de programação
- ✓ Conhecimento básico de linguagem de programação
- ✓ Programação Orientada a Objetos



DIGITAL  
INNOVATION  
ONE

# Aula 1: Instalando e executando Java

Development environment

# Objetivos

1. Entendendo a JVM
2. Instalando a JVM
3. Executando um programa Java

Aula 1 | Etapa 1:

# Entendendo a JVM

Introduction to Java Ecosystem &  
Documentation



# Entendendo a JVM

O que é a JVM?

“JVM (Java Virtual Machine) é uma máquina virtual responsável pela tradução dos ByteCodes oriundos do compilador Javac (Java Compiler) em código de máquina de cada sistema operacional.”

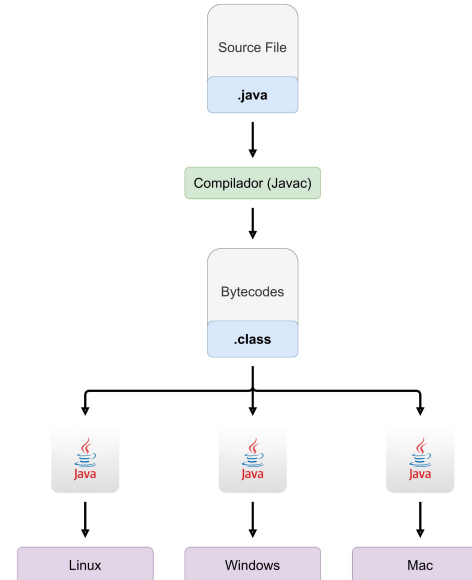
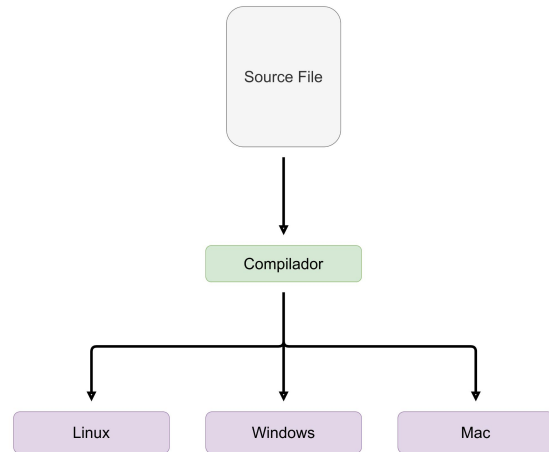
# Entendendo a JVM

## Características:

- Execução de pilhas
- Gerenciamento de memória
- Gerenciamento de threads
- Otimização de código (Compilação JIT - Just In Time)
- Garbage Collector (GC)

# Entendendo a JVM

Diferença Java para outras linguagens compiladas:



# Entendendo a JVM

Quais as diferenças entre JRE e JDK?

- **JRE (Java Runtime Environment)** - responsável por executar os programas em Java.
- **JDK (Java Development Kit)** - utilitários que permite o desenvolvimento de programas em Java. Já possui a JVM para executar os programas.

# Entendendo a JVM

Quais os tipos de plataformas Java?

- **Java SE (Java Standard Edition)** - contém as especificações do Java e pode ser implementado por diversas empresas como Oracle, OpenJDK, Azul Zulu, etc.
- **Java EE (Java Enterprise Edition)** - contém todas as especificações do Java SE e um número de programas úteis para execução em servidores. Em 2019 foi renomeado para Jakarta EE.
- **Java ME (Java Micro Edition)** - contém especificações para desenvolvimento de programas para dispositivos pequenos como celulares, PDAs, etc.



# Entendendo a JVM

Quais as implementações Java SE?

- **OpenJDK**
- Oracle JDK
- AdoptOpenJDK
- Amazon Corretto
- GraalVM CE
- Azul Zulu

Aula 1 | Etapa 2:

# Instalando a JVM

Introduction to Java Ecosystem &  
Documentation

# Instalando a JVM

Instalando curl:



```
sudo apt install curl
```



# Instalando a JVM

Instalando Java Version Manager:

- <https://github.com/shyiko/jabba>



```
curl -sL https://github.com/shyiko/jabba/raw/master/install.sh | bash && . ~/.jabba/jabba.sh
```

# Instalando a JVM

Utilizando Java Version Manager:



```
jabba ls-remote
```

```
jabba install openjdk@1.11
```

```
jabba use openjdk@1.11
```

```
java -version
```

**Aula 1 | Etapa 3:**

# **Executando um programa Java**

Introduction to Java Ecosystem &  
Documentation

# Executando um programa Java

Criando classe principal (src/main/java/com/dio/BankApi.java):

```
package com.dio;

import com.dio.entity.User;
import com.dio.service.StatementService;

import java.util.UUID;

public class BankApi {

    public static void main(String[] args) {
        final User userA = new User(UUID.randomUUID(), "Usuario 1");
        final User userB = new User(UUID.randomUUID(), "Usuario 2");
        final StatementService statementService = new StatementService(userA, userB);

        statementService.findAllByUser(userA).forEach(System.out::println);
        statementService.findAllByUser(userB).forEach(System.out::println);
    }
}
```

# Executando um programa Java

Compilando:



```
jabba use openjdk@1.11
```

```
javac -d target/ -sourcepath src/main/java/ src/main/java/com/dio/BankApi.java
```

# Executando um programa Java

Executando:



```
cd target/
```

```
java com.dio.BankApi
```

# Dúvidas?

- > Fórum do curso
- > Comunidade online (discord)