



CAPACITAÇÕES  
**JAVA PARA  
A PDPJ-Br**

# FICHA DO CURSO

JAVA BÁSICO

40 HORAS – MODALIDADE EaD - AUTOINSTRUCIONAL





## INTRODUÇÃO

O curso possui diversos recursos e propostas de atividades que irão contribuir com o processo de aprendizagem do(a) cursista.

Nesta Ficha de Curso, serão apresentados os conteúdos a serem estudados; o ambiente que apoiará o desenvolvimento do curso; os materiais de estudo; e outros pontos importantes para os(as) cursistas.

## 1. RESUMO DO CURSO

Informações gerais sobre o curso:

|                               |                                                                                                                                                                      |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>CURSO</b>                  | Java Básico                                                                                                                                                          |
| <b>MODALIDADE</b>             | Educação a Distância - EaD / Autoinstrucional                                                                                                                        |
| <b>CARGA HORÁRIA DO CURSO</b> | 40 horas                                                                                                                                                             |
| <b>PRÉ-REQUISITOS</b>         | Noções básicas de informática: navegar pela internet; realizar o download e a instalação de programas; e utilizar as funcionalidades básicas do sistema operacional. |

## 2. OBJETIVOS

### OBJETIVO GERAL

Ao final do curso, espera-se que o(a) cursista esteja habilitado(a) a desenvolver e manter soluções de software, utilizando a linguagem de programação Java, tornando-o(a) apto(a) a apoiar a implantação e utilização da Plataforma Digital do Poder Judiciário (PDPJ-Br), no seu Tribunal.

## OBJETIVOS ESPECÍFICOS

Ao final do curso, o(a) cursista deverá demonstrar capacidade no uso dos conceitos básicos da linguagem de programação Java, conhecendo, entre outros:

- a) Configuração de um ambiente de desenvolvimento Java;
- b) Programação Java orientada a objetos avançada;
- c) Programação funcional em Java (expressões lambda);
- d) Utilização dos principais pacotes da implementação Java Standard Edition;
- e) Depuração de código.

## 3. PROGRAMA

O programa foi desenvolvido de forma a atender os objetivos do curso. Ele será abordado de forma contextualizada, convergente e aplicada à prática.

| Unidades Didáticas (UD)                                                                                                 | Conteúdo                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. Conceitos de programação, fundamentos da linguagem de programação Java e configuração do ambiente de desenvolvimento | 1.1. Programas de computador e algoritmos<br>1.2. Linguagens de programação<br>1.3. Compilação, interpretação, código fonte, código objeto e máquinas virtuais<br>1.4. Histórico da linguagem de programação Java<br>1.5. Estrutura de uma aplicação Java<br>1.6. Ambientes integrados de desenvolvimento (IDE – Integrated Development Environments)<br>1.6.1. Eclipse IDE<br>1.6.2. Visual Studio Code<br>1.7. Máquina Virtual do Java (JVM – Java Virtual Machine)<br>1.8. Instalação do Java<br>1.8.1. Java Runtime Environment (JRE)<br>1.8.2. Java SE Development Kit (JDK) |



|                                                          |                                                                                                                                                                                                                                                                                                                               |
|----------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2. Tipos de dados, variáveis e operadores                | 2.1. Tipos de dados primitivos<br>2.2. Variáveis<br>2.3. Restrições e convenções para nomes<br>2.4. Comentários<br>2.5. Operadores<br>2.6. Expressões<br>2.7. Sequências de caracteres (Java String)<br>2.8. Dados e operações matemáticas (Java Math)<br>2.9. Dados e operações lógicas (Java Boolean)<br>2.10. Type casting |
| 3. Estruturas de controle                                | 3.1. Comandos condicionais<br>3.1.1. Comando if...else<br>3.1.2. Comando switch...case<br>3.1.3. Expressão condicional ternária<br>3.2. Comandos de iteração<br>3.2.1. Comando while...loop<br>3.2.2. Comando for...loop<br>3.2.3. Comando do...while<br>3.2.4. Comandos break e continue                                     |
| 4. Programação orientada a objetos básica                | 4.1. Classes e objetos<br>4.2. Atributos<br>4.3. Métodos<br>4.4. Palavra reservada this<br>4.5. Construtores<br>4.6. Modificadores de acesso a atributos, métodos e construtores<br>4.7. Encapsulamento: getters e setters<br>4.8. Classes aninhadas                                                                          |
| 5. Operações com Datas e entrada e saída básica de dados | 5.1. Operações com Datas<br>5.2. Entrada e saída básica de dados                                                                                                                                                                                                                                                              |
| 6. Depuração de código e enumerações                     | 6.1. Depuração de código no Eclipse IDE<br>6.2. Depuração de código no Visual Studio Code<br>6.3. Enumerações                                                                                                                                                                                                                 |
| 7. Coleções, pacotes e APIs                              | 7.1. Introdução a coleções<br>7.2. Collection, Set, List e Map<br>7.3. Iteradores<br>7.4. Comando de iteração for each.<br>7.5. Pacotes e APIs                                                                                                                                                                                |
| 8. Operações com arquivos                                | 8.1. Operações com arquivos                                                                                                                                                                                                                                                                                                   |
| 9. Gerência de memória e recursividade                   | 9.1. Alocação de memória<br>9.2. Garbage collector e regras de escopo<br>9.3. Recursividade                                                                                                                                                                                                                                   |

|                                              |                                                                                                                                                                                                                                                                                                                                                                            |
|----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 10. Programação orientada a objetos avançada | 10.1. Herança<br>10.2. Upcasting e downcasting<br>10.3. Sobrecarga<br>10.4. Sobreposição e anotação @Override<br>10.5. Palavra reservada super<br>10.6. Modificadores de classes, métodos e atributos<br>10.7. Classes abstratas<br>10.8. Métodos abstratos<br>10.9. Polimorfismo<br>10.10. Delegação, composição e agregação<br>10.11. Boxing, unboxing e classes wrapper |
| 11. Interfaces e tratamento de exceções      | 11.1. Introdução a interfaces<br>11.2. Herança versus interfaces<br>11.3. Métodos Default<br>11.4. Interfaces de grande utilidade<br>11.5. Estrutura try...catch<br>11.6. Comando finally<br>11.7. Exceções personalizadas (throw)<br>11.8. Pilha de chamadas de métodos (stack trace)                                                                                     |
| 12. Generics e expressões Lambda             | 12.1. Generics<br>12.2. Expressões Lambda                                                                                                                                                                                                                                                                                                                                  |

#### 4. PROCEDIMENTOS DIDÁTICOS E CRONOGRAMA DO CURSO

Os procedimentos didáticos (recursos para aprendizagem) que serão adotados no curso envolvem material de estudo, videoaulas de aulas de imersão e aulas práticas e materiais complementares.

O curso é dividido em 4 (quatro) Trilhas de Aprendizagem, com carga horária de 10 horas cada.

Na Trilha de Aprendizagem, estarão disponíveis as informações dos conteúdos que serão abordados, os objetivos de aprendizagem, o e-book, o caderno de atividades, as videoaulas e os materiais complementares, que serão apresentados a seguir.

O curso foi previamente ofertado na modalidade síncrona a um público selecionado do Poder Judiciário. A presente versão encontra-se no formato autoinstrucional, não havendo, portanto, interação com um tutor e/ou com outros/as cursistas. O prazo para a conclusão das atividades e do curso, como um todo, dependerá do empenho, dedicação e

disponibilidade do/a cursista.

A cada semana recomenda-se que os/as cursistas desenvolvam as seguintes atividades:

- a) Estudar o material disponibilizado na trilha como forma de preparo para a aula;
- b) Assistir às videoaulas programadas para a semana;
- c) Realizar as atividades práticas propostas no material de estudo;
- d) Reservar um horário para os estudo individual dos materiais indicados no ambiente do curso;
- e) Realizar as atividades avaliativas, respondendo aos questionários aplicados.

## 5. CARGA HORÁRIA DO/A CURSISTA

A carga horária do curso é dividida em trilhas de aprendizagem, no total de 04 (quatro) trilhas, compondo 40 horas totais de curso.

A Tabela 1, a seguir, apresenta a distribuição da carga horária semanal total aproximada do(a) cursista, de 10 horas, considerando as atividades semanais previstas na vigência do curso.

Tabela 1 – Carga horária semanal do(a) cursista (aproximadamente)

| <b>Atividade semanal</b>               | <b>Carga horária (horas)</b> | <b>Fração (%)</b> |
|----------------------------------------|------------------------------|-------------------|
| Estudo individual de conteúdo pré-aula | 01:00                        | 10,0%             |
| Assistir às videoaulas                 | 06:00                        | 60,0%             |
| Estudo individual pós-aula             | 01:00                        | 10,0%             |
| Atividades Avaliativas                 | 02:00                        | 20,0%             |
| <b>Total</b>                           | <b>10:00</b>                 | <b>100%</b>       |



## 6. ATIVIDADES AVALIATIVAS

A atividade avaliativa será aplicada em forma de questionário com questões de múltipla escolha. As questões serão contextualizadas e abordarão os temas de estudo da trilha. Para cada trilha será aplicada 01 avaliação que é composta por 05 questões, totalizando, ao final das 04 trilhas, 20 (vinte) questões.

O(a) cursista poderá avançar nas demais trilhas sem a necessidade de concluir as atividades avaliativas, porém o Certificado de Conclusão do Curso só será emitido para os cursistas que concluírem as atividades avaliativas com aproveitamento mínimo de **70%**, ou seja, o mínimo de 14 questões corretas das 20 questões propostas.

O questionário permitirá **03 tentativas**, possibilitando ao cursista conseguir a verificação do aproveitamento necessário para a conclusão do curso. Para fins de cálculo de aproveitamento será considerada a nota maior das tentativas realizadas.

## 7. AVALIAÇÃO DE REAÇÃO

Ao final do curso será aplicada uma avaliação de reação, por meio da qual os(as) cursistas responderão a um formulário de reação com questões relativas ao curso, seu conteúdo e didática do professor, com o intuito de avaliar a percepção dos(as) cursistas quanto ao curso, incluindo: materiais disponibilizados, qualidade das videoaulas, carga horária e demais recursos pedagógicos oferecidos pelo curso.

## 8. INFORMAÇÕES SOBRE O CONTEUDISTA

### **Professor Mestre Ronaldo Pinheiro Gonçalves Junior**

Possui graduação em Ciência da Computação (2011) e mestrado em Informática Aplicada (2015) ambos pela Universidade de Fortaleza.

Atualmente, estuda Engenharia de Software pela The University of Texas at Dallas (UTD) para a obtenção do título de PhD. Tem experiência em desenvolvimento Full Stack de aplicativos web e para nuvem. Atua como docente do Centro de Ciências Tecnológicas da Universidade de Fortaleza e participa de projetos com foco nas áreas de Computação em Nuvem, Engenharia de Software e Sistemas de Recomendação. Trabalha como professor conteudista e orientador na modalidade EAD e utiliza a linguagem Java em disciplinas EAD e presenciais, como Projeto e Arquitetura de Sistemas, Ambiente de Dados, Estrutura de Dados, Construção e Análise de Algoritmos, entre outras. Atuou como assistente de ensino (2019) na disciplina de pós-graduação Análise e Projeto de Sistemas Orientada a Objetos pela UTD. Trabalhou como desenvolvedor Web (2020) no Computer Science Mentor Center e posteriormente como gerente de banco de dados e gerente de projeto (2021), realizando atividades de desenvolvimento ágil. Possui Certificação Java - Certified Java Programmer for the Java 2 Platform SE (SCJP) e Certificação Java para a Web - Certified Web Component Developer for the Java 2 Platform EE (SCWCD) pela Sun Microsystems.

## 9. REFERÊNCIAS BIBLIOGRÁFICAS

CARNELL, John. **Spring Microservices in Action**. Manning Publications, 2021.

LECHETA, Ricardo R. **Web Services RESTful**. Novatec, 2015.

LUKSA, Marko. **Kubernetes in Action**. Manning Publications, 2018.

MACHADO, Rodrigo Prestes; FRANCO, Márcia Islabão; BERTAGNOLLI, Silvia de Castro. **Desenvolvimento de software III: programação de sistemas web orientada a objetos em Java**. Bookman, 2016.

OBE, Regina O.; HSU, Leo S. **PostgreSQL: Up and Running**. O'Reilly Media, 2014.

POULTON, Nigel. **Docker Deep Dive**. CreateSpace Independent Publishing Platform, 2018.

RICHARDSON, Chris. **Microservices Patterns: With Examples in Java**.

Manning Publications, 2018.

SILVEIRA, Guilherme. **Introdução à arquitetura de design de software:** uma visão sobre a plataforma Java. Elsevier, 2012.

SILVERMAN, R. E. Git. **Guia Prático.** O'Reilly Novatec, 2013.

SONATYPE. **Maven:** The Definitive Guide. O'Reilly Media, 2018.

WALLS, Craig. **Spring Boot in Action.** Manning Publications, 2016.