



CAPACITAÇÕES
**JAVA PARA
A PDPJ-Br**

JAVA AVANÇADO

FICHA DO CURSO

Ronaldo Pinheiro Gonçalves Junior





CAPACITAÇÕES
**JAVA PARA
A PDPJ-Br**

FICHA DO CURSO

Java Avançado
40 horas
Modalidade EaD
(autoinstrucional)



1. 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

2. RESUMO DO CURSO

Informações gerais sobre o curso:

CURSO	Java Avançado
MODALIDADE	Educação a Distância - EaD / Autoinstrucional
CARGA HORÁRIA DO CURSO	40 horas
PRÉ-REQUISITOS	Java Básico: Tipos de dados; Estruturas de controle; Enumerações; Operações com data; Entrada e saída básica de dados.



CAPACITAÇÕES
**JAVA PARA
A PDPJ-Br**

3. OBJETIVOS

OBJETIVO GERAL DO CURSO

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.

OBJETIVO ESPECÍFICO DO CURSO

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

- a) Gestão de código-fonte e versionamento de projetos e de APIs REST;
- b) Autenticação de acesso utilizando o protocolo OAuth2;
- c) Desenvolvimento de APIs (Application Programming Interfaces) REST (Representational State Transfer) utilizando Spring Boot e Spring Cloud;
- d) Desenvolvimento de APIs REST de acesso a bancos de dados e arquivos;
- e) Desenvolvimento de APIs REST para gerência de filas de mensagens;
- f) Desenvolvimento de microsserviços;
- g) Documentação de APIs REST; e
- h) Desenvolvimento e consumo de APIs REST.

4. PROGRAMA

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



UNIDADES DIDÁTICAS (UD)	CONTEÚDO
1. Introdução ao Spring Boot e Spring Cloud e configuração do ambiente de desenvolvimento	1.1. O que é o framework Spring 1.2. O que é o Spring Boot 1.3. O que é o Spring Cloud 1.4. Ambientes integrados de desenvolvimento (IDE – Integrated Development Environments) 1.4.1. Eclipse IDE 1.4.2. Visual Studio Code 1.5. Máquina Virtual do Java (JVM – Java Virtual Machine) 1.6. Instalação do Java 1.6.1. Java Runtime Environment (JRE) 1.6.2. Java SE Development Kit (JDK)
2. Criação de um projeto Spring e gerência de build e integration utilizando Apache Maven	2.1. Criação de um projeto Spring 2.2. Instalação e configuração 2.3. Configuração de projeto e dependências
3. Gerência e versionamento de código-fonte utilizando Git	3.1. Git, Github e Gitlab 3.2. Instalação do Git 3.3. Conexão ao Github ou Gitlab 3.4. Comandos de básicos 3.4.1. git init 3.4.2. git add 3.4.3. git rm 3.4.4. git commit 3.4.5. git status 3.5. Comandos avançados 3.5.1. git clone 3.5.2. git pull 3.5.3. git push 3.5.4. git fetch 3.5.5. git log 3.5.6. git tag 3.6. Trabalhando com branches 3.6.1. git branch 3.6.2. git checkout 3.6.3. git merge 3.6.4. git stash e git pop 3.6.5. git reset e git clean 3.6.6. git cherry-pick



	<ul style="list-style-type: none">3.6.7. git rebase3.7. Trabalhando com o git-flow<ul style="list-style-type: none">3.7.1. git-flow init3.7.2. git-flow feature3.7.3. git-flow release3.7.4. git-flow hotfix
4. Servidor de aplicações WildFly e injeção de dependência	<ul style="list-style-type: none">4.1. Instalação e configuração4.2. Configuração de projeto para execução no WildFly4.3. Conceito e aplicações da injeção de dependência4.4. Como utilizar injeção de dependência com Spring Boot
5. Desenvolvimento de APIs REST utilizando Spring Boot e Spring Cloud	<ul style="list-style-type: none">5.1. O que são web services e APIs?5.2. Web services SOAP x REST5.3. Protocolo HTTP (HyperText Transfer Protocol)<ul style="list-style-type: none">5.3.1. Request e Response5.3.2. Verbos HTTP5.3.3. HTTP Status Codes5.3.4. HTTP Headers5.4. Publicação de endpoints5.5. Controllers e Services5.6. Verbos HTTP e Status Codes<ul style="list-style-type: none">5.6.1. Recuperação de dados (GET)5.6.2. Inserção de dados (POST)5.6.3. Atualização de dados (PUT e PATCH)5.6.4. Exclusão de dados (DELETE)5.7. Configuração do CORS (Cross-Origin Resource Sharing)5.8. Tratamento de exceções
6. Autenticação e operações com arquivos	<ul style="list-style-type: none">6.1. O protocolo OAUTH26.2. Autenticação com Spring Security6.3. Autenticação com Spring Security e Keycloak6.4. Upload de arquivos6.5. Upload de múltiplos arquivos6.6. Download de arquivos6.7. Tratamento de exceções



7. Sistema de Gerência de Bancos de Dados (SGBD) PostgreSQL	7.1. Criação de databases e schemas 7.2. A linguagem SQL (Structured Query Language) do PostgreSQL 7.3. SQL DDL (Data Definition Language) 7.4. SQL DQL (Data Query Language) 7.5. SQL DML (Data Manipulation Language)
8. Acesso a bancos de dados e versionamento de bancos de dados utilizando FlyWay	8.1. Criação de Datasource e conexão a uma instância de SGBD 8.2. Persistência de dados utilizando Spring Data JPA, Hibernate e JDBC 8.3. Utilização do Repository 8.4. Entities e DTO (Data Transfer Objects) utilizando MapStruct 8.5. DAO (Data Access Object) 8.5.1. Recuperação de dados (GET) 8.5.2. Utilização de Query Parameters 8.5.3. Paginação de resultados 8.5.4. Ordenação de resultados 8.5.5. Inserção de dados (POST) 8.5.6. Atualização de dados (PUT e PATCH) 8.5.7. Exclusão de dados (DELETE) 8.6. Validação de dados 8.7. Gerência de transações de banco de dados 8.8. Tratamento de exceções 8.9. Operação via linha de comando 8.10. Operação via Apache Maven
9. Arquitetura de Microsserviços e processamento de filas de mensagens utilizando Rabbit MQ	9.1. O que são microsserviços? 9.2. Princípios fundamentais do projeto de microsserviços 9.3. Configuração 9.4. Filas Producer e Consumer 9.5. Reprocessamento de filas



10. Execução de APIs REST utilizando Docker Containers	10.1. O que é Docker? 10.2. Containers x Virtual Machines (VMs) 10.3. Instalação e configuração 10.4. Execução de containers 10.5. Criação de imagens utilizando Dockerfile 10.6. Criação de imagens utilizando Docker Compose 10.7. Repositórios de imagens (Docker Hub e Docker Registries)
11. Documentação de APIs REST e versionamento de APIs REST	11.1. HATEOAS 11.2. Documentação Swagger (Open API) 11.3. Gerando automaticamente a documentação Open API 11.4. Ferramenta Postman 11.5. O que é versionamento semântico? 11.6. Alternativas de implementação de versionamento de REST APIs
12. Criação e manutenção de logs de execução utilizando Apache Log4j2 e execução de APIs REST utilizando Amazon Elastic Kubernetes Service (Amazon EKS)	12.1. Criação e manutenção de logs de execução utilizando Apache Log4j2 12.2. O que é Kubernetes? 12.3. Instalação e configuração 12.4. Criação e manutenção de clusters EKS 12.5. Criação e manutenção de recursos computacionais (nodes e Fargate) 12.6. Configurações de armazenamento, rede e segurança 12.7. Gerência de clusters



5. PROCEDIMENTOS DIDÁTICOS

Os procedimentos didáticos (recursos para aprendizagem) que serão adotados no curso envolvem material de estudo, videoaulas 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.

MATERIAL DE ESTUDO

O material de estudo é composto por um **e-book** produzido especificamente para este curso, com linguagem dialógica, de fácil entendimento e com exemplos, associando teoria e prática; um **caderno de atividades** para o(a) cursista experienciar a abordagem prática do curso; e **materiais complementares** selecionados pelo professor, para colaborar com o aprendizado e ampliá-lo.

Para cada trilha, serão disponibilizados um e-book e um caderno de atividades, que poderão ser acessados na forma virtual, diretamente na Web, ou off-line, realizando o download dos materiais.

VIDEOAULA PRÁTICA

Nas videoaulas práticas, o(a) cursista experimentará o conteúdo por meio da criação de artefatos funcionais, similar a um tutorial de ensino. Os artefatos propostos terão caráter prático e utilitário, sempre que possível ligados às práticas do Poder Judiciário brasileiro.



CAPACITAÇÕES
**JAVA PARA
A PDPJ-Br**

EXPERIÊNCIA ORACLE

Como forma de ampliar os recursos de estudos, o ambiente virtual do curso conta com um espaço exclusivo de introdução à plataforma Oracle.

Esse recurso é adicional aos estudos. Recomendamos que o(a) cursista reserve um tempo para conhecer a plataforma Oracle e aprofundar os seus conhecimentos.

A plataforma Oracle será trabalhada da seguinte forma:

Aulas autoinstrucionais: Os (As) cursistas encontrarão, nos materiais, referências a conteúdos extras ligados à Oracle University.

ATIVIDADE AVALIATIVA

A atividade avaliativa é um componente obrigatório para a conclusão do curso e, para cada trilha, será aplicada uma avaliação para verificação do conhecimento adquirido.

A avaliação proposta é alinhada com o objetivo de cada Trilha de Aprendizagem e aplicada, por meio de questionário, com questões de múltipla escolha.

ORIENTAÇÕES GERAIS

De maneira resumida, recomendamos ao(à) cursista:

- a) Estudar o material disponibilizado na trilha;
- b) Realizar as atividades práticas propostas no material de estudo;
- c) Reservar um horário para o estudo individual dos materiais indicados no ambiente do curso;
- d) Realizar as atividades avaliativas, respondendo aos questionários aplicados.



CAPACITAÇÕES
**JAVA PARA
A PDPJ-Br**

6. PLATAFORMA EAD

O curso será desenvolvido a partir do ambiente virtual e outras plataformas serão usadas como apoio ao processo de ensino e de aprendizagem durante o curso. Plataformas que serão utilizadas:

PLATAFORMA EAD - AMBIENTE VIRTUAL

Os (As) cursistas terão acesso aos Materiais de Estudo por meio da plataforma Moodle, do Centro de Formação e Aperfeiçoamento de Servidores do Poder Judiciário (CEAJUD).

Na plataforma, estarão todos os recursos, todas as atividades e orientações necessárias ao processo de aprendizagem do(a) cursista. É necessário que o(a) cursista desenvolva o hábito de entrar na plataforma e reserve um tempo para o estudo individual.

PLATAFORMA ORACLE (complementar)

O curso contará com um conteúdo extra chamado Oracle Experience, opcional ao(à) cursista, que conta com uma plataforma e materiais complementares. Esses materiais ajudarão o(a) cursista a aprofundar-se em assuntos de seu interesse. O material que compõe o Oracle Experience é o Oracle University.

A Oracle University é acessível através do link <https://learn.oracle.com/ols/user-portal>. Para acessar, basta o(a) cursista criar um login, verificar seu e-mail e seguir o link *Free Oracle Training*. A trilha gratuita recomendada para este projeto é a *Java Explorer*, <https://mylearn.oracle.com/ou/learning-path/java-explorer/79726> (em inglês).



CAPACITAÇÕES
**JAVA PARA
A PDPJ-Br**

7. MONITORAMENTO DE PARTICIPAÇÃO

Os (As) cursistas terão o acompanhamento da sua participação da seguinte forma:

PROGRESSO DO CURSO

No ambiente virtual, o(a) cursista contará com o recurso de progresso do curso, onde poderá consultar, a qualquer momento, o andamento do seu rendimento.

O progresso do curso mostrará todas as atividades a serem realizadas pelo(a) cursista e, conforme sejam concluídas, estas serão atualizadas automaticamente na barra de progresso.

O(A) cursista consegue, em tempo real, saber os recursos e as atividades já concluídas e também as que necessitam ser vistas, estudadas e/ou realizadas

8. REQUISITOS TÉCNICOS

É necessário que o(a) cursista tenha conhecimentos básicos de programação e noções básicas de informática, como: navegar pela internet; realizar o download e a instalação de programas; e utilizar as funcionalidades básicas do sistema operacional.

Nos primeiros momentos, o curso contempla os principais conceitos básicos de programação e fundamentos da linguagem de programação Java. É importante ressaltar que o sucesso no aprendizado depende do entendimento da proposta do curso EaD e do comprometimento com o processo.



9. CARGA HORÁRIA DO CURSO

A carga horária do curso é dividida em trilhas de aprendizagem, no total de 4 (quatro) trilhas, compondo 40 horas totais de curso. A distribuição dos recursos e das atividades acompanha a dinâmica da trilha, que é composta por:

1. Material de estudo (e-book / caderno de atividade / material complementar)
2. Videoaula Prática
3. Experiência Oracle
4. Atividade Avaliativa

Cada trilha tem carga horária de 10 horas, distribuídas da seguinte forma:

Atividade	Carga Horária (trilha)
Estudo do Conteúdo	3h
Estudo da Prática	3h
Videoaula Prática	3h
Experiência Oracle	opcional
Atividade Avaliativa	1h

10. AVALIAÇÃO DE DESEMPENHO

A proposta metodológica dos cursos está apoiada no estudo dos conteúdos em trilhas de aprendizagem, no total de 4 (quatro) trilhas por curso. Para cada trilha, o(a) cursista terá 1 (uma) atividade avaliativa, associada aos temas em



CAPACITAÇÕES
**JAVA PARA
A PDPJ-Br**

estudo.

A atividade avaliativa será aplicada em forma de questionário, com questões de múltipla escolha contextualizadas, que abordarão os temas de estudo da trilha. Para cada trilha, serão aplicadas 5 (cinco) questões de múltipla escolha, totalizando 20 (vinte) questões no curso.

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

O questionário permitirá 2 (duas) tentativas, possibilitando ao(à) 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.

11. AVALIAÇÃO DE REAÇÃO

Como forma de verificar a qualidade e o impacto da aplicação do curso aos(às) cursistas, será realizada uma avaliação de reação, em forma de questionário, com questões de múltipla escolha.

12. INFORMAÇÕES SOBRE O PROFESSOR

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, na 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



CAPACITAÇÕES
**JAVA PARA
A PDPJ-Br**

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.

13. 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, Sílvia 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



CAPACITAÇÕES
**JAVA PARA
A PDPJ-Br**

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.