



PROJETO PEDAGÓGICO DE CURSO
Pós-Graduação *Lato Sensu*
Tecnologia Java

Sumário

1. Nome do Curso e Área do Conhecimento.....	3
2. Características Técnicas do Curso	3
3. Público Alvo	4
4. Critérios de Seleção.....	4
5. Justificativa do Curso.....	4
6. Objetivos do Curso	5
7. Competências e Habilidades do Egresso.....	6
8. Metodologia de Ensino e Aprendizagem	6
Composição do Corpo Docente	9
9. Estágio Não Obrigatório	9
10. Matriz Curricular	10
11. Carga Horária.....	11
12. Conteúdo programático	11
13. Infraestrutura Física e Pedagógica	27

1. Nome do Curso e Área do Conhecimento

Nome do Curso: Tecnologia Java

Área de Avaliação (CAPES): CIÊNCIA DA COMPUTAÇÃO

Grande Área (CAPES): 10000003 CIÊNCIAS EXATAS E DA TERRA

Área do Conhecimento (CAPES): 10300007 CIÊNCIA DA COMPUTAÇÃO

Classificação OCDE: Computação e Tecnologias da Informação e Comunicação (TIC)

2. Características Técnicas do Curso

Modalidade: Educação a Distância

Número máximo de vagas por Polo/Unidade: 1000 alunos

Período de Oferecimento: O curso possui entrada intermitente, respeitadas as datas de início e de fim cadastradas na oferta, bem como observado o período indicado para a sua integralização.

Limitações legais

Resolução CNE/CES Nº 1, de 06 de abril de 2018, que estabelece normas para o funcionamento de cursos de pós-graduação lato sensu.

O candidato deverá ser graduado com diploma devidamente registrado segundo as normas estabelecidas pelo MEC.

3. Público Alvo

O curso é destinado a graduados com conhecimentos em programação e tecnologia, e para profissionais de outras áreas que desejam adquirir habilidades e competências para atuar com soluções de arquitetura em JAVA.

4. Critérios de Seleção

O ingresso na pós-graduação será realizado por meio de inscrição no portal da Instituição e entrega dos documentos solicitados. Em seguida, analisados pela área competente.

5. Justificativa do Curso

Atualmente em nosso cotidiano, utilizamos distintas plataformas nos diversos dispositivos que usamos. Em um desktop geralmente são utilizadas plataformas específicas, como também nos dispositivos móveis como os *smartphones* e *tablets*. Mesmo com os avanços rápidos das tecnologias, dos próprios aplicativos como também das linguagens de programação, seja uma inovação ou atualizações, nos quais também não podemos deixar de considerar as modernizações dos sistemas operacionais. Diante desse cenário bastante mutável, é importante o uso de linguagem que possa ser utilizada nas diferentes plataformas, e sendo assim, serem visualizadas nos diversos dispositivos, mesmo que sejam inovações ou atualizações. A linguagem JAVA que foi criada em 1990, até na atualidade cumpre esse papel de ser multiplataforma, pode ser utilizada em sistemas operacionais distintos como, por exemplo, Windows, Linux, Android. Todos esses benefícios fazem com que a linguagem JAVA seja conhecida em todo o mundo, e uma necessidade para os profissionais que trabalham com desenvolvimento possuir conhecimentos específicos.

6. Objetivos do Curso

6.1. Objetivo Geral

- Estabelecer conhecimentos com a arquitetura de desenvolvimento nas tecnologias usadas na área de desenvolvimento para desktop e Web.
- Entender o uso de linguagens de programação Javascript.
- Compreender a arquitetura de interfaces gráfica para aplicações multiplataformas em Java.
- Gerenciar e aplicar frameworks usados na arquitetura de projetos Java Multiplataformas.
- Conhecer a arquitetura de aplicativos em Java Android utilizando recursos do dispositivo.

6.2. Objetivos Específicos

- Conhecer software orientado a objetos.
- Identificar linguagem e padrões web.
- Construir projetos com arquitetura de desenvolvimento Java Web.
- Construir projetos avançados em Java.
- Conhecer a Enterprise Java Beans (JEE).
- Identificar a entrega e Implantação contínua (DevOps).
- Conhecer Design Patterns e gerência de configuração
- Dominar interface de programação de aplicações (API) e Web Services.
- Conhecer o desenvolvimento de aplicações mobile no sistema operacional Android.

7. Competências e Habilidades do Egresso

Competências

Atuar no desenvolvimento de projetos multiplataformas de maneira integrada, dos diversos sistemas operacionais como Windows da Microsoft, Android da Google, ou Linux. Nos quais podem ser visualizados nos diversos dispositivos como desktops (computadores pessoais e corporativos), notebooks, dispositivos móveis como *smatphones* e *tablets*.

Habilidades

- Identificar linguagem e padrões web e software orientado a objetos.
- Criar projetos com arquitetura de desenvolvimento Java Web, simples e complexos.
- Utilizar a Enterprise Java Beans (JEE) e DevOps.
- Aplicar o Design Patterns e gerência de configuração.
- Utilizar interface de programação de aplicações (API) e Web Services.
- Criar aplicações mobile no sistema operacional Android.

8. Metodologia de Ensino e Aprendizagem

O desenvolvimento das disciplinas do curso se dará no ambiente virtual, onde o aluno cumprirá 40 horas por disciplina.

Estas 40 horas são compostas pelo estudo dos diversos insumos pedagógicos disponibilizados, como:

- ✓ materiais de leitura;
- ✓ videoaulas;
- ✓ slides;
- ✓ podcasts;

- ✓ indicações de leituras extras como artigos e capítulos de livros.

Como parte do modelo acadêmico, também contabilizamos:

- ✓ a interação com os tutores para esclarecimentos de dúvidas pedagógicas (via Fórum);
- ✓ a realização da avaliação disciplinar pelo aluno.

Vale ressaltar que cada aluno é único e tem seu próprio tempo para aprendizagem. Alguns conseguem assimilar o conteúdo com mais agilidade, outros precisam retomar as leituras e videoaulas para tornar sua aprendizagem efetiva.

O tempo gasto pelo aluno para o entendimento de cada disciplina depende também da sua familiaridade com o tema, ou seja, o conhecimento prévio que o aluno tem em relação a temática abordada.

No ambiente virtual, o aluno encontrará o conteúdo das disciplinas, organizados em temas/webaulas.

Para cada um deles, o aluno realizará um conjunto de atividades baseadas em leitura de textos de fundamentação teórica e acesso a recursos audiovisuais.

Um tutor apoiará as atividades realizadas no ambiente virtual, atendendo o aluno nas suas dúvidas por meio de ferramentas de comunicação.

O aluno, ao iniciar os seus estudos, terá um encontro presencial para acolhida/ambientação; esse encontro terá como objetivos:

- ✓ Integrar o aluno ao curso de Pós-Graduação.
- ✓ Dialogar e esclarecer as dúvidas sobre a proposta pedagógica do curso e as regras acadêmicas.
- ✓ Apresentar ao aluno o Ambiente Virtual de Aprendizagem (o primeiro acesso; o envio de documentos; os serviços de secretaria e financeiro; a disciplina Ambientação; a tutoria online; o boletim acadêmico; as disciplinas e seus conteúdos; a biblioteca virtual; entre outros).
- ✓ Proporcionar um momento de Network aos pós-graduandos.

Avaliação do Desempenho do Aluno

O aluno deverá realizar as atividades propostas no ambiente virtual. A realização das atividades irá compor sua frequência no curso, que será considerada para a sua aprovação.

A atividade avaliativa que o aluno realizará para compor a sua média é a Avaliação Virtual (AV); essa atividade é obrigatória e estará disponível no Ambiente Virtual de Aprendizagem – AVA, conforme cronograma de seu curso.

Para a aprovação em cada uma das disciplinas, o aluno deverá obter frequência de, no mínimo, 75% (setenta e cinco por cento) e nota igual ou superior a 7,0 (sete).

As notas devem ser expressas no intervalo de 0 (zero) a 10 (dez).

O aluno que obtiver média inferior a 7,0 (sete) nas disciplinas terá direito ao Programa de Dependência e Recuperação – PDR, mediante a solicitação de requerimento e respeitando o período de jubramento do curso.

O PDR será realizado no ambiente virtual de aprendizagem, sendo que o aluno terá acesso ao conteúdo da disciplina e realizará uma Avaliação Virtual - AV, e a nota obtida substituirá a média do aluno.

Para a obtenção do **Certificado** de Pós-graduação *Lato Sensu* – especialização, o aluno deverá cumprir todas as condições seguintes:

- ✓ Frequência mínima de 75% (setenta e cinco por cento) em todas as disciplinas;
- ✓ Nota igual ou superior a 7,0 (sete) em todas as disciplinas.

Certificação

O Certificado de conclusão de curso de Especialização será acompanhado por histórico escolar, em cumprimento às exigências da Resolução CNE/CES nº1, de 06 de abril de 2018, da Câmara de Educação Superior do Conselho Nacional de Educação.

Composição do Corpo Docente

O corpo docente do curso é constituído por profissionais qualificados, com comprovado saber em sua área de atuação, conforme Resolução CNE/CES nº1, de 06 de abril de 2018, sendo integrado, no mínimo, por 30% (trinta por cento) de portadores de título de pós-graduação *stricto sensu*, isto é, portadores de títulos de Mestrado e Doutorado, obtidos em programas de pós-graduação *stricto sensu* devidamente reconhecidos pelo poder público em território nacional, ou revalidados, conforme legislação vigente. Os demais docentes são certificados em nível de especialização, pós-graduação *lato sensu*, de reconhecida capacidade técnico-profissional.

9. Estágio Não Obrigatório

O estágio curricular não obrigatório tem como finalidade estimular o aluno a desenvolver atividades extracurriculares, para que possa inter-relacionar os conhecimentos teóricos e práticos adquiridos durante o curso e aplicá-los na solução de problemas reais da profissão, proporcionando o desenvolvimento da análise crítica e reflexiva para os problemas socioeconômicos do país, de acordo com a Resolução de Estágio curricular não obrigatório vigente na Instituição.

Os principais objetivos da prática do estágio curricular não obrigatório são:

- I. proporcionar o exercício do aprendizado comprometido com a realidade socioeconômica-política do país;
- II. propiciar a realização de experiências de ensino e aprendizagem visando à educação profissional continuada, alicerçada no desenvolvimento de competências e habilidades e ao exercício do pensamento reflexivo e criativo; e
- II. incentivar o trabalho de pesquisa e investigação científica, visando ao desenvolvimento da ciência, da tecnologia e da cultura.

A carga horária é definida pela concedente de estágio, não podendo ultrapassar a carga horária máxima de 6 (seis) horas diárias e 30 (trinta) horas semanais, as quais podem ser realizadas em empresas públicas ou privadas, instituição de pesquisa, órgãos governamentais e não governamentais, e as próprias unidades da Universidade, desde que sigam às condições adequadas para que o estagiário possa aprofundar os seus conhecimentos teóricos e práticos adquiridos no curso.

Para o Curso de Pós-Graduação EAD, a prática do estágio curricular não obrigatório é permitida durante a vigência do curso, não podendo exceder em um mesmo campo de estágio o período de 2 (dois) anos.

Os estágios curriculares não obrigatórios devem estar apoiados em Termo de Compromisso e de comum acordo com a Instituição, devendo explicitar não somente os aspectos legais específicos, como também os aspectos educacionais e de compromisso com a realidade social.

O Planejamento do Estágio Curricular Não Obrigatório é de responsabilidade do coordenador de curso/professor orientador e também do Departamento de Estágios.

10. Matriz Curricular

DISCIPLINAS	CH PRÁTICA	CH TEÓRICA	CH TOTAL
Ambientação	0 h	0 h	0 h
Linguagem e padrões web	0 h	40 h	40 h
Programação orientada a objetos	0 h	40 h	40 h
Desenvolvimento Java Web	0 h	40 h	40 h
Desenvolvimento avançado em Java	0 h	40 h	40 h
Enterprise Java Beans (JEE)	0 h	40 h	40 h
Entrega e Implantação contínua (DevOps)	0 h	40 h	40 h
Design Patterns e gerência de configuração: do projeto ao controle de versões	0 h	40 h	40 h
Interface de programação de aplicações (API) e Web Services	0 h	40 h	40 h
Desenvolvimento de aplicações mobile - Android	0 h	40 h	40 h

11. Carga Horária

A carga horária de 360h constitui o conteúdo ministrado em 9 (nove) disciplinas.

12. Conteúdo programático

Disciplina: Ambientação

Ementa: Ensino a distância: características desta modalidade de estudo. A tecnologia e o ensino à distância. Legislação do Ensino a Distância no Brasil. Aspectos relacionados ao perfil no Ensino a Distância.

Competências e Habilidades:

- Compreender a modalidade de ensino a distância.
- Identificar uma relação entre tecnologia e educação no contexto do EAD;
- Discutir elementos da legislação do ensino a distância no Brasil;
- Refletir sobre o perfil do aluno em ensino a distância.

Conteúdo Programático 1: Introdução ao Ensino a Distância.

Conteúdo Programático 2: Tecnologia e educação.

Conteúdo Programático 3: Legislação do Ensino a Distância no Brasil.

Conteúdo Programático 4: Perfil do aluno no Ensino a Distância.

Bibliografia básica:

BRASIL. Ministério da Educação. **Lei de Diretrizes e Bases da Educação Nacional - Lei n. 9.394/96**. Brasília, 1996. Disponível em: https://www.planalto.gov.br/ccivil_03/leis/l9394.htm. Acesso em: 6 dez. 2022.

BRASIL. Ministério da Educação. **Resolução CNE/CP n. 7, de 18 de dezembro de 2018**. Brasília, 2018. Disponível em: https://normativasconselhos.mec.gov.br/normativa/view/CNE_RES_CNECESN72018.pdf. Acesso em: 6 dez. 2022.

GIL, H. A passagem da Web 1.0 para a Web 2.0 e... Web 3.0: potenciais consequências para uma 'humanização' em contexto educativo. 2014. **Boletim informativo Cybercentro Castelo Branco**. Castelo Branco, 2014. Disponível em: <https://repositorio.ipcb.pt/bitstream/10400.11/2404/1/A%20passagem%20da%20Web%20Henrique.pdf>. Acesso em: 6 dez. 2022.

MEYER, A. I. da S. (2022). Conceituando a Educação a Distância. **Revista Ibero-Americana de Humanidades, Ciências e Educação**, v. 8, n. 1, p. 590-601, [s. l.], 2022. Disponível em: <https://periodicorease.pro.br/rease/article/view/3835>. Acesso em: 6 dez. 2022.

VAZ, M. L. de L.; RIBEIRO, F.; COSTA, L. A. da. Os desafios da educação a distância on-line e a remotividade na nova engenharia educacional. **Brazilian Journal of Science**, v. 1, n. 4, p. 79-86. Goiás, 2022. Disponível em: <https://www.brazilianjournalofscience.com.br/revista/article/view/79>. Acesso em: 6 dez. 2022.

VIEIRA, D. M. L.; COSTA, L. A.; VAZ, M. L. L. Um novo olhar para a educação a distância. In: MACHADO, G. E.; COSTA, S. C.; SILVA, K. R. P. **Debates contemporâneos: perspectivas e reflexões atuais**. Santa Maria: Arco, 2021.

Bibliografia Complementar:

BAPTISTA, M. M. Internet: auxílio à educação. **BIBLOS - Revista do Instituto de Ciências Humanas e da Informação**, v. 16, p. 37- 44, [s. l.], 2004. Disponível em: <http://hdl.handle.net/20.500.11959/brapci/23717>. Acesso em: 6 dez. 2022.

MOORE, M.; KEARSLEY, G. **Educação a distância: uma visão integrada**. São Paulo: Thomson Learning, 2007.

MORAN, J. M.; MASETTO, M. T.; BEHRENS, M. A. **Novas tecnologias e mediação pedagógica**. 21. ed. Campinas: Papirus, 2013.

PIVA, D. J.; PUPO, R.; GAMEZ, L. et al. **EAD na prática**: planejamento, métodos e ambientes de educação on-line. São Paulo: Elsevier, 2011.

Disciplina: Linguagem e padrões Web

Ementa: Marcações (tags) básicas e avançadas para criação de páginas em linguagem de marcação de hipertexto (HTML); Propriedades e utilização de folhas de estilo em cascata (CSS); linguagem de programação interpretada JavaScript; criação de projetos de páginas web com boas práticas, de acordo com os padrões da W3C.

Competências e Habilidades:

- Conhecer e utilizar as marcações essenciais em HTML, na criação de páginas estáticas da Web;
- Conhecer as propriedades essenciais das folhas de estilo em cascata e seu uso, eliminando duplicidades e conflitos de declarações, bem como boas práticas na sua adoção;
- Integrar às páginas em HTML as folhas de estilos em cascata, permitindo a criação de páginas atrativas considerando conceitos de experiência do usuário;
- Conhecer a linguagem de programação interpretada JavaScript, para dinamizar funcionalidades e interações nas páginas desenvolvidas com HTML e CSS;
- Construir e Implementar páginas web de qualidade utilizando as linguagens HTML, CSS e JavaScript, de acordo com o layout da interface e seguindo os padrões recomendados pelo W3C

Conteúdo Programático 1: Dominando a linguagem de marcação de hipertexto – HTML. Estrutura básica de um documento HTML e comparações com XHTML e XML.

Conteúdo Programático 2: Aprimoramento das páginas Web: as folhas de estilo em Cascata. Evolução, gramática e convenções, herança, escopo e estrutura, *Box Model*, responsividade na Web.

Conteúdo Programático 3: Linguagem JavaScript: do básico ao avançado. Conceitos, tipos de dados e variáveis, operadores e expressões, controle de fluxo com condicionais e loops. Trabalhando com funções, manipulando o documento *Object Model*. Apresentando bibliotecas e *frameworks*.

Conteúdo Programático 4: Implementação de páginas Web robustas: integrando HTML, CSS e JavaScript.

Bibliografia Básica:

SILVA, Mauricio Samy. **HTML5: a linguagem de marcação que revolucionou a web**. São Paulo: Novatec Editora, 2019. 336p.

SILVA, Maurício Samy. **CSS3: desenvolva aplicações web profissionais com uso dos poderosos recursos de estilização das CSS3**. São Paulo: Novatec Editora, 2019. 496p.

SILVA, Maurício Samy. **JavaScript - Guia do Programador: Guia completo das funcionalidades de linguagem JavaScript**. . São Paulo: Novatec Editora, 2020. 608p.

Bibliografia Complementar:

TITTEL, Ed. e NOBLE Jeff. **HTML, XHTML e CSS Para Leigos**. Rio de Janeiro: Alta Books Editora, 2018. 412p.

LOUDON, Kyle. **Desenvolvimento de Grandes Aplicações Web: Produzindo código capaz de crescer e prosperar**. São Paulo: Novatec Editora, 2019. 328p.

DE SOUSA, Claudiany Calaça; SILVA, Ennio Willian Lima. O ensino de programação web utilizando dispositivos móveis: um relato de experiências com alunos do ensino médio integrado ao curso técnico em informática/Web programming teaching using mobile devices: a report of experiences with high school students integrated to the computer technical course. **Brazilian Journal of Development**, v. 5, n. 8, p. 13415-13425, 2019.

Disciplina: Programação orientada a objetos

Ementa: Histórico sobre a orientação a objetos, conceitos essenciais sobre a programação orientada a objetos, por meio de seus pilares: abstração, herança, polimorfismo e encapsulamento; objetos: atributos e métodos; classes e seus tipos; instanciação de objetos, encapsulamento a nível de classes, pacotes, métodos e atributos; método construtor; herança, generalização e especialização; polimorfismo com sobrecarga e sobrescrita de métodos e implementação prática de programas orientados a objetos.

Competências e Habilidades:

- Compreender os pilares da orientação a objetos, para se desenvolver softwares e aplicações modernas robustas e com qualidade;
- Ser capaz de analisar e melhorar os elementos e conceitos envolvidos no desenvolvimento orientado a objetos, empregando ainda boas práticas de programação;
- Compreender e criar projetos que integrem classes, classes abstratas e outros elementos da orientação a objetos, aplicando de modo prático a abstração, herança, polimorfismo e encapsulamento.

Conteúdo Programático 1: Histórico sobre o paradigma de orientação a objetos e os pilares que fundamentam tal paradigma no processo de desenvolvimento de software e aplicações, exemplificados utilizando a Linguagem de Modelagem Unificada (UML).

Conteúdo Programático 2: Aplicação do pilar de abstração para o projeto e criação de objetos, considerando seus atributos e métodos, expressado em classes, com ou sem herança.

Conteúdo Programático 3: Instanciação e encapsulamento de classes, pacotes, métodos e atributos em projetos orientados a objetos, utilizando ainda método construtor com declarações implícitas e explícitas em classes que implementam herança e, conseqüentemente, generalização e especialização.

Conteúdo Programático 4: Implementação de programa orientado a objetos que implementam métodos com polimorfismo de sobrecarga e sobrescrita.

Bibliografia Básica:

ALVES, WILLIAM PEREIRA. **Análise e Projeto de Sistemas**. Saraiva Educação SA, 2018.

ALVES, WILLIAM PEREIRA. **Programação Python: aprenda de forma rápida**. Saraiva Educação AS, 2021.

CHAVES, Luan Junior; DE PAULA, Nilton César. PROGRAMAÇÃO ORIENTADA A OBJETOS UTILIZANDO A LINGUAGEM JAVA. ANAIS DO SEMEX, n. 12, 2019.

GUERRA, Eduardo. **Design Patterns com Java: Projeto orientado a objetos guiado por padrões**. Editora Casa do Código, 2014.

JECK, David. **Introduction to programming using Java**. Hobart and William Smith Colleges, 2021.

MANZANO, JOSÉ AUGUSTO NG; JÚNIOR, ROBERTO AFFONSO DA COSTA. **Programação de Computadores com Java**. Saraiva Educação SA, 2014.

PRESSMAN, Roger S.; MAXIM, Bruce R. **Engenharia de software-9**. McGraw Hill Brasil, 2021.

ZIVIANI, Nivio. **Projeto de Algoritmos com Implementações em Java e c++**. Cengage Learning Edições Ltda., 2010.

Bibliografia complementar:

BARRY, Paul. **Use a cabeça! Python**. Alta Books, 2019.

GAMMA, Erich. **Padrões de projetos: soluções reutilizáveis**. Bookman editora, 2009.

H. M. Deitel, P. J. Deitel. **Java: Como Programar**, 8ª. Edição. Pearson, 2010

SIERRA, Kathy; BATES, B. **Java Use a Cabeça! Java**. 1ª Edição. 2005.

WAZLAWICK, Raul. **Engenharia de software: conceitos e práticas**. Elsevier Editora Ltda., 2019.

Disciplina: Desenvolvimento Java Web

Ementa: Introdução a Java EE; Servidores de aplicação; Introdução à tecnologia de *Servlet*; Introdução a *Java Server Pages* (JSP); Arquiteturas baseadas no

modelo MVC (*Model View Controller*); Introdução a JDBC; DAO (Data Access Object); XML; JSF (*JavaServer Faces*); API de Persistência Java (JPA) e Hibernate.

Competências e Habilidades:

- Desenvolvimento de sistemas Web utilizando a linguagem de programação Java e tecnologias relacionadas;
- Compreender as diferentes tecnologias que possuem Java como base para a implementação de aplicações web robustas;
- Identificar e Implementar soluções Java Web com diferentes recursos que permitam a conexão com bases de dados e a otimização da persistência e manipulação de dados;

Conteúdo Programático 1: Introdução às Tecnologias *Java Enterprise Edition*. Como servidores de aplicação Java funcionam e como as diferentes tecnologias Java pode ser adotadas.

Conteúdo Programático 2: *Servlets* e *Java Server Pages*

Conteúdo Programático 3: JSF, Arquiteturas MVC e Introdução a persistência de dados.

Conteúdo Programático 4: Configurando o *Hibernate* e Integração JSF e JPA.

Bibliografia Básica:

DEITEL, Paul J; DEITEL, Harvey M. **Java: Como Programar**. 10ed. São Paulo: Pearson, 2016. 968p.

KURNIAWAN, Budi; DECK, Paul. **Servlet, JSP and Spring MVC**. Quebec: Brainy Software Inc, 2015. 420p.

COELHO, Hébert. **JPA Eficaz: As melhores práticas de persistência de dados em Java**. São Paulo: Editora Casa do Código, 2014. 183p.

Bibliografia Complementar:

DUARTE, Nuno Filipe Brandão. **Frameworks e Bibliotecas Javascript**. (Mestrado em Engenharia Informática) - Instituto Politécnico do Porto. Porto, 70p., 2015.

DEITEL, Paul J; DEITEL, Harvey M. Ajax. **Rich Internet Applications e Desenvolvimento Web para Programadores**. São Paulo: Pearson Prentice Hall, 2009. 747p.

VOHRA, Deepak. **JavaServer Faces 2.0: Essential Guide for Developers**. Toronto: Nelson Education, 2014. 432p.

SIERRA, Kathy; BATES, Bert. **Use a cabeça!: Java**. Rio de Janeiro: Alta Books, 2009. 513p.

JANDL JUNIOR, Peter. **Java Guia do Programador**. 3ª Edição: Atualizado Para Java 8. 2015.

Disciplina: Desenvolvimento avançado em Java

Ementa: Abordar o desenvolvimento avançado na linguagem Java, sobre paradigmas e interfaces funcionais, incluindo conceitos de lambdas, classes anônimas, funções e operators. Novidades das atualizações do JAVA.

Competências e Habilidades:

- Entender o Paradigma Funcional no Java;
- Aprender como utilizar uma lambda e API lambda do Java 8;
- Entender o funcionamento da API de Threads do Java.
- Entender o funcionamento da função PararelStreams no Java.

Conteúdo Programático 1: Conceitos iniciais sobre estrutura de dados, arrays e registro: Entender o Paradigma Funcional no Java; aprender como utilizar uma lambda e API Lambda do Java; Entender paradigma da recursividade (Taill Call Optimization e Memorization).

Conteúdo Programático 2: Interfaces Funcionais: Entender o Paradigma Funcional no Java; aprender como utilizar uma lambda e API lambda do Java 8.

Conteúdo Programático 3: Processamento Assíncrono e Paralelo: Entender o conceito de síncrono e assíncrono; entender o funcionamento da API de Threads do Java; Entender o funcionamento da função PararelStreams no Java.

Conteúdo Programático 4: Por dentro da Modularização do Java: Entender o que é o Jigsaw. Novidades do Java 10. Novidades do Java 11.

Bibliografia Básica:

BARNES, David J.; KÖLLING, Michael. **Programação orientada a objetos com Java:** uma introdução prática utilizando o Blue J. São Paulo: Pearson Prentice Hall, 2004.

FERREIRA, Haline Kelly; ZUCHI, Jederson Donizete. **Análise Comparativa entre Frameworks Frontend baseados em JavaScript para Aplicações Web.** Taquaritinga: Revista Interface Tecnológica, v. 15, n. 2, p. 111-123, 2018.

SILVA, Maurício Samy. **JavaScript - Guia do Programador: Guia completo das funcionalidades de linguagem JavaScript.** . São Paulo: Novatec Editora, 2020. 608p.

Bibliografia Complementar:

DEITEL, Paul J; DEITEL, Harvey M. **Java: Como Programar.** 10ed. São Paulo: Pearson, 2016. 968p.

LECHETA, Ricardo R. **Web Services RESTful: Aprenda a criar web services RESTful em Java na nuvem do Google.** São Paulo: Novatec Editora, 2015. 432p.

VOHRA, Deepak. **JavaServer Faces 2.0: Essential Guide for Developers.** Toronto: Nelson Education, 2014. 432p.

Disciplina: Enterprise Java Beans (JEE)

Ementa: Enterprise Java Beans, desenvolvimento e fundamentos tecnológicos, abordando beans de sessão (state-less - sem informação de estado e state-full

- com informação de estado), beans de entidade (persistência gerenciada por bean e por contêiner), beans de mensagem, realização de transações atômicas, segurança com Java Authentication and Authorization Service – JASS. EJB na prática.

Competências e Habilidades:

- Conhecer a Enterprise Java Beans desenvolvimento e fundamentos tecnológicos;
- Dominar Bean de Sessão e Bean de Entidade relacionado à persistência gerenciada por Bean e persistência gerenciada por contêiner;
- Entender Bean de Mensagem e a realização de transações utilizando Java Transaction Service – JTS
- Identificar segurança aplicando o Java Authentication and Authorization Service – JASS

Conteúdo Programático 1: Introdução a Enterprise Java Beans; Visão arquitetural; Visão do cliente. Injeção de dependências e CDI.

Conteúdo Programático 2: Interceptadores; Stateless Sessions Beans, Stateful Session Beans; Beans de Sessão e Beans de Entidade. Timers.

Conteúdo Programático 3: Message-Driven Beans; Bean de Mensagem e transações utilizando Java Transaction Service – JTS; mensageria com JMS. Chamadas assíncronas. Exceções.

Conteúdo Programático 4: Configuração de transações; Configuração de segurança; Segurança utilizando o Java Authentication and Authorization Service – JASS; Persistência de dados na plataforma JEE. Integração com o ambiente J2EE.

Bibliografia Básica:

HUNT, Andrew. O programador pragmático [recurso eletrônico] : de aprendiz a mestre / Andrew Hunt, David Thomas ; tradução: Aldir José Coelho Corrêa da

Silva ; revisão técnica: Henrique J. Brodbeck. – Dados eletrônicos. – Porto Alegre : Bookman, 2010

ROCHA, Luiz. DESENVOLVIMENTO DE SOFTWARE PARA WEB. IESDE BRASIL S.A. 2020

SANTANA, Eduardo. Back-end Java: Microsserviços, Spring Boot e Kubernetes. Casa do Código. 2021

Bibliografia Complementar:

VOHRA, Deepak. **JavaServer Faces 2.0: Essential Guide for Developers**. Toronto: Nelson Education, 2014. 432p.

BARNES, David J.; KÖLLING, Michael. **Programação orientada a objetos com Java: uma introdução prática utilizando o Blue J**. São Paulo: Pearson Prentice Hall, 2004.

SILVA, Maurício Samy. **JavaScript - Guia do Programador: Guia completo das funcionalidades de linguagem JavaScript**. . São Paulo: Novatec Editora, 2020. 608p.

Disciplina: Entrega e Implantação contínua (DevOps)

Ementa: Fundamentos de entrega contínua e implantação contínua (DevOps). Conceitos de CI/CD. Desenvolvimento de software Distribuído. Implantação de espaços de desenvolvimento, testes, integração, produção. Desenvolvimento de espaços de integração. Gerenciamento de repositórios. Ferramentas. testes automatizados e contínuos. Gerência de configuração. Projetos com arquitetura de microsserviços. Projetos com arquitetura sob contêineres. Efeitos na entrega contínua no desenvolvimento de software.

Competências e Habilidades:

- Identificar os fundamentos de entrega contínua, implantação contínua (DevOps).

- Conhecer o desenvolvimento de software distribuído, implantação de espaços de desenvolvimento, testes, integração, produção.
- Compreender o desenvolvimento de espaços de integração e as ferramentas.
- Conhecer a gerência de configuração e efeitos na entrega contínua no desenvolvimento de software.

Conteúdo Programático 1: Introdução aos fundamentos de entrega contínua, implantação contínua (DevOps). Conceitos de CI/CD.

Conteúdo Programático 2: Desenvolvimento de software distribuído, implantação de espaços de desenvolvimento, testes, integração, produção. Desenvolvimento de espaços de integração. Gerenciamento de repositórios.

Conteúdo Programático 3: Gerenciamento de repositórios. Ferramentas, testes automatizados e contínuos. Gerência de configuração.

Conteúdo Programático 4: Projetos com arquitetura de microsserviços. Projetos com arquitetura sob contêineres. Efeitos na entrega contínua no desenvolvimento de software.

Bibliografia Básica:

MUNIZ; Antonio. Jornada DevOps. Brasport; 2ª edição, 2020

WILLIS, John. The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations. It Revolution Press. 2016

ZAAL, Sjoukje. Azure DevOps Explained: Get started with Azure DevOps and develop your DevOps practices (English Edition). Packt Publishing. 2020

Bibliografia Complementar:

FREEMAN, Emily. Devops for Dummies. For Dummies. 2019

SATO, Danilo. DevOps na prática: Entrega de software confiável e automatizada. Casa do Código 2014

THORPE, ETHAN. DEVOPS: A COMPREHENSIVE BEGINNERS GUIDE TO LEARN DEVOPS STEP BY STEP. ETHAN THORPE 2019

Disciplina: *Design Patterns* e gerência de configuração: do projeto ao controle de versões.

Ementa: Apresentação, diferenciação e aplicação dos principais *Design Patterns* existentes (*Gang of Four*). Conceitos básicos sobre Gerência de Configuração, atividades essenciais: controle de mudança, controle de versão e integração contínua. Ferramentas de controle de versões. Integração entre o *Design Patterns* e Gerência de Configuração com o processo de Desenvolvimento de Software.

Competências e Habilidades:

- Conhecer e diferenciar os principais padrões de projeto, ou *Design Patterns* existentes no Grupo elementar de padrões chamado *Gang of Four* propiciando sua adoção em projetos de software;
- Entender a Gerência de Configuração, no que tange as atividades essenciais de: Controle de Mudança, Controle de Versão e Integração Contínua, e como esta gerência pode ser aplicada no desenvolvimento de software.
- Reconhecer e utilizar ferramentas de controle de versão, favorecendo a atividade de Controle de Versão no contexto da Gerência de Configuração;
- Propor projetos de software que integrem *Design Patterns* e a Gerência de Configuração, visando qualidade.

Conteúdo Programático 1: *Design Patterns*: conhecendo os principais padrões.

Conteúdo Programático 2: Gerência de Configuração: atividades, ferramentas e desafios.

Conteúdo Programático 3: Controle de Versões na Prática com Ferramentas de Versionamento.

Conteúdo Programático 4: Integrando *Design Patterns*: gerência de configuração em projetos de software.

Bibliografia Básica:

SHALLOWAY, Alan e TROTT, James. *Design patterns explained: a new perspective on object-oriented design*. Pearson Education, 2004. 480p.

SOMMERVILLE, Ian. **Engenharia de Software**; tradução Luiz Cláudio Queiroz; revisão técnica Fábio Levy Siqueira. 10. ed. São Paulo: Pearson Education do Brasil, 2018.

WEILL, Peter; ROSS, Jeanne W. **Governança de TI - Tecnologia da Informação**. M. Books, 2020. 296p.

Bibliografia Complementar:

PRESSMAN, Roger S. **Engenharia de Software**: uma abordagem profissional; tradução João Eduardo Nóbrega Tortello; revisão técnica Reginaldo Arakaki, Julio Arakaki, Renato Manzan de Andradel. 8. ed. Porto Alegre: AMGH, 2016.

BORGES, Simone de Sousa. **Apoio de gerência de configuração de software ao ARAGAwCRe**. Universidade Federal de São Carlos. Dissertação de Mestrado. São Carlos: UFSCar, 2009. 102f.

SILVERMAN, Richard E. **Git - Guia Prático**. São Paulo: Novatec Editora, 2019. 208p.

Disciplina: Interface de programação de aplicações (API) e Web Services.

Ementa: Diferenciando e caracterizando as APIs e *Web Services*. Arquiteturas de APIs e *Web Services*. Tecnologias para o desenvolvimento de APIs e *Web Services*. Relação de projetos *front end* com APIs e *Web Services*. Decisões tecnológicas para o desenvolvimento e testes de APIs e *Web Services*.

Competências e Habilidades:

- Conhecer as diferenças entre APIs e *Web Services*, em relação a diferentes especificidades;
- Analisar, planejar e projetar Arquiteturas de APIs e *Web Services*, bem como selecionar tecnologias para a implementação de tais soluções;
- Implementar e testar APIs e *Web Services* básicos e entender o processo de consumo e conexão com o *front end*.

Conteúdo Programático 1: Caracterizando as APIs dos *Web Services*.

Conteúdo Programático 2: Análise de Projetos de Arquiteturas de APIs e *Web Services* e Seleção de Tecnologias para Implementação.

Conteúdo Programático 3: Consumo das APIs e *Web Services* por soluções do *Front End*.

Conteúdo Programático 4: Decisões tecnológicas para o desenvolvimento e testes de APIs e *Web Services*.

Bibliografia Básica:

LECHETA, Ricardo R. **Web Services RESTful: Aprenda a criar web services RESTful em Java na nuvem do Google**. São Paulo: Novatec Editora, 2015. 432p.

MASSE, Mark. **REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces**. Sebastopol: O'Reilly Media, Inc., 2011. 116p.

MITCHELL, Lorna Jane. **PHP Web Services: APIs for the Modern Web**. Sebastopol: O'Reilly Media, Inc., 2016.

Bibliografia Complementar:

AGNOL, Giovani Dall; PAVAN, Willingthon. **Web Services–Como Integrar Empresas Através dos Recursos da Internet**. In IX Congreso Argentino de Ciencias de la Computación. 2003.

Zanelato, Ana Paula Ambrósio. **Web Services: uma Nova Arquitetura da Internet**. ETIC - Encontro De Iniciação Científica. v. 6, n. 6, 2010.

RICHARDSON, Leonard; RUBY, Sam. **RESTful web services**. Sebastopol: O'Reilly Media, Inc., 2008. 288p.

PEREIRA, Caio Ribeiro. **Construindo APIs REST com Node.js**: Caio Ribeiro Pereira. São Paulo: Editora Casa do Código, 2016. 193p.

Disciplina: Desenvolvimento de aplicações mobile – Android

Ementa: O cenário do mercado da plataforma Android e seus aplicativos; Especificidades de dispositivos com a plataforma Android; Base estrutural de um aplicativo Android; Ferramentas para desenvolvimento Android; Principais componentes padrões da plataforma Android; Estratégias de desenvolvimento de aplicativos Android; O Android e seu mercado de trabalho para os próximos anos.

Competências e Habilidades:

- Assimilar o cenário atual e futuro da tecnologia *mobile* Android;
- Conhecer a estrutura básica de um programa Android;
- Conhecer e utilizar os componentes mais utilizados no desenvolvimento de aplicações Android;
- Dominar técnicas para o desenvolvimento de aplicações Android.

Conteúdo Programático 1: O cenário do mercado da plataforma Android e seus aplicativos.

Conteúdo Programático 2: Base estrutural de um aplicativo Android.

Conteúdo Programático 3: Ferramentas para desenvolvimento Android.

Conteúdo Programático 4: Componentes padrões da plataforma Android.

Bibliografia Básica:

DEITEL, Paul; DEITEL, Harvey; WALD, Alexander. **Android 6 para programadores**: uma abordagem baseada em aplicativos. 3. ed. Porto Alegre: Bookman, 2016. 422 p. (Deitel developer series). ISBN 9788582604113.

LECHETA, Ricardo R. **Google Android:** aprenda a criar aplicações para dispositivos móveis com Android SDK. 3. ed. São Paulo: Novatec, 2013. 824 p. ISBN 978-85-7522-344-4.

OEHLMAN, Damon; BLANC, Sébastien. **Aplicativos Web Pro Android:** Desenvolvimento Pro Android Usando HTML5, CSS3 & JavaScript. 1 ed. Rio de Janeiro: ciência moderna, 2012. 455p. ISBN 9788539902507.

Bibliografia Complementar:

NUDELMAN, Greg. **Padrões de projeto para o Android:** soluções de projetos de interação para desenvolvedores. São Paulo: Novatec, 2013. 456 p. ISBN 9788575223581.

QUERINO FILHO, Luiz Carlos. **Desenvolvendo seu primeiro aplicativo Android:** Entre de cabeça no mundo dos aplicativos moveis, criando e publicando seu próprio programa para o sistema líder do mercado. 1.ed. São Paulo 2013. 247p. ISBN 9788575223789.

SHACKLES, Greg. **Construindo aplicativos móveis com C#.** São Paulo Novatec 2012. 200 p. ISBN 9788575223222.

|

13. Infraestrutura Física e Pedagógica

O aluno encontrará todo o conteúdo do curso e assistirá às aulas gravadas no ambiente virtual. Para assistir às aulas é fundamental que as especificações abaixo sejam obedecidas, possibilitando, assim, uma recepção de maior qualidade dos vídeos.

Hardware:

- Processador Intel Core 2 Duo ou superior.
- 2Gb de Memória RAM.
- Placa de vídeo com resolução 1024x768, qualidade de cor 32 bits e compatível com Microsoft DirectShow.

- Microsoft DirectX 9.0c ou posterior.

Software:

- Navegador: Firefox, Google Chrome, Internet Explorer (sempre atualizado).
- Sistema Operacional: Windows XP ou posterior.
- Adobe Flash Player (atualizado).
- *Plugin* de vídeos SilverLigth (atualizado)

Rede:

- Conexão com a Internet banda larga de no mínimo 2 MB.
- Em caso de acesso em ambientes corporativos além da velocidade, é necessário verificar as condições de segurança de rede de sua empresa e se certificar que o site não estará bloqueado.

Adicionalmente, é prevista a utilização da biblioteca virtual para consultas bibliográficas e pesquisa de assuntos referentes às disciplinas ministradas.

