



PROJETO PEDAGÓGICO DE CURSO
Pós-Graduação *Lato Sensu*
Análise e Desenvolvimento de Sistemas

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	5
8. Metodologia de Ensino e Aprendizagem	6
9. Estágio Não Obrigatório	8
10. Matriz Curricular	10
11. Carga Horária	10
12. Conteúdo programático	10
13. Infraestrutura Física e Pedagógica	12

1. Nome do Curso e Área do Conhecimento

Nome do Curso: Análise e Desenvolvimento de Sistemas

Área de Avaliação (CAPES): Ciência da Computação

Grande Área (CAPES): CIÊNCIAS EXATAS E DA TERRA

Área do Conhecimento (CAPES): 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 de especialização em Análise e Desenvolvimento de Sistemas está destinado a profissionais das áreas de negócios e engenharia que desejam conhecer processos de desenvolvimento e análise de sistemas.

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

Os sistemas de informação representam o ferramental básico para a execução dos processos de negócio em sua totalidade. Para que estes sistemas contribuam com as estratégias de negócios é necessário que todo o processo de desenvolvimento e implementação seja fundamentado em metodologias e boas práticas.

Além de ter o processo de desenvolvimento bem estabelecido, é necessário que nas organizações exista a cultura de análise adequada dos sistemas, a fim de aumentar cada vez mais eficiência dos mesmos, evitando problemas, indisponibilidade que só cooperam com o caminho contrário ao aumento das vantagens competitiva.

Desta forma, a Especialização em Análise e Desenvolvimento de Sistemas se configura como uma oportunidade para os profissionais de negócios e engenharia que desejam conhecer o desenvolvimento e a análise de sistemas de informação, de forma alinhada as estratégias organizacionais.

6. Objetivos do Curso

6.1. Objetivo Geral

- Capacitar profissionais para desenvolver e analisar sistemas de informação, conhecendo os frameworks de desenvolvimento e aplicando para os diferentes sistemas existentes.

6.2. Objetivos Específicos

- Apresentar os fundamentos básicos de *frameworks back end* e *front end*.
- Fornecer conhecimentos sobre linguagens de programação utilizadas no desenvolvimento de sistemas.
- Compreender os processos mais comuns na análise de sistemas de informação.

7. Competências e Habilidades do Egresso

Competências

Atuar no contexto empresarial desenvolvendo sistemas de informação alinhados as estratégias de negócios, atendendo as expectativas dos clientes e usuários.

Habilidades

- Desenvolver sistemas de informação utilizando linguagens de programação de mercado.
- Identificar, mapear, planejar, executar e monitorar problemas relacionados a sistemas de informação.
- Aplicar técnicas e ferramentas no desenho e implementação de sistemas de informação.

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
Lógica de programação	0 h	40 h	40 h
Linguagem e padrões web	0 h	40 h	40 h
Frameworks back end	0 h	40 h	40 h
Frameworks front end	0 h	40 h	40 h
Desenvolvimento Java Web	0 h	40 h	40 h
Programação PHP	0 h	40 h	40 h
Programação orientada a objetos	0 h	40 h	40 h
Modelagem do sistema com a análise orientada a objetos	0 h	40 h	40 h
Aplicações móveis	0 h	40 h	40 h

11. Carga Horária

A carga horária de 360 h 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: Lógica de programação.

Ementa: Ensino e aprendizagem do “pensar” algorítmico e sistêmico fundamental para a modelagem e desenvolvimento de soluções computacionais. Introdução à Lógica de Programação. Conceitos fundamentais. Programação

Estruturada. Algoritmos e seus elementos. Fluxograma. Tipos de dados. Variáveis. Operadores. Constantes. Comentários. Expressões. Estrutura sequencial. Seleção simples e múltipla. Repetição contada, com pré-condição e póscondição. Sub-rotinas sem retorno de valor e com retorno de valor. Passagem de parâmetros. Variáveis locais e globais. Estruturas de dados homogêneas. Estruturas de dados heterogêneas.

Competências e Habilidades:

- Introduzir os conceitos básicos de desenvolvimento de algoritmos e lógica de programação
- Desenvolver no aluno o raciocínio lógico aplicado à solução de problemas em nível computacional.
- Ter visão crítica e sistemática sobre resolução de problemas.
- Compreender os conceitos fundamentais de algoritmos como forma de solução de problemas
- Capacidade de desenvolver e modelar algoritmos visando soluções práticas.

Conteúdo Programático 1: Estrutura de Dados Básicas. Tipos básicos de dados. Operadores aritméticos, relacionais e lógicos. Comandos básicos de entrada e saída e atribuição.

Conteúdo Programático 2: Conceito de bloco de comandos. Estruturas de controle de fluxo - condicionais (se, se-senão e caso). Estruturas de controle de fluxo - repetições (para, enquanto e repita -enquanto).

Conteúdo Programático 3: Estruturas de Dados Homogêneas: Vetores e Matrizes. Estruturas de Dados Heterogêneas: Registros. Cadeias de caracteres.

Conteúdo Programático 4: Modularização. Variáveis locais e globais. Funções. Passagem de parâmetros por valor e por referência. Funções recursivas. Biblioteca de funções.

Bibliografia Básica:

SOUZA, Marco. **Algoritmos e lógica de programação**: um texto introdutório para engenharia. São Paulo: Cengage, 2019. (BV

<https://integrada.minhabiblioteca.com.br/reader/books/9788522128150/pageid/>

0)

SILVA, Flávio. **Lógica para computação**. São Paulo: Cengage, 2017. (BV
<https://integrada.minhabiblioteca.com.br/reader/books/9788522127191/pageid/0>)

XAVIER, Gley. **Lógica de Programação**. São Paulo: Editora Senac SP, 2017. (BV
<https://www.bibliotecadigitalsenac.com.br/?from=busca%3FcontentInfo%3D724%26term%3Dlogica%252520de%252520programa%2525C3%2525A7%2525C3%2525A3o&page=2§ion=0#/legacy/724>)

GUEDES, Sérgio (org). **Lógica de programação algorítmica**. São Paulo: Pearson, 2014. (BV
<https://plataforma.bvirtual.com.br/Acervo/Publicacao/22146>)

Bibliografia Complementar:

MEIO, Ana Cristina. **Princípios de linguagem de programação**. São Paulo: Blucher, 2003. (BV
<https://integrada.minhabiblioteca.com.br/reader/books/9788521214922/pageid/4>)

GOLDSTEIN. Laur. **Lógica: conceitos chave em filosofia**. Porto Alegre: Bookman, 2007. (BV
<https://integrada.minhabiblioteca.com.br/reader/books/9788536309651/pageid/0>)

ALVES, William. **Linguagem e lógica de programação**. São Paulo: Érica, 2014. (BV
<https://integrada.minhabiblioteca.com.br/reader/books/9788536519371/pageid/0>)

Disciplina: Linguagem e padrões Web

Ementa: Marcação (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: Frameworks back end

Ementa: Diferenciando o *front end* do *back end* e diferenciando as decisões tecnológicas existentes. Visão geral sobre *frameworks back end*. O Framework JavaScript Express. O Framework Spring Boot para Java. O Framework Laravel para PHP.

Competências e Habilidades:

- Conhecer e selecionar *frameworks back end* apropriados para o desenvolvimento de projetos do lado servidor, principalmente para soluções distribuídas;
- Utilizar, na prática, os *frameworks back end*, na criação de APIs para o *front end* e outras possíveis aplicações consumidoras;
- Gerenciar equipes que podem atuar utilizando *frameworks back end*.

Conteúdo Programático 1: Conhecendo e Selecionando *Frameworks Back End*.

Conteúdo Programático 2: Express: Um *Framework Back End* com JavaScript e Node.js.

Conteúdo Programático 3: Potencializando o PHP com o *Framework Back End* Laravel.

Conteúdo Programático 4: Java com o *Framework Back End* Spring Boot.

Bibliografia Básica:

AMBLER, Tim; CLOUD, Nicholas; HAWKES, Robin Andrew. **JavaScript Frameworks for Modern Web Dev**. Nova Iorque: Apress, 2015. 485p.

BROWN, Ethan. **Web development with node and express: leveraging the JavaScript stack**. Sebastopol: O'Reilly Media, 2019.

GABARDO, Ademir C. **Laravel para ninjas**. São Paulo: Novatec Editora, 2017. 480p.

Bibliografia Complementar:

LEONARD, Anghel. **Spring Boot Persistence Best Practices**. Nova Iorque: Apress, 2020. 1027p.

CORDEIRO, Alexandre Costa. **Dyfocus: Desenvolvimento do Back- End de um Aplicativo Mobile para Smartphone**. Monografia (Graduação em Engenharia de Controle e Automação) - Universidade Federal de Santa Catarina. Florianópolis, 71p., 2016.

MORAES, William Bruno. **Construindo aplicações com NodeJS**. São Paulo: Novatec Editora, 2018. 216p.

Disciplina: *Frameworks front end*

Ementa: Apresentação dos principais *frameworks front end* atuais, suas vantagens e desvantagens; Conceitos essenciais sobre o *framework* Vue.js.

Utilização do *framework* Angular com conceitos sobre *TypeScript*.

Compreendendo a estrutura do *framework* React e utilizando seus componentes. Conceitos sobre migração entre *frameworks*. Consumo de interfaces de programação de aplicações (API) e conexão com bancos de dados relacionais e não relacionais com *frameworks*.

Competências e Habilidades:

- Conhecer e identificar as vantagens e desvantagens de *frameworks* utilizados para o desenvolvimento *front end*;

- Identificar a sintaxe e semântica de cada *framework*, bem como sua estrutura de arquivos, construção e comportamentos;
- Implementação de páginas web com a adoção de *frameworks*;
- Entender e realizar a migração de *frameworks*, segundo demandas;
- Dominar o modo de estabelecer conexões com bancos de dados relacionais e não relacionais e o consumo de API.

Conteúdo Programático 1: Diferenciando e escolhendo um *Framework Front End*. Apresentar os *frameworks* *Vue.js*, *Angular* e *React*, identificando criadores, vantagens, desvantagens e fornecendo base teórica para a escolha de um deles.

Conteúdo Programático 2: Compreendendo e utilizando o *Framework* *Vue.js*.

Conteúdo Programático 3: *Angular*: utilizando o *TypeScript* com o *Framework* da Google.

Conteúdo Programático 4: *React*: desenvolvimento de componentes com o *Framework* do Facebook.

Bibliografia Básica:

ALVES, William Pereira. **Desenvolvimento de Aplicações Web com Angular**. Rio de Janeiro: Alta Books Editora, 2019. 384p.

PETERSON, Kathleen. **React Js: The Ultimate Beginner's Guide to Learn React Js Programming Step by Step**. Michigan: Independently Published, 2020. 161p.

FILIPOVA, Olga. **Learning Vue. js 2**. Birmingham: Packt Publishing Ltd, 2016. 334p.

Bibliografia Complementar:

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.

DE CAMARGOS, João Gabriel Colares et al. **Uma Análise Comparativa entre os Frameworks Javascript Angular e React**. Belo Horizonte: Computação & Sociedade, v. 1, n. 1, 2019.

JAPIKSE, Philip; GROSSNICKLAUS, Kevin; DEWEY, Ben. **Introduction to TypeScript**. In: Building Web Applications with Visual Studio 2017. Apress, Berkeley, CA, 2017. p. 241-280.

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: Programação PHP

Ementa: Conceitos, técnicas, comandos e instruções em PHP; Revisão de Clientes/servidores/TCP; Revisão/contextualização de programação de computadores: HTML e PHP..

Competências e Habilidades:

- Programar na linguagem de desenvolvimento PHP

Conteúdo Programático 1: Instalando o MySQL, Apache e PHP, será apresentado a instalação das ferramentas que serão utilizadas durante a disciplina, mostrando todos dos parâmetros.

Conteúdo Programático 2: Conhecer o ambiente PHP e utilizar os comandos da linguagem PHP conforme a solução web.

Conteúdo Programático 3: Relacionamentos com ambiente web, desenvolver scripts do lado servidor em PHP.

Conteúdo Programático 4: AJAX + PHP Relacionamentos com ambiente web.

Conteúdo Programático 5: Banco de dados com PHP, utilização do banco com a linguagem PHP.

Conteúdo Programático 6: Desenvolvimento de sites em PHP, desenvolver e manter aplicações dinâmicas para Internet com interface web, criando sites dinâmicos com a linguagem PHP.

Conteúdo Programático 7: Sistemas open-source em PHP, criação e apresentação.

Conteúdo Programático 8: Práticas com frameworks PHP criação e apresentação.

Bibliografia Básica:

CONVERSE, Tim; PARK, Joyce. **PHP a bíblia**. Rio de Janeiro: Campus, 2003.

DEITEL, Paul J.; DEITEL, Harvey M. Ajax, Rich. **Internet Applications e desenvolvimento web para programadores**. São Paulo: Pearson Prentice Hall, 2009.

NIEDERAUER, Juliano. **Web Interativa com Ajax e PHP**. São Paulo: Novatec, 2013.

Bibliografia Complementar:

LEMAY, Laura. **Aprenda a criar páginas web com HTML e XHTML em 21 dias**. São Paulo: Makron Books, 2002.

SHEPERD, George. **Microsoft ASP.NET 3.5 - Passo a Passo**. Porto Alegre: Bookman, 2009.

Disciplina: Programação orientada a objetos

Ementa: Histórico sobre a orientação a objetos, conceitos essenciais sobre os pilares da orientação a objetos: 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.

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.

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

Bibliografia complementar:

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

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: Modelagem do sistema com a análise orientada a objetos

Ementa: Modelagem do negócio, análise e desenvolvimento do software orientado a objetos. Utilização de padrões em métodos com a Linguagem de Modelagem Unificada (UML) para especificação, visualização e documentação da solução de software. Modelagem Funcional e de Comportamento: Diagrama de Casos de Uso, de Colaboração, de Sequência, de Atividades e de Estados. Principais diagramas da Modelagem Estrutural: Diagrama de Classes, Diagrama de Objetos, de Componentes, de Pacotes e de Implantação.

Competências e Habilidades:

- Analisar e inferir com a modelagem orientada a objetos;
- Saber como aplicar a linguagem UML em projetos de softwares;
- Utilizar os diagramas da UML a fim de produzir a modelagem funcional e comportamental do software;
- Utilizar os diagramas da UML para produzir a modelagem estrutural do software.

Conteúdo Programático 1: Modelagem do negócio, análise e desenvolvimento do software orientado a objetos.

Conteúdo Programático 2: Utilização de padrões em métodos com a Linguagem de Modelagem Unificada (UML) para especificação, visualização e documentação.

Conteúdo Programático 3: Modelagem Funcional e de Comportamento: Diagrama de Casos de Uso, de Colaboração, de Sequência, de Atividades e de Estados.

Conteúdo Programático 4: Principais diagramas da Modelagem Estrutural: Diagrama de Classes, Diagrama de Objetos, de Componentes, de Pacotes e de Implantação.

Bibliografia básica:

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

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.

NOGUEIRA, José H. M. **Engenharia de Software**: Métodos Orientados a Objetos e Ágeis. Fortaleza, CE: Amazon KDP, 2018.

Bibliografia Complementar:

PAULA Filho, Wilson de P. **Engenharia de Software**: projetos e processos. 4. ed. Rio de Janeiro: LTC, 2019.

GUEDES, Gilleanos T. A. **UML 2**: uma abordagem prática. 3. ed. São Paulo: Novatec, 2018.

BOOCH, Grady; RUMBAUCH, James; JACOBSON, Ivar. **UML**: guia do usuário; tradução: Fábio Freitas da Silva e Cristina Amorim Machado. 2. ed. Rio de Janeiro: Elsevier, 2012.

Disciplina: Aplicações móveis

Ementa: Desenvolvimento de aplicativos móveis com mentalidade digital para negócios de próxima geração.

Conteúdo Programático 1: Introdução à Mobilidade Digital. Revolução da Mobilidade na vida humana; a transformação mobile; bases de dispositivos móveis; smartphones, tablets, relógios inteligentes e outras tecnologias.

Conteúdo Programático 2: Noções Sobre Processos De Desenvolvimento De Software. O Pensamento Lean; o manifesto agile; os princípios de UX Design; técnicas de Design thinking.

Conteúdo Programático 3: Linguagens de Programação. Programação orientada a objetos; linguagens nativas; Android - Java, iOS - Swift e Objective-c, (Windows Phone - C#); ferramentas de desenvolvimento (IDEs, SDKs Frameworks).

Conteúdo Programático 4: Desenvolvimento de aplicações para dispositivos móveis. Desenvolvimento de Mobile Native App; plataforma operacional (Android, iOS, Windows, etc.); desenvolvimento de plataformas híbridas; ferramentas de desenvolvimento híbrido.

Conteúdo Programático 5: Arquitetura e Ambiente de Desenvolvimento. Plataforma operacional (Android, iOS, Windows, etc.); a era da nuvem para dispositivos móveis (Cloud Computing, Cloud Storage, Cloud Services; e, Open Data, Open API, Open Source).

Conteúdo Programático 6: Desenvolvimento Navite App Android. Paas para Aplicativos e Back-end Android; Ferramentas de Desenvolvimento Android; Android Developer Fundamentals.

Conteúdo Programático 7: Teste De Software. Teste De Aplicativos Sob Demanda Em Um Serviço Na Nuvem.

Conteúdo Programático 8: Boas Práticas De Gestão De Projetos. Critérios para a Escolha do Hosting; diferenças Native App vs. Híbrid; adequação do cloud plataforma ao projeto/negócio; adequação do cloud plataforma ao projeto/negócio.

Bibliografia:

ALVES, Thiago Salhab. **Tecnologias para web e para dispositivos móveis.** Londrina: Editora e Distribuidora Educacional S/A, 2016.

ASCENCIO, Ana Fernanda Gomes; CAMPOS, Edilene Aparecida Veneruchi de. **Fundamentos da programação de computadores.** 2. ed. São Paulo: Pearson Prentice Hall, 2007.

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

BONGIORNO, Renato. **Implementação de negócios digitais.** São Paulo: Editora Senac São Paulo, 2017.

BREITMAN, Karin. **Web semântica: a internet do futuro**. Rio de Janeiro: LTC, 2005.

GOMES, Rafael Caveari; FERNANDES, Jean Alves R.; FERREIRA, Vinícius Corrêa. **Sistema Operacional Android**. 2012. 30f. TCC (graduação) – Curso de Engenharia de Telecomunicações. Universidade Federal Fluminense. Rio de Janeiro, 2012.

HOWARD, Michael. **Escrevendo Código Seguro: estratégias e técnicas práticas para codificação segura de aplicativos em um mundo em rede**. 2. ed. Porto Alegre: Bookman, 2005.

LECHETA, Ricardo R. **Google Android: aprenda a criar aplicações para dispositivos móveis com o Android SDK**. 3. ed. São Paulo: Novatec, 2013.

LEE, Valentino; SCHNEIDER, Heather; SCHELL, Robbie. **Aplicações móveis: arquitetura, projeto e desenvolvimento**. São Paulo: Pearson Education do Brasil, 2005.

SUTTER, Herb. **Programação avançada em C++**. São Paulo: Pearson Mkron Books, 2006.

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.

