



PROJETO PEDAGÓGICO DE CURSO
Pós-Graduação *Lato Sensu*
Engenharia de *Software*

Sumário

1. Nome do Curso e Área do Conhecimento.....	3
2. Características Técnicas do Curso.....	3
3. Público-Alvo	3
4. Critérios de Seleção	4
5. Justificativa do Curso	4
6. Objetivos do Curso	4
7. Competências e Habilidades do Curso	5
8. Metodologia de Ensino e Aprendizagem	5
9. Estágio Não Obrigatório.....	8
10. Matriz Curricular	10
11. Carga Horária.....	11
12. Conteúdo Programático.....	11
13. Infraestrutura Física e Pedagógica.....	20

1. Nome do Curso e Área do Conhecimento

Nome do Curso: Engenharia de *software*

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: 200 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 designado para bacharéis em Engenharia, Administração, Ciência da Computação, Sistema de Informação, Tecnólogos da área da Tecnologia da Informação e para profissionais com graduação superior que atuam no setor corporativo em papel técnico ou de liderança: de startups até empresas

de grande porte, já envolvidos com processos de Engenharia de Software ou atuando como empreendedores no setor da Tecnologia da Informação, que buscam desenvolver e/ou manter suas próprias soluções.

4. Critérios de Seleção

O ingresso na pós-graduação será realizado por meio de inscrição no Portal Pós, entrega da documentação pessoal e do diploma da graduação autenticado. Em seguida, analisados pela Instituição.

5. Justificativa do Curso

A dependência por *softwares* no cotidiano das pessoas e no ambiente corporativo da área pública e privada está em constante crescimento. Não se consegue imaginar atualmente o mundo sem a utilização de aplicativos que facilitem e agilizem a vida das pessoas e das empresas.

A Engenharia de Software vem aplicar princípios de gestão, técnicas e ferramentas para todo o processo de desenvolvimento do *software*, desde o levantamento do problema até a entrega da solução.

No Brasil, o mercado busca cada vez mais profissionais capacitados para o desenvolvimento de *softwares*. Analistas de sistemas, programadores, administradores de banco de dados, coordenadores de projeto, entre outros profissionais da área de TI, necessitam ampliar suas competências em relação à Engenharia de Software.

Este curso vem atender a esta demanda por conhecimento da gestão do ciclo de vida do *software*, abrangendo: a análise, a modelagem, a construção, os testes, a implantação e a manutenção. Com o foco na produção de *softwares* em um menor prazo, consequentemente com menor custo e com maior qualidade, através da aplicação de métodos e ferramentas atuais para o desenvolvimento de *softwares*.

6. Objetivos do Curso

6.1. Objetivos Gerais

- Preparar profissionais capazes de criar e manter soluções computacionais, utilizando os recursos adequados em todas as etapas do desenvolvimento de *softwares*, para atender os diversos segmentos do mercado, minimizando custos através do aumento da produtividade e melhorando a qualidade do *software*.

6.2. Objetivos Específicos

- Apresentar os conceitos gerais do ciclo de vida de um *software* em um ambiente corporativo.
- Proporcionar conhecimentos para gerenciar, implantar e manter projetos de *softwares* nas organizações.
- Propiciar um conjunto de conhecimentos para a compreensão do problema através de técnicas de levantamento de requisitos, utilizando abstração para a modelagem do sistema, sua construção e implantação.
- Apresentar as principais metodologias de desenvolvimento de *software*, principalmente as Metodologias Ágeis, a fim de permitir a construção de *softwares* com custo menor e com melhor qualidade.
- Propiciar o desenvolvimento de um profissional inovador, com respeito aos princípios éticos, atuando individualmente ou fazendo parte de uma equipe.

7. Competências e Habilidades do Curso

Competências

Estar apto a identificar problemas do mundo real e a propor soluções em *softwares*, compreendendo todo o processo de desenvolvimento de forma qualitativa e sua manutenção, utilizando ferramentas automatizadas ou não e dominando os métodos e as técnicas para a produção de *software*.

Habilidades

- Analisar, especificar, modelar, validar e gerenciar os requisitos do *software*.
- Apresentar linguagens de programação e sistemas gerenciadores de banco de dados para a construção do sistema.

- Identificar e gerenciar casos de testes.
- Aplicar metodologias de gestão e apresentar ferramentas para o desenvolvimento do *software*.
- Identificar a necessidade de manutenção do *software* visando correções de erros ou propor melhorias para a evolução do mesmo.
- Compreender a importância da negociação e do trabalho em equipe, exercendo a boa comunicação com os *stakeholders* do sistema.

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.

O estágio curricular não obrigatório do curso de Pós-Graduação *Lato Sensu* segue as diretrizes estabelecidas pela Lei Nº 11.788/2008.

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

I. proporcionar o exercício do aprendizado compromissado 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	0h	0h	0h

Conceitos gerais e principais abordagens de desenvolvimento do <i>software</i>	0h	40h	40h
CASE: as principais ferramentas no ciclo de desenvolvimento de <i>softwares</i>	0h	40h	40h
Engenharia de requisitos	0h	40h	40h
Modelagem do sistema com a análise orientada a objetos	0h	40h	40h
Codificação do projeto: padrões de programação e projeto do banco de dados	0h	40h	40h
Gerenciamento ágil dos sistemas	0h	40h	40h
Validação do <i>software</i> : testes de <i>softwares</i> e aplicações de segurança no sistema	0h	40h	40h
Evolução dos <i>softwares</i> : aplicação da Engenharia de Software em sistemas emergentes	0h	40h	40h
Arquitetura de software	0h	40h	40h

11. Carga Horária

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

12. Conteúdo programático

Disciplina: Ambientação

Ementa: Histórico da Educação a Distância. Legislação da Educação a Distância no Brasil. Potencialidades da Educação a Distância. Flexibilidade de Acesso. Tecnologias para apoio à pesquisa. Aprendizagem colaborativa. Características do aluno na EAD. Boa convivência virtual: netiquetas.

Competências e Habilidades:

- Identificar as características do aluno da modalidade de educação a distância;
- Utilizar ferramentas tecnológicas de apoio a pesquisa;

- Comunicar-se com clareza e assertividade oralmente e por escrito;
- Gerenciar tempo e atividades de estudo.

Conteúdo Programático 1: Histórico da Educação a Distância.

Conteúdo Programático 2: Legislação da Educação a Distância no Brasil.

Conteúdo Programático 3: Potencialidades da Educação a Distância.

Conteúdo Programático 4: Flexibilidade de Acesso.

Conteúdo Programático 5: Tecnologias para apoio à pesquisa.

Conteúdo Programático 6: Aprendizagem colaborativa.

Conteúdo Programático 7: Características do aluno na EAD.

Conteúdo Programático 8: Boa convivência virtual: netiquetas.

Bibliografia básica:

GOTTARDI, M. de L. A autonomia na aprendizagem em educação a distância: competência a ser desenvolvida pelo aluno. **Associação Brasileira de Educação a Distância**, São Paulo, v. 14, n. 8, p. 110-123, dez, 2015.

MORAN, J. M. MASETTO, M. T. BEHRENS, M. A. **Novas Tecnologias e Mediação Pedagógica**. 21. ed. Campinas, SP: Papirus, 2013.

PIVA, D. J. PUPO, R. GAMEZ, L. OLIVEIRA, S. **EAD na Prática: Planejamento, métodos e ambientes de educação online**. São Paulo: Elsevier, 2011.

Bibliografia Complementar:

ALVES, L. Educação a distância: conceitos e história no Brasil e no mundo. **Associação Brasileira de Educação A Distância**, São Paulo, v. 10, n. 7, p.85-92, out. 2011.

MAIA, Carmen; MATTAR, João. **ABC da EaD: a educação a distância hoje**. São Paulo: Prentice Hall, 2007.

MOORE, Michael G., **Educação a distância: sistemas de aprendizagem on-line**. 3. ed., São Paulo: Cengage Learning, 2013.

Disciplina: Conceitos gerais e principais abordagens de desenvolvimento do software

Ementa: Conceitos gerais da Engenharia de Software: aplicações do software no mundo corporativo e o papel do engenheiro de software, o ciclo de desenvolvimento do software: levantamento de requisitos, análise, modelagem, construção, testes, implantação e manutenção e os principais modelos de certificações: ISO e CMMI. As metodologias clássicas: cascata, prototipação, espiral, incremental e o Processo Unificado (UP). As metodologias ágeis: o Manifesto Ágil e as principais abordagens: FDD e XP. A metodologia SCRUM: o ciclo de desenvolvimento e as principais características: equipe, papéis e artefatos.

Competências e Habilidades:

- Aplicar os conceitos centrais da Engenharia de Software como as abordagens clássicas e as mais utilizadas atualmente no mercado;
- Compreender a importância de desenvolver softwares que atendam às necessidades dos usuários;
- Conhecer os principais componentes do processo de desenvolvimento de software;
- Compreender os princípios e fundamentos das metodologias ágeis como XP (Extreme Programming) e Scrum.

Conteúdo Programático 1: Conceitos gerais da Engenharia de Software: aplicações do software no mundo corporativo e o papel do engenheiro de software, o ciclo de desenvolvimento do software: levantamento de requisitos, análise, modelagem, construção, testes, implantação e manutenção e os principais modelos de certificações: ISO e CMMI.

Conteúdo Programático 2: As metodologias clássicas: cascata, prototipação, espiral, incremental e o Processo Unificado (UP).

Conteúdo Programático 3: As metodologias ágeis: o Manifesto Ágil e as principais abordagens: FDD e XP.

Conteúdo Programático 4: A metodologia SCRUM: o ciclo de desenvolvimento e as principais características: equipe, papéis e artefatos.

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.

VINCENZI, Auri M. R. [et al.] **Automatização de Teste de Software com Ferramentas de Software Livre**. 1.ed. Rio de Janeiro: Elsevier, 2018.

Disciplina: CASE: as principais ferramentas no ciclo de desenvolvimento de softwares

Ementa: A automação, um conjunto de ferramentas que auxiliam o processo de desenvolvimento de softwares: ferramentas para levantamento de requisitos, desenvolvimento de interfaces, construção de banco de dados e programação orientada a objetos. Ferramentas para a modelagem: MS Visio, Astah, IDE Eclipse (UML) e NetBeans. As principais características das ferramentas de gestão do desenvolvimento do software: MS Visual Studio Team System (VSTS), GitHub e RedMine. Ferramentas para automação de testes de software.

Competências e Habilidades:

- Determinar ferramentas para auxiliar o processo de desenvolvimento de softwares;
- Criar processos de automação de testes;
- Conhecer algumas das principais ferramentas CASE aplicadas atualmente;
- Compreender a importância de tornar o processo de desenvolvimento de software mais automatizado para obter custos menores e com melhor qualidade.

Conteúdo Programático 1: A automação, um conjunto de ferramentas que auxiliam o processo de desenvolvimento de softwares: ferramentas para levantamento de requisitos, desenvolvimento de interfaces, construção do banco de dados e para programação orientada a objetos.

Conteúdo Programático 2: Ferramentas para a modelagem: MS Visio, Astah, IDE Eclipse (UML) e NetBeans.

Conteúdo Programático 3: As principais características das ferramentas de gestão do desenvolvimento do software: MS Visual Studio Team System (VSTS), Github e RedMine.

Conteúdo Programático 4: Ferramentas para automação de testes de software.

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.

HIRAMA, Kechi. **Engenharia de Software: qualidade e produtividade com tecnologia**, Rio de Janeiro: Elsevier, 2011.

VINCENZI, Auri M. R. [et al.] **Automatização de Teste de Software com Ferramentas de Software Livre**. 1.ed. Rio de Janeiro: Elsevier, 2018.

SILVERMAN, Richard E. **Git. Guia Prático**. São Paulo: Novatec, 2013.

Disciplina: Engenharia de requisitos

Ementa: Definição e características da Engenharia de Requisitos com a visão Ágil e o analista de requisitos. Tipos de requisitos: requisitos funcionais e não funcionais, requisitos de domínio, de usuário e de sistemas. Levantamento de requisitos: técnicas de reunião com o usuário e especificação de requisitos. Análise de requisitos, validação, mudança e gerência de requisitos.

Competências e Habilidades:

- Compreender que a Engenharia de Requisitos é importante como base para gerar um software de qualidade;
- Conhecer os tipos de requisitos necessários para a construção do software;
- Aplicar técnicas de reunião com o usuário para abstrair os requisitos do software;
- Analisar e gerenciar os requisitos levantados.

Conteúdo Programático 1: Definição e características da Engenharia de Requisitos com a visão Ágil e o analista de requisitos.

Conteúdo Programático 2: Tipos de requisitos: requisitos funcionais e não funcionais, requisitos de domínio, de usuário e de sistemas.

Conteúdo Programático 3: Levantamento de requisitos: técnicas de reunião com o usuário e especificação de requisitos.

Conteúdo Programático 4: Análise de requisitos, validação, mudança e gerência de requisitos.

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.

HIRAMA, Kechi. **Engenharia de Software**: qualidade e produtividade com tecnologia. Rio de Janeiro: Elsevier, 2011.

MACHADO, Felipe N. R. **Análise e Gestão de Requisitos de Software**: onde nascem os sistemas. 3. ed. São Paulo: Érica, 2016.

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: Codificação do projeto: padrões de programação e projeto do banco de dados

Ementa: Procedimentos de programação com a utilização das principais linguagens orientadas a objeto: Java, C#, Python e Ruby. Projeto da interface do usuário: análise do usuário e do conteúdo. Modelagem de dados e o projeto do fluxo de dados. Projetos de sistemas fazendo uso de tecnologias de Big Data e de Cloud Computing.

Competências e Habilidades:

- Escolher linguagens de programação para o desenvolvimento de softwares orientados a objetos;
- Analisar e modelar a interface do usuário com as funcionalidades e os fluxos de navegações;
- Saber modelar o banco de dados e produzir o fluxo de dados;
- Conhecer ambientes de sistemas em Big Data e ambientes que fornecem funcionalidade de Cloud Computing.

Conteúdo Programático 1: Procedimentos de programação com a utilização das principais linguagens orientadas a objeto: Java, C#, Python e Ruby.

Conteúdo Programático 2: Projeto da interface do usuário: análise do usuário e do conteúdo das funcionalidades.

Conteúdo Programático 3: Modelagem de dados e o projeto do fluxo de dados.

Conteúdo Programático 4: Projetos de sistemas fazendo uso de tecnologias de Big Data e Cloud Computing.

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.

HIRAMA, Kechi. **Engenharia de Software**: qualidade e produtividade com tecnologia. Rio de Janeiro: Elsevier, 2011.

TEOREY, Tobey J. ...[et al.] **Projeto e Modelagem de Banco de Dados**. 2. ed. Rio de Janeiro: Elsevier, 2014.

RODRIGUES MACHADO, Felipe Nery. **Big Data: O futuro dos dados e aplicações**. São Paulo: Érica, 2018.

CHEE, Brian J. S. e FRANKLIN JR, Curtis. **Computação em Nuvem – Cloud Computing – Tecnologias e Estratégias**. São Paulo: MBooks, 2013.

Disciplina: Gerenciamento Ágil dos sistemas

Ementa: Controlando o escopo: as métricas e as estimativas. Gestão de riscos e de custos com COCOMO II. Gestão da qualidade. Administração da manutenção do software.

Competências e Habilidades:

- Saber como administrar o escopo do sistema, determinando métricas e estimativas para o desenvolvimento;
- Conhecer métodos de gestão de riscos e de modelagem de custos com COCOMO II;

- Escolher métodos para controlar a qualidade de um software;
- Saber como controlar as alterações do software.

Conteúdo Programático 1: Controlando o escopo: as métricas e as estimativas.

Conteúdo Programático 2: Gestão de riscos e de custos com COCOMO II.

Conteúdo Programático 3: Gestão da qualidade.

Conteúdo Programático 4: Administração da manutenção do software.

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.

HIRAMA, Kechi. **Engenharia de Software**: qualidade e produtividade com tecnologia. Rio de Janeiro: Elsevier, 2011.

VINCENZI, Auri M. R. ...[et al.] **Automatização de Teste de Software com Ferramentas de Software Livre**. 1. ed. Rio de Janeiro: Elsevier, 2018.

Disciplina: Validação do software: Testes de softwares e aplicações de segurança no sistema

Ementa: Qualidade do software, documentação de testes e o modelo TMMi. Gerenciamento e automação dos testes de software com ferramentas CASE de

licença pública. Procedimentos para garantir a segurança do software.

Ferramentas CASE para testes de segurança do software.

Competências e Habilidades:

- Compreender a importância da aplicação de testes bem realizados, como uma das garantias da qualidade do software;
- Saber utilizar ferramentas para controlar a aplicação de testes;
- Saber como aplicar procedimentos de segurança em projetos de software;
- Conhecer algumas ferramentas para testar a segurança do software.

Conteúdo Programático 1: Qualidade do software, documentação de testes e o modelo TMMi.

Conteúdo Programático 2: Gerenciamento dos testes de software com ferramentas CASE de licença pública.

Conteúdo Programático 3: Procedimentos para garantir a segurança do software.

Conteúdo Programático 4: Ferramentas CASE para testes de segurança do software.

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.

HIRAMA, Kechi. **Engenharia de Software: qualidade e produtividade com tecnologia**. Rio de Janeiro: Elsevier, 2011.

VINCENZI, Auri M. R.[et al.] **Automatização de Teste de Software com Ferramentas de Software Livre**. 1. ed. Rio de Janeiro: Elsevier, 2018.

Disciplina: Evolução dos softwares: aplicação da Engenharia de Software em sistemas emergentes.

Ementa: Evolução e melhorias dos processos da Engenharia de Software. Engenharia de Software em aplicativos para dispositivos móveis e aplicações WEB (WebApps). Engenharia de Software no desenvolvimento de software de games. Desenvolvimento e gestão de projetos com DevOps.

Competências e Habilidades:

- Conhecer as novas tendências e melhorias nos processos da Engenharia de Software;
- Saber como aplicar a Engenharia de Software no desenvolvimento de aplicações WEB e aplicativos móveis;
- Saber como aplicar a Engenharia de Software no desenvolvimento de games;
- Conhecer a nova cultura DevOps para o desenvolvimento de softwares.

Conteúdo Programático 1: Evolução e melhorias dos processos da Engenharia de Software.

Conteúdo Programático 2: Engenharia de Software em aplicativos para dispositivos móveis e aplicações WEB (WebApps).

Conteúdo Programático 3: Engenharia de Software no desenvolvimento de softwares de games.

Conteúdo Programático 4: Desenvolvimento e gestão de projetos com DevOps.

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.

HIRAMA, Kechi. **Engenharia de Software**: qualidade e produtividade com tecnologia. Rio de Janeiro: Elsevier, 2011.

WAZLAWICK, Raul S. **Engenharia de Software**: conceitos e práticas. 2. ed. Rio de Janeiro: Elsevier, 2019.

Disciplina: Arquitetura de software

Ementa: Conceitos fundamentais. Projeto de arquiteturas. Modelagem de arquiteturas. Análise de arquiteturas. Implementação e implantação. Estilos arquiteturais aplicados. Projetos em requisitos de qualidade.

Competências e Habilidades:

- Compreender conceitos fundamentais acerca da arquitetura de *software*;
- Ser capaz de implementar projetos de arquitetura, desde a concepção até a gestão de diferentes processos;
- Modelar e analisar arquiteturas;
- Conhecer diferentes estilos arquiteturais aplicados.

- Compreender a relação dos requisitos de qualidade na escolha e modelagem de arquiteturas de software.

Conteúdo Programático 1: Conceitos Fundamentais (contextualização, arquiteturas, componentes, configurações, estilos e padrões arquiteturais).

Conteúdo Programático 2: Análise e Projeto de Arquiteturas de Software (concepção, ferramentas, arquiteturas para domínios específicos, metas, escopo).

Conteúdo Programático 3: Modelagem de arquiteturas (conceitos, aspectos em ambiguidade e precisão, técnicas comuns, nível de formalidade e automação, *frameworks*).

Conteúdo Programático 4: Estilos e Projetos Arquiteturais integrados a Requisitos de Qualidade (Implementação, implantação, e projetos arquiteturais de sistemas distribuídos, descentralizadas, orientados a serviços, orientado a microsserviços, requisitos de qualidade).

Bibliografia Básica:

BUSCHAMANN et al. **A System of Patterns**. 1. ed. West Sussex: Wiley, 1996.

FOROUZAN, Behrouz A. **Comunicação de Dados e Redes de Computadores**. 4. ed. São Paulo: McGraw-Hill, 2008.

GARLAN e SHAW. "Introduction to Software Architecture," *Advances in Software Engineering and Knowledge Engineering, Volume I*, edited by V.Ambriola and G.Tortora, **World Scientific Publishing Company**, New Jersey, 1996. Disponível em: <https://www.cs.cmu.edu/afs/cs/project/able/ftp/intro_softarch/intro_softarch.pdf>. Acesso em: 19 ago. 2018.

Lobo, Edson J. R. **Curso de Engenharia de Software**. 1. ed. São Paulo: Digerati Books, 2008.

PERRY e WOLF (1992, P. 44), **Foundations for the Study of Software Architecture**. Disponível em: <<http://users.ece.utexas.edu/~perry/work/papers/swa-sen.pdf>>. Acesso em: 19 ago. 2018.

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

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.

