



PROJETO PEDAGÓGICO DE CURSO

Pós-Graduação *Lato Sensu*

**Engenharia de software com ênfase em qualidade
e teste 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.....	5
7. Competências e Habilidades do Curso	6
8. Metodologia de Ensino e Aprendizagem	6
9. Matriz Curricular	9
10. Carga Horária.....	10
11. Conteúdo Programático.....	10
12. Infraestrutura Física e Pedagógica.....	26

1. Nome do Curso e Área do Conhecimento

Nome do Curso: Engenharia de software com ênfase em qualidade e teste 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: 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 profissionais formados em Ciência da Computação, Engenharia da Computação, Engenharia de Software, Sistemas de

Informação ou em outros cursos da área de Tecnologias da Informação, que queiram se aperfeiçoar na área de Engenharia de Software, no que tange a melhoria e garantia de Qualidade e Teste de Software. Profissionais egressos de outras áreas poderão se matricular, caso possuam conhecimento básicos em Engenharia de Software, Projeto de Software ou Teste de Software.

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

A busca pela criação de produtos de qualidade ocorre em diferentes setores da economia. Logo, esta busca atinge também os produtos de software. Com a Crise do Software, na década de 60, a Engenharia de Software tornou-se conhecida e difundida, pois concentra esforços em diferentes ferramentas, métodos e processos para atingir-se a qualidade esperada.

A evolução do hardware e aumento do tráfego de dados por meio da rede mundial de computadores, como quaisquer outras mudanças tecnológicas na história, leva ao surgimento de novas demandas de soluções de software, como serviços disponíveis na nuvem, integração com objetos e coisas – a chamada Internet das Coisas, a possibilidade de minerar e identificar oportunidades e auxiliar a tomada de decisões por meio de bases de dados amplas.

É nítida a complexidade destas novas demandas e também a ampliação da indústria de software, com destaque para a possibilidade de trabalho em *home office* ou escritórios geograficamente distantes. Agora, além da preocupação com a complexidade em se desenvolver softwares, há também

a necessidade de se gerir equipes que trabalham em diferentes fusos horários.

A divisão das equipes gera dificuldades principalmente no que tange desenvolver e testar os softwares. Nesta perspectiva, metodologias e técnicas que possam ser facilmente difundidas, compartilhadas e geridas se destacam. E, conseqüentemente, profissionais que detêm o conhecimento para exercer tais tarefas são nitidamente bem quistos e procurados por diversas empresas à nível nacional e internacional.

Contudo, para se tornar um profissional qualificado, experiência e formação são essenciais. Uma vez que garantir a qualidade considerando as especificidades do processo de desenvolvimento de software é um processo árduo, que somente o aprendizado em um conjunto de métodos, processos e ferramentas da Engenharia de Software, integradas e provenientes tanto da academia, quanto da indústria podem permitir tal qualificação.

Destarte, é nesta perspectiva e na necessidade de especializar profissionais de diversas áreas que este curso busca integrar disciplinas atuais, focadas em práticas da indústria em sinergia com fundamentos teóricos da academia, a suprir a carência do mercado por profissionais que dominem como atingir-se a qualidade de software, em especial por meio de testes de software, que este curso de especialização é proposto.

6. Objetivos do Curso

6.1. Objetivo Geral

- Capacitar e aperfeiçoar profissionais na análise, aperfeiçoamento e adoção de métodos, processos e ferramentas atuais da Engenharia de Software com ênfase em garantir uma maior qualidade no processo de desenvolvimento de software, desde o planejamento à evolução, estendendo tal qualidade e dinamismo também ao gerenciamento de equipes.

6.2. Objetivos Específicos

- Compreender o processo de desenvolvimento de software e suas especificidades no que tange a necessidade de se empregar métodos que garantam melhor qualidade;
- Analisar e criar documentos de requisitos funcionais e não funcionais otimizados a garantir o melhor atendimento possível às requisições dos clientes e suas necessidades, visando ainda a criação de projetos de software mais concisos, bem como casos de testes derivados de tais requisitos;
- Desenvolver a capacidade de gerenciar, com a adoção de práticas ágeis, as equipes de desenvolvimento de sistemas de software;
- Aplicar boas práticas no decorrer de todas as etapas de desenvolvimento, considerando métodos e processos emergentes, como o design de experiência de usuário, *design patterns* ou padrões de projeto, gerência de configuração, versionamento e automatização de tarefas;
- Aprender a inspecionar e testar softwares manualmente e de modo automatizado, visando soluções robustas e de excelência, considerando a integração de práticas da indústria e da academia.

7. Competências e Habilidades do Egresso

Competências

Atuar como profissional especializado na análise, seleção e aplicação de métodos, processos e ferramentas da Engenharia de Software para melhorar a qualidade de produtos de software e gerenciamento de equipes.

Atuar como gestor de equipes de qualidade, alinhar às necessidades dos projetos às demandas de qualidade necessárias a se atingir a solução solicitada e especificada nos requisitos.

Atuar coletando e analisando requisitos funcionais e não funcionais, para criação de projeto de software mais conciso e de maior qualidade. Permitindo ainda, gerar artefatos que serão utilizados nas etapas de inspeção e testes.

Atuar como analista de qualidade em indústrias de software, propondo e adotando meios mais assertivos de se obter qualidade no processo de desenvolvimento de software.

Habilidades

- Criar documentos de requisitos funcionais e não funcionais conciso e que possam gerar melhores projetos e artefatos para testes, inspeção e melhoria da qualidade de soluções de software;
- Compreender e estruturar as etapas do processo de desenvolvimento e as equipes envolvidas, integrando métodos, processos e ferramentas provenientes da Engenharia de Software, resultando em melhor qualidade;
- Aplicar práticas de gerenciamento de equipes e de suas atividades com foco em qualidade, por meio de práticas ágeis, *design patterns* ou padrões de projeto e gerência de configuração;
- Inspeccionar e Testar softwares de modo sistemático, por meio de práticas difundidas amplamente na indústria e na academia;
- Realizar o versionamento de software por meio de ferramentas que possibilitam a automatização de tarefas, incluindo testes;
- Selecionar ferramentas e métodos que melhor se adequem à necessidade do projeto e da equipe, no que tange a garantia de qualidade;

- Saber implementar o gerenciamento dos processos de testes de software e o modelo de maturidade de teste de software.

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 júbilamento 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 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

III. 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

Conceitos gerais e principais abordagens de desenvolvimento do software	0 h	40 h	40 h
Engenharia de requisitos	0 h	40 h	40 h
Gerenciamento ágil dos sistemas	0 h	40 h	40 h
Design de experiência do usuário - UX	0 h	40 h	40 h
Evolução dos softwares: aplicação da Engenharia de Software em sistemas emergentes	0 h	40 h	40 h
Validação do software: testes de softwares e aplicações de segurança no sistema	0 h	40 h	40 h
Teste e inspeção de software: técnicas e automatização	0 h	40 h	40 h
<i>Design Patterns</i> e gerência de configuração: do projeto ao controle de versões	0 h	40 h	40 h
Qualidade de software com <i>Clean Code</i> e técnicas de usabilidade	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: 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: 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: 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: Design de Experiência do Usuário - UX.

Ementa: Conceitos sobre ergonomia e psicologia cognitiva; Diretrizes para a concepção de interfaces WEB; Acuidade visual; Conceitos sobre design e análise de interface digital. Métodos e técnicas de usabilidade; Critérios e concepção de navegabilidade; Técnicas de avaliação e UX – Experiência do usuário. Web Design responsivo.

Competências e Habilidades:

- Entender conceitos sobre ergonomia e psicologia cognitiva;
- Desenvolver sistemas com um ambiente agradável ao usuário;
- Atentar para acuidade visual;
- Apreender conceitos básicos sobre design digital e Web Design Responsivo.

Conteúdo Programático 1: Conceitos sobre ergonomia e Usabilidade.

Conteúdo Programático 2: Interfaces WEB e técnicas de avaliação de IHC.

Conteúdo Programático 3: Análise da interface digital.

Conteúdo Programático 4: Acuidade visual, Design digital e Web Design Responsivo.

Bibliografia Básica:

AGNER, Luiz. **Ergodesign e arquitetura de informação**: trabalhando com o usuário. Rio de Janeiro: Quartet, 2006. 173p.

Dillon, A. **User interface design**. MacMillan Encyclopedia of Cognitive Science, London: MacMillan, 2003.

IIDA, Itiro. **Ergonomia**: projeto e produção. São Paulo: Edgard Blücher, 2001. 465p.

ISO 13407. Human-centred design processes for interactive systems. 1999.

ISO 9241. Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on Usability. 1998.

JASTRZEBOWSKI, W. An Outline of Ergonomics, or the Science of Work based on the truths drawn from the Science of Nature 1857. **Central Institute for Labour Protection**. 2000.

KO, Amy J. How to evaluate analytically. **UW Faculty Web Server**. Disponível no seguinte link: <http://faculty.washington.edu/ajko/books/design-methods/how-to-evaluate-analytically.html>.

Meister D. **Human Factors**: Theory and Practice. New York: Wiley-Interscience, 1971; 10.

NIELSEN, Jakob; LORANGER, Hoa. **Usabilidade na Web**. Rio de Janeiro: Elsevier, 2007.

SOMMERVILLE, Ian. **Software Engineering**. 8. Ed. England: Addison-Wesley, 2007.

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

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: Teste e inspeção de software: técnicas e automatização

Ementa: Conceitos sobre testes e Inspeção de software. Técnicas de teste de Software: unitário, de componentes, teste de sistema e outras técnicas de teste. Técnicas de inspeção de software: de leitura ad hoc, de leitura baseada em *checklists*, de leitura baseada em cenários, de leitura baseada em perspectivas e outras técnicas de inspeção. Selecionar e executar processos e casos de testes automatizados.

Competências e Habilidades:

- Compreender os conceitos básicos sobre teste de software e inspeção de software, identificando suas diferenças, vantagens e desvantagens na adoção dos mesmos;
- Conhecer, entender e propor os casos de testes;
- Utilizar uma ou mais técnicas de teste e de inspeção de software e os relatórios de registros de testes;
- Identificar as especificidades do projeto de software e selecionar técnicas de teste ou inspeção, de acordo com a solução a ser criada;
- Conhecer o objetivo, os processos e aplicabilidade dos testes automatizados.

Conteúdo Programático 1: Conceituando Teste e Inspeção de Software

Conteúdo Programático 2: Técnicas de Teste de Software

Conteúdo Programático 3: Técnicas de Inspeção de Software.

Conteúdo Programático 4: Processos e Casos de Testes automatizados.

Bibliografia Básica:

DELAMARO, Márcio, JINO, Mario, MALDONADO, José. **Introdução Ao Teste De Software**. 2 ed. Rio de Janeiro: Campus, 2016. 448p.

RIOS, Emerson, MOREIRA, Trayahú. **Teste de Software**. Rio de Janeiro: Alta Books Editora, 2013. 304p.

MALDONADO, José, DELAMARO, Márcio, e VINCENZI, Auri Marcelo Rizzo. **Automatização de teste de software com ferramentas de software livre**. São Paulo: Elsevier, 2018. 256p.

Bibliografia Complementar:

MATIELI, Luciana Vago. **Estimativa de Teste de Software: Levantamento Bibliográfico em Veículos Brasileiros e Internacionais**. Universidade Federal Fluminense. Relatórios de Pesquisa em Engenharia de Produção. v. 15 n. 1, 14p. 2016.

SANTOS, Luiz Diego Vidal, OLIVEIRA, Catuxe Varjão de Santana. **Introdução à garantia de qualidade de software**. Timburi: Cia do eBook, 2017. 39p.

PERES, Hugo. **Automatizando Testes de Software Com Selenium**. Rio de Janeiro: Simplissimo Livros Ltda, 2016. 88p.

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.

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: Qualidade de software com *clean code* e técnicas de usabilidade

Ementa: Fundamentos da Qualidade de Software. Conformidade de requisitos funcionais e de qualidade com produtos de software. Conhecendo e aplicando a filosofia de desenvolvimento *Clean Code*. Técnicas de Usabilidade: técnicas de concepção, de análise e de avaliação. Integrando a filosofia *Clean Code* e Técnicas de Usabilidade para melhoria da Qualidade de Software. Falácias e como evitá-las na obtenção da qualidade de software.

Competências e Habilidades:

- Entender a Qualidade de Software e sua importância no contexto da relação custo versus esforços de trabalho em equipes de desenvolvimento e como obter a qualidade de software por meio da conformidade de requisitos funcionais e não funcionais;
- Aplicar a filosofia *Clean Code* nas equipes de desenvolvimento;

- Identificar e selecionar técnicas de usabilidade e baseadas em experiência de usuário (UX) para alcançar melhor qualidade nos produtos de software;
- Analisar e compreender meios de integração da filosofia *Clean Code* com técnicas de usabilidade para mitigação de falácias no contexto de obtenção de qualidade de software.

Conteúdo Programático 1: Qualidade de Software: fundamentos e conformidade de requisitos.

Conteúdo Programático 2: *Clean Code*: a Filosofia do “Código Limpo”.

Conteúdo Programático 3: Técnicas de Usabilidade: melhorando a qualidade da experiência do usuário.

Conteúdo Programático 4: Integrando *Clen Code* e técnicas de usabilidade para ampliar a qualidade.

Bibliografia Básica:

MARTIN, Robert C. **Arquitetura Limpa: O guia do artesão para estrutura e design de software**. Rio de Janeiro: Alta Books, 2019. 432p.

CYBIS, Walter; BETIOL, Adriana Holtz; FAUST, Richard. **Ergonomia e usabilidade: conhecimentos, métodos e aplicações**. São Paulo: Novatec editora, 2017. 496p.

ZANIN, Aline et al. **Qualidade de Software**. Porto Alegre: SAGAH, 2018. 112p.

Bibliografia Complementar:

WAZLAWICK, Raul. **Engenharia de software: conceitos e práticas**. São Paulo: Elsevier Editora Ltda., 2019. 360p.

DE MESQUITA SPINOLA, Mauro; DE PAULA PESSÔA, Marcelo Schneck. **Gestão da qualidade no desenvolvimento de software**. Editora Senac São Paulo, 2019.

FEATHERS, Michael C. **Trabalho eficaz com código legado**. Porto Alegre: AMGH Editora, 2013. 408p.

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.

