



PROJETO PEDAGÓGICO DE CURSO
Pós-Graduação *Lato Sensu*
MBA em Gestão da Qualidade de Software

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	11
13. Infraestrutura Física e Pedagógica	13

1. Nome do Curso e Área do Conhecimento

Nome do Curso: MBA em Gestão da Qualidade de Software

Área de Avaliação (CAPES): Interdisciplinar

Grande Área (CAPES): Multidisciplinar

Área do Conhecimento (CAPES): Multidisciplinar

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 MBA em Gestão da Qualidade de Software está destinado a profissionais da área de TI que desejam se aprofundar na gestão da qualidade 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

Logo no início do uso da TI nas organizações o desenvolvimento de software era extremamente artesanal e não existia grandes preocupações com a qualidade do produto entregue. Foi aí que por volta da década 1970 começaram os primeiros movimentos em busca da qualidade do produto software de forma a atender os padrões e necessidades de negócios.

Ao longo dos últimos anos tem-se percebido cada vez mais a busca pelo desenvolvimento de software com qualidade e alinhado a estratégia de negócio. No entanto, para que de fato o software tenha a qualidade garantida, é necessária uma gestão adequada.

Neste sentido o MBA em Gestão da Qualidade de Software objetiva formar profissionais capazes de gerenciar a qualidade de software de forma que as entregas atendam aos requisitos de negócio.

6. Objetivos do Curso

6.1. Objetivo Geral

- Capacitar profissionais da área de TI a trabalhar com gerenciar a qualidade do software desenvolvido nas organizações.

6.2. Objetivos Específicos

- Apresentar os modelos e frameworks de gestão qualidade de software.
- Fornecer conhecimentos sobre o planejamento e execução de processo de software com qualidade.
- Compreender estrategicamente a utilização de ferramentas e tecnologias utilizadas na gestão da qualidade de software.

7. Competências e Habilidades do Egresso

Competências

Atuar no contexto empresarial identificando as ferramentas e tecnologias de gestão da qualidade de software que entreguem valor aos negócios atendendo aos requisitos propostos.

Habilidades

- Gerenciar a qualidade do software desenvolvido e entregue para as organizações.
- Identificar, mapear, planejar, avaliar, executar e monitorar a qualidade do software.
- Aplicar técnicas e ferramentas de testes e validação 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 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

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
Arquitetura de software	0 h	40 h	40 h
Evolução dos softwares: aplicação da Engenharia de Software em sistemas emergentes	0 h	40 h	40 h
CASE: as principais ferramentas no ciclo de desenvolvimento de softwares	0 h	40 h	40 h
Qualidade de software com Clean Code e técnicas de usabilidade	0 h	40 h	40 h
Teste e inspeção de software: técnicas e automatização	0 h	40 h	40 h
Validação do software: testes de softwares e aplicações de segurança no sistema	0 h	40 h	40 h
Gerenciamento ágil dos sistemas	0 h	40 h	40 h
Design Patterns e gerência de configuração: do projeto ao controle de versões	0 h	40 h	40 h
Gestão da qualidade e produtividade	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: 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: NcGraw-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.

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

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: 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: 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.

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: 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: 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 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: Gestão da qualidade e produtividade.

Ementa: Embasamento histórico e conceitual acerca de produtividade. Círculos de controle da qualidade. Definição de organização do trabalho e o arranjo físico, planejamento e controle da produção, controle estatístico de processo (CEP) -

Técnicas PERT e CPM – e planejamento das necessidades de materiais (MRP). Fundamentação de *Just in Time*, *Kanban* e *Kaisen* e de indicadores de gestão - *Key performance indicators* (KPI).

Competências e Habilidades:

- Reconhecer os conceitos de produtividade e a evolução dos processos.
- Planejar sistemas de controle da qualidade.
- Compreender e implantar técnicas de controle da qualidade.

Conteúdo Programático 1: Histórico e conceitos de produtividade.

Conteúdo Programático 2: *Layout* o impacto do arranjo físico.

Conteúdo Programático 3: CEP, MRP, *Just in Time*, *Kanban* e *Kaisen*.

Conteúdo Programático 4: *Key performance indicators* (KPI).

Bibliografia Básica:

CALDEIRA, J. **100 Indicadores de gestão: *Key performance indicators* (KPI).** Coimbra: Conjuntura Actual Editora SA., 2012.

NEUMANN, C. **Projeto de fábrica e layout.** 1.ed. Rio de Janeiro: Elsevier, 2015.

RAMOS, E. M. L. S.; AMMEIDA, S. S.; ARAÚJO, A. R. **Controle estatístico da qualidade.** Porto Alegre: Bookman Companhia Editora Ltda., 2013.

Bibliografia Complementar:

CHIAVENATO, I. **Gestão da produção: uma abordagem introdutória.** 3.ed. Barueri: Manole, 2014.

LOBO, R. N. **Gestão da qualidade.** 1.ed. São Paulo: Editora Érica, 2010.

PALADINI, E.P. **Gestão e avaliação da qualidade: uma abordagem estratégica.** São Paulo: Atlas, 2019.

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.

