



PROJETO PEDAGÓGICO DE CURSO
Pós-Graduação *Lato Sensu*
DevOps

Sumário

1. Nome do Curso e Área do Conhecimento.....	3
2. Características Técnicas do Curso	3
3. Público-alvo	4
4. Critérios de Seleção.....	4
5. Justificativa do Curso.....	4
6. Objetivos do Curso	5
7. Competências e Habilidades do Egresso.....	6
8. Metodologia de Ensino e Aprendizagem	7
Composição do Corpo Docente	9
9. Estágio Não Obrigatório	9
10. Matriz Curricular	10
11. Carga Horária.....	11
12. Conteúdo programático	11
13. Infraestrutura Física e Pedagógica	26

1. Nome do Curso e Área do Conhecimento

Nome do Curso: DevOps

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

Profissionais da área de desenvolvimento de sistemas e tecnologias ou áreas afins e também para pessoas que desejam atualizar seus conhecimentos com o mais atual gerenciamento de TI.

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 transformação digital aliada a metodologias ágeis possibilitou uma alta demanda por profissionais de TI em muitas empresas. Desenvolver sistemas computacionais de forma organizada, cooperada, rápida e com qualidade são requisitos primordiais para profissionais que desejam atuar em um mercado que cresce exponencialmente a cada ano.

Estima-se que até 2024 haverá mais vagas disponíveis no mercado de tecnologia do que profissionais capacitados. Outro fator determinante é que empresas de TI estão buscando por profissionais que possuem além de habilidades técnicas que envolvem rapidez e qualidade, pessoas que tenham como fundamento o trabalho de forma cooperada, organizada e com conhecimento compartilhado de forma remota.

Com a imensa necessidade da criação de sistemas ligados a modelos ágeis já é uma grande realidade nas empresas de software por isso que o mercado busca profissionais que saibam executar Devops para a integração e controle de sistemas de forma ágil, cooperada e segura.

Neste contexto, torna-se fator fundamental capacitar profissionais para atuar no mercado de para sanar gargalos que envolvam desenvolvimento e versionamento de projetos e principalmente formar cidadãos capazes para conectar com pessoas, trabalhar de forma cooperada, favorecendo a interação e tornando os projetos ainda mais dinâmicos.

6. Objetivos do Curso

6.1. Objetivo Geral

- Formar profissionais com conhecimentos fundamentais na integração de sistemas computacionais, desenvolvimento de software de forma cooperada e apoio as atividades que envolvem a realização de tarefas de forma ágil e contínua, otimizar a entrega de software de forma organizada e controlada, com uso de ferramentas que auxiliem o processo de controle de versões, melhorando a confiabilidade de produtos e serviços.

6.2. Objetivos Específicos

- Capacitar o aluno a elaborar e implementar ações referentes a integração contínua de processos;
- Capacitar o aluno em ferramentas, técnicas e metodologias em para automatizar processos e soluções tecnológicas;
- Promover a cultura de DevOps em processos de automatização, monitoramento e controle de informações.
- Capacitar o alunos em processos para agilizar e otimizar a criação e o gerenciamento da estrutura das aplicações.
- Auxiliar o aluno no controle de documento, código e relatórios que envolvam ferramentas de versionamento.

7. Competências e Habilidades do Egresso

Competências

Atuar em segmentos utilizando ferramentas que auxiliem o controle de versões, trazendo mais agilidade e qualidade aos projetos envolvidos. Reestabelecer feedbacks entre a equipe a fim de proporcionar mais relacionamento e resolução de problemas relacionados a versões de software e pro fim manipular ferramentas ágeis e de DevOps presentes nos ambientes e empresas proporcionando mais muito mais dinamismo.

Habilidades

O profissional com formação em DeVops estará apto a possuir habilidades que norteiam não só processos computacionais, mas como aqueles que englobam a criação e execução de ferramentas de versionamento e metodologias quem envolvam rapidez e qualidade, como:

- Executar o controle de versões em código para uma maior organização do projeto
- Identificar em ferramentas de versionamento processos que envolvam alteração no projeto
- Aplicar conceitos relativos a metodologia ágeis e cultura DeVops
- Aplicar processos de colaboração mutua entre os entes envolvidos na criação de projetos.
- Buscar o trabalho em times multidisciplinares de forma organizada e cooperada.
- Identificar e executar ferramentas de controle de versões e sua utilidade de acordo com as metodologias empregadas dentro das empresas.

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 condições seguintes:

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

Certificação

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

Composição do Corpo Docente

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

9. Estágio Não Obrigatório

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

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

- I. proporcionar o exercício do aprendizado comprometido com a realidade socioeconômica-política do país;

II. propiciar a realização de experiências de ensino e aprendizagem visando à educação profissional continuada, alicerçada no desenvolvimento de competências e habilidades e ao exercício do pensamento reflexivo e criativo; e

II. incentivar o trabalho de pesquisa e investigação científica, visando ao desenvolvimento da ciência, da tecnologia e da cultura.

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

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

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

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

10. Matriz Curricular

DISCIPLINAS	CH PRÁTICA	CH TEÓRICA	CH TOTAL
Ambientação	0 h	0 h	0 h
Evolução dos softwares: aplicação da Engenharia de Software em sistemas emergentes	0 h	40 h	40 h
Práticas da cultura DevOps no desenvolvimento de sistemas	0 h	40 h	40 h
Teste e inspeção de software: técnicas e automatização	0 h	40 h	40 h

Cloud computing	0 h	40 h	40 h
Qualidade de software com Clean Code e técnicas de usabilidade	0 h	40 h	40 h
Entrega e Implantação contínua (DevOps)	0 h	40 h	40 h
Arquiteturas e infraestruturas em nuvem	0 h	40 h	40 h
Interface de programação de aplicações (API) e Web Services	0 h	40 h	40 h
Site Reliability Engineering (SRE)	0 h	40 h	40 h

11. Carga Horária

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

12. Conteúdo programático

Disciplina: Ambientação

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

Competências e Habilidades:

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

Conteúdo Programático 1: Introdução ao Ensino a Distância.

Conteúdo Programático 2: Tecnologia e educação.

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

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

Bibliografia básica:

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

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

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

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

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

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

Bibliografia Complementar:

BAPTISTA, M. M. Internet: auxílio à educação. **BIBLOS - Revista do Instituto de Ciências Humanas e da Informação**, v. 16, p. 37- 44, [s. l.], 2004.

Disponível em: <http://hdl.handle.net/20.500.11959/brapci/23717>. Acesso em: 6 dez. 2022.

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

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

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

Disciplina: 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: Práticas da cultura DevOps no desenvolvimento de Sistemas

Ementa: Compreendendo os pilares da Cultura DevOps no contexto do desenvolvimento Full Stack; Ferramentas do Ciclo de Vida do DevOps com foco na Integração Contínua. Tecnologias Git com Integração Contínua na Prática. Gerenciamento de equipes no desenvolvimento web com DevOps.

Competências e Habilidades:

- Compreender e Adotar a Cultura DevOps para departamentos de Tecnologia da Informação e Equipes de Desenvolvimento;
- Integrar as práticas da cultura DevOps no contexto de desenvolvedores Full Stack e das soluções implementadas;
- Conhecer e selecionar ferramentas específicas para etapas diferentes do processo de desenvolvimento de software;
- Utilizar ferramentas Git e de integração contínua na prática, com projetos ponto a ponto.

Conteúdo Programático 1: Conceitos Essenciais sobre a Cultura DevOps aplicada ao Desenvolvimento de Sistemas.

Conteúdo Programático 2: Selecionando Conhecendo Ferramentas do DevOps no Desenvolvimento Web.

Conteúdo Programático 3: Utilizando Tecnologia de Versionamento Git e de Integração Contínua na Prática.

Conteúdo Programático 4: Adoção das Práticas DevOps em Equipes de Desenvolvimento.

Bibliografia Básica:

KIM, Gene; HUMBLE, Jez; DEBOIS, Patrick; WILLIS, John. **Manual De DevOps: Como obter agilidade, confiabilidade e segurança em organizações tecnológicas.** Rio de Janeiro: Alta Books, 2020. 464p.

MORAES, Gleicon. **Caixa de Ferramentas DevOps: Um guia para construção, administração e arquitetura de sistemas modernos.** São Paulo: Editora Casa do Código, 2015. 176p.

SATO, Danilo. **DevOps na prática: Entrega de software confiável e automatizada**. São Paulo: Editora Casa do Código, 2014. 248p.

Bibliografia Complementar:

LWAKATARE, Lucy Ellen et al. **DevOps in practice: A multiple case study of five companies**. Information and Software Technology, v. 114, p. 217-230, 2019.

RATO, Francisco de Oliveira. **DevOps em sistemas de informação: implementação em operações de tecnologias de informação**. Dissertação (Mestrado em Gestão de Informação) - Universidade Nova de Lisboa. Lisboa, 50p., 2018.

BRAGA, Filipe Antônio Motta. **Um panorama sobre o uso de práticas DevOps nas indústrias de software**. Dissertação (Mestrado em ciência da computação) - Universidade Federal de Pernambuco. Recife, 124p., 2015.

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: Cloud computing

Ementa: Introdução a *cloud computing* (nuvem). Arquitetura de *cloud computing*. Governança de TI e *cloud computing*. Infraestrutura de nuvem. Serviços de nuvem. Aplicações.

Competências e Habilidades:

- Conhecer as vantagens e desvantagens da computação em nuvem.
- Tomar decisões de projeto e aquisição de serviços em nuvem.
- Desenvolver serviços em nuvem cumprindo requisitos de segurança da informação.

Conteúdo Programático 1: Introdução à *cloud computing*: Arquitetura e infraestrutura.

Conteúdo Programático 2: Governança de TI e *cloud computing*.

Conteúdo Programático 3: Serviços de nuvem.

Conteúdo Programático 4: Principais aplicações e futuro em *cloud computing*.

Bibliografia Básica:

CHEE, Brian J. S.; FRANKLIN, Curtis. **Computação em nuvem:** cloud computing: tecnologias e estratégias. São Paulo: M. Books do Brasil, 2013.
DIÓGENES, Yuri; VERAS, Manoel. **Certificação Cloud Essentials:** guia preparatório para o exame CLO - 001. Rio de Janeiro: Novaterra, 2014.
VELTE, Anthony T.; VELTE, Toby J.; ELSENPETER, Robert. **Cloud computing:** computação em nuvem, uma abordagem prática. Rio de Janeiro: Alta Books, 2011.

Bibliografia Complementar:

MORAES, Alexandre Fernandes de. **Telemática convergência de dados, voz e imagem.** São Paulo: Erica, 2014.

TAURION, Cezar. **Cloud computing**: computação em nuvem: transformando o mundo da tecnologia da informação. Rio de Janeiro: Brasport, 2009.

VERAS, Manoel. **Cloud Computing**: nova arquitetura da TI. Rio de Janeiro: Brasport, 2012.

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: Entrega e Implantação contínua (DevOps)

Ementa: Conceitos teóricos sobre Implementação contínua, benefícios de CI/CD ferramentas de entrega e implementação continua, diferença entre Integração, Entrega e Implementação contínua, Integração Contínua vs. Entrega vs. Melhores práticas e monitoramento de CI/CD, servidor de CI.

Competências e Habilidades:

- Compreender as diferenças que norteiam Integração, Entrega e Implantação Contínuas.

- Fornecer feedback rápido de cada estágio de CI/CD
- Executar de forma prática a entrega e implementação contínua em Devops

Conteúdo Programático 1: Visão geral de integração e entrega em DevOps

Conteúdo Programático 2: Implementação e utilização de ferramentas de entrega e implementação contínua

Conteúdo Programático 3: Melhores práticas e monitoramento CI/CD,

Conteúdo Programático 4: Servidor de CI.

Bibliografia Básica (inserir 3 bibliografias – atuais):

Kim, Gene. **Manual de DevOps: como obter agilidade, confiabilidade e segurança em organizações tecnológicas**. 1ed. Alta Books. 2018

Sato, Danilo. **Devops Na Pratica**. Editora Casa do código. 2014.

MUNIZ, Antonio et al. **Jornada DevOps: unindo cultura ágil, Lean e tecnologia para entregar software com qualidade**. 2ed. Brasport. 2020

Bibliografia Complementar (inserir 3 bibliografias atuais):

KIM, Gene; BEHR, Kevin; SPAFFORD, George. **O Projeto Fênix: Um romance sobre TI, DevOps e sobre ajudar o seu negócio a vencer**. Alta Books, 2017.

MUNIZ, Antonio et al. **Jornada RH Ágil: entenda como as Relações Humanizadas colaboram para construir times protagonistas e resultados de valor**. Brasport, 2021.

Muniz, A., & Irigoyen, A. (2019). **Jornada Ágil e Digital**. Brasport. 2019

Disciplina: Arquiteturas e infraestruturas em nuvem

Ementa: Conceito de arquitetura em nuvem. Principais provedores de nuvem e tipos de serviços gerais. Participação no mercado de nuvem. Modelos de nuvens: SaaS, PaaS e IaaS. Estudos de caso de arquiteturas em nuvem aplicadas a Banco de dados, Internet das Coisas e aplicações. Provisionamento de custos de nuvem.

Competências e Habilidades:

- Analisar as arquiteturas disponíveis para implementá-las em projetos de virtualização nas empresas.
- Descrever as principais oportunidades de migração de dados para ambiente de nuvem considerando os custos envolvidos.
- Desenhar modelos de arquiteturas em *high level* para apoiar os executivos na tomada de decisão quanto ao uso de seus ativos em nuvem.
- Gerenciar, especificar e documentar arquiteturas de serviços em nuvens, indicando as tecnologias e as ferramentas mais apropriadas para cada tipo de projeto.

Conteúdo Programático 1: Conceito de arquitetura em nuvem. Principais provedores de nuvem e tipos de serviços gerais. Federação. Participação no mercado de nuvem. Relação da arquitetura com a infraestrutura local, híbrida e em nuvem.

Conteúdo Programático 2: Apresentação do modelo SaaS – Software as a Service, objetivos, arquitetura e aplicações. Apresentação do modelo PaaS – Plataforma as a Service, objetivos, arquitetura e aplicações. Apresentação do modelo IaaS – Infraestrutura as a Service, objetivos, arquitetura e aplicações.

Conteúdo Programático 3: Estudo de caso de arquiteturas para provisionar armazenamento de banco de dados. Estudo de caso de arquiteturas para provisionar infraestrutura e serviços em IoT – Internet das Coisas. Estudo de caso de arquiteturas para provisionar serviços de armazenamento e aplicações.

Conteúdo Programático 4: Projeto de provisionamento com arquiteturas mistas e projeção financeira de custos.

Bibliografia Básica:

LOPER, Adriene Aparecida; SILVA, Nathalia dos Santos; LOPES, Giancarlo Michelino Gaeta. **Projeto de redes e sistemas distribuídos**. Londrina: Editora e Distribuidora, 2019.

VERAS, M. **Computação em Nuvem. Nova arquitetura de TI**. Rio de Janeiro: Brasport, 2015. eBook.

VERAS, M. **Virtualização. Tecnologia Central de Datacenter**. Rio de Janeiro: Brasport, 2016.

Bibliografia Complementar:

FERREIRA, Sergio da Costa. **Sistemas de informação em segurança**. Londrina: Editora e Distribuidora Educacional, 2017.

DANIEL, R. **Cloud Computing. Planejando a sua jornada**. Auto publicação, 2017.

FOWLER, S. J. **Microserviços prontos para a produção**. São Paulo: Novatec, 2017.

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

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

Competências e Habilidades:

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

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

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

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

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

Bibliografia Básica:

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

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

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

Bibliografia Complementar:

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

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

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

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

Disciplina: Site Reliability Engineering (SRE).

Ementa: Introdução à Engenharia de Confiabilidade de Site, Princípios, responsabilidades e práticas fundamentais da SRE: ciclos virtuosos, Princípios

e práticas fundamentais da SRE: O lado humano da SRE, diferença entre SREs e DevOps, o engenheiro de SREs, Framework SRE

Competências e Habilidades:

- Capacidade de medir indicadores de performance, risco, custo, entrega, qualidade e disponibilidade de uma aplicação e/ou produto.
- Ser capaz de estabelecer limites de nível de serviço]
- Estabelecer práticas de aprimoramento de serviço de confiabilidade por meio de um misto de automações, métodos de trabalho e realinhamento organizacional.
- Compreender e fomentar para fortalecer a Cultura DevOps numa organização criando uma Comunidade de Prática SRE

Conteúdo Programático 1: SRE: Engenharia de Confiabilidade do site, Relação entre DevOps e SRE e diferença entre profissionais.

Conteúdo Programático 2: XX ciclos virtuosos, Princípios e práticas fundamentais da SRE.

Conteúdo Programático 3: Aplicação e diferenças de SLIs, SLOs e Error Budget

Conteúdo Programático 4: Plano de recuperação de desastres e Framework SRE.

Bibliografia Básica (inserir 3 bibliografias – atuais):

Bayer, Betsy; Murphy, Richard Niall; Petooff, Jennifer; Jones, Cris. **Engenharia de Confiabilidade do Google: Como o Google administra seus sistemas de produção**. 1d. Novatec Editora. 2016.

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

Bibliografia Complementar (inserir 3 bibliografias atuais):

DE SIQUEIRA, Iony Patriota. **Manutenção centrada na confiabilidade: manual de implementação.** Qualitymark, 2005.

SCAPIN, Carlos Alberto. **Análise sistêmica de falhas.** 2d. Falconi Editora, 2014
LAFRAIA, João Ricardo Barusso. **Manual De Confiabilidade, Manutenibilidade: E DISPONIBILIDADE.** Qualitymark Editora Ltda, 2001

MUNIZ, Antonio et al. **Jornada DevOps: unindo cultura ágil, Lean e tecnologia para entregar software com qualidade.** 2d. Brasport. 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.

