**CURSO DE ENGENHARIA DE SOFTWARE**

|  |  |
| --- | --- |
| **1. CARACTERIZAÇÃO DA DISCIPLINA** | |
| Nome da Disciplina: **Prática Profissional I** | Ano/semestre: **2022/1** |
| Código da Disciplina: **08933** | Período: **7º** |
| Carga Horária Total: **80h/a** | Carga Horária Prática: **80h/a**  Carga Horária Teórica: **0h/a** |
| Pré-Requisito: **Não se aplica** | Co-Requisito: **Não se Aplica** |
| **2. PROFESSOR(ES)** | |
| William P. S. Júnior, Me. | |
| **3. EMENTA** | |
| Fábrica de Software, modelos e componentes. Metodologias e processos para gestão e desenvolvimento de software. Análise e projetos de software orientados a objetos (problemas reais). Gerir projetos de software. Processos de software. Ferramentas. Gestão de projetos ágil e/ou tradicional (Integração, Escopo, Tempo, Custos, Qualidade, Recursos Humanos, Comunicações, Riscos, Aquisições e Partes Interessadas). Modelos de ciclo de vida. | |
| **4. OBJETIVO GERAL** | |
| Praticar a gestão e desenvolvimento de um software com base em metodologias ágeis, assim como na análise e projeto orientados a objetos. | |

|  |  |
| --- | --- |
| **5. OBJETIVOS ESPECÍFICOS** | |
| **Unidades** | **Objetivos Específicos** |
| Fábrica de Software | - Conhecer a estrutura organizacional de uma Fábrica de Software;  - Compreender os modelos e componentes de uma Fábrica de Software. |
| Gestão de Software | - Compreender as metodologias ágeis para gestão e desenvolvimento de software; - Apontar atividades, técnicas, métodos e ferramentas que auxiliam na gestão, processo, análise e projeto de software. |
| Processo de software OpenUP/Basic | - Compreender o ciclo de vida, disciplinas, práticas, artefatos e demais componentes do processo de desenvolvimento de software iterativo OpenUP. |
| Análise e projeto de software. | - Recordar os conceitos fundamentais da modelagem de sistemas orientados a objetos; - Compreender os principais conceitos utilizados em análise e projeto de software orientado a objeto;  - Analisar e projetar projetos de software orientado a objetos; |
| Utilização de ferramentas CASE. | - Aplicar os conceitos adquiridos na resolução de projetos reais usando as ferramentas adequadas. |
| Aplicação das fundamentações teóricas. | - Praticar os conceitos da gerência de projetos de software ágil usando a metodologia Scrum; - Elaborar produtos de trabalho (artefatos), tendo como base escopos de projetos reais; - Desenvolver habilidades de comunicação e trabalho em grupo, por meio de atividades práticas (individuais e em grupo). |
| Formação geral. | - Formar um cidadão reflexivo, autônomo e consciente, com responsabilidade social, civil e penal sob a tutela da informação. |

|  |
| --- |
| **6. HABILIDADES E COMPETÊNCIAS** |
| O Curso de Bacharelado em Engenharia de Software visa à formar profissionais que possuam competências e habilidades gerais, bem como competências e habilidades específicas para atuação profissional.  **Habilidades e competências gerais:**  G.3. Tomar decisões, avaliando criticamente, soluções computacionais, consciente dos aspectos humanos, éticos, legais e ambientais decorrentes;  G.4. Gerir a própria aprendizagem e desenvolvimento pessoal-profissional, realizando trabalho em equipe, com visão trans e interdisciplinar;  G.5. Desenvolver trabalhos e soluções, adotando metodologias diversificadas;  G.6. Identificar novas oportunidades de negócios e empreender, desenvolvendo soluções inovadoras, baseado em experiências e experimentos, exercendo liderança na área de atuação profissional;  G.8. Resolver problemas usando ambientes de programação.  **Habilidades e competências específicas:**  E.1. Investigar, compreender e estruturar as características de domínios de aplicação em diversos contextos para a construção de sistemas de software, considerando questões éticas, sociais, legais e econômicas, individualmente e/ou em equipe;  E.2. Compreender e aplicar processos, técnicas e procedimentos de construção inerentes à produção e utilização de software, conhecendo os direitos e propriedades intelectuais;  E.3. Avaliar a qualidade e evolução de sistemas de software, aplicando adequadamente normas técnicas, através de padrões e boas práticas no desenvolvimento de software;  E.4. Identificar e analisar problemas, avaliando as necessidades dos clientes, especificar os requisitos de software, projetar, desenvolver, implementar, verificar, integrar e documentar soluções de software baseadas no conhecimento apropriado de teorias, modelos e técnicas. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **7. CONTEÚDO PROGRAMÁTICO** | | | | | |
| **Semana** | **Data** | **Conteúdo** | **Estratégia de ensino-aprendizagem** | **Aula**  **Teórica/**  **Prática** | **Local** |
| **1** | 08/02 | Apresentação da disciplina (plano de ensino, discussão sobre background, presença, horário, avaliação e comunicação professor-alunos) | Leitura da referência bibliográfica  Objeto de aprendizagem  Aula  Atividade pré-aula  Atividade pós-aula  Atividade Prática Laboratorial | Prática | AVA / FTT |
| **2** | 15/02 | Definição dos Grupos e Projetos | Leitura da referência bibliográfica  Objeto de aprendizagem  Aula  Atividade pré-aula  Atividade pós-aula  Atividade Prática Laboratorial | Prática | AVA / FTT |
| **3** | 22/02 | Fábrica de Software: | Leitura da referência bibliográfica  Objeto de aprendizagem  Aula  Atividade pré-aula  Atividade pós-aula  Atividade Prática Laboratorial | Prática | AVA / FTT |
| **4** | 08/03 | Processo de Software | Leitura da referência bibliográfica  Objeto de aprendizagem  Aula  Atividade pré-aula  Atividade pós-aula  Atividade Prática Laboratorial | Prática | AVA / FTT |
| **5** | 15/03 | ComVocAÇÃO  Apresentar o processo elaborado. | Leitura da referência bibliográfica  Objeto de aprendizagem  Aula  Atividade pré-aula  Atividade pós-aula  Atividade Prática Laboratorial | Prática | AVA / FTT |
| **6** | 22/03 | Metodologias ágeis: Scrum, XP, FDD, ASD e Iconix Process.  Elaborar processo híbrido para o projeto contendo: fases, atividades, boas práticas, ferramentas, artefatos, papéis | Leitura da referência bibliográfica  Objeto de aprendizagem  Aula  Atividade pré-aula  Atividade pós-aula  Atividade Prática Laboratorial | Prática | AVA / FTT |
| **7** | 29/03 | Análise orientada a objetos. Diagramas da UML: histórico, conceitos e visão geral.  Organização dos grupos para o Seminário sobre a UML 2.5 | Leitura da referência bibliográfica  Objeto de aprendizagem  Aula  Atividade pré-aula  Atividade pós-aula  Atividade Prática Laboratorial | Prática | AVA / FTT |
| **8** | **05/04** | **1ª Verificação de Aprendizagem** | **VA** | Prática | Presencial. |
| **9** | **12/04** | 1ª Sprint – documentação (pacote de trabalho 1): - Visão do produto (Elevator statement, Vision box ou ConOps) - Requisitos (Product backlog inicial priorizado – CRC ou história de usuário e Casos de Uso)  Elaborar uma proposta de um MVP do projeto de acordo com o exemplo do "Caso do FoodTruck Mexicano", postado em Objeto de Aprendizagem. | Leitura da referência bibliográfica  Objeto de aprendizagem  Aula  Atividade pré-aula  Atividade pós-aula  Atividade Prática Laboratorial | Prática | AVA / FTT |
| **10** | **19/04** | **Apresentação da Proposta do MVP** | Leitura da referência bibliográfica  Objeto de aprendizagem  Aula  Atividade pré-aula  Atividade pós-aula  Atividade Prática Laboratorial | Prática | AVA / FTT |
| **11** | **26/04** | 1ª Sprint – documentação(pacote de trabalho 2):  - Fluxo geral do processo de negócio do produto (BPMN/BizAgi)  - Riscos (Planilha OpenUP)  - Plano de projeto ágil, incluindo as estimativas (Planilha OpenUP)  - Quadro de tarefas (Kanban board), incluindo o Product Backlog atualizado e priorizado  - Diagramas da UML  - Documento de Arquitetura  - Lições aprendidas até o momento  - Outros, a depender do processo definido | Leitura da referência bibliográfica  Objeto de aprendizagem  Aula  Atividade pré-aula  Atividade pós-aula  Atividade Prática Laboratorial | Prática | AVA / FTT |
| **12** | **03/05** | Implementação dos Projetos | Leitura da referência bibliográfica  Objeto de aprendizagem  Aula  Atividade pré-aula  Atividade pós-aula  Atividade Prática Laboratorial | Prática | AVA / FTT |
| **13** | **10/05** | **SINACEM** | Leitura da referência bibliográfica  Objeto de aprendizagem  Aula  Atividade pré-aula  Atividade pós-aula  Atividade Prática Laboratorial | Prática | AVA / FTT |
| **14** | **17/05** | **2ª Verificação de Aprendizagem** | VA. | Prática | Presencial |
| **15** | **24/05** | Entrega da Segunda Verificação de Aprendizagem.  Planejamento e execução da 2ª Sprint (pacote de trabalho 3):  Código-fonte (pelo menos uma funcionalidade que agregue valor ao produto)  - Plano de testes ágil  - Critérios de aceitação  - Casos de testes unitários  - Demais artefatos atualizados  - Outros, a depender do processo definido | Leitura da referência bibliográfica  Objeto de aprendizagem  Aula  Atividade pré-aula  Atividade pós-aula  Atividade Prática Laboratorial | Prática | AVA / FTT |
| **16** | **31/05** | Implementação dos Projetos | Leitura da referência bibliográfica  Objeto de aprendizagem  Aula  Atividade pré-aula  Atividade pós-aula  Atividade Prática Laboratorial | Prática | AVA / FTT |
| **17** | **07/06** | Revisão e retrospectiva da 2ª Sprint Preparação para a banca examinadora (Scrum checklist) | Leitura da referência bibliográfica  Objeto de aprendizagem  Aula  Atividade pré-aula  Atividade pós-aula  Atividade Prática Laboratorial | Prática | AVA / FTT |
| **18** | **14/06** | Banca examinadora | Leitura da referência bibliográfica  Objeto de aprendizagem  Aula  Atividade pré-aula  Atividade pós-aula  Atividade Prática Laboratorial | Prática | AVA / FTT |
| **19** | 21/06 | Banca examinadora.  Fixação de conteúdo específico e de formação geral. | Leitura da referência bibliográfica  Objeto de aprendizagem  Aula  Atividade pré-aula  Atividade pós-aula  Atividade Prática Laboratorial | Prática | AVA / FTT |
| **20** | **28/06** | **3ª Verificação de Aprendizagem** | **VA** | Teórica | Presencial |

|  |
| --- |
| **8. PROCEDIMENTOS DIDÁTICOS** |
| **Metodologia:** Atividade avaliativa on-line no Ambiente virtual de Aprendizagem, prova de verificação de aprendizagem, aula expositiva dialogada, sala invertida, tempestade de ideias, mapa conceitual, retomada de conteúdo, estudo de caso, e Tecnologias da Informação e Comunicação – AVA – plataforma Moodle com Vídeo do YouTube, Videoaula de introdução do professor/animação, Imagem explicativa ou Infográfico dentre outros. Sistema Academico Lyceum. Artefaots da disciplina (BrModelo/Bizagi/MySQL...), Questionário online (socrative/mentimeter/survio/googleForms).  **Recursos educativos:** Quadro-branco/pincel, projetor multimídia, livros, ebook, artigos científicos, computador, celular e internet.  **Recursos de Acessibilidade disponíveis aos acadêmicos :** O curso assegura acessibilidade metodológica, digital, comunicacional, atitudinal, instrumental e arquitetônica, garantindo autonomia plena do discente. |
| **9. ATIVIDADE INTEGRATIVA** |
| A interdisciplinaridade no curso de Engenharia de Software é construída com o amparo das disciplinas de Prática Interdisciplinar. Estas promovem a associação entre os diferentes conteúdos, habilidades e cenários em projetos que favoreçam a construção do conhecimento científico, tecnológico e de prática profissional aliado à autoaprendizagem, proatividade, resolução conjunta de problemas, trabalho em equipe e reflexividade, entre outros.   A proposta de cada disciplina de Prática Interdisciplinar é variável, mas, obrigatoriamente, deve evoluir em uma constante de maturidade pessoal, interpessoal, científica e prática. Para o desenvolvimento dos trabalhos interdisciplinares o aluno percorre três momentos: ensino – por meio do diálogo entre as áreas de conhecimento; pesquisa - seguindo os rigores metodológicos necessários à construção do conhecimento científico e de extensão – oportunizando o compartilhamento dos projetos desenvolvidos para o público interno e externo.  Em função disto, as atividades de cada trabalho estão detalhadas nos Planos de Ensino de Prática Interdisciplinar. |
| **10. PROCESSO AVALIATIVO DA APRENDIZAGEM** |
| **1ª Verificação de aprendizagem (V. A.)** – valor 0 a 100 pontos  Avaliação com valor 0 a 50 pontos.  Avaliações processuais totalizam 0 a 50 pontos distribuídos da seguinte forma:  **APS:** 12 pts divididos em 8 atividades  **Aprendendo a Resolver Problemas - ARP**: 10 pts em uma atividade.  **Atividade Livre**: 28 pts a serem distribuídos a critério do professor.  A média da 1ª V. A. será a somatória da nota obtida na avaliação teórica (0 a 50 pontos) e as notas obtidas nas avaliações processuais (0 a 50 pontos).  (a devolutiva será realizada conforme Cronograma).  **2ª Verificação de aprendizagem (V. A.)** – valor 0 a 100 pontos  Ex.: Avaliação com valor 0 a 50 pontos.  Avaliações processuais totalizam 0 a 50 pontos distribuídos da seguinte forma:  **APS**: 7,5 pts divididos em 5 atividades  **Aprendendo a Resolver Problemas - ARP:** 10 pts em uma atividade.  **Atividade Livre**: 32,5 pts a serem distribuídos a critério do professor.  A média da 2ª V. A. será a somatória da nota obtida na avaliação teórica (0 a 50 pontos) e a nota obtida nas avaliações processuais (0 a 50 pontos).  (a devolutiva será realizada conforme Cronograma).  **3ª Verificação de aprendizagem (V. A.)** – valor 0 a 100 pontos  Ex.: Avaliação com valor 0 a 50 pontos.  Avaliações processuais totalizam 0 a 50 pontos distribuídos da seguinte forma:  **APS**: 6 pts divididos em 4 atividades  **Aprendendo a Resolver Problemas - ARP:** 10 pts em uma atividade.  **Atividade Livre**: 34 pts a serem distribuídos a critério do professor.  A média da 3ª V. A. será a somatória da nota obtida na avaliação teórica (0 a 50 pontos) e nota obtida nas avaliações processuais (0 a 50 pontos).  A devolutiva será realizada conforme Cronograma. Caso o aluno tenha a disciplina Projeto Interdisciplinar na grade curricular deste semestre, a composição da 3ªVA (0 a 100 pontos) descrita anteriormente corresponderá a 70% da 3ªVA e a nota SITES a 30%.  **ORIENTAÇÕES ACADÊMICAS**   * Nas três VAs – O pedido para avaliação substitutiva tem o prazo de 3 (três) dias úteis a contar da data de cada avaliação com apresentação de documentação comprobatória (§ 1º e § 2º do art. 39 do Regimento Geral do Centro Universitário UniEVANGÉLICA). **A solicitação deverá ser feita através do Sistema Acadêmico Lyceum obrigatoriamente.** * Nas três VAs – O pedido para Revisão de nota tem o prazo de 3 (três) dias úteis a contar da data da publicação, no Sistema Acadêmico Lyceum, do resultado de cada avaliação. (Art. 40 do Regimento Geral do Centro Universitário UniEVANGÉLICA). * Atribui-se nota zero ao aluno que deixar de submeter-se às verificações de aprendizagem nas datas designadas, bem como ao que nela se utilizar de meio fraudulento. (Capítulo V Art. 39 do Regimento Geral do Centro Universitário UniEVANGÉLICA)   **Participação em eventos científicos:**  **Portaria – Frequência e nota dos alunos que apresentarem trabalhos em eventos científicos**  Seguir as orientações presentes na Portaria Nº 01, de 7 de fevereiro de 2019, dos Bacharelados em Computação, que dispõe sobre os procedimentos de justificativa de ausência para alunos que apresentarem trabalhos em eventos científicos.  **Condição de aprovação**  Considera-se para aprovação do (a) acadêmico (a) na disciplina, frequência mínima igual ou superior a 75% da carga horária e nota igual ou superior a sessenta (60) obtida com a média aritmética simples das três verificações de aprendizagem. |

|  |
| --- |
| **11. BIBLIOGRAFIA** |
| **Básica:**  Um Guia para o Conhecimento em Scrum (Guia SBOKTM). Edição 2016. SCRUMStudy.  COHN, M. Desenvolvimento de Software com Scrum – Aplicando Métodos Ágeis com Sucesso. Bookman, 2012.  SOMMERVILLE, Ian. Engenharia de Software. 9. ed. São Paulo: Pearson Addison Wesley, 2011. 529 p.  **Complementar:**  PHAM, A.; PHAM, Phuong-Van. Scrum em Ação: Gerenciamento e Desenvolvimento Ágil de Projetos de Software. Novatec, 2011. 288 p.  FERNANDES, A. A.; TEIXEIRA, D. S. Fábrica de Software – Implantação e Gestão de Operações. São Paulo: Atlas, 2011.  Um Guia Do Conhecimento Em Gerenciamento De Projetos (Guia PMBOK®) – Quinta Edição. Editora Saraiva. 2017  GUEDES, Gilleanes T. A. UML 2: uma abordagem prática. 3ª. ed. São Paulo, SP, Brasil: Novatec, 2018. 484 p.  SCHACH, Stephen R. Engenharia de Software: os paradigmas clássico e orientado a objetos. 7 ed. São Paulo: MCGraw-Hill, 2009. 618 p. |

Anápolis, 27 de janeiro de 2022.

**Profa. M.e Natasha Sophie Pereira**

COORDENADORA DO CURSO DE ENGENHARIA DE SOFTWARE DA UniEVANGÉLICA

**Profº. M.e. William P. Santos Júnior**

PROFESSORA RESPONSÁVEL PELA DISCIPLINA