

CURSO DE ENGENHARIA DE SOFTWARE

1. CARACTERIZAÇÃO DA DISCIPLINA

Nome da Disciplina: Prática Profissional II	Ano/semestre: 2021/2
Código da Disciplina: 08939	Período: 8º
Carga Horária Total: 80h/a	Carga Horária Teórica: 00h/a Carga Horária Prática: 80h/a
Pré-Requisito: Não se Aplica	Co-Requisito: Não se Aplica

2. PROFESSOR(ES)

William Pereira dos Santos Júnior, M.e

3. EMENTA

Desenvolvimento de Projetos reais de Software aplicados em um ambiente de desenvolvimento acadêmico.

4. OBJETIVO GERAL

Proporcionar que o discente exercite as técnicas de desenvolvimento, aprendidas ao longo do curso, em projetos acadêmicos.

5. OBJETIVOS ESPECÍFICOS

Unidades	Objetivos Específicos
Requisitos e Banco de Dados	Aplicar conceitos de mapeamento objeto relacional para o desenvolvimento de software.
Metodologia de desenvolvimento, arquitetura de software e padrões de projeto	Aplicar os padrões para o desenvolvimento de software e desenvolvimento web.
Metodologia de desenvolvimento.	Criar aplicação guiada por resultados.
Políticas públicas: educação, habitação, saneamento, saúde, transporte, segurança, defesa e questões ambientais	Formar um cidadão reflexivo, autônomo e consciente.

6. HABILIDADES E COMPETÊNCIAS

O curso de Bacharelado em Engenharia de Computação deverá possibilitar uma formação profissional que revele as habilidades e competências nos âmbitos Pessoal, Interpessoal e Técnico.

Habilidades e Competências Pessoais:

1. Desenvolver o pensamento sistêmico;
2. Desenvolver o pensamento crítico;
3. Possuir disciplina pessoal;
4. Ter persistência;
5. Desenvolver o auto aprendizado;
6. Identificar novas oportunidades de negócios e desenvolver soluções inovadoras; e
7. Possuir abertura a mudanças.

Habilidades e Competências Interpessoais:

1. Realizar trabalho em equipe;
2. Possuir comunicação verbal e escrita eficientes; e
3. Capacidade para resolução conjunta de problemas.

Habilidades e Competências Técnicas:

1. Compreender e aplicar processos, técnicas e procedimentos de construção de software;
2. Analisar e selecionar tecnologias adequadas para a construção de softwares;
3. Integrar sistemas;

4. Conceber, aplicar e validar princípios, padrões e boas práticas no desenvolvimento de software;
5. Gerenciar o processo de informatização; e
6. Analisar e criar modelos relacionados ao desenvolvimento de sistemas web.

7. CONTEÚDO PROGRAMÁTICO

Semana	Data	Conteúdo	Estratégia de ensino-aprendizagem	Aula Teórica/Prática	Local
1	04/08	Apresentação do plano de disciplina, da metodologia de trabalho e do projeto a ser desenvolvido por equipe.	Aula Presencial e Online	Prática	Fábrica de Software
2	11/08	Proposta de trabalho: elaborar a apresentação do projeto (abertura), metodologia de desenvolvimento definido e artefatos necessários.	Resolução de problemas e análise das produções. Presencial e Online	Prática	Fábrica de Software
3	18/08	Apresentação do planejamento do projeto e da primeira sprint. Orientações quanto ao planejamento do processo de desenvolvimento definido.	Seminário e resolução de problemas. Presencial e Online	Prática	Fábrica de Software
4	25/08	Convocação Reunião de alinhamento dos projetos e tutoria em arquitetura e tecnologias para desenvolvimento web. Proposta de trabalho: estudo de estimativa em projetos de software.	Resolução de problemas. Presencial e Online	Prática	Fábrica de Software
5	01/09	Acompanhamento dos projetos e apresentação de fechamento de sprints. Orientações sobre o uso de estimativa de software, tanto para tarefas como sprints.	Resolução de problemas. Presencial e Online	Prática	Fábrica de Software
6	08/09	Revisão de conteúdo sobre: inteligência artificial; fundamentos de programação e linguagens de programação; e algoritmos e estruturas de dados.	Resolução de problemas Presencial e Online	Prática	Sala de aula
7	15/09	Convocação Apresentação do documento de arquitetura. Discussão sobre padrões de projeto. Orientações sobre a revisão das metas dos projetos em função de cumprir prazos e entregar resultados	Resolução de exercícios e tempestade de ideias. Presencial e Online	Prática	Fábrica de Software
8	22/09	1ª Verificação de Aprendizagem	Verificação de Aprendizagem	Prática	Sala de aula
9	29/09	Devolução e resolução da verificação de aprendizagem. Recuperação de conteúdo.	Resolução de problemas.	Prática	Fábrica de Software
10	06/10	Acompanhamento dos projetos. Orientações sobre o papel do product owner..	Resolução de exercícios e tempestade de ideias. Presencial e Online	Prática	Fábrica de Software
11	13/10	Paleta Learn Inception	Workshop Presencial e Online	Prática	Fábrica de Software
12	20/10	Acompanhamento e Orientação dos projetos.	Workshop Presencial e Online	Prática	Fábrica de Software
13	27/10	Apresentação de resultados desenvolvidos no projeto	Workshop Presencial e Online	Prática	Sala de aula
14	03/11	Revisão de Conteúdo	Resolução de problemas. Presencial e Online	Prática	Fábrica de Software
15	10/11	2ª Verificação de Aprendizagem	Verificação de aprendizagem	Prática	Fábrica de Software
16	17/11	Devolução e resolução da verificação de aprendizagem. Recuperação de conteúdo Sistemas em camadas, sistemas SOA e em workflows: arquitetura, tecnologias e exemplos.	Verificação de aprendizagem. Oficina em laboratório Presencial e Online	Prática	Fábrica de Software
17	24/11	SINACEN	Workshop Presencial e Online	Prática	Fábrica de Software
18	01/12	Resumo dos conhecimentos adquiridos.	Atividade avaliativa – Socrative	Prática	Fábrica de Software
19	08/12	Entrega final do projeto e atividade avaliativa	Atividade avaliativa – Socrative	Prática	Fábrica de Software
20	15/12	3ª Verificação de Aprendizagem	Verificação de Aprendizagem	Prática	Sala de aula

8. PROCEDIMENTOS DIDÁTICOS

Metodologias de ensino/aprendizagem

A disciplina deve ser ministrada através da prática de desenvolvimento de projetos reais. As aulas serão enriquecidas com desafios propostos pelo professor, no qual irão promover a investigação por soluções nas áreas gerenciais e técnicas de um projeto de desenvolvimento de software. Algumas discussões de conteúdo em sala irão complementar a formação de alguns conceitos fundamentais de arquitetura e desenvolvimento web para que os alunos tenham êxito na execução dos seus respectivos projetos.

Os projetos serão diariamente avaliados em termos das competências desenvolvidas pelos alunos nas áreas gerenciais e técnicas do desenvolvimento de um software:

- Planejamento
- Documentação de software
- Estimativa de software
- Uso de padrões no desenvolvimento
- Domínio tecnológico
- Participação efetiva da equipe
- Resultados técnicos alcançados

Os métodos de ensino utilizados são:

- Metodologia Ativa: Seminário, problematização, aula expositiva dialogada, oficina em laboratório, mapa conceitual.
- Tecnologia da Informação e comunicação (TICs): ferramentas de modelagem (Ashta/Cacao/Draw.io/Bizagi/BrModelo/Workbech MySQL), ambientes de programação, sistemas gerenciadores de banco de dados (Postgre SQL/MySQL).
- Ambiente de Aprendizagem – AVA; Zoom;

Recursos educativos

Quadro-branco/pincel, projetor multimídia, livros, ebook, fábrica de software, 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 atividade integrativa nos Cursos Superiores de Computação é realizada através do Projeto Interdisciplinar, em cada um dos períodos oferecidos na Matriz Curricular. O objetivo geral é possibilitar ao discente a intercomunicação entre as disciplinas estudadas, em nível curricular vertical e horizontal, aplicando e traduzindo os conhecimentos teóricos, técnicos e práticos, adquiridos durante sua formação acadêmica, traduzindo-os de forma concreta na elaboração de um projeto específico para melhor compreensão da realidade em que se insere social e profissionalmente.

Sendo assim, a disciplina promove a interdisciplinaridade através da participação nas disciplinas Projeto Interdisciplinar onde os projetos são detalhados em seus respectivos Planos de Ensino.

10. PROCESSO AVALIATIVO DA APRENDIZAGEM

1ª Verificação de aprendizagem (V. A.) – valor 0 a 100 pontos

Avaliação no formato ENADE, com valor de 0 a 50 pontos.

Avaliações processuais totalizam 50 pontos distribuídos da seguinte forma:

- Atividades que utilizam TICs (softwares de diagramação, programação) e metodologia ativa na fábrica de software – 0 a 50.

A média da 1ª V. A. será a somatória da nota obtida na avaliação teórica e prática (0-60 pontos) e as notas obtidas nas avaliações processuais (0-40 pontos).

2ª Verificação de aprendizagem (V. A.) – valor 0 a 100 pontos

Avaliação no formato ENADE, com valor de 0 a 60 pontos, contemplando questões relacionadas ao componente específico e de formação geral.

Avaliações processuais totalizam 40 pontos distribuídos da seguinte forma:

- Atividades que utilizam TICs (softwares de diagramação e programação) e metodologia ativa na fábrica de software – 0 a 40.

A média da 2ª V. A. será a somatória da nota obtida na avaliação teórica e prática (0-60 pontos) e as notas obtidas nas avaliações processuais (0-40 pontos).

3ª Verificação de aprendizagem (V. A.) – valor 0 a 100 pontos

Avaliação no formato ENADE, com valor de 0 a 40 pontos, contemplando questões relacionadas ao componente específico e de formação geral.

Avaliações processuais totalizam 60 pontos distribuídos da seguinte forma:

- Atividades que utilizam TICs (softwares de diagramação e programação) e metodologia ativa na fábrica de software – 0 a 20 pontos.
- Atividade avaliativa (Socrative) – 0 a 10 pontos

- Projeto da Disciplina – 0 a 30 pontos.

A média da 3ª V. A. será a somatória da nota obtida na avaliação teórica e prática (0-40 pontos) e as notas obtidas nas avaliações processuais (0-60 pontos).

Observação 3ª VA: Alunos que não estão matriculados em disciplina de projeto interdisciplinar terão a nota da 3ª VA calculada através da seguinte fórmula:

$$= (VA + Ativ) * 100/70$$

onde,

VA = Nota da 3ª Verificação de Aprendizagem

Ativ = Nota da(s) atividade(s) realizadas na fábrica.

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.**

A aplicação de avaliação substitutiva será conforme calendário do curso aprovado em colegiado:

- 1ª VA: 10/Fevereiro
 - 2ª VA: 12/Maio
 - 3ª VA: 16/Junho
- 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). A solicitação deverá ser feita através DE PROCESSO FÍSICO na Secretaria Geral do Centro Universitário de Anápolis - UniEVANGÉLICA com a avaliação original em anexo, obrigatoriamente.
 - Proibido uso de qualquer material de consulta durante a prova. Os equipamentos eletrônicos deverão ser desligados e qualquer manuseio deles será entendido como meio fraudulento de responder as questões. "Atribui-se nota zero ao acadêmico que deixar de submeter-se às verificações de aprendizagens nas datas designadas, bem como ao que nela utilizar-se de meio fraudulento" (Capítulo V, art. 39 do Regimento Geral do Centro Universitário de Anápolis, 2015).

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:

FERNANDES, Aguinaldo A.; TEIXEIRA, D. S. **Fábrica de software**. São Paulo: Atlas, 2011

DELAMARO, Mário Eduardo; MALDONADO, José Carlos; JINO, Mario. **Introdução ao teste de software**. Rio de Janeiro: Elsevier, 2007.

<https://integrada.minhabiblioteca.com.br/reader/books/9788595155732>

MENDES, Antônio. **Arquitetura de software: desenvolvimento orientado para arquitetura**. Ed. CAMPUS, 2002.

Complementar:

CARVALHO, Ariadne M. B. R.; CHIOSSI, Thelma C. S. **Introdução à Engenharia de Software**. Unicamp, 2001.

ASTELS, David; MILLER, Granville; NOVAK, Miroslav. **Extreme programming: guia prático**. Ed. CAMPUS, 2002.

MEYER, Bertrand. **Object-oriented software construction**. 2ed. New Jersey: Prentice Hall, 1997. 1254 p

GHEZZI, C.; JAZAYERI, M.; MANDRIOLI, D. **Fundamentals of software engineering**. 2. Ed. New Jersey: Prentice Hall, 2003. 604 p.

PETERS, J. F.; PEDRYCZ, W. **Engenharia de Software: teoria e prática**. Rio de Janeiro: Campus, 2001. 602 p.

REZENDE, Denis Alcides. **Engenharia de software e sistemas de informação**. 3 ed. Rio de Janeiro: Brasport, 2005. 358p.

Anápolis, 04 de Agosto de 2021.



Profa. M.e Natasha Sophie Pereira

COORDENADORA DO CURSO DE ENGENHARIA DE SOFTWARE DA UniEVANGÉLICA



Prof. M.e William Pereira dos Santos Júnior

COORDENADOR PEDAGÓGICO DO CURSO DE ENGENHARIA DE SOFTWARE DA UniEVANGÉLICA



Prof. M.e William Pereira dos Santos Júnior

PROFESSOR RESPONSÁVEL PELA DISCIPLINA

