

PLANO DE ENSINO

1. IDENTIFICAÇÃO

Campus: Avançado Quedas do Iguaçu

Eixo tecnológico: Informação e Comunicação

Curso: Técnico em Informática Integrado ao Ensino Médio

Componente curricular: Engenharia de Software

Docente: Odair Moreira de Souza

Carga horária: 40 h/a e 33 h/r

Turno: Matutino

Número de aulas na semana: 1 aula

Período letivo: 2019

Turma (s): 2ºA

Coordenador do curso: Odair Moreira de Souza

2. EMENTA

A Natureza do Software. Introdução à Engenharia de Software. Processos de Software. Desenvolvimento Ágil. Engenharia de Requisitos. Modelagem de Sistemas. Projeto de Arquitetura. Projeto e Implementação. Testes de Software. Evolução de Software. Confiança e Proteção. Gerenciamento de Projetos. Planejamento de Projeto. Gerenciamento de Qualidade. Gerenciamento de Configuração. Introdução à UML. Conceitos de orientação a objetos.

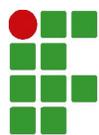
3. OBJETIVOS

3.1 Objetivo geral

Compreender os conceitos de engenharia de software, processos de desenvolvimento de software, modelagem de projetos de software e qualidade de produtos e processos.

3.2 Objetivos específicos

- Conhecer os conceitos fundamentais da engenharia de software;
- Compreender as metodologias de desenvolvimento de software;
- Aprender a elaboração de engenharia de requisitos;
- Dominar a modelagem de projetos de software;
- Entender e implementar os principais tipos de teste de software;
- Aprender as abordagens para o planejamento e gerenciamento de projetos de software; e,
- Conhecer as medidas de qualidade de software.



4. CONTEÚDO PROGRAMÁTICO

4.1. Introdução à Engenharia de Software

- 4.1.1. Conceitos de Processo e Produto de Software
- 4.1.2. Etapas do Processo de Desenvolvimento de Software
- 4.1.3. Ciclo de Vida de Software

4.2. Processo de Desenvolvimento de Software

- 4.2.1. Modelos de Processo Prescritivo
- 4.2.2. O Processo Unificado
- 4.2.3. Desenvolvimento Ágil

4.3. Engenharia de Requisitos

- 4.3.1. Requisitos Funcionais e Não-Funcionais
- 4.3.2. Processo de Especificação de Requisitos
- 4.3.3. Documentos de Requisitos

4.4. Modelagem de Software

- 4.4.1. Linguagem UML
- 4.4.2. Projetos de Arquitetura
- 4.4.3. Projeto de Projetos

4.5. Teste de Software

- 4.5.1. Verificação e Validação de Software
- 4.5.2. Processo de Teste de Software
- 4.5.3. Tipos de Testes de Software
- 4.5.4. Teste de caixa-branca e caixa-preta

4.6. Gerenciamento e Planejamento de Software

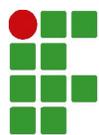
- 4.6.1. Gerenciamento de Projetos de Software
- 4.6.2. Planejamento de Projetos de Software
- 4.6.3. Gerenciamento de Configuração
- 4.6.4. Manutenção de Software

4.7. Introdução a Qualidade de Software

- 4.7.1. Garantia da Qualidade de Software
- 4.7.2. Qualidade de Produto e Processo
- 4.7.3. Modelos de Qualidade de Produto
- 4.7.4. Melhorias de Processo de Software

5. METODOLOGIA DE ENSINO E AVALIAÇÃO

A metodologia a ser empregada no processo de ensino-aprendizagem consiste em aulas expositivas e dialogadas, incentivando a participação dos discentes durante a explanação do conteúdo, resolução de problemas reais, construção e contextualização dos conhecimentos abordados.



O conteúdo teórico será abordado utilizando como referencial teórico a bibliografia sugerida, por meio de exemplos e estudos de caso e ao final de cada conteúdo, ou conjunto de assuntos relacionados, será proposto aos discentes o desenvolvimento de exercícios individuais ou em dupla para fixação da teoria apresentada.

As aulas práticas serão ministradas no laboratório de informática. Para todos os conteúdos abordados, serão apresentados exemplos desenvolvidos em ambiente computacional. Em seguida solicita-se aos discentes que apliquem os conceitos expostos, com o intuito de incentivar a reflexão e a habilidade de raciocínio para resolução de problemas.

As atividades de desenvolvimentos de trabalhos compreenderão a construção de soluções computacionais para os problemas propostos e que propiciem a fixação dos conteúdos previamente identificados.

Nesta disciplina os discentes irão desenvolver um projeto de software (com escolha livre e orientada pelo docente), desde o levantamento de requisitos, modelagem do software, projeto arquitetural, gerenciamento das atividades, projeto de teste, análise de qualidade e a aplicação de um processo de desenvolvimento de software, nesse projeto os discentes terão vários papéis, tais como, analista de requisitos, engenheiro de software, analista de teste, gerente de projetos e entre outros. Além de ministrar as aulas o docente atuará nos projetos como cliente levantando as demandas de requisitos, validação de artefatos e gerente de projetos, orientando o projeto real de engenharia de software das equipes.

5.1 Recursos didáticos

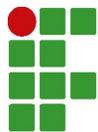
Será utilizado laboratório de informática, projetor multimídia e plataformas online de ensino-aprendizagem de programação.

6. CRITÉRIOS DE AVALIAÇÃO

A avaliação será realizada seguindo as orientações da Resolução nº 50/2017, que estabelece as normas de avaliação dos processos de ensino e aprendizagem no IFPR. Em cada bimestre, o conceito do aluno será composto por meio de sua participação e desempenho nos seguintes pontos:

- Participação e desempenho nos trabalhos individuais;
- Desenvolvimento de aplicações práticas;
- Desempenhos em avaliações individuais teóricas e práticas;
- Participação, assiduidade e proatividade.

Considera-se que para cada bimestre serão aplicados, no mínimo duas atividades avaliativas práticas de desenvolvimento (40% do conceito bimestral) em



equipe ou individual e uma ou duas avaliações práticas de desenvolvimentos de software (60% do conceito bimestral). Ressalta-se que os trabalhos e listas devem ser apresentados para a turma e/ou somente para o professor em formato de arguição, se for para a turma será aberto para debate sobre a solução algorítmica desenvolvida.

Os resultados serão apresentados a cada atividade avaliativa, sendo explicitado o diagnóstico feito pelo docente. De acordo com as normas da Instituição, os alunos receberão os conceitos A, B, C ou D nos períodos determinados pelo IFPR e no final do conteúdo de cada área curricular.

Interdisciplinaridade

Essa disciplina poderá ter atividades avaliativas interdisciplinares, envolvendo conteúdos relacionados com as disciplinas de Banco de Dados, Física e Matemática por meio de atividade avaliativa prática de desenvolvimentos.

Critérios de Aprovação

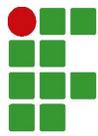
Conforme previsto na Resolução nº 50/2017 do IFPR, Art. 16. A aprovação dos estudantes ocorrerá considerando os seguintes critérios:

I – obtenção de conceito A, B ou C no componente curricular e frequência igual ou superior a 75% (setenta e cinco por cento) da carga horária total no período letivo dos cursos técnicos de nível médio.

7. RECUPERAÇÃO

I - Programa de Atividades e de Orientação: A recuperação do estudante poderá ser realizada no decorrer do módulo da disciplina através de atendimento mais direto e individualizado e com atividades diferenciadas que possam contemplar sua aprendizagem, baseada na Resolução nº 50/2017 do IFPR. O aluno com conceito insuficiente em um bimestre será convocado a participar de recuperação paralela contínua durante o próximo bimestre, exceto no último bimestre, pois isso possibilitará ao alunos tempo e atendimento para compreender a conteúdo antes da avaliação de recuperação. Caso a aprendizagem ainda for considerada insuficiente, o estudante cursa a disciplina novamente como progressão, em horários previamente combinados.

II - Formas de Avaliação: As atividades avaliativas de recuperação serão ofertadas em contraturno dos estudantes. Além disso, os alunos contam com atendimentos individualizados pelo professor. As recuperações paralelas serão agendadas com no mínimo 15 (quinze) dias de antecedência da aplicação.



III - Direito de Realizar as Avaliações de Recuperação: Conforme Resolução CONSUP/IFPR nº 50/2017, serão oferecidos estudos de recuperação paralela ou retomada dos conteúdos a todos os estudantes, independente do conceito atingido ser B, C ou D.

8. BIBLIOGRAFIA

8.1 Bibliografia Básica

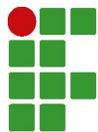
1. BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML: Guia do Usuário**. Rio de Janeiro, 2ª ed. Elsevier, 2012.
2. FILHO, Wilson de P. P. **Engenharia de Software: Fundamentos, Métodos e Padrões**. Rio de Janeiro, 3ª ed. LTC, 2015.
3. PRESSMAN, Roger S.; MAXIM, Bruce R. **Engenharia de Software: Uma Abordagem Profissional**. Porto Alegre, 8ª ed. AMGH, 2016.
4. SOMMERVILLE, I. **Engenharia de Software**. São Paulo, 9ª ed. Pearson Prentice Hall, 2011.

8.2 Bibliografia Complementar

1. BEZERRA, E. **Princípios de análise e projetos de sistemas com UML**. Rio de Janeiro, 3ª ed. Elsevier, 2015.
2. CARDOSO, C. **UML na prática: do problema ao sistema**. Rio de Janeiro, 1ª ed. Ciência Moderna, 2003.
3. ENGHOLM JR, H. **Engenharia de Software na prática**. São Paulo, 1ª ed. Novatec, 2010.
4. NOGUEIRA, M. **Engenharia de Software: Um Framework para a Gestão de Riscos em Projetos de Software**. Rio de Janeiro, 1ª ed. Ciência Moderna,

9. OBSERVAÇÕES

- A distribuição do conteúdo das aulas é uma previsão e poderá ser adequado



durante o andamento do ano letivo para atender aos reajustes em função do desempenho da turma ou para tratar eventualidades.

- O Google Classroom será o canal de comunicação digital entre o docente e os discentes, para disponibilização dos materiais das aulas, listas de exercícios, implementações de exemplos, submissão de atividades, avisos para a turma, agendamento de atividades e avaliações.
- O acesso dos estudantes aos laboratórios para o desenvolvimento de atividades em horários alternativos aos das aulas deverá ser solicitado ao professor e ter anuência do responsável pelo laboratório.
- O local e os horários de atendimentos do professor será divulgado nos murais de avisos do campus e na seção de avisos da turma do Google Classroom.

Quedas do Iguaçu, 12 de fevereiro de 2019.

Odair Moreira de Souza
Docente