

CÓDIGO DO CURSO	CURSO
GSISNF	BACHARELADO EM SISTEMAS DE INFORMAÇÃO

CÓDIGO DA DISCIPLINA	NOME DA DISCIPLINA	ANO/SEMESTRE
GS19603NF	PROJETO INTEGRADOR DE SISTEMAS	2022/1

CARGA HORÁRIA	CRÉDITOS	AULAS/SEMANA				PERÍODO
75 horas (90 h-a)	4	TEÓRICA	PRÁTICA	ESTÁGIO	NÃO PRESENCIAL	7º
		3	2	0	0	

PRÉ - REQUISITO (S)	CORREQUISITO (S)
Programação de Clientes Web Teste e Manutenção de Software Programação para a Web OU Programação de Aplicações Corporativas	-----

EMENTA
Aspectos práticos de qualidade de software. Gerência de configuração. Controle de versão. Revisão de conteúdos para a construção ou manutenção de projetos de software. Construção ou manutenção de projeto(s) de software utilizando práticas e controles de qualidade.

BIBLIOGRAFIA
BIBLIOGRAFIA BÁSICA
1. LARMAN, Craig. Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo . 3.ed. Porto Alegre, RS: Bookman, 2007. xiv, 695p., il. ISBN 9788560031528 (broch.). 2. PILONE, Dan; MILES, Russ. Use a cabeça: desenvolvimento de software . Rio de Janeiro: Alta Books, 2008. 379p., il. ISBN 9788576082125. 3. SOMMERVILLE, Ian. Engenharia de software . 9.ed. São Paulo: Pearson, 2011. xiii, 529p., il. ISBN 9788579361081.
BIBLIOGRAFIA COMPLEMENTAR

1. COHN, Mike. **Desenvolvimento de software com scrum: aplicando métodos ágeis com sucesso**. Porto Alegre: Bookman, 2011. xii, 496. ISBN 9788577808076:(broch.).
2. FOWLER, Martin. **Refatoração: aperfeiçoando o projeto de código existente**. Porto Alegre: Bookman, 2004. 365p., il. ISBN 9798536303955.
3. LOUDON, Kyle. **Desenvolvimento de grandes aplicações Web**. São Paulo: Novatec, 2010. 325 p. ISBN 9788575222515 (broch.).
4. MCCONNELL, Steve. **Code complete: um guia prático para construção de software**. 2.ed. Porto Alegre: Bookman, 2005. xv, 928p., il. ISBN 9788536305042.
5. POPPENDIECK, Mary; POPPENDIECK, Thomas David. **Implementando o desenvolvimento lean de software: do conceito ao dinheiro**. Porto Alegre: Bookman, 2011. 279p., il. ISBN 9788577807567.

OBJETIVOS GERAIS

1. Construir um ou mais projetos de software que envolvam todas as etapas do ciclo de vida de software, adotando práticas de controle de qualidade durante todas as etapas.
2. Discutir conceitos e práticas relacionados ao controle de qualidade de software;
3. Discutir aspectos que influenciam a qualidade de um produto de software e a maturidade de equipes e empresas de desenvolvimento de software.

METODOLOGIA

Aulas expositivas, dialogadas e práticas, com o auxílio de recursos audiovisuais. Todas as aulas serão realizadas em laboratório devidamente equipado com ferramentas de apoio.

CRITÉRIO DE AVALIAÇÃO

Trabalhos práticos constituem o principal meio de avaliação. Seminários podem ser utilizados para compor a nota.

PROGRAMA

1. Introdução
 - 1.1. Atividades do desenvolvimento de software
 - 1.2. Fases do desenvolvimento de software
 - 1.3. Custo médio da correção de defeitos
 - 1.4. Índices de detecção de defeitos por atividade de controle de qualidade
 - 1.5. Técnicas de construção colaborativa e percentual típico de defeitos encontrados
 - 1.6. Práticas comuns na indústria
 - 1.7. Fatores de qualidade interna e externa
2. Gerência de configuração
 - 2.1. Atividade de gerência de configuração
 - 2.2. Conceitos
 - 2.2.1. Papéis
 - 2.2.2. Artefatos
 - 2.2.3. Solicitação de mudança (formulário)
 - 2.2.4. Referencial (baseline)
 - 2.2.5. Item de configuração
 - 2.2.6. Banco de dados de configuração
 - 2.2.7. Repositório de itens de configuração

- 2.3. Versões para sistemas
 - 2.3.1.1. Conceitos (variante, liberação, versão estável, versão alfa, versão beta, versão candidata a liberação (RC), versão oficial)
 - 2.3.1.2. Estilos de identificação (numérico, alfanumérico, data)
- 2.4. Versões para componentes e APIs
 - 2.4.1. Estilo de identificação numérico
 - 2.4.2. Versionamento semântico
- 2.5. Evolução e suporte de versões
- 2.6. Construção de bancos de dados de configuração
 - 2.6.1. Tipos de bancos de dados de configuração
 - 2.6.2. Ferramentas comuns no mercado
- 2.7. Repositórios de itens de configuração
 - 2.7.1. Introdução (o que são, para que servem, como funcionam em geral)
 - 2.7.2. Conceitos
 - 2.7.2.1. Revisão
 - 2.7.2.2. Diferença (*diff*)
 - 2.7.2.3. Revisão baseada em alterações por arquivo
 - 2.7.2.4. Revisão baseada em instantâneo de conteúdo
 - 2.7.2.5. Identificações de revisão (numérica, hash)
 - 2.7.2.6. Revisão mais recente (*head*)
 - 2.7.2.7. Cópia de trabalho
 - 2.7.3. Repositórios em arquitetura cliente-servidor versus arquitetura distribuída
 - 2.7.4. Como ocorre o armazenamento e a manipulação de repositórios
 - 2.7.5. Organização de repositórios
 - 2.7.5.1. Linha principal (*master/trunk*)
 - 2.7.5.2. Linhas paralelas (*branches*)
 - 2.7.5.3. Linhas ou marcações estáveis (*tags*)
 - 2.7.6. Uso concorrente
 - 2.7.6.1. Resolução por travamento
 - 2.7.6.2. Resolução por junção
 - 2.7.6.3. Estratégias de trabalho
 - 2.7.7. Práticas recomendadas
 - 2.7.8. Ferramentas comuns no mercado
3. Controle de versões com Git
 - 3.1. Uso básico
 - 3.1.1. Clonando um repositório
 - 3.1.2. Ignorando arquivos (.gitignore)
 - 3.1.3. Verificando o status
 - 3.1.4. Adicionando e retirando arquivos
 - 3.1.5. Gravando um conjunto de mudanças
 - 3.1.6. Obtendo atualizações
 - 3.1.7. Fazendo uma junção
 - 3.1.8. Modificando uma mensagem
 - 3.1.9. Enviando mudanças para o repositório
 - 3.2. Uso de ramos (*branches*)
 - 3.2.1. Listagem, criação, integração, remoção
 - 3.3. Uso de marcações estáveis (*tags*)
 - 3.3.1. Listagem, criação, remoção
4. Revisão de conteúdos para a construção de projetos de software
5. Acompanhamento de projetos de software

PROFESSOR RESPONSÁVEL

NOME - SIAPE Thiago Delgado Pinto – 1683010

ASSINATURA

COORDENADOR DO CURSO

NOME - SIAPE Rodrigo Reis Gomes – 1644791

ASSINATURA

LOCAL E DATA

Nova Friburgo, 19 de novembro de 2021.