



Plano de Ensino

Dados Gerais

Curso:

217 - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas (Campus Colinas do Tocantins)

Disciplina:

TCN.0726 - Arquitetura de Software - Graduação [33 h/40 Aulas]

Ano/Período Letivo:

2024/2

Turma:

20242.3.217.1N

Professor(es):

Ronneesley Moura Teles (1867392) (Servidor)

Data da Aprovação:

14/10/2024

Data da Homologação:

15/10/2024

Ementa

Arquiteturas de software: Camadas, Cliente-Servidor, MVC, Micro-serviços, Pipes-and-filters, peer-to-peer, Service-Oriented Architecture e Publish-Subscribe. Conceitos de Construção de Software: Pré-requisitos para a construção. Boas práticas para definição dos requisitos. Paradigmas e seleção da linguagem de programação. Convenções de programação. Acoplamento e Coesão. Classes e pacotes. Rotinas (métodos). Motivos para se criar uma rotina. Bons nomes para uma rotina. Tamanho adequado de uma rotina. Programação defensiva. Problemas gerais no uso de variáveis. Nomes adequados para variáveis. Tipos de dados fundamentais. Estruturas. Tratamento de erro e exceção. Estratégias para Tolerância a falhas.

Justificativa

Capacitar o aluno a entender a importância das arquiteturas de software e capacitá-lo no desenvolvimento delas.

Objetivo Geral

Compreender e propor a aplicação de modelos arquiteturais para o desenvolvimento de software

Objetivos Específicos

Selecionar tecnologias e habilidades necessárias à equipe voltada ao seu desenvolvimento da aplicação, de forma apta a maximizar o valor do software, a fim de gerar valor agregado ao negócio em que se aplica;
Conhecer as principais arquiteturas, paradigmas e ferramentas de desenvolvimento de software;
Dimensionar equipes de desenvolvimento baseado no conjunto de habilidades necessárias para a concepção do software.

Conteúdo Programático

O que são e qual a importância das arquiteturas de software? Projetos sem arquitetura definida. Organização de projetos em camadas. Arquitetura Cliente-Servidor. Arquitetura em Camadas. Arquitetura MVC. Arquitetura de microsserviços. Outras arquiteturas. Paradigmas e linguagens de programação. Conversões e Regras. Acoplamento e Coesão. Classes, pacotes, métodos, funções. Nomenclatura. Tipos de dados. Tratamento de erros e exceção.

Metodologia



As aulas serão expositivas e dinâmicas com utilização de quadro, pincel e computador. Para isso, serão utilizados laboratórios de informática para as práticas de codificação.

Nesta disciplina os alunos terão que se organizar em grupos de até 5 alunos.

Cada grupo deverá desenvolver o mesmo projeto nas seguintes formas:

- 1) Software utilizando somente o paradigma procedural;
- 2) Software utilizando somente programação orientada a objetos (POO);
- 3) Software utilizando POO com MVC;
- 4) Software utilizando microsserviços;

Assim a média dos bimestres do aluno serão:

$$M1 = P1 + P2$$

$$M2 = P3 + P4$$

onde M1 e M2 são, respectivamente, as médias do primeiro e segundo bimestre, e Pi representa cada uma das i-formas em que o projeto deve ser entregue.

Informações Adicionais

Referencia Bibliografica Básica

- VALENTE, Marco Tulio. Engenharia de Software Moderna. (Livro Digital), 2020
- MARTIN, Robert C. Arquitetura Limpa: O guia do artesão para estrutura e design de software. Alta Books Editora, 2019.
- MARTIN, Robert C. O Codificador Limpo: Um Código de Conduta para Programadores Profissionais. Alta Books Editora, 2012.

Referencia Bibliografica Complementar

- FREEMAN, E. Use a Cabeça: Padrões de Projetos. 2ª Edição. Editora Alta Books, 2007.
- ENGHOLM, Hélio. Análise e design orientados a objetos. Novatec Editora, 2017.
- GIRIDHAR, Chetan. Aprendendo Padrões de Projeto em Python. Editora Novatec, 2016.