|  |
| --- |
| **Disciplina:** Desenvolvimento Full Stack |
| **Professor:** Caio Eduardo do Prado Ireno |
| **Competências:** |
| Capacidade de diagnosticar, sugerir e defender técnicas, métodos e soluções para desenvolvimento de aplicações em modelo distribuído, desde sua concepção inicial em projetos novos quanto em soluções que necessitem de mudança arquitetural para suporte ao crescimento de um negócio digital.  Autonomia para propor soluções visando confiabilidade de um software a um cenário de negócios digitais com necessidade de escala, gerenciar tecnicamente um projeto e/ou equipe que atuará em mudanças de arquitetura e apoiar gestores de negócios na definição do melhor caminho técnico a ser mantido. |
| **Habilidades:** |
| Construir e estilizar interfaces web responsivas com HTML e CSS.  Implementar chamadas HTTP para consumo de APIs com JavaScript.  Projetar e modelar banco de dados relacionais simples (SQLite).  Criar e organizar projetos seguindo boas práticas de arquitetura (Blueprints no Flask, separação de camadas).  Realizar o controle de versões utilizando Git. |
| **Conteúdo programático:** |
| **Fundamento de sistemas e serviços web:** Introdução à programação em redes; camadas e protocolos do modelo TCP/IP; aplicações da internet web; terminologias; Protocolo DNS; Protocolo HTTP; Padrão REST; Formato JSON; API Requests; API Flask; Conectores de bancos de dados.  **Arquitetura de sistemas:** Padrões de projeto; arquitetura MVC; modelos cliente-servidor.  **Desenvolvimento Front-End:** Estrutura de páginas HTML; estilização com CSS; interatividade com JavaScript; consumo de APIs REST usando Fetch API.  **Aula 1 – Introdução ao Desenvolvimento Web e Front-End Básico**   * Introdução ao ambiente de desenvolvimento (VSCode, Python, SQLite, navegador) * Estrutura de arquivos e pastas para projetos web * Criação de páginas HTML básicas * Introdução à CSS para estilização * Introdução ao JavaScript para manipulação da página   **Aula 2 – Consumo de API com JavaScript**   * Conceito de API e método HTTP (GET, POST, PUT, DELETE) * Fetch API no JavaScript para consumir serviços REST * Montagem dinâmica de tabelas/listas com dados de API * Criação de formulários para envio de dados via API   **Aula 3 – Desenvolvimento de Back-End com Flask e SQLite**   * Conceito de servidor web e API RESTful * Estrutura de um projeto Flask (Model, Controller, Repository) * Criação de Models e Configuração do Banco de Dados (SQLite) * Implementação de rotas de CRUD com Flask e Blueprint   **Aula 4 – Integração Completa e Boas Práticas**   * Comunicação Front-End ↔ Back-End * Testes básicos de integração * Organização e boas práticas de projeto * Deploy local e orientações para deploy em servidores |
| **Metodologia de ensino:** |
| * Aulas expositivas utilizando apresentações em *slides* e demonstrações *práticas*. * Debates em grupo. * Exercícios de implementação diversos. * Discussão de dúvidas de forma aberta e dinâmica |

|  |
| --- |
| **Recursos necessários:** |
| * Computador com navegadores de internet. * Python 3. * NodeJS * Visual Studio Code * Git para controle de versão |
| **Critérios de avaliação:** |
| **Alinhar com a coordenação** |
| **Bibliografia básica:** |
| RODRIGUES, Thiago N.; SILVA, Lídia P C.; NEUMANN, Fabiano B.; et al. Integração de Aplicações. Porto Alegre: Grupo A, 2020.  MACIEL, Francisco Marcelo de B. Python e Django. Rio de Janeiro: Editora Alta Books, 2020.  OLIVEIRA, Cláudio Luís V.; ZANETTI, Humberto Augusto P. Node.js: programe de forma rápida e prática. Rio de Janeiro: Grupo GEN, 2021.  SHAW, Zed A. Aprenda Python 3 do Jeito Certo. Rio de Janeiro: Editora Alta Books, 2019.  BREITMAN, Karin K. Web Semântica - A Internet do Futuro. Rio de Janeiro: Grupo GEN, 2005. |