|  |  |
| --- | --- |
| **PLANO DE APRENDIZAGEM** | |
| **CURSO:** Sistemas da Informação. | **SEMESTRE LETIVO:** 2022/02 |
| **DISCIPLINA/PROJETO:** Integração de Equipe Técnica de DevOps Aplicadas a Projeto de Software. | **CH:** 45 horas |
| **MATRIZ CURRICULAR:** | |
| **PROFESSOR:** James Alves | |

|  |
| --- |
| **EMENTA** |
| DEVOPS – Conceitos de DevOps, práticas e metodologias ágeis. Gerenciamento de pacotes maven e node, controle de versão e repositório de binários. Continous Integration, Continous Delivery e Continous Deployment com Gitlab e Jenkins. Gerenciamento e agregação de logs usando o stack ELK (Elasticsearch, Logstash e Kibana) Emprego de containers Docker para avaliação rápida e simples de novas ferramentas. Linux - Interfaces gráficas e Shell. Gerenciamento de pacotes de software e atualizações. Instalação e configuração das ferramentas básicas para automação: Git/Gitlab (controle de versão), Jenkins/Ansible (automação), Artifactory (repositório) |

|  |
| --- |
| **OBJETIVO GERAL** |
| Desenvolver compreensão sobre os conceitos de produção de software em equipes. O aluno desenvolverá conceitos teóricos e práticos para o planejamento de projeto de software. |

|  |
| --- |
| **OBJETIVOS ESPECÍFICOS** |
| • Entender como funciona o versionamento software.  • Desenvolver sistemas colaborativamente.  • Elaborar um ciclo de Devops |

|  |
| --- |
| **COMPETÊNCIAS PROFISSIONAIS** |
| * Automotivação * Trabalho em equipe * Inteligência emocional * Adaptabilidade * Gestão do conhecimento * Gestão na solução de problemas |

|  |
| --- |
| **COMPETÊNCIAS COGNITIVAS** |
| * Capacidade de memorização, percepção, atenção e compreensão |

|  |
| --- |
| **COMPETÊNCIAS SOCIOEMOCIONAIS** |
| * Responsabilidade, criatividade, comunicação, autonomia, sociabilidade, ética e organização. |

|  |
| --- |
| **CONTEÚDO PROGRAMÁTICO** |
| 1. Sistemas de controle de versão 2. Ambientes de dsenvolvimento. 3. Continuous integration/Continuous delivery 4. Docker 5. Gerenciamento de entregas |

|  |
| --- |
| **METODOLOGIA DE APRENDIZAGEM** |
| A aprendizagem será baseada na solução de problemas propostos (BPL) conceituando a busca de conhecimento por parte dos alunos (individual e coletivamente). A abordagem também contará com elementos de jogos (gamificação) onde equipes (grupos) deverão propor soluções de demandas provenientes de outras equipes atuando como ‘consultores’. |

|  |
| --- |
| **RECURSOS DIDÁTICOS** |
| Serão utilizados textos de pesquisa, apresentações online, leituras de artigos acadêmicos, trechos de livros, busca em sites (mídias digitais), gravações feitas pelo professor além de ferramentas online de averiguação de aprendizagem. |

|  |  |
| --- | --- |
| **AVALIAÇÃO** | |
| A avaliação será realizada da seguinte maneira:    A pontuação é dividida em:   1. 10 pontos – avaliação de performance 2. 30 pontos – prova (a critério do professor) 3. 60 pontos – trabalho (s) em grupo | |
| **Data** | **Atividade Avaliativa e Feedbacks** |
|  | Avaliação do conteúdo individual. |
|  | Demonstração de competência em grupo. |
|  | Entrega dos Exercícios das Unidades de Aprendizagem. |
|  | Avaliação do portfólio produzido. |

|  |  |  |
| --- | --- | --- |
| **TRILHA DE APRENDIZAGEM / CRONOGRAMA** | | |
| As etapas do projeto mesclam conteúdos teóricos de aprendizagem e práticas com entregas programadas | | |
| **Data** | **Situações e Atividades de Aprendizagem** | **Conteúdo online** |
|  | Apresentação da disciplina  Fundamentos do Devops | - Introdução ao DevOps  - O papel do DevOps  - Comunicação e colaboração na cultura DevOps |
|  | Fundamentos do Devops  Desenvolvimento Contínuo | - Integração da equipe de Devs com o Git  - Criação de branches e commits com o Git  - Tipos de controles de versão e verbetes |
|  | Teste Contínuo | - Introdução aos testes de software  - Casos de teste  - Ferramentas de teste |
|  | Integração e entrega Contínua | - Fundamentos do Jenkins  - Pipeline de integração e entrega contínuas |
|  | Monitoramento Contínuo | - Uso do Docker para microsserviços  - Configuração e manutenção de contêineres usando o Docker |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| Clique ou toque aqui para inserir uma data. |  |  |

|  |
| --- |
| **BIBLIOGRAFIA BÁSICA** |
| Sommervile, **Engenharia de Software**, 10 Edição, pearson , Disponível em: https://plataforma.bvirtual.com.br/Leitor/Publicacao/168127/pdf/0  Fagonde, F. Dias, C. **DevOps**. Porto Alegre. 2021  CARVALHO, Ariadne M. B. R.; CHIOSSI, Thelma C. dos Santos. Introdução a Engenharia de Software, Unicamp: 2001.  MAGELA, Rogerio. **Engenharia de Software Aplicada: Princípios (volume 1)**. Alta Books. 2006.  REZENDE, Denis Alcides. Engenharia de Software e Sistemas de Informação,Brasport:2005 |

|  |
| --- |
| **BIBLIOGRAFIA COMPLEMENTAR** |
| Clique ou toque aqui para inserir o texto.BEZERRA, Eduardo; Princípios de análise e projeto de sistemas com UML, 2/E. 2ªEdição. Campus, 2006.  MEDEIROS, E.; Desenvolvendo Software com UML 2.0 : definitivo. São Paulo: Pearson Makron Books, 2004.  MOLINARI, Leonardo. Gerência de Configuração: Técnicas e Práticas no Desenvolvimento do Software. Florianópolis: Visual Books, 2007.  HALL, M. & BROWN, L. Core Servlets e Java Server Pages - Tecnologias Core. 2. ed. Rio de Janeiro: Ciência Moderna, 2005. v. 1.  CORNELL, Gary. Core Java 2. Pearson Education do Brasil, 2004. |

|  |
| --- |
| **SITES DE INTERESSE** |
| Clique ou toque aqui para inserir o texto. |