

INTEGRAÇÃO E ENTREGA CONTÍNUA

Apresentação da Disciplina

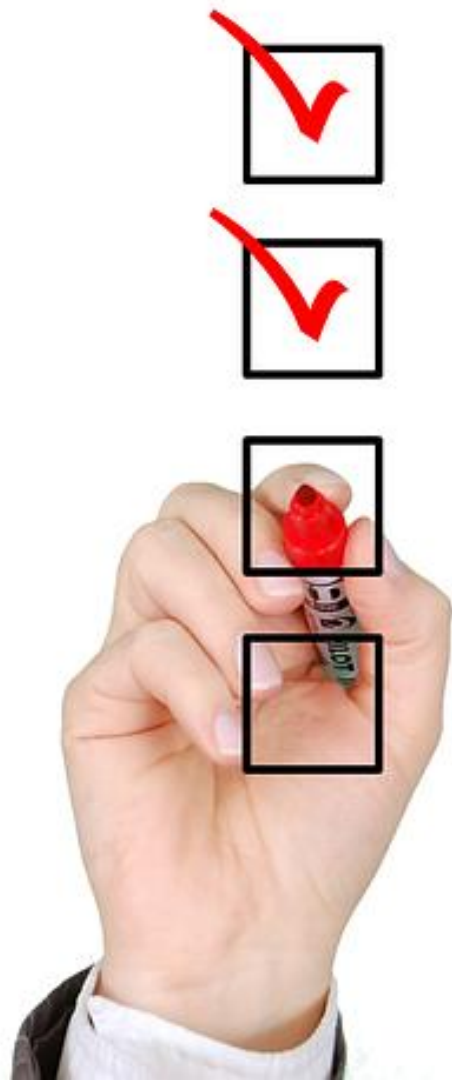
Semestre: 1º/2026

Professora: Lucineide Pimenta

INTEGRAÇÃO E ENTREGA CONTÍNUA

WILLKOMMEN
欢迎 स्वागत
BIENVENIDA
WELCOME
BIENVENUE ようこそ
добро пожаловать
ترحيب BEM-VINDO

Tópicos da aula



- ☐ O(a) Professor(a)
- ☐ Os Alunos
- ☐ A Disciplina
 - ☐ Ementa
 - ☐ Objetivos
 - ☐ Aprendizagem Baseada em Projetos
 - ☐ Plano de Aulas
 - ☐ Cronograma de Provas e critérios de Avaliação
 - ☐ Bibliografia Básica e Complementar
- ☐ Considerações Finais

O(A) Professor(a)

Prof.^a Lucineide Nunes Pimenta



Professora de Ensino Superior
Sistemas de Informação e
outros cursos
Unidade Passos
(2018-2022)

UEMG

Prof.^a E. Técnico e Tecnológico
Integrado E. Médio e
Graduação
Campus Jacareí
(2022-2024)

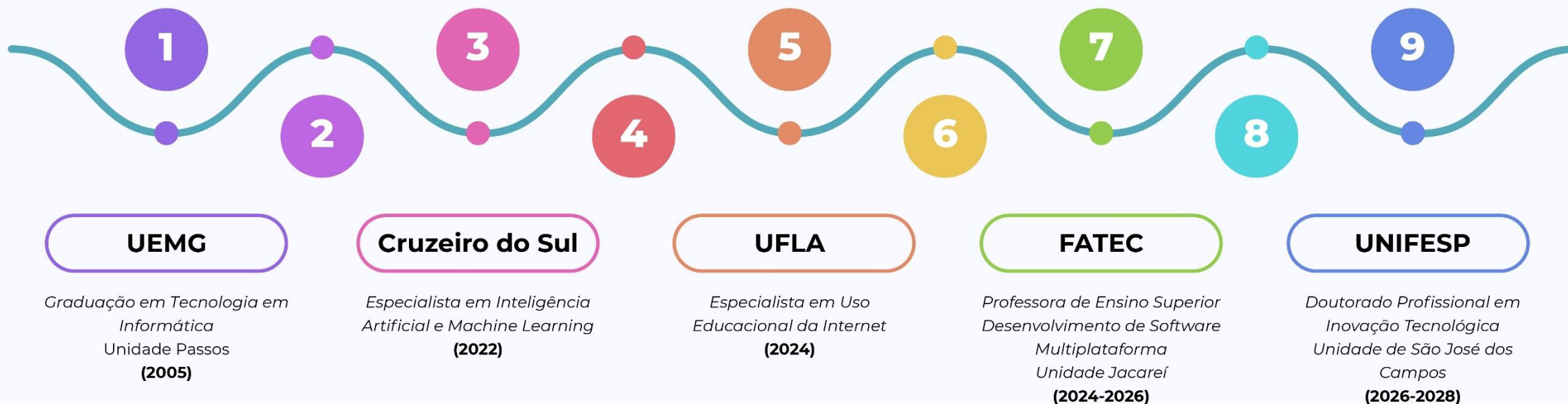
IFSP

Mestrado Profissional em
Inovação Tecnológica
Unidade de São José dos
Campos
(2025)

UNIFESP

Especialização em Formação
Didático Pedagógica para
Cursos de Modalidade a
Distância
(2025-2026)

UNIVESP



Os(as) Alunos(as)

❑ Apresentação:

- ❑ Nome e Cidade;
- ❑ Trabalha ou faz estágio na área de Tecnologia?

“Onde meus talentos e paixões encontram as necessidades do mundo, lá está meu caminho, meu lugar”.

Aristóteles



Informações Gerais da Disciplina

| A Disciplina | |
|---|----------------------------|
| Curso: Desenvolvimento de Software Multiplataforma | |
| Aulas: Segunda-feira das 22h15 às 23h05 / Sexta-feira das 18h45 às 21h15 | |
| Disciplina: Integração e Entrega Contínua | Ano Letivo: 2026/1 |
| Professor(a): Lucineide Pimenta | |
| Carga Horária | |
| Semanal: 4 horas/aula | Semestral: 80 aulas |

Ementa

- ☐ *Conceitos de Integração e entrega contínua de software.*
- ☐ *Gerencia de Configuração.*
- ☐ *Ferramentas de integração e entrega contínua de software.*
- ☐ *Pipeline: para construção de aplicações, para verificação da qualidade do código, configuração dos requisitos).*
- ☐ *Scripts de compilação e implantação.*
- ☐ *Controle de versão avançado.*
- ☐ *Ferramentas de versionamento de código.*
- ☐ *Estágios de COMMIT.*
- ☐ *Testes automatizados.*
- ☐ *Implantação de entrega de versões de aplicações.*
- ☐ *Gerencia de Dados.*
- ☐ *Gerenciamento de Componentes e dependências.*
- ☐ *Gerenciamento da entrega contínua.*



Objetivo

- ❑ Compreender e aplicar conceitos, práticas e padrões relacionados à integração e entrega contínua.
- ❑ Compreender a relação entre testes e integração contínua.
- ❑ Avaliar as vantagens e desvantagens da integração e entrega contínua no desenvolvimento.
- ❑ Criar e avaliar gráficos.
- ❑ Empregar técnicas de monitoramento de infraestrutura e logs da aplicação.
- ❑ Selecionar e avaliar ferramentas para integração e entrega contínua.
- ❑ Compreender e empregar ferramentas e infraestrutura do ambiente integrado (Containers e Docker).





Experiência do Usuário

Aprendizagem Baseada em Projetos (ABP)

Aprendizagem Baseada em Projetos (ABP)

❑ Catálogo de Conhecimentos

- ❑ Manipulação de arrays e JSON.
- ❑ Organização do código em funções.
- ❑ Modelagem de dados.
- ❑ Segurança da informação.
- ❑ CI/CD (Integração e Entrega Contínua).
- ❑ Autenticação e autorização de usuários.
- ❑ Desenvolvimento de APIs.
- ❑ Consumo de APIs.
- ❑ Manipulação de mapas interativos.



Aprendizagem Baseada em Projetos (ABP)

- ❑ **Catálogo de Tecnologias para o semestre**
 - ❑ **Linguagens de Programação:** JavaScript, TypeScript
 - ❑ **Banco de Dados:** PostgreSQL (relacional), NoSQL (MongoDB ou Firebase)
 - ❑ **Ferramentas de Design:** Figma
 - ❑ **Controle de Versionamento:** GitHub
 - ❑ **Desenvolvimento Frontend:** React Native (TypeScript)
 - ❑ **Desenvolvimento Backend:** Node.js (TypeScript)
 - ❑ **Plataformas:** Arduino (para IoT), Cloud para CI/CD
 - ❑ **Outras Ferramentas:** Trello ou GitHub Projects para gestão de tarefas



Aprendizagem Baseada em Projetos (ABP)

- ❑ Ferramentas que suportam **CI/CD, versionamento e automação de deploy**:
 1. **GitHub e GitHub Actions** – Para controle de versão e automação de pipelines CI/CD.
 2. **Docker** – Para criação e gerenciamento de ambientes de desenvolvimento e produção.
 3. **Firebase Hosting** – Para deploy automatizado da aplicação.
 4. **PostgreSQL/MongoDB** – Para integração de banco de dados no processo de entrega contínua.
 5. **Node.js com TypeScript** – Para configurar testes automatizados e garantir qualidade antes do deploy.
 6. **Trello ou GitHub Projects** – Para gerenciamento do fluxo de trabalho ágil e organização das tarefas de CI/CD.
- ❑ Essas ferramentas permitirão que os estudantes configurem pipelines completos para **teste, build e deploy automatizados** do projeto.

Aprendizagem Baseada em Projetos (ABP)

- ❑ Tema para as aulas
- ❑ **Aplicativo Móvel de Monitoramento e Comunicação de Eventos Climáticos e Ambientais Críticos para a População.**
- ❑ O aplicativo será desenvolvido para o **INPE**, com foco em alertas de queimadas, inundações, desmatamento, mudanças climáticas e coleta de dados locais da população em tempo real.



Requisitos da Disciplina de IEC (Integração e Entrega Contínua)

- ❑ Com base na **ementa da disciplina** e no **desafio da ABP**, foram definidos quatro requisitos principais que serão aplicados no desenvolvimento dos projetos.
- ❑ Esses requisitos garantirão que os grupos implementem corretamente práticas de CI/CD e versionamento no projeto.



Requisitos da Disciplina de IEC (Integração e Entrega Contínua)

IEC.01 – Configuração do Ambiente de Integração Contínua e Versionamento

- ❑ Criar um repositório no GitHub para o projeto do grupo, configurando **Git Flow** (ou fluxo equivalente), **proteção de branch principal** e integração inicial com **GitHub Actions** para build e análise de qualidade do código.
- ❑ **Objetivo:** aplicar fundamentos de versionamento e CI desde o início do desenvolvimento.

Requisitos da Disciplina de IEC (Integração e Entrega Contínua)

IEC.02 – Implementação de Testes Automatizados

- ❑ Desenvolver e integrar ao pipeline testes automatizados (unitários e, quando aplicável, de integração) no projeto, utilizando biblioteca/framework adequado à tecnologia (ex.: Jest para React Native).
- ❑ **Objetivo:** garantir que os testes sejam executados automaticamente no pipeline, cobrindo as funcionalidades críticas do app.
- ❑ **Atende diretamente à exigência do coordenador** para preparar os alunos para Segurança da Informação.

Requisitos da Disciplina de IEC (Integração e Entrega Contínua)

IEC.03 – Pipeline de Entrega Contínua e Implantação

- ❑ Configurar um pipeline completo de CI/CD que realize build, execução de testes, análise de qualidade, empacotamento e implantação automática em ambiente de homologação usando **Docker** e **Firebase Hosting**.
- ❑ Objetivo: simular um fluxo de entrega contínua profissional, aplicando as boas práticas aprendidas.



Integração e Entrega Contínua (IEC)

Plano de Aulas

Plano de Aulas – Integração e Entrega Contínua (IEC)

 **Duração:** 26 dias letivos dedicados as aulas.

 **Distribuição:**

- **Segunda-feira:** 1 aula
- **Sexta-feira:** 3 aulas

 **Metodologia:**

- Aulas expositivas e dialogadas
- Estudos de caso
- Exercícios práticos individuais
- Requisitos da ABP (IEC.01 a IEC.03)
- Atividades práticas e avaliações contínuas



Plano de Aulas – Integração e Entrega Contínua (IEC)

- ❑ **Objetivos Gerais**
- ❑ Compreender e aplicar conceitos, práticas e padrões de Integração e Entrega Contínua (CI/CD).
- ❑ Configurar pipelines de build, teste e implantação contínua.
- ❑ Empregar técnicas de controle de versão avançado, testes automatizados, monitoramento e entrega de software de forma segura e ágil.

Plano de Aulas – Integração e Entrega Contínua (IEC)

- ❑ **Requisitos da Disciplina**
- ❑ **IEC.01 – Configuração do Ambiente de Integração Contínua e Versionamento**
Entrega: **A definir com o Focal Point.**
- ❑ **IEC.02 – Implementação de Testes Automatizados**
Entrega: **A definir com o Focal Point.**
- ❑ **IEC.03 – Pipeline de Entrega Contínua e Implantação**
Entrega: **A definir com o Focal Point.**

Plano de Aulas – Integração e Entrega Contínua (IEC)

❑ Cronograma Detalhado

| Dia Letivo | Tema | Fundamentação Teórica | Referências | Exercício/Prática | Entrega |
|------------|---|---|--|---|---------|
| 1 | Introdução à Integração e Entrega Contínua | Conceitos de CI/CD, benefícios, riscos e relação com metodologias ágeis | Atlassian – CI/CD ; Humble & Farley, Entrega Contínua (Cap. 1 e 2) | Discussão em grupos sobre aplicação no projeto; diagrama do fluxo ideal de integração e entrega | - |
| 2 | Controle de Versão Avançado (Git e GitHub) | Git Flow, GitHub Flow, pull requests, code review, branches protegidas | Git Flow ; GitHub Flow | Criar repositório, configurar branch principal e branch de desenvolvimento | - |
| 3 | Ferramentas de Integração Contínua – GitHub Actions | Fundamentos de pipelines, workflows, jobs e triggers | GitHub Actions | Criar workflow simples que executa build | - |
| 4 | Configuração de Build e Análise de Qualidade | Linters, formatação automática, análise estática de código | ESLint | Configurar ESLint e Prettier no projeto | - |
| 5 | Monitoramento de Qualidade no Pipeline | Métricas de qualidade, SonarCloud | SonarCloud | Integrar SonarCloud e configurar análise automática | - |
| 6 | Conclusão do Módulo 1 e Aplicação no Projeto | Revisão de conceitos iniciais e aplicação no ABP | - | Implementar práticas no projeto | - |
| 7 | Fundamentos de Testes Automatizados | Tipos de testes, pirâmide de testes | Pirâmide de Testes – Martin Fowler | Configurar Jest para React Native, criar 1º teste unitário | - |
| 8 | Testes Unitários em React Native | Boas práticas, mocks e spies | Testing Library RN | Criar testes para componente com props | - |

Plano de Aulas – Integração e Entrega Contínua (IEC)

□ Cronograma Detalhado

| | | | | | |
|----|----------------------------------|---|-------------------------------|---|---|
| 9 | Testes de Integração | Uso e cenários reais | Jest Docs | Testar tela que consome API | - |
| 10 | Integração de Testes ao Pipeline | Configuração de workflows para execução de testes | CI com Jest | Criar pipeline que executa testes e falha se não passarem | - |
| 11 | Cobertura de Testes | Conceito, importância, relatórios automáticos | Jest Coverage | Gerar relatório e enviar para SonarCloud | - |
| 12 | Testes com Mock de API | Testes offline e simulação de serviços externos | - | Criar mock para API de alertas climáticos | - |
| 13 | Revisão e Ajustes de Testes | Revisão coletiva de testes | - | Aplicar boas práticas no projeto | - |
| 14 | Entrega de Testes Automatizados | Revisão final de testes no projeto | - | Ajustar cobertura e integração no pipeline | - |
| 15 | Fundamentos de Docker para CI/CD | Conceitos de contêineres e imagens | Docker Docs | Criar Dockerfile básico | - |
| 16 | Docker na Integração Contínua | Uso em pipelines, build automatizado | - | Integrar Dockerfile no pipeline | - |
| 17 | Deploy com Docker | Publicar contêiner em repositório | - | Configurar imagem para deploy | - |

Plano de Aulas – Integração e Entrega Contínua (IEC)

❑ Cronograma Detalhado

| 18 | Fundamentos de Deploy Contínuo | Conceitos e estratégias de entrega | <u>Firebase Hosting</u> | Criar conta e configurar Firebase Hosting | - |
|----|--|------------------------------------|-------------------------|--|---|
| 19 | Deploy no Firebase Hosting | Integração com GitHub Actions | - | Pipeline com deploy automático | - |
| 20 | Deploy com Ambientes de Staging e Produção | Separação de ambientes | - | Configurar dois ambientes no Firebase Hosting | - |
| 21 | Monitoramento de Logs e Infraestrutura | Ferramentas de logging e métricas | - | Criar logs para ações críticas no app | - |
| 22 | Conclusão e Entrega de Pipeline Completo | Revisão e ajustes finais | - | Testar pipeline completo de build, testes e deploy | - |
| 23 | Refinamento do Projeto | Correção de problemas detectados | - | Revisar código e pipeline | - |
| 24 | Preparação para Apresentação | Estrutura de entrega do projeto | - | Criar roteiro de apresentação | - |
| 25 | Simulação de Apresentação | Treino e ajustes | - | Apresentar para outro grupo | - |
| 26 | Apresentação Final | Encerramento da disciplina | - | Apresentar projeto final | - |

Instrumentos de Avaliação



Datas e Pesos das Avaliações 2026/1 - 4o semestre

| Datas | | | | | Pesos | | | | |
|-------------|------------------------------|--------------------------|------------------------------|------------------------|---------|---------|------------|-----|------------|
| Requisito 1 | Prova 1 e Conj. Exercícios 1 | Requisitos 2 e 3 (Final) | Prova 2 e Conj. Exercícios 2 | Exame | Prova 1 | Prova 2 | Exercícios | ABP | Portifólio |
| A definir | 10/04/26 (6a-feira) | A definir | 19/06/26 (6a-feira) | 06/07/26 (2a-feira) | 25% | 30% | 10% | 25% | 10% |

Sub-Exame: ?

Bibliografia Básica

- ❑ HUMBLE J; PRIKLANDNICKI R. **Entrega Contínua:** Como Entregar Software de Forma Rápida e Confiável. São Paulo: Bookman, 2013.
- ❑ MUNIZ, A.; et al. **Jornada DevOps:** Unindo Cultura Ágil, Lean e Tecnologia Para Entrega de Software Com Qualidade. São Paulo: Brasport, 2019.
- ❑ SATO D. **DevOps na prática:** entrega de software confiável e automatizada. São Paulo: Casa do Código, 2014.
- ❑ SILVA, R. **Entrega contínua em Android:** Como automatizar a distribuição de apps. São Paulo: Casa do Código, 2016.

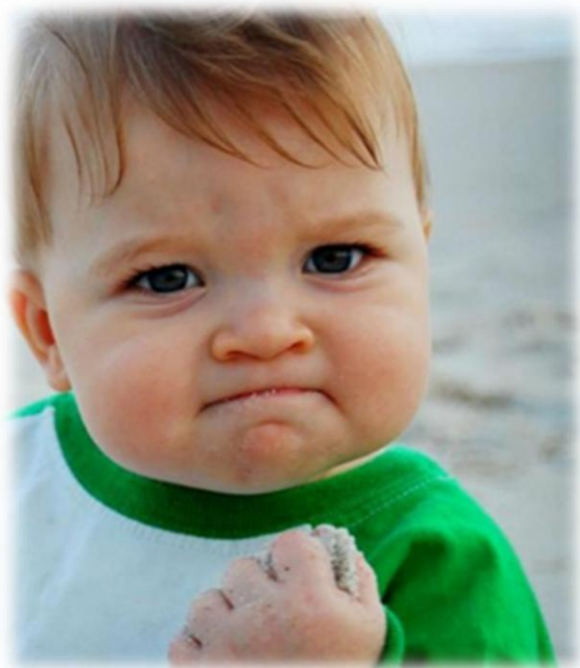
Bibliografia Complementar

- ❑ ARUNDEL, J. DOMINGUS, J. **DevOps nativo de nuvem com Kubernetes**. São Paulo: Novatec, 2019.
- ❑ MORAES, G. **Caixa de Ferramentas DevOps**: Um guia para construção, administração e arquitetura de sistemas modernos. São Paulo: Casa do Código, 2015. PIRES, A.; MILITÃO, J. **Integração Contínua com Jenkins**. São Paulo: Casa do Código, 2019.
- ❑ VITALINO, J. F. N.; CASTRO, M. A. N. **Descomplicando o Docker**. 2 ed. São Paulo: Brasport, 2018.
- ❑ SILVERMAN, R. E. **Git**: guia prático. São Paulo: Novatec, 2019.

Dúvidas?



Considerações Finais



**Professora:
Lucineide Pimenta**

Bom semestre à todos!

