

# Metodos de Desenvolvimento de Software - Plano de ensino

---

**DISCIPLINA:** Metodos de Desenvolvimento de Software

**CARGA HORÁRIA:** 60 horas

**PROFESSOR:** Carla Rocha

**CREDITOS:** 04

**SEMESTRE/ANO:** 01/2020

## Objetivos da Disciplina

Métodos de desenvolvimento de software podem ser entendidos como conjuntos estruturados de boas práticas, podendo ser repetíveis durante o processo de produção do software.

Nesse contexto, a disciplina Metodos de Processos de Software se faz importante para os futuros Engenheiros de Software por apresentar diferentes métodos de desenvolvimento, com enfoque especial aos diferentes ciclos de vida e técnicas de desenvolvimento de software. Os principais objetivos são:

- Capacitar o aluno a compreender os diferentes métodos, ferramentas, procedimentos e paradigmas de desenvolvimento de software.
- Capacitar o aluno a aplicar / adaptar processos de desenvolvimento de software a resolução de problemas de software.
- Capacitar os estudantes para construir sistemas complexos e distribuídos, utilizando metodologias de desenvolvimento e tecnologias web / mobile atuais.

## Ementa do Programa

Modelos de ciclo de vida e de processos; Processo Unificado. Métodos Ágeis de desenvolvimento de software. Outras abordagens de desenvolvimento de software (orientado a dados, orientado a funções, orientado a objetos, orientado a aspectos). Ferramentas.

## Formação das equipes

---

- Planilha para definição dos grupos e temas [aqui](#)

## Canais de Comunicação

---

Toda a disciplina será realizada de modo remoto. Serão disponibilizado tanto material assíncrono quanto aulas síncronas. A presença das aulas assíncrona será realizada por meio de exercícios disponibilizado junto ao material síncrono.

## Dúvidas, conversas rápidas, avisos

- Grupo do telegram - [t.me/mds2020\\_1](https://t.me/mds2020_1)

## Aulas síncronas

- Video conferência no jitsi - <https://meet.jit.si/mds2020-1> (Senha de entrada: código da disciplina)
- Todas as aulas síncronas serão gravadas e disponibilizadas no [canal do youtube](#)

## Aulas assíncronas

- Vídeos disponibilizadas no youtube - [canal youtube]  
([https://www.youtube.com/channel/UC\\_VXpS5GIL8NdJNkwNeAorw/videos?view\\_as=subscriber](https://www.youtube.com/channel/UC_VXpS5GIL8NdJNkwNeAorw/videos?view_as=subscriber))
- Leituras sugeridas na sprint - disponibilizados no planejamento das aulas

## Planejamento das aulas

---

- O planejamento das aulas semanais, discriminando se são assíncronas ou síncronas, e qual canal vai ser atualizado no início da semana no [link](#)

## Descrição do Programa

### Processos de Desenvolvimento de Software

- Modelos de Processo de Desenvolvimento de Software (ciclo de vida)
- Atividades de Processo

### Fundamentos do Extreme Programming

- O manifesto Ágil
- Os Quatro valores e as Quatro variáveis
- Práticas ágeis
- O jogo do planejamento
- Releases Pequenas
- A metáfora
- Histórias do Usuário
- Desenho simples
- Testes (unitário, aceitação)
- Refatoração
- Programação em Pares
- Desenvolvimento Coletivo

### Fundamentos do Processo Unificado de Desenvolvimento de Software

- Conceitos
- Fases: Iniciação, Elaboração, Construção e Transição
- Disciplinas (Modelagem de Negócio, Requisitos, Análise e Desenho, Implementação, Teste, Gerenciamento de Projeto, Gerência de Configuração e Mudanças, Implantação e Ambiente)

## Avaliações e Critérios de Avaliação

A avaliação será feita por meio de:

- EP1 a EPn: Entregas do Projeto.
- MT1 a MTn: MiniTeste Individual e Presencial (Prova).
- P1: Participação em sala de aula

O objetivo do Projeto simula uma situação real de desenvolvimento e engenharia de Produto de Software. Os alunos de MDS irão se concentrar na execução metodologia de desenvolvimento através da especificação de requisitos, codificação e testes. Haverá duas avaliações formais das releases a serem desenvolvidas.

Os pesos atribuídos aos diferentes eventos de avaliação são indicados abaixo.

| Evento da Avaliacao            | Peso |
|--------------------------------|------|
| Projeto (avaliacao individual) | 70%  |
| Criterio Extra de avaliacao*   | 10%  |
| Avaliacao Individual           | 20%  |

- **Criterio Extra de avaliacao** consiste em criterios propostos a cada semestre, a serem implementados de forma opcional.
- **Avaliacao Individual** pode ser tanto por meio dos minitestes, como contribuicao para a wiki da disciplina (conteudo abordado em sala de aula, issues abertas no repositorio da disciplina, tutoriais), ou contibuicao para outros repositórios, a nao ser o repositorio da disciplina. Os alunos serao previamente avisados sobre o criterio de avaliacao individual adotado no semestre.

Para o aluno satisfazer os seguintes requisitos para obter a aprovação na disciplina:

- Aprovação se MF  $\geq$  5,0 e se Percentual de faltas (PF) for PF  $\leq$  25%. Onde PF é dado pelo número de aulas com faltas registradas dividido pelo número de aulas ministradas.
- Reprovação se MF  $<$  5,0 ou se PF  $>$  25%. Nessa situação o aluno será considerado reprovado por nota ou por falta.

Os criterios avaliados individualmente no projeto esta destacado na tabela abaixo

| Evento da Avaliacao Individual no projeto          |
|--|
| Codigo/ Entrega                                    |
| Documentação                                       |
| Coerência - Documentos e Código                    |
| Critério Extra                                     |
| Histórias e Planejamento da Release                |
| Testes Automatizados e Cobertura de Código $>$ 90% |
| Tracking   |

## Evento da Avaliacao Individual no projeto

|  |
|--|
| Wiki Atualizada                                |
| Software Implantado e Disponível para Uso      |
| PA - pareamento                                |
| PA - reuniao de planejamento da sprint         |
| PA - planning poker                            |
| PA - sprint time box                           |
| PA - participacao nas daylies                  |
| PA - review com o cliente                      |
| PA - retrospectiva na sprint                   |
| PA - user stories                              |
| PA - risco sustentavel de trabalho             |
| PA - codigo escrito com padroes                |
| PA - plano de comunicacao                      |
| PA - comunicacao tecnica nas issues            |
| PA - pull requests educativos                  |
| PA - praticas de comunidades de software livre |

## Avisos

- Também são considerados critérios de avaliação da participação: assiduidade; pontualidade; interesse; participação em sala.
- Os documentos referentes à disciplina, estarão disponíveis em: <https://github.com/fga-eps-mds/A-Disciplina>
- Os casos não previstos de perda de avaliação serão avaliados individualmente, de acordo com as circunstâncias.
- Os projetos são avaliados continuamente
- A cobertura de código deverá ser 90%, excetuando a camada de apresentação
- O tamanho dos times deve respeitar o limite máximo de 6 membros.
- As atividades do projeto deverão ser organizadas por meio de issues e milestones.
- O código-fonte e demais artefatos elaborados deverão ser revisados utilizando pull/merge requests e issues.
- A presença será realizada computada pela realização de atividades (exercícios propostos) tanto em aulas síncronas quanto aulas assíncronas.

## Cronograma

O cronograma das aulas e os detalhamento da avaliação das Releases 1 e 2 estão disponíveis [aqui](#)

## Datas das Releases 1 e 2

- Release 1 - 08 de outubro de 2020
- Release 2 - 10 de dezembro de 2020

## Bibliografia Basica

- Beck, K., Programacao Extrema (XP) Explicada, 1st ed. Bookman, 2004
- Jacobson, I., Booch G., Rumbaugh J., The Unified Software Development Process, 1st ed., Addison-Wesley, 1999.
- Sommerville, I., Engenharia de software. 8th ed., Pearson Addison Wesley, 2007. – Pfleeger, S. L., Engenharia de software: teoria e pratica. 2nd ed., Prentice Hall, 2004. – Pressman, R. S., Engenharia de software. 6th ed., McGraw-Hill, 2006. – Ambler, S., Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process, 1st ed., Wiley, 2002 – Jacobson, I., Booch G., Rumbaugh J., UML: Guia do Usuario, 2nd ed., Elsevier, 2005. OPENACCESS ScrumXPdiretodasTrincheiras. (<http://www.infoq.com/br/minibooks/scrum-xp-from-the-trenches>)

## Bibliografia Basica:

- Beck, K., Programação Extrema (XP) Explicada, 1st ed. Bookman, 2004.
- Jacobson, I., Booch G., Rumbaugh J., The Unified Software Development Process, 1st ed., Addison-Wesley, 1999.
- [EBRARY] Lano, K., UML 2 Semantics and Applications, 1st ed., Wiley, 2009.

## Bibliografia Complementar :

- Sommerville, I., Engenharia de software. 8th ed., Pearson Addison Wesley, 2007.
- Pfleeger, S. L., Engenharia de software: teoria e prática. 2nd ed., Prentice Hall, 2004.
- Pressman, R. S., Engenharia de software. 6th ed., McGraw-Hill, 2006.
- Ambler, S., Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process, 1st ed., Wiley, 2002
- Jacobson, I., Booch G., Rumbaugh J., UML: Guia do Usuário, 2nd ed., Elsevier, 2005.
- [OPEN ACCESS] Scrum e XP direto das Trincheiras. (<http://www.infoq.com/br/minibooks/scrum-xp-from-the-trenches>)