

Cronograma turma Carla

Aqui serão atualizados o cronograma (data da aula), tipo de aula (síncrona/assíncrona), material de apoio e forma de avaliação do semestre.

As aulas síncronas terão o timebox de 40 minutos.

Sprint 0 - 17/08 e 19/08

- 18/08 - aula síncrona no <https://meet.jit.si/mds2020-1> (Senha de entrada: código da disciplina) - Apresentação do plano de ensino disciplina + apresentação sobre o trabalho da disciplina - definição dos times (máximo 6 membros - balanceado)
- 20/08 - aula síncrona no <https://meet.jit.si/mds2020-1> (Senha de entrada: código da disciplina) - Apresentação do projeto (definição dos temas)
- Material assíncrono - Vídeo sobre a metodologia da disciplina apresentada na campus party 2020 - <https://youtu.be/MIGYHI3lyyg> + Responder ao questionário <https://forms.gle/qjiNRJPWSLnKFVz5A>
- Material Assíncrono - aula sobre git e gitflow da [canal do BOSS](#)

Sprint 1 - 25/08 - 27/08

- 25/08 - Aula síncrona -
- Conteúdo assíncrono - Projeto de software e Modelos de ciclo de vida e processos de desenvolvimento de software - disponíveis no canal do youtube + formulários avaliativos
- Síncrono - stand up com os líderes do projeto (30 minutos)

Sprint 2

- Introdução aos métodos ágeis de desenvolvimento
- Práticas - XP (Artefatos, rituais, papéis)

Sprint 3

- Práticas ágeis - XP - planning poker
- Práticas ágeis - XP - história de usuário + critérios de aceitação

Sprint 4

- Práticas Ágeis

Sprint 5

- Práticas ágeis

Sprint 6

- Práticas Ágeis

Sprint 7 - Entrega Release 1

- **A Release 1** é realizada no horário do almoço (11h30 -14h00) durante toda a semana. A ordem de apresentacao é sorteada, mas os grupos sao podem trocar entre si a ordem.
- **Peso da R1** - 40% da nota de projeto
- **Tempo de apresentação:** 15 minutos
- **Artefatos avaliados:** (I) Documento de Visão do Projeto, (II) Planejamento/Comunicação Interna e Externa (agenda de trabalho + ferramentas), Documento de Arquitetura do Projeto, (III) Especificação das historias de usuários (critérios de aceitação), (IV) Configuração do repositório de acordo com os padrões de comunidade de software livre (Github), (V) Wiki atualizada (VI) Protótipo de alta fidelidade
- **Praticas avaliadas:** pareamento, produtividade, participação nos rituais, desempenho
- **Detalhamento da avaliação:** Código-Fonte/entregas (30%), (II) Coerência entre a documentação e implementação (10%), (III) Documentação: Doc. visão (15%), doc arq (15%), Requisitos Especificados (10%), Protótipo (10%), (IV) critério extra(10%)

Sprint 8

- Praticas ageis - testes unitarios (automacao)
- Praticas ageis - testes de aceitacao

Sprint 9

- Métodos ágeis - praticas ágeis - XP: integração continua

Sprint 10

- Métodos ágeis - praticas ágeis - Engenharia de Produto - full stack
- Métodos ágeis - praticas ágeis - Engenharia de Produto - visao geral de pipeline full stack

Sprint 11

- Métodos ágeis - praticas ágeis - Engenharia de produto: entrega continua (pipeline) + gitflow
- Métodos ágeis - praticas ágeis - Engenharia de produto: entrega continua

Sprint 12

- Métodos ágeis - praticas ágeis - XP: monitoramento e controle - retrospectiva + velocity + PR
- Métodos ágeis - praticas ágeis - indicadores e métricas no contexto de desenvolvimento ágil

Sprint 13

- Métodos ágeis - praticas ágeis - Entrega Continua - indicador e metricas ageis
- Métodos ágeis - praticas ágeis - Entrega Continua - Qualidade de codigo-fonte

Sprint 14

- Entrega continua - integração continua vs deploy continuo (papel do devops)
- Entrega Continua - Implantação (papel do devops)

Sprint 15

- Buffer de Contingência

Sprint 16 - Release 2

- **A Release 2** é realizada no horario do almoco (11h30 -14h00) durante toda a semana. A ordem de apresentacao é sorteada, mas os grupos sao podem trocar entre si a ordem.
- **Peso da R2** - 60% da nota de projeto
- **Tempo de apresentacao:** 15 minutos
- **Detalhe da avaliacao:** (I) Código-fonte entregue (60%): features, implantacao, qualidade, cobertura de testes, testes de aceitacao, (II) documentação/evidências da execução metodologia de desenvolvimento contínua (30%), (III) Tracking (10%)

Tutoriais

[Web Developer Roadmap](#) [Spellbook of Modern WebDev](#) [System Design Primer](#)

Ferramentas

Referências

Engenharia de Software

[Engenharia de Software Moderna](#)

[Code Complete](#) [Steve McConnell]

[Clean Code: A Handbook of Agile Software Craftsmanship](#) [Uncle Bob]

[The Pragmatic Programmer: From Journeyman to Master](#) [Andy Hunt, Dave Thomas]

Arquitetura de Software

[Clean Architecture](#) [Uncle Bob]

[Building Evolutionary Architectures: Support Constant Change](#) [Neal Ford, Rebecca Parsons, Patrick Kua]

Microserviços

[Building Microservices: Designing Fine-Grained Systems](#) [Sam Newman]

Design

[The Modern Web Design Process \[Jeff Cardello, John M. Williams\]](#)

Métodos Ágeis

[Scaling Agile @ Spotify with Tribes, Squads, Chapters & Guilds \[Henrik Kniberg, Anders Ivarsson\]](#)

[Software Engineering: A Practitioner's Approach](#)

[Kanban em 10 passos \[Jesper Boeg\]](#)

[Extreme Programming Explained: Embrace Change \[Kent Beck, Cynthia Andres\]](#)

DevOps

[The DevOps Handbook \[Gene Kim, Jez Humble, Patrick Debois, John Willis\]](#)

[Site Reliability Engineering: How Google Runs Production Systems \[Betsy Beyer, Chris Jones, Jennifer Petoff, Niall Murphy\]](#)

[Accelerate: The Science of Lean Software and DevOps \[Nicole Forsgren, Jez Humble, Gene Kim\]](#)