



## Projeto Final: Fase 1

13/10/2025

# 1 Visão Geral do Processo de Desenvolvimento

**Objetivo.** Desenvolver, em grupos de 2 a 3 alunos, um MVP (*Minimum Viable Product*) de um software, passando por todas as etapas da Engenharia de Software estudadas nesta disciplina.

**Repositório.** Cada grupo usará um repositório do GitHub Classroom<sup>1</sup> contendo:

- Pasta `doc/` para a documentação do projeto.
- Pasta `src/` com código-fonte.
- Arquivo `README.md` com resumo do projeto e instruções de instalação e uso.

**Processo.** Durante a disciplina de Engenharia de Software, cada grupo de alunos será responsável pelo desenvolvimento de um software, seguindo um processo incremental que acompanha os conteúdos abordados ao longo da disciplina. A proposta é que teoria e prática avancem lado a lado: à medida que novos temas são explorados em sala, o projeto evolui com base nessas etapas.

**Fases.** O projeto será organizado em fases orientadas por marcos de entrega, iniciando com a ideiação e levantamento de requisitos, passando pela modelagem e projeto, até a implementação de um MVP e a realização de testes. Toda a documentação e o código serão armazenados no GitHub Classroom, com instruções detalhadas fornecidas a cada fase.

# 2 Fase 1: Iniciação e Levantamento de Requisitos

Nesta primeira etapa do projeto, o grupo deve definir a ideia do sistema que irá desenvolver e iniciar a documentação dos requisitos. A proposta deve ser clara, viável para um MVP (produto mínimo viável) e relevante para algum perfil de usuário. Em seguida, será feito o levantamento dos requisitos e a modelagem inicial com base nas técnicas apresentadas em aula.

## Objetivos da Etapa:

- Definir tema e escopo do sistema.
- Identificar usuários, problema e propósito da aplicação.
- Levantar requisitos funcionais e não funcionais.
- Identificar os principais casos de uso e representá-los textual e graficamente.
- Estruturar os documentos do projeto no repositório.

---

<sup>1</sup>Link informado pelo professor via sistema acadêmico.

**Organização do Repositório.** No repositório GitHub do projeto, crie a seguinte estrutura inicial:

```
doc/
|-- requisitos.md
|-- casos-de-uso/
    |-- casos-de-uso.md
    |-- caso-de-uso-01-nome_do_caso_de_uso.md
    |-- caso-de-uso-02-nome_do_caso_de_uso.md
    |-- caso-de-uso-03-nome_do_caso_de_uso.md
    |-- ...
|-- diagramas/
    |-- diagrama-casos-de-uso.png
src/
README.md
```

### O que deve conter cada arquivo:

- README.md
  - Nome do sistema
  - Nomes dos integrantes do grupo
  - Descrição geral do projeto (1 a 2 parágrafos)
  - Público-alvo ou usuários principais (envolvidos/*stakeholders*)
  - Problema a ser resolvido e motivação
- requisitos.md
  - Lista de requisitos funcionais numerada
  - Lista de requisitos não funcionais numerada
  - Lista de regras de negócio numerada
  - Redija cada requisito com clareza e objetividade
  - Evite termos vagos como “deve ser fácil de usar” – procure quantificar ou qualificar
- casos-de-uso/casos-de-uso.md
  - Este arquivo deve ter uma lista dos casos de uso identificados do sistema, e deve exibir o diagrama de casos de uso do sistema.
  - Na listagem de casos de uso, ordene-os por prioridade – os primeiros casos de uso devem ser os essenciais para o MVP.
- casos-de-uso/
  - Cada arquivo de caso de uso deve descrever um caso de uso do sistema.
  - Para cada caso de uso, informar, ao menos:
    - \* Nome do caso de uso
    - \* Ator principal/primário
    - \* Atores secundários (se houver)
    - \* Meta no contexto do projeto
    - \* Pré-condições
    - \* Fluxo principal / Cenários
    - \* Exceções e Fluxos alternativos

- \* Prioridade
  - \* Frequência de uso
  - \* Canal de interação com atores primário e secundários
  - \* Questões em aberto (se houver)
- diagramas/diagrama-casos-uso.png
    - Diagrama de casos de uso desenhado com a ferramenta CASE StarUML

#### Dicas:

- Conversem com potenciais usuários ou colegas para simular entrevistas ou coleta de requisitos.
- Priorizem casos de uso centrais para o sistema.
- Mantenham os documentos simples e editáveis – eles poderão ser revisados em fases posteriores.

**Prazo de entrega.** A ser definido e comunicado em aula ou via ambiente virtual da disciplina.

#### Plágio não é tolerado



Você deve ser o único(a) responsável por fazer a entrega para essa atividade. Todo o código ou texto deverá ser produzido exclusivamente por você, exceto trechos de códigos que possam ter sido fornecidos como parte do enunciado.

Você pode discutir com outros estudantes com o intuito de esclarecer pontos, isso é até incentivado, porém você não poderá copiar trechos de códigos, textos ou soluções de qualquer fonte (e.g. colegas da mesma turma ou de turmas anteriores, repositórios de códigos na Internet ou soluções providas por serviços como Copilot e ChatGPT).