

JIRA

# ACESSO

- <https://www.atlassian.com/br/software/jira>
- Cada membro do grupo deve se cadastrar no Jira
- Somente um membro irá criar o projeto e convidar os outros para participar

# Passos

- Usar projeto Scrum
- Usar tarefas e histórias de usuário
- Esse é um exemplo do meu espaço Jira
  - <https://joaochoma.atlassian.net/>

# Projeto no Jira

- No **Jira**, um **projeto Scrum** é um tipo de projeto pensado para trabalhar em **sprints** (iterações com começo e fim), usando principalmente **Backlog + Sprint + Board** (quadro).

# Dentro do projeto Scrum

- **Backlog**: onde ficam os itens “a fazer”, ordenados por prioridade; você arrasta itens para um **Sprint** quando for planejar.
- **Sprints**: “pacotes de trabalho” com duração fixa (1–4 semanas, típico) e objetivo. Sprint Planning define o que entra e como será feito.
- **Board**: visualização do sprint (To do / In progress / Done ou fluxo similar), para acompanhar o andamento.

# Recomendação

- **Epic** → **(Story/Task/Bug)** → **Sub-task**
- **Epic**: entrega grande (ex.: “Módulo de Orçamentos e Reimpressão”) que vai gerar várias entregas menores.
- **Story**: algo em “linguagem do usuário” (valor/funcionalidade).
- **Task**: trabalho técnico/atividade interna (ex.: “Configurar pipeline CI”, “Refatorar serviço X”).
- **Sub-task**: passos menores para concluir uma Story/Task (ex.: “Criar endpoint”, “Criar tela”, “Escrever testes”).

# **Epic: “Listagem e reimpressão de orçamentos”**

- listar orçamentos com filtros
- visualizar detalhes
- reimprimir (PDF/print)
- logs/testes e validações concluídos

# Stories (adaptando casos de uso)

- **UC-01 / Story:** “Como usuário, quero listar orçamentos para localizar rapidamente um registro”
  - Aceitação: paginação, filtro por data/cliente, ordenação
- **UC-02 / Story:** “Como usuário, quero abrir detalhes do orçamento para conferir itens e valores”
- **UC-03 / Story:** “Como usuário, quero reimprimir o orçamento para enviar ao cliente”

# Tasks técnicas (se precisar)

- “Padronizar template de impressão”
- “Criar endpoint de exportação”
- “Ajustar permissões de acesso”

# **Sub-tasks por story**

- UI (tela / componente)
- API (endpoint/serviço)
- Persistência (query / index)
- Testes (unit/integration)
- Revisão/QA

# Como isso vira Sprint (planejamento)

- No **Sprint Planning**, você:
  - define o **Sprint Goal** (objetivo do sprint),
  - seleciona **Stories/Tasks** do backlog que cabem na capacidade,
  - detalha “como fazer” (sub-tasks), estimando esforço.

# Como encaixar “Casos de Uso” (UML) dentro do Jira

- **Opção 1 (mais comum): Caso de Uso = Story (ou conjunto de Stories)**
- **Epic** = “Módulo X”
- **Caso de Uso (UC)** vira uma **Story** (ou uma Story “guarda-chuva”)
- Cenários/fluxos (principal e alternativos) viram **sub-tasks** ou **critérios de aceitação** da Story

# Exemplo

- **Epic:** “Orçamentos”
  - **Story (UC-01):** “Cadastrar orçamento”
  - Critérios de aceitação: fluxo principal + alternativos
  - Sub-tasks: UI, API, testes, persistência

## Opção 2: Caso de Uso = Task e as Stories viram “fatias”

- Use quando o “caso de uso” é amplo e você quer entregar em partes:
- **Task (UC-01)** descreve o caso de uso completo (referência/controle)
- Várias **Stories** entregam o UC em incrementos (MVP → melhorias)