

# Padronização e Gerenciamento de Projetos

Definição, organização e explicação de processos, termos, práticas e padrões que serão utilizados no LJIT.

- [Gerenciamento de Projeto](#)
  - [Glossário](#)
  - [Definição de Scrum](#)
  - [Definição do Board](#)
  - [Definição de Tarefas](#)
  - [Ciclo de Vida das Tarefas](#)
  - [Padrões e Tipos de Branch](#)
  - [Padrões e Tipos de Commits](#)
  - [Padrão de Pull Requests](#)

# Gerenciamento de Projeto

Aqui se encontram documentos que tratam sobre o gerenciamento de projetos e padronização de tarefas, Scrum e processos.

# Glossário

Existem termos que serão utilizados aqui, porém que existem outras definições ou utilizações fora do contexto do Laboratório.

Aqui estão relatados as definições de acordo com o contexto em que utilizamos.

<b>Termo</b>	<b>Significado</b>
Board	Quadro de tarefas
Task	Tarefa única, com requisitos, critérios de aceitação e responsável
Backlog	Repositório de tarefas a serem refinadas, descritas, atuadas e concluídas
Sprint	Período de 2 a 4 semanas, que possui um objetivo a ser atingido e uma quantidade de tarefas a serem realizadas
Daily	Reunião diária de manutenção do time, realizada para relatar o que foi realizado no dia anterior, o que será realizado no dia e se há algum problema na realização para cada um dos integrantes. No nosso contexto, é um momento diário, onde os relatos ocorrerão via mensagem na plataforma oficial de comunicação (Discord)
Branch	Ramo criado a partir de um ponto no repositório de código de um projeto
Deploy	Lançamento de uma versão do projeto, normalmente vinculado à um lançamento de versão de código e uma branch somente para isso
Pacote	Coleção de tarefas vinculadas à uma versão ou sprint
Pull Request	São propostas de merge, vinculadas de uma branch à outra, com o intuito de facilitar a revisão das mudanças realizadas
Merge	Integração de código em uma branch diferente
Commit	Uma instância de versionamento do código no repositório
Ambiente	Local onde está sendo executado um projeto específico, são isolados e possuem uma finalidade

Termo	Significado

# Definição de Scrum

## O que é Scrum

- É um processo ágil que nos permite focar em entregar o maior valor de negócio no menor tempo.
  - Permite inspecionar software em funcionamento mais rapidamente e repetidamente (em intervalos de duas semanas a um mês).
  - O negócio define as prioridades. Times se auto organizam para determinar a melhor forma de entregar as features com mais alta prioridade.
  - Em intervalos de duas semanas a um mês, qualquer um pode ver software em funcionamento e decidir fazer o release dele da forma como está ou continuar a melhorá-lo para outra sprint.
- 

## Características

- Times auto-organizados
- 

## Manifesto Ágil

- Indivíduos e interações mais do que Processos e ferramentas
- Software em funcionamento mais do que Documentação abrangente
- Colaboração com cliente mais do que Negociação de contratos
- Responder a mudanças mais do que Seguir um Plano

## Componentes

- Backlog - Funcionalidades esperadas do software
  - Sprint - Desenvolvimento do software, dura normalmente de uma semana à 4 semanas
  - Objetivo da sprint - O que deve ser feito em uma sprint
  - Incremento de produto - é o produto potencialmente entregável
  - Diário
    - Daily Scrum Meeting - Reunião de alinhamento
- 

## Papéis

Product Owner - PO

- Define o que será feito na sprint, representa a ponte entre o cliente e o produto
- Prioriza features de acordo com o seu valor de mercado
- Aceita ou rejeita os resultados do trabalho

## Scrum Master - SM

- Representa o gerenciamento para o projeto
- Responsável por propagar as práticas do Scrum
- Remove impedimentos
- Garante que a equipe esteja totalmente funcionando e produtiva
- Permite estreita cooperação entre os papéis e funções

## Time Dev

- Tipicamente de 5 a 9 pessoas
- Múltiplas habilidades
- Programadores, testadores, UX designers, etc.
- Membros devem estar alocados full-time (geralmente)
- Times auto-organizáveis
- Composição do time só deve ser modificada entre sprints

---

## Cerimônias (Eventos)

### Planejamento da sprint

- Itens
  - Capacidade do Time
  - Backlog do produto
  - Condições do negócio
  - Produto atual
  - Tecnologia
- Priorização da sprint
  - Avalia o product backlog e define o objetivo da sprint
- Planning
  - Decide como atingir o objetivo
  - Criar o sprint backlog a partir do product backlog
- Definição
  - Time seleciona itens com o qual podem se comprometer
  - É criado o sprint backlog
  - Design de alto nível é definido

Ex:

História: Eu como Viajante, quero ver fotos dos hotéis

Tarefas:

- Codificar a camada intermediária
- Codificar UI
- Corrigir bugs existentes

## Daily Scrum

- Diário
- 15 minutos
- Todos em pé (para agilizar e ter a atenção dos participantes)
- Não é para resolver problemas
- Todos são convidados a participar
- Somente Time Dev, Scrum Master, Product Owner podem se manifestar
- Ajuda a evitar reuniões desnecessárias
- Todos respondem 3 questões
  - O que você fez ontem?
  - O que vai fazer hoje?
  - Existe algo te atrapalhando?
- **Não** é um status para o Scrum Master
  - São compromissos com os outros membros

## Sprint Review

- Time apresenta o que foi concluído durante a Sprint
- Geralmente na forma de uma demonstração das novas funcionalidades ou arquitetura
- Informal
  - 2 Horas
  - Sem slides
- Todo o time participa
- Todos são convidados

## Sprint Retrospective

- Periodicamente avaliar o que está e o que não está funcionando
- Tipicamente de 15 a 30 minutos
- Feito após cada sprint
- Todos participam

Uma das técnicas de review é o:

### **Começar / Parar / Continuar**

- Todo o time debate sobre o que eles gostariam de:
    - Começar a fazer
    - Parar de fazer
    - Continuar a fazer
-

# Artefatos

## Backlog do produto (Product Backlog)

- Os requisitos
- Lista com todo o trabalho desejado para o projeto
- Priorizado pelo Product Owner
- Repriorizado ao início de cada Sprint
- Escrito de forma a especificar os requisitos

## Objetivo da Sprint (Sprint Goal)

- Uma declaração do foco do trabalho durante a sprint
- Pode ser algo como um Épico (Tarefa geral ou módulo)

## Sprint Backlog

- Membros se comprometem com o trabalho escolhido por eles mesmos
  - Trabalho nunca é delegado por alguém
- Estimativa de trabalho restante é atualizada diariamente
- Qualquer membro do time pode adicionar, deletar ou modificar o Sprint Backlog
- O trabalho da Sprint emerge

## Gráfico de Sprint Burndown

- É um gráfico que demonstra a quantidade de horas restantes em cada semana da sprint
- À medida que o tempo passa, as horas diminuem
- Importante para tomar decisões críticas para o andamento do projeto

## Gráfico de Tarefas Cumpridas

- É um gráfico que demonstra a quantidade de horas cumpridas, em tarefas
- Útil para reconhecer gargalos no Time Dev

---

## Escalabilidade

- Tipicamente times de 7 +/- 2 pessoas
- Fatores importantes ao escalar
  - Tipo de aplicação
  - Tamanho do time
  - Dispersão do time
  - Duração do projeto
- Utilizando o Scrum of Scrums
  - Escala em equipes de scrums as equipes de scrums utilizando os representantes de cada time como um membro da equipe maior



# Definição do Board

## Fluxo

O fluxo do board segue de cima para baixo, da esquerda para direita.

### **Importante**

Assim que uma tarefa sai de uma coluna da esquerda pra direita, ela nunca pode voltar para a coluna anterior ou para colunas à esquerda (*vide casos especiais*), mesmo que hajam correções a ser feitas.

O intuito é sempre manter o fluxo das tarefas seguindo sem bloqueios.

## Colunas

O nosso board consiste de 5 colunas: **To-do**, **Fazendo**, **Revisando**, **Staging**, **Finalizadas**.

### To-do

Aqui ficam as tarefas da **Sprint** que ainda serão escolhidas por um dos integrantes para realizá-la.

### Fazendo

Aqui ficam as tarefas da **Sprint** que estão sendo trabalhadas por algum dos integrantes. Integrantes devem mover as tarefas de “To-do” para esta coluna somente quando começarem a atuar sobre a tarefa.

Todas as tarefas presentes nessa coluna deverão ter uma **branch** vinculada à elas.

Tarefas só poderão sair da coluna “**Fazendo**” após a finalização das ações descritas na tarefa.

### Revisando

Aqui ficam as tarefas da **Sprint** que foram finalizadas por um dos integrantes e estão sendo revisadas por algum outro integrante que não realizou a tarefa. Essa coluna ajuda a ver quais tarefas necessitam da ação de outro integrante do time.

A revisão é extremamente importante e reforça a qualidade das tarefas que realizamos.

Todas as tarefas presentes nessa coluna deverão ter uma **Pull Request** vinculada à elas.

Tarefas só poderão sair da coluna “**Revisando**” após passarem por uma revisão de outro integrante, testando e validando os **Critérios de Aceite** desta tarefa.

## Staging

Aqui ficam as tarefas da Sprint que já foram finalizadas, testadas e revisadas, e estão somente esperando o “deploy”.

Todas as tarefas presentes nessa coluna deverão estar “**mergeadas**” na **branch** de desenvolvimento.

Tarefas só poderão sair da coluna “**Staging**” após criada um **pacote** com todas as tarefas de “Staging” para a criação de uma nova versão.

## Finalizada

Aqui ficam as tarefas da Sprint que foram lançadas e não possuem nenhuma ação a ser realizadas por nenhum integrante.

---

## Bloqueios

Bloqueios são sinalizações de que uma tarefa há um bloqueio ou impedimento na conclusão da mesma. Bloqueios podem ocorrer em qualquer momento do “Ciclo de Vida” de uma tarefa.

Bloqueios devem ser sinalizados apropriadamente, tanto na ferramenta de Board quanto nas reuniões de Daily via mensagem no canal de comunicação oficial.

# Definição de Tarefas

As tarefas são a parte mais atômica do processo, elas podem ser categorizadas em 4 tipos:

## Tipos

### 1. Épicos

Descrevem um objetivo final massivo, como a finalização de uma funcionalidade complexa, um módulo, ou uma etapa do desenvolvimento como a **Prototipagem** ou **Levantamento de Requisitos**. Possui **Histórias** vinculadas.

### 2. Histórias

Descrevem uma História de Usuário, têm como objetivo descrever uma funcionalidade atômica, um fluxo de usuário ou uma interação do mesmo com o sistema a ser desenvolvido. Possui **Tarefas** vinculadas.

### 3. Tarefas

Descrevem o que deve ser feito pelo integrante do time. Geralmente são tarefas que compõe o desenvolvimento de uma funcionalidade (História) como a criação de uma tela, a criação de certas rotas de API, ou a criação de uma classe específica. Devem ser bem descritas utilizando o padrão descrito na Seção **Padrões**.

### 4. Bug

Descrevem um bug, erro ou inconsistência no sistema. Têm maior prioridade e possuem um ciclo de vida diferente do que a das tarefas comuns.

## Padrões

### 1. Épicos

1. Intenção - Deve descrever o motivo do porque esse Épico existe, a finalidade e o impacto que esse Épico provê.
2. Descrição Geral - Deve descrever de forma geral o objetivo das tarefas e histórias vinculadas à esse Épico.
3. Critérios de Aceite - Critérios gerais que definem se esse Épico está pronto para ser finalizado ou não
4. Recursos - Links, textos ou ferramentas de apoio para a realização deste Épico

### 2. História

1. Descrição Geral - Deve seguir o modelo de descrição de História de Usuário  
Como usuário quero fazer x, visualizar y para chegar em z.

2. Descrição Técnica - Deve descrever o que é necessário desenvolver para que o usuário realize a ação descrita na **Descrição Geral**
3. Critérios de Aceite - Critérios gerais que definem se essa História está pronta para ser finalizada ou não, devem cobrir o que foi descrito na **Descrição Geral**
4. Recursos (Opcional) - Links, textos ou ferramentas de apoio para a realização desta História

### 3. **Tarefas**

1. Descrição Geral - Descreve de forma geral o que será realizado nesta Tarefa
2. Descrição Técnica - Descreve o que deve ser feito a nível técnico, quais padrões devem ser utilizados, onde deverá ser criado, nome de módulo, ou quaisquer técnicas específicas da tarefa
3. Possíveis Impactos - Lista possíveis locais e funcionalidades que irão ser impactadas com a realização dessa tarefa
4. Critérios de Aceite - Critérios específicos que definem o que será revisado e testado nesta Tarefa
5. Recursos (Opcional) - Links, textos ou ferramentas de apoio para a realização desta Tarefa

### 4. **Bugs**

1. Severidade - Descreve a severidade do bug, têm 4 severidades, da maior para a menor: Urgente, Alta, Média e Baixa.
2. Descrição do Comportamento - Descreve como reproduzir o bug, como foi encontrado esse comportamento, (possivelmente com um vídeo demonstrando a reprodução)
3. Descrição Técnica - Descreve quais partes são possivelmente afetadas e o que deve ser feito para a resolução desse bug
4. Critérios de Aceite - Critérios que definem se o bug realmente foi resolvido e se será reproduzível novamente

# Ciclo de Vida das Tarefas

Cada tarefa possui um processo da sua criação à conclusão que comumente chamamos de “Ciclo de Vida”.

Esse Ciclo de Vida dita as ações que serão realizadas sobre as tarefas ao longo do seu processo de conclusão.

## Definição

O Ciclo, normalmente, se dá assim:

1. Cria-se a tarefa no Backlog
  1. Descreve em poucas palavras o intuito da tarefa em seu título assim como seu escopo, seguindo o padrão em [Definição de Tarefas](#).
  2. Descreve em alto nível, sem muito detalhamento, de acordo com o tipo e o template da tarefa, as informações necessárias ao template, como “**Descrição Geral**”, “**Descrição Técnica**” ou “**Critérios de Aceite**”.
2. A tarefa é discutida e possivelmente selecionada para ser realizada em uma Sprint
3. A tarefa entra no Board na coluna “**To-do**”
4. O integrante do time lê e escolhe a tarefa e a vincula a si
5. O mesmo integrante, ao começar a realizar a tarefa, move-a para a coluna “**Fazendo**”
  1. Caso seja uma tarefa de código, o desenvolvedor deve:
    1. Criar uma branch seguindo o padrão de branches descrito em [Padrão e Tipos de Branch](#).
    2. Vincular a branch à tarefa na plataforma de gerenciamento de tarefas.
    3. Realiza a tarefa.
6. Ao finalizar a tarefa, move-a para a coluna “**Revisando**”
  1. Caso seja uma tarefa de código, o desenvolvedor deve:
    1. Criar uma **Pull Request** descrevendo as alterações e utilizando o [Padrão de Pull Requests](#).
    2. Pedir para um outro desenvolvedor fazer o Code Review (Revisão de Código) das alterações.
    3. O outro desenvolvedor deve então testar a tarefa em sua máquina e verificar se a mesma possui e cumpre os **Critérios de Aceite** descritos na tarefa.
    4. Só então o outro desenvolvedor poderá aprovar a **Pull Request**.

5. Após a aprovação da Pull Request, o desenvolvedor deve realizar o merge da Pull Request na branch **develop**.
7. Após a revisão de pares, a tarefa poderá ser movida para a coluna “**Staging**”
8. Ao final da Sprint, é criado um pacote com as tarefas realizadas, onde será listadas todas as tarefas realizadas e quais funcionalidades foram adicionadas.
  1. Sobre a criação de versões, ler Criação de Versões.
9. Após o lançamento do pacote de tarefas, a tarefa poderá ser movida para a coluna “**Finalizada**”

# Padrões e Tipos de Branch

No momento utilizamos o padrão de pacotes de branch, ou seja, temos um padrão de nomenclatura de branches apropriado que ajuda na organização, empacotamento, versionamento e entrega das tarefas e do desenvolvimento.

Utilizamos o padrão de [Conventional Commits](#) para a nomenclatura do **intuito** das tarefas de acordo com o tipo de alteração realizada no sistema.

## Branches comuns

As branches têm de ser criadas com o seguinte padrão:

<intuito-da-tarefa>/<codigo-do-projeto>-<numero-da-tarefa>

Exemplo:

feat/SOL-132

fix/NEB-532

### Importante

O padrão é **obrigatório**, visto que a aplicação do board só vincula as tarefas às branches que possuem a **nomenclatura correta**.

## Branches especiais

Existem algumas branches especiais que têm propósitos específicos.

São essas:

## Branches de Ambiente

- develop - Branch integradora de códigos para o ambiente de desenvolvimento.

- staging - Branch integradora de códigos para o ambiente de homologação.
- prod - Branch integradora de códigos para o ambiente de produção.

## Branches Esporádicas

- release - Branch utilizada para integrar códigos antes do lançamento de uma versão em algum dos ambientes
- hotfix - Branch utilizada para criar correções rápidas e precisas, saem diretamente da master/main sem precisar de trazer código não finalizado dos outros ambientes

# Padrões e Tipos de Commits

Aqui utilizamos uma padronização das mensagens de commits baseado no padrão do [Conventional Commits](#).

## Padrão

As mensagens seguem o seguinte padrão:

<tipo>(escopo-opcional): <descrição-do-commit>

Exemplo:

feat(Carrossel): Criação da estrutura inicial do Carrossel

fix(Carrossel): Bug do Carrossel não passando imagens

## Tipos

Os principais tipos usados são:

- **feat** - descreve a adição de uma nova funcionalidade à base de código
- **fix** - descreve a correção de um problema ou bug existente na base de código
- **chore** - descreve tarefas “obrigatórias”, que não possuem um intuito de funcionalidade novas
- **refactor** - descreve a refatoração, limpeza de código ou renomeação de funções e variáveis
- **docs** - descreve a alteração ou criação de novas documentações na base de código ou em documentos como README
- **test** - descreve alterações em testes automatizados ou em testes de integração
- **merge** - descreve alterações de integração de código (deve ser usado com pouca frequência)

## Escopo

Descreve o contexto em que a mudança está sendo realizada, por exemplo, qual funcionalidade ou qual a página.

# Descrição

A descrição deve ser sucinta, **não passando mais de 15 palavras** e explicando a alto nível o que foi alterado.

# Padrão de Pull Requests

Pull Requests são uma parte importante do desenvolvimento de software, devem ser feitos com muita intencionalidade e esporadicamente.

As Pull Requests devem seguir um padrão de descrição vinculado ao tipo de tarefa que é vinculado.

## Padrões

“

### Tarefa

#### Descrição Geral

Descrição das alterações de forma geral

#### Funcionalidades Alteradas

Lista das funcionalidades alteradas

#### Recursos

Links, textos ou contexto para a Pull Request

#### Link para a Tarefa

Link para a tarefa na plataforma de gerenciamento de tarefas

#### Critérios de Aceite

Critérios de aceite da tarefa (Devem ser copiados da tarefa na plataforma de gerenciamento de tarefas)

# Bug

## Severidade do Bug

Severidade do Bug (Urgente, Alta, Média ou Baixa)

## Descrição Geral

Descrição das alterações de forma geral

## Funcionalidades Alteradas

Lista das funcionalidades alteradas

## Recursos

Links, textos ou contexto para a Pull Request

## Link para a Tarefa

Link para a tarefa na plataforma de gerenciamento de tarefas

## Critérios de Aceite

Critérios de aceite da tarefa (Devem ser copiados da tarefa na plataforma de gerenciamento de tarefas)