

---

# Modelo de Processo Ágil BJR

**Alexandre L'Erario, Breno Angelotti, Gabriel Romero, Jean  
Gonçalves, João Goulart, Mateus Merscher, Renan Batel**

*Universidade Tecnológica Federal do Paraná - Câmpus Cornélio Procópio*

---

**E**ste modelo de processo de software apresenta uma metodologia de desenvolvimento ágil e cíclico para pequenos times de desenvolvimento com foco em projetos SaaS.

*26 de setembro de 2018*

## Lista de figuras

1	Exemplo de figura com escala horizontal . . . . .	12
2	Exemplo de figura sem escala . . . . .	13
3	Exemplo de figura rotacionada . . . . .	14

## Lista de tabelas

1	Modifique a legenda e crie um label . . . . .	11
---	---	----

# Sumário

<b>1</b>	<b>Introdução</b>	<b>4</b>
<b>2</b>	<b>Processo</b>	<b>4</b>
2.1	Fases . . . . .	4
2.1.1	Levantamento de Requisitos . . . . .	4
2.1.2	Desenvolvimento . . . . .	4
2.1.3	Análise e Teste . . . . .	4
2.1.4	Entrega . . . . .	5
2.1.5	Manutenção . . . . .	5
2.2	Papéis . . . . .	5
2.3	Tarefas . . . . .	6
2.4	Disciplinas . . . . .	7
2.5	Artefatos . . . . .	8
2.6	Guidance . . . . .	8
2.7	Interações . . . . .	8
2.8	Iterações . . . . .	8
2.9	Milestones . . . . .	8
2.10	Ferramentas . . . . .	8
2.11	Feedback . . . . .	9
<b>3</b>	<b>Ciclo de vida</b>	<b>9</b>
<b>4</b>	<b>Referências Bibliograficas</b>	<b>11</b>
<b>5</b>	<b>Elementos textuais - Alguns exemplos</b>	<b>11</b>
5.1	Colocar elementos em itens . . . . .	11
5.1.1	Uma subseção de terceiro nível . . . . .	11
5.2	Tabelas . . . . .	11
5.3	Figuras . . . . .	12

# 1 Introdução

O modelo de processo de software apresentado neste documento é de estrutura ágil para equipes pequenas com projetos de empresas que tenham Software as a Service (SaaS). A proposta desse modelo é a autonomia do time de desenvolvimento e a visibilidade que cada desenvolvedor tem dentro do projeto, tudo isso contido dentro de um ciclo de desenvolvimento rápido e eficaz.

- Modelo de Processo Ágil BJR;
- Integrantes: Breno Angelotti; Gabriel Romero; Jean Carlos Gonçalves; João Victor Goulart de Almeida; Mateus Merscher; Renan Batel.
- Github: <https://github.com/goulartt/template>;

## 2 Processo

O modelo de processo de software desenvolvido é de categoria ágil com ciclo de vida cíclico. O foco desse processo é desenvolver de forma rápida, e dando autonomia para equipe de desenvolvimento, ideal para empresas que contenham SaaS e várias equipes pequenas (squads) de desenvolvimento.

### 2.1 Fases

#### 2.1.1 Levantamento de Requisitos

A fase inicial deste processo é a fase de Levantamento de Requisitos, onde é realizada a comunicação com o cliente. Essa comunicação é de extrema importância, sendo realizada através de reuniões onde serão estabelecidos parâmetros para toda a parte de planejamento, nestas reuniões devem ser levantados os requisitos do cliente e os objetivos que a entrega deve atingir, quanto mais informações forem obtidas mais claros ficarão os objetivos.

Ainda na fase de Levantamento de Requisitos ocorre o planejamento das fases seguintes com base no que foi levantado com o cliente, durante o planejamento deve ficar claro quais serão as prioridades para que se possa agregar valor à entrega, uma boa execução desta fase facilitará o fluxo das outras.

#### 2.1.2 Desenvolvimento

Finalizada a fase inicial do processo, os artefatos necessários para a construção do projeto já estão formalizados, sendo assim a próxima fase trata do Desenvolvimento.

A fase de Desenvolvimento começa com uma discussão sobre os itens que deverão ser produzidos. Durante essa fase são definidos quais itens serão feitos e qual será a ordem de execução baseado nas prioridades do cliente. Assim que os itens são definidos começa-se a produção dos mesmos.

#### 2.1.3 Análise e Teste

Após o término do Desenvolvimento, os itens produzidos devem ser testados, nesse momento temos a fase de Análise e Teste. Nesta etapa o time é responsável por verificar se o conteúdo gerado a partir da fase anterior possui erros ou pendências técnicas.

Quando o fluxo não apresenta erros ou falhas, é necessário validar se o conteúdo cumpre aquilo especificado no planejamento, podemos dizer que essa fase verifica de modo detalhado se os itens estão preparados para a fase seguinte ou não.

### 2.1.4 Entrega

A próxima etapa, de Entrega, depende fortemente de suas antecessoras, pois para que toda as suas tarefas sejam feitas, a especificação deve ter sido clara, a implementação deve ter ocorrido, e todos os itens devem ter sido validados e testados, nesta fase é registrado como os itens foram construídos e também o que realmente foi produzido nas duas fases anteriores.

Também é nesta fase onde ocorre o versionamento dos artefatos gerados, a versão considerada estável do produto é implantada no ambiente de produção e, para finalizar, o resultado do acompanhamento de todo o ciclo é apresentado para a equipe.

### 2.1.5 Manutenção

Ao final do nosso ciclo do processo temos a fase de Manutenção, nesta fase o time se preocupa em manter o funcionamento do sistema, nela é de grande importância o monitoramento do produto para a geração de relatórios, com foco especial nos itens do último pacote entregue, o relatório auxiliará mais tarde na priorização e construção de correções.

## 2.2 Papéis

O ideal para utilização desse modelo é uma equipe de 5 a 7 pessoas, sendo que toda equipe deve conter um Customer (cliente), um Agile Coach, um Product Owner, uma equipe de desenvolvimento e um Line Manager (Opcional), cuja suas respectivas funções são:

- **Customer:** Aquele que irá ditar o que será entregue.
- **Agile Coach:** Ajuda a manter o fluxo de trabalho, facilitando as retrospectivas, as reuniões de planejamento do ciclo, organizar a equipe de desenvolvimento e garantir a entrega prevista do ciclo.
- **Product Owner:** Responsável pelo backlog e define prioridades das histórias, se comunicando diretamente com os stakeholders e apresentar ao cliente o andar do desenvolvimento do software.
- **Squad:** Desenvolvedores responsáveis por um conjunto de funcionalidades.
- **Line Manager:** responsável por no máximo 5 squads, irá alinhar com as equipes o que está sendo feito, participará em algumas reuniões para acompanhamento do squad sem influenciar no processo. Responsável pelo coaching e desenvolvimento de carreira individual de cada membro dos squads. Acompanha reclamações e problemas para criar planos de ação que melhorem o processo.

## 2.3 Tarefas

Este processo de desenvolvimento não é orientado a atividades, e sim a tarefas diretamente, como se trata de um processo ágil com um ciclo de vida cíclico e de duração indeterminada, os deadlines de cada tarefa é definido por quem o aplica, entretanto, para um melhor aproveitamento do processo, é aconselhável que o deadline das tarefas de planejamento(Especificação de requisitos, Construção da história e Definição de backlogs e prioridades) não ultrapasse o início tarefa de Planejamento do Desenvolvimento, já as demais tarefas, o deadline sugerido é fase de Entrega. As definições de cada tarefa e seus papéis relacionados se encontram logo abaixo:

- **Especificação de requisitos:** Durante uma reunião entre o Product Owner e o Customer para estabelecer os requisitos do cliente, o Product Owner deverá extrair o máximo de informações possíveis do Customer, após essa reunião o Product Owner deve analisar o que foi obtido e identificar os requisitos do projeto, quanto mais informações forem obtidas mais fácil será identificar os requisitos do projeto para satisfazer os requisitos do cliente;
- **Construção da história:** O Product Owner construirá as histórias, com base nos requisitos já levantados por ele, as Histórias servirão para esclarecer melhor os requisitos e as funcionalidades, elas devem ser simples, curtas e claras, devem ser feitas utilizando do ponto de vista de um usuário;
- **Definição de backlogs e prioridades:** O Product Owner deve montar o backlog com o que será feito e definir as prioridades de cada um dos itens com base nas prioridades do cliente;
- **Manter fluxo:** Durante o caminhar do projeto o Agile Coach deve ajudar a manter o fluxo de execução das fases e tarefas seguindo como o planejado e ajudar na comunicação entre as equipes de desenvolvimento;
- **Planejamento de Desenvolvimento:** Em uma reunião o Squad junto do Agile Coach definirão os itens que a serem desenvolvidos durante este ciclo do projeto e a duração do ciclo com base nas estimativas de tempo levantadas pelo Squad para cada item, 20% do tempo estimado será dedicado aos testes e correções e 10% do tempo estimado deve ser definido como respiro;
- **Implementação da Sprint:** Os itens definidos na tarefa de Planejamento de Desenvolvimento devem ser desenvolvidos pelo Squad dentro do tempo que foi planejado;
- **Correções:** O Squad responsável realizará as correções que forem identificadas durante o desenvolvimento ou durante os testes no tempo dedicado a correções definido no Planejamento de Desenvolvimento;
- **Criação dos casos de teste:** : os casos de testes são especificados pelo Squad a partir das funções desenvolvidas, as especificações incluem testes unitários e testes de integrações entre as funções, os casos devem ser registrados de forma rastreável aos requisitos;
- **Construção dos testes:** À partir da especificação dos casos de teste, o Squad deve realizar a construção dos scripts de teste;

- **Execução dos testes:** Após a criação dos testes os mesmos deverão ser executados pelo Squad no produto nos diferentes ambientes necessários, e caso encontrado algum comportamento diferente do esperado, é necessário rastrear a causa e realizar a correção;
- **Validação:** O Squad analisa os resultados dos testes e das correções e o pacote funcional é liberado para entrega, caso ainda haja falhas elas não devem estar incluídas no pacote a ser entregue;
- **Documentação dos artefatos:** Todos os artefatos são registrados (P.O, Agile Coach, Squad);
- **Versionamento:** Os artefatos são versionados (P.O, Agile Coach, Squad);
- **Implantação:** Após todas as validações um pacote sem falhas e estável é enviado para produção pelo Squad;
- **Análise e priorização da manutenção:** o Agile Coach junto com o Squad é responsável pela análise do relatório de erros coletados do sistema, o Agile Coach analisa as incidências e número de usuários afetados para apontar a relevância do erro, o Squad deve discutir o nível de dificuldade para e qual o impacto da implementação da correção, feito isso, os números são analisados e as ações/correções são priorizadas;
- **Análise e priorização da manutenção:** o Agile Coach deve realizar uma nova análise sobre as ações/correções priorizadas e definir quando as mesmas serão implementadas.

## 2.4 Disciplinas

O componente de disciplina se aplica ao conjunto de atividades relacionadas, as quais podemos associar:

- **Desenvolvimento:** Implementação da Sprint, Correções, Versionamento, Implantação;
- **Teste de Software:** Criação dos casos de teste, Construção dos testes, Execução dos testes;
- **Sustentação:** Planejamento da Manutenção, Feedback, Análise e priorização da manutenção;
- **Gestão de Projetos:** Feedback, Manter Fluxo, Planejamento de Sprint, Validação, Documentação, Versionamento;
- **Análise de Requisitos:** Especificação de requisitos, Construção da História, Definição de Backlog e priorização, Validação;

## 2.5 Artefatos

## 2.6 Guidance

## 2.7 Interações

O processo possui apenas duas principais interações com os Stakeholders do projeto.

Primeiramente temos uma interação constante entre Line Manager, P.O e Customer, onde são sempre discutidas as demandas do último. É responsabilidade do Line Manager e do P.O a extração de informações das demandas do Customer para a definição dos requisitos, é nessa interação também onde os dois validam as entregas realizadas para verificar se a os requisitos estão sendo atendidos e se novas demandas estão surgindo.

A segunda interação presente no processo trata-se do alinhamento dos Squads com o Line Manager e Product Owner para um feedback dos Squads, dos membros individualmente e do ciclo como um todo, pontos positivos, pontos negativos, o que se pode aproveitar ou não, coisas novas que foram testadas e demonstraram uma melhoria, ou algo que falhou durante o ciclo. Esta reunião serve como base para um histórico de aperfeiçoamento do processo ao longo das Iterações, a cada ciclo o processo deve ser refinado até atingir os níveis de desempenho desejados, isso não impede que novos métodos sejam testados;

## 2.8 Iterações

## 2.9 Milestones

Em cada ciclo teremos todas as nossas fases do processo realizadas novamente, deste modo, para que não haja milestones desnecessários ou que possam tornar o processo confuso e/ou lento, foram definidos apenas dois marcos principais em nosso processo. O primeiro se dá ao fato de Definir o Backlog, pois esta tarefa irá definir tudo que será utilizado nas fases e tarefas posteriores, ou seja, com um backlog definido em mãos temos o conteúdo necessário para prosseguir no processo e uma informação para ser comparada com a entrega.

Quando os itens definidos no backlog passarem pelas fases de desenvolvimento, análise e teste, finalmente poderemos realizar nossa entrega dos artefatos, que é o outro milestone do nosso ciclo. Na entrega temos todo nosso esforço durante o ciclo em forma de conteúdo já validado pela fase de Análise e Teste, provando que o processo foi aplicado de maneira esperada e tal conteúdo entregue atinge os objetivos previamente definidos.

## 2.10 Ferramentas

Os instrumentos recomendados para execução desse processo são:

- **Kanban:** Utilizar algum software que implementa a ideia de Kanban ou post-its para organização das ideias e ciclos.
- **Versionador de Documentos:** Os artefatos gerados durante o ciclo serão atualizados constantemente, é recomendado o uso de algum versionador de documentos como SharePoint;
- **Versionador de Código:** Todo código-fonte gerado durante o desenvolvimento deve ser versionado para manter um controle de entrega por ciclo, pode ser considerado opções como GIT e SVN.



## 2.11 Feedback

Feedback é um dos pilares mais importantes propostos no processo descrito neste documento, pois ele deverá caminhar em conjunto com os demais componentes do ciclo de vida para que possamos tomar diferentes posturas e propor melhorias ao processo e aos envolvidos. Dentro desse componente, temos duas vertentes principais, os feedbacks internos e externos em perspectiva ao Squad.

- **Feedback Interno:** Através dos indicadores gerados a partir dos artefatos o Agile Coach irá propor diferentes posturas dos membros envolvidos no processo e caso necessário realizar pequenas mudanças no processo para o próximo ciclo;
- **Feedback Externo:** O Line Manager obterá informações através de alinhamentos com os Agile Coach's e acompanhamento do processo e conteúdo desenvolvido dos diferentes Squad's. Com os dados obtidos a partir deste controle o Line Manager irá realizar feedbacks coletivos e individuais visando melhorar o desempenho individual dos envolvidos assim como melhorias no processo de cada Squad baseadas nas suas características e ajudará no desenvolvimento da carreira individual dos membros da equipe.

## 3 Ciclo de vida

Nosso processo foi estruturado de forma que seu ciclo de vida será composto por pequenas iterações que irão englobar todas as fases que serão descritas posteriormente, visando a agilidade nas entregas e melhoria no processo como um todo. Como se trata de um processo ágil, a decisão de que o mesmo seria cíclico foi fundamental, onde teremos constantes revisões nas tarefas ao decorrer de cada ciclo.

O nascimento do nosso processo caminha em conjunto com as primeiras fases definidas, que envolvem tarefas associadas à planejamento e construção do escopo. Essa etapa de nosso ciclo de vida serve de base para todas etapas seguintes, como à cada ciclo teremos um novo planejamento e uma nova definição do conteúdo a ser trabalhado, o acompanhamento da eficiência desse processo se torna muito mais ágil, pois conseguiremos analisar em pequenas etapas se o processo está sendo efetivo, ou seja, se o conteúdo que está sendo entregue é realmente o que havia sido proposto.

O fato de todas nossas fases estarem agrupadas de forma cíclica somado à milestones e entregas bem definidas, fazem com que uma possível manutenção no ciclo de vida traga resultados muito mais rápido, já que no ciclo seguinte haverá todo um pensamento no planejamento e quais pontos podem ser ajustados nas demais fases.

Em certos modelos de processo de software apenas parte do ciclo de vida é cíclica, ou seja, um esforço muito grande pode ser empenhado nas primeiras fases para que só então se inicie uma fase cíclica, como por exemplo, o desenvolvimento. Uma vantagem do processo proposto nesse documento é uma desativação menos impactante (tempo e custos), pois caso seja necessário a interrupção do processo, apenas o ciclo em andamento será prejudicado, pois todas as entregas produzidas de ciclos anteriores terão suas fases completas, desde a concepção até sua entrega.

Os benefícios de uma desativação menos agressiva vão além do tempo poupado e menor prejuízo relacionados aos custos. Em cada ciclo temos documentações relacionadas aos artefatos, versionamentos e informação gerada através dos feedbacks. Ou seja, mesmo

após um acontecimento inesperado ou negativo, os dados coletados de ciclos prévios irão enriquecer de forma positiva a base histórica da empresa que implementa o processo.

## 4 Referências Bibliograficas

Utilize o mendley, o jabref ou diretamente o bibtex para gerenciar suas referências bibliográficas. As referências são criadas automaticamente de acordo com o uso no texto.

Exemplo: Redes de computadores, segundo [?] é considerada..... Já [?] apresenta uma versão...

Analizando os pressupostos de [?] e [?] concluimos que....

## 5 Elementos textuais - Alguns exemplos

Esta seção apresenta exemplos de elementos textuais. **Remova-a da versão final do texto.**

### 5.1 Colocar elementos em itens

Texto antes da lista

- First item in a list
- Second item in a list
- Third item in a list

#### 5.1.1 Uma subseção de terceiro nivel

Exemplo de uma subseção

### 5.2 Tabelas

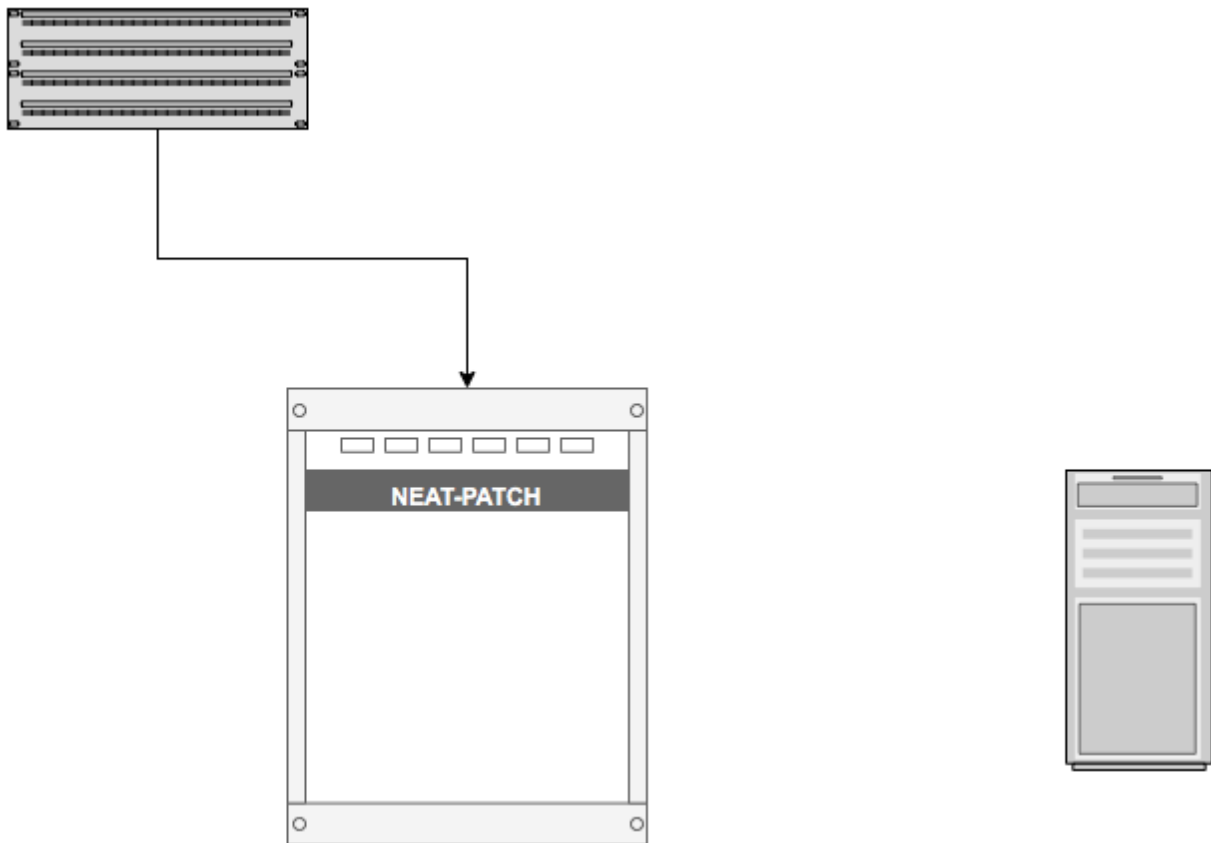
Utilize o site <http://www.tablesgenerator.com/> para elaborar as tabelas de seu trabalho. Para adicionar uma tabela utilize: a tag input, passando o arquivo da tabela como parametro

**Tabela 1:** *Modifique a legenda e crie um label*

Este é um exemplo de tabela	C1		C2	
Você pode criar a tabela no excel	1	2	3	4
Exportar para CSV	5	6	7	8
E importar no Table Generator	9	10		
<i>Gere o tex, e adicione em seu arquivo</i>				

Dentro do arquivo você deve definir o label e pode utilizá-lo para referenciar. Exemplo: Na tab 1 temos a relação de ....

Você também pode modificar a tabela manualmente, incluindo, por exemplo h! dentro de sua definição. Veja no exemplo tab2.tex



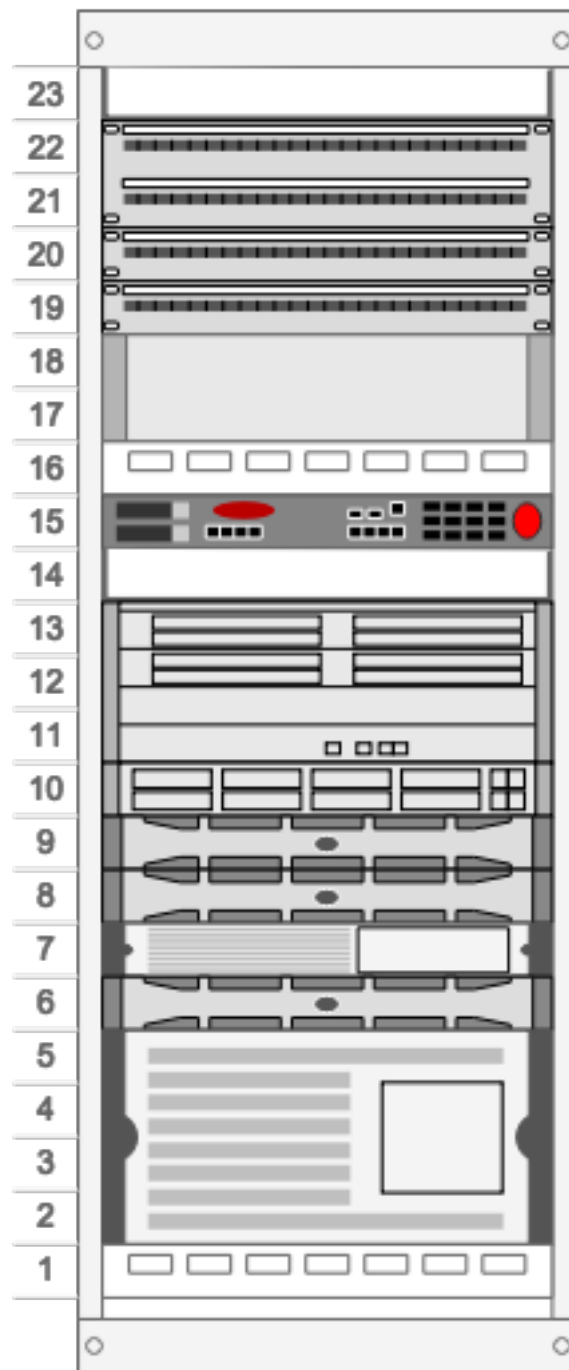
**Figura 1:** *Exemplo de figura com escala horizontal*

## 5.3 Figuras

As figuras podem ser no formato PDF, JPG, PNG. Você pode referenciá-las da mesma maneira que tabelas. Exemplo: A figura 1 apresenta.....

Não se preocupe o local em que a figura será renderizada em seu texto. Preocupe-se em criar referência para ela, ou seja, toda figura e tabela deve conter pelo menos uma referência no texto.

Você pode rotacionar figuras também. Para isso utilize o parâmetro `angle=-90`. Repare que a escala da figura foi modificada pelo parametro `height`. Você também pode utilizar `scale`



**Figura 2:** *Exemplo de figura sem escala*



**Figura 3:** *Exemplo de figura rotacionada*