

---

# Relatório de projeto

***Gabriel Grazionale, Mayara Queiroz, Yago Militão***

*Universidade Tecnológica Federal do Paraná - Câmpus Cornélio Procópio*

---

**P**rocesso de Desenvolvimento detalhado voltado para Software, com o objetivo de instruir um profissional da área nas etapas de desenvolvimento de um sistema.

*7 de novembro de 2018*

## Lista de figuras

1	Template de Test Case [1] . . . . .	6
2	Especificação do Processo com BPMN. . . . .	7
3	. . . . .	8

## Lista de tabelas

1	Refinamento de Requisitos Funcionais . . . . .	5
2	Refinamento de Requisitos Não Funcionais . . . . .	5
3	Requisitos Funcionais aceitos pelo usuário . . . . .	5
4	Requisitos Não Funcionais aceitos pelo usuário . . . . .	5

# Sumário

<b>1</b>	<b>Processo</b>	<b>4</b>
1.1	Artefatos . . . . .	4
1.2	Templates . . . . .	4
1.2.1	Reunião com cliente . . . . .	4
1.2.2	Refinamento de requisitos . . . . .	5
1.2.3	Documento de Requisitos . . . . .	5
1.2.4	Testes unitários e integrados . . . . .	6
1.2.5	Teste de aceitação do usuário . . . . .	6
1.3	Organização e Responsabilidades . . . . .	6
1.4	Especificação do Processo com BPMN . . . . .	7
<b>2</b>	<b>Atualizações do Software e Versionamento</b>	<b>7</b>
2.1	Git e Github . . . . .	8
2.2	Novas funcionalidades . . . . .	8
2.3	Correções de bugs . . . . .	8
2.4	Novos campos no banco de dados . . . . .	9
<b>3</b>	<b>Referências bibliográficas</b>	<b>9</b>

# 1 Processo

O processo que será utilizado é baseado em Cascata. Cascata é um modelo de processo ideal para quando os requisitos de um determinado problema são bem definidos e compreendidos, também conhecido como ciclo de vida clássico ou tradicional [2].

Para o desenvolvimento de software esse modelo sugere uma abordagem sequencial e sistemática, iniciado pelo levantamento de requisitos, que é realizada junto ao cliente. Posteriormente é definido as estimativas do projeto, cronogramas e acompanhamento e por fim a modelagem, desenvolvimento e implantação do software.

## 1.1 Artefatos

Para cada fase do processo de desenvolvimento será gerado um ou mais artefatos de saída, a seguir pode-se analisar as fases do processo proposto, de acordo com a figura 2:

- Levantamento e Análise de Requisitos;
- Implementação;
- Verificação;
- Implantação.

Na fase inicial de Levantamento e Análise de Requisitos serão gerados três artefatos: Uma pauta de Reuniões com os Clientes, onde será levantado todos os requisitos do sistema; um documento com o Refinamento dos Requisitos, onde os requisitos previamente levantados serão otimizados, também junto aos clientes e um Documento Final de Requisitos, que será aprovado previamente pelos clientes.

Durante a fase de Implementação será gerado um único artefato: os Códigos Fonte do sistema. Esse código fonte se trata das linhas de código de desenvolvimento dos requisitos solicitados no documento de requisitos.

A fase de Verificação irá gerar dois artefatos: Os Testes Unitários e Integrados, que serão realizados pelo próprio desenvolvedor do sistema ao concluir o desenvolvimento de uma funcionalidade, e o Teste de Aceitação do Usuário, que será realizado pelo próprio usuário em um ambiente de homologação, para verificar se todas as funcionalidades estão de acordo com o que foi solicitado no documento de requisitos.

Por fim, caso o sistema final seja aprovado em todos os testes o processo passará para a fase de Implantação, onde o sistema será implantado no ambiente de produção.

## 1.2 Templates

### 1.2.1 Reunião com cliente

Seguir as perguntas abaixo:

- 1) Qual é a sua linha de negócio? (Tentar descobrir com o que o cliente trabalha, "especular")
- 2) Atualmente, como você trabalha? (Tentar descobrir a forma que o cliente desempenha suas atividades)
- 3) Quais recursos você utiliza para exercer suas funções/serviços? (Tentar descobrir quais os principais atividades que o cliente realiza)

4) Dentro da sua linha de negócio, o que gostaria de automatizar? (Tentar descobrir as dificuldades que o cliente enfrenta em seu modelo de negócio)

- lista de requisitos funcionais.

Caso necessário, poderá adicionar mais questões ao longo da reunião. Lembre-se: Anote neste template todo o andamento da reunião.

### 1.2.2 Refinamento de requisitos

Lista de requisitos funcionais e não funcionais

**Tabela 1:** *Refinamento de Requisitos Funcionais*

Requisitos Funcionais	Aceito pelo usuário?
Requisito 1	Sim
Requisito 2	Não
Requisito 3	Sim

**Tabela 2:** *Refinamento de Requisitos Não Funcionais*

Requisitos Não Funcionais	Aceito pelo usuário?
Requisito Não Funcional 1	Sim
Requisito Não Funcional 2	Não
Requisito Não Funcional 3	Sim

### 1.2.3 Documento de Requisitos

Lista de requisitos funcionais e não funcionais aceitos pelo usuário

**Tabela 3:** *Requisitos Funcionais aceitos pelo usuário*

Identificador	Requisito Funcional	Descrição	Prioridade
RF01	Nome do Requisito	Descrição do Requisito	Alta
RF02	Nome do Requisito	Descrição do Requisito	Média
RF03	Nome do Requisito	Descrição do Requisito	Baixa

**Tabela 4:** *Requisitos Não Funcionais aceitos pelo usuário*

Identificador	Requisito Não Funcional	Descrição	Prioridade
RNF01	Nome do Requisito	Descrição do Requisito	Alta
RNF02	Nome do Requisito	Descrição do Requisito	Média
RNF03	Nome do Requisito	Descrição do Requisito	Baixa

Eu, (nome do cliente), portador do RG (num), CPF (num), firmo por meio deste que aceito e estou de acordo com todos os requisitos especificados neste documento e portanto os mesmos estão aptos a serem desenvolvidos.

Assinatura do Cliente, Data e Local

### 1.2.4 Testes unitários e integrados

Test case, na Figura 1, usado para testes unitários e integrados, todos os campos devem ser preenchidos.

Test Case ID		Created Date	
Priority		Created By	
Module Name		Requirement#	
Title			
Pre Conditions			
Step#	Test Step	Test Data	Expected Result
Tear Down			

### Execution History

Execution Date	Environment	Tester Name	Status	Defect#

Figura 1: *Template de Test Case [1]*

### 1.2.5 Teste de aceitação do usuário

Nesta etapa o usuário terá a versão de homologação do software para realizar os testes de funcionalidades diretamente no mesmo, caso todas as funcionalidades estejam de acordo com o que o usuário deseja, o termo abaixo será assinado:

Eu, (nome do cliente), portador do RG (num), CPF (num), firmo por meio deste que aceito e estou de acordo com todas as funcionalidades do sistema e portanto o mesmo está apto a ser implantado.

Assinatura do Cliente, Data e Local

## 1.3 Organização e Responsabilidades

Gabriel:

Mayara:

Yago Militão:

## 1.4 Especificação do Processo com BPMN

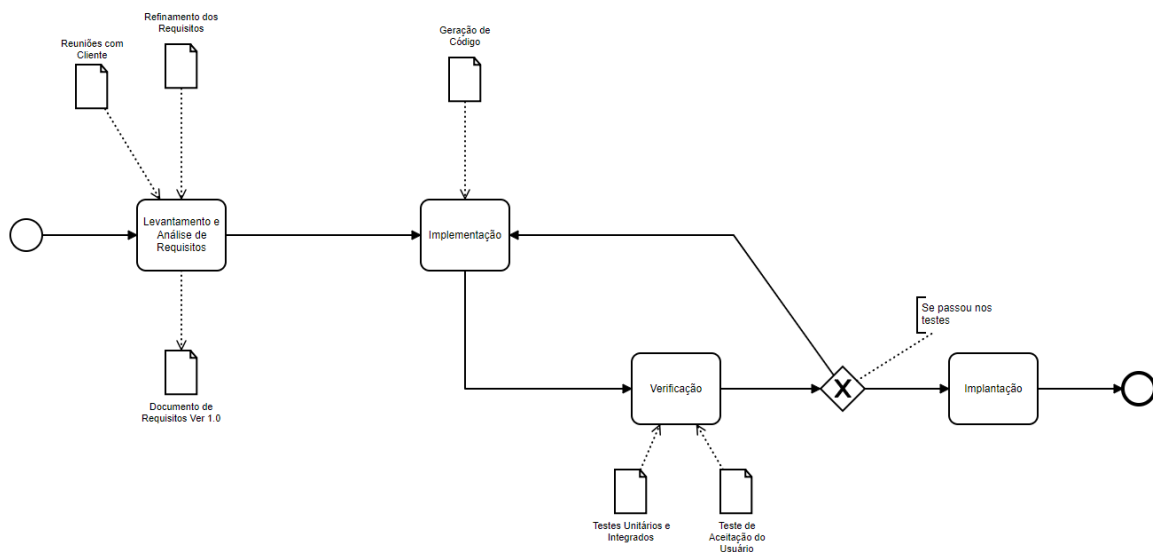
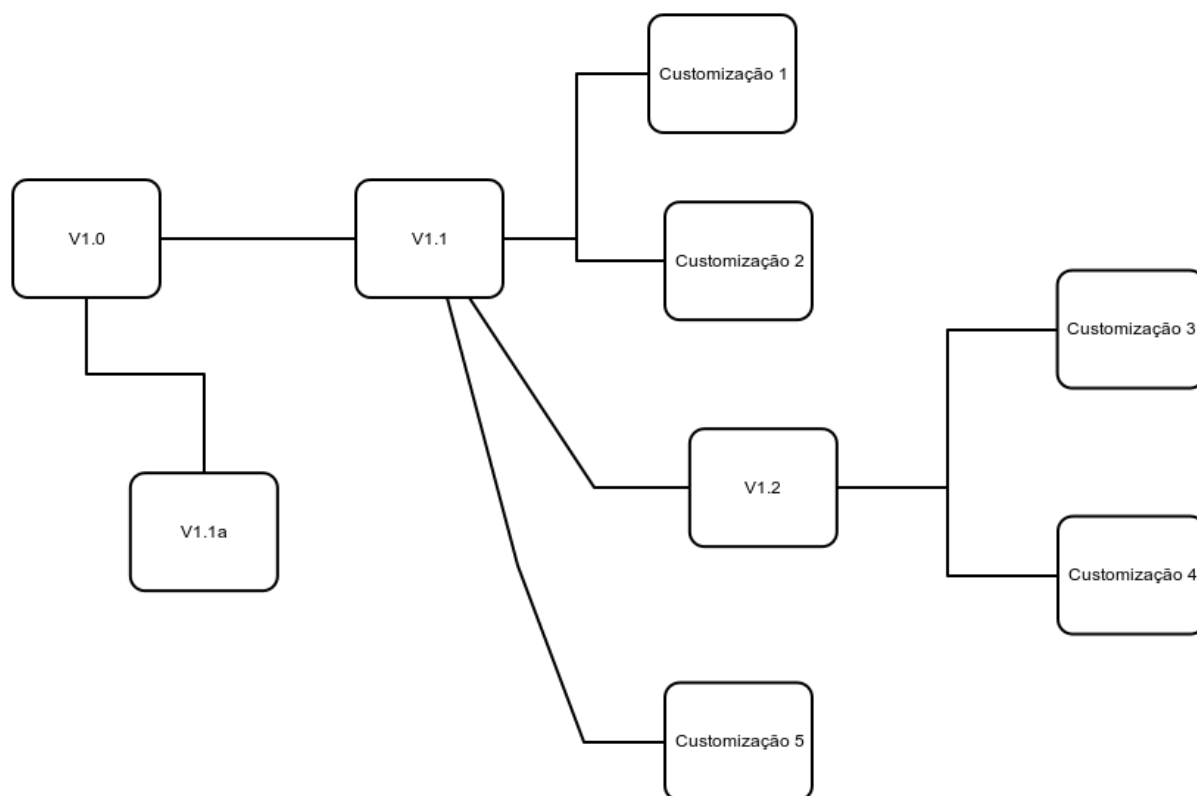


Figura 2: Especificação do Processo com BPMN.

## 2 Atualizações do Software e Versionamento

Um item, ao ser desenvolvido, evolui até que atinja um estado em que atenda aos propósitos para o qual foi criado. Isso implica em diversas alterações, gerando uma nova versão do item a cada estado. Para estabelecer o controle sobre as diversas versões, todas as versões devem ser armazenadas e identificadas. Para isso será usado funcionalidades do Github (repositório central dos ICs) e da ferramenta Git. Como o software Klassic pode ser customizado para determinados clientes, a estrutura de derivação de versões será no seguinte esquema:



**Figura 3:** .

## 2.1 Git e Github

O GitHub é uma plataforma de hospedagem de projetos de software que oferece, além de armazenamento de repositórios GIT e/ou SVN, uma área do repositório específica para documentação online (Portal Wiki) e outra para controlar tarefas e mudanças no projeto (issues). A fim de se ter melhor aproveitamento da plataforma e agilidade. Todos os artefatos de software e o gerenciamento do projeto será feito no GitHub e/ou em ferramentas integradas ao GitHub por meio de cada desenvolvedor.

## 2.2 Novas funcionalidades

Para realizar a implementação de uma nova funcionalidade, deverá ser preenchido um relatório de novos requisitos. Os mesmos passarão pela aprovação do gerente de configuração. Caso aprovado, será designado a um desenvolvedor e o mesmo deverá realizar um clone da aplicação atual em um ambiente local. Por fim, após a implementação da nova funcionalidade, o desenvolvedor deverá realizar os testes e o preenchimento do relatório de testes unitários e de integração.

## 2.3 Correções de bugs

Caso seja um bug reportado pelo cliente ou usuário da aplicação, o gerente de configuração deverá realizar o preenchimento de um relatório onde especifica qual o problema detectado. Posteriormente, esse relatório deve ser enviado a um desenvolvedor para que seja realizado as correções. Por fim, o desenvolvedor deverá realizar o clone da aplicação local, os



testes e o preenchimento do relatório de testes unitários e de integração, assim como no desenvolvimento de uma nova funcionalidade.

Caso o bug seja reportado pela própria equipe de desenvolvimento, o mesmo deve ser informado ao gerente de configuração, para que os passos acima sejam realizados.

## **2.4 Novos campos no banco de dados**

Cada desenvolvedor deverá realizar o backup do banco de dados e da aplicação. Após a realização dos backups, o desenvolvedor deverá rodar a aplicação em ambiente local e realizar as alterações. Após as alterações serem realizadas, as mesmas deverão ser relatadas ao gerente de configurações, este será responsável por aprovar as mudanças e posteriormente realizar o merge de todas as alterações do banco de dados.

## **3 Referências bibliográficas**

[1] Admin, “Test case templates,” 2018.

[2] H. D. Media, “Introdução ao modelo cascata,” 2013.