

Universidade de Brasília - UnB
Faculdade do Gama - FGA



**Processo de Validação do Projeto *Nexite* Através da
Técnica Revisão por Pares**

Djorkaeff Alexandre Vilela Pereira - 160026822

Gabriela Medeiros da Silva - 160121817

Ignacio Sanabria Alonso de Caso - 190055511

João Lucas Sousa Reis - 160009758

Larissa Siqueira Sales - 160130883

Lucas Gomes Silva - 160133505

Luciano dos Santos Silva - 160013321

Brasília - DF

2019

Sumário

1. Introdução	2
1.1 Artefato avaliado	2
1.2 Conceito de Validação	2
2. Metodologia	3
3. Objetivo	4
4. Verificação e Validação de Software	5
4.1 Conceitos	5
4.2 Técnicas de Verificação e Validação	5
5. Revisão	8
5.1 Revisão por Pares	9
5.2 Outros métodos de RTF	9
6. Objeto de análise	10
6.1 Escopo	10
6.2 Público-alvo	10
7. Avaliação	11
8. Conclusão	13
9. Referências bibliográficas	14
APÊNDICE I - Mapa Conceitual	15
APÊNDICE II - Diagrama de Contribuição por Palavras	16
APÊNDICE II - Diagrama de Contribuição por Tópico	17

1. Introdução

Este trabalho irá apresentar conceitos sobre verificação e validação de um software, além do processo de revisão por pares utilizado como uma metodologia na técnica de validação. Para isso, serão fornecidas características e descrição de cada recurso utilizado e estudado e o resultado obtido a partir disso.

Essas informações serão apresentadas como parte do processo de testes relacionados a um projeto de software e o procedimento de sua aplicação.

1.1. Artefato avaliado

Para a avaliação do projeto estudado, foi utilizado o documento de visão do aplicativo Nexte, um dos trabalhos da disciplina de Métodos de Desenvolvimento de Software da Universidade de Brasília, realizado no primeiro semestre de 2018 por uma das integrantes do grupo, enquanto aluna do curso de Engenharia de Software.

O aplicativo citado foi criado com o objetivo de integrar jogadores de tênis amador de uma determinada região, melhorando o engajamento e a competitividade entre eles e expandindo o alcance do esporte.

1.2. Conceito de Validação

Validação de software é um conceito aplicado juntamente com o cliente, tendo em vista que é conceituado como a análise das funcionalidades do produto em relação às expectativas das partes interessadas.

No contexto em que esse projeto foi desenvolvido, não se faz possível o acesso direto ao cliente da aplicação. Para se sanar esse requisito, o grupo decidiu utilizar da alternativa de se portar como um usuário da aplicação, buscando obter um resultado o mais próximo possível da realidade dos usuários.

2. Metodologia

Para efetuar a validação, os desenvolvedores apresentaram o produto à pessoa que fez o papel de usuário, mostrando todas as funcionalidades implementadas e elaborando um questionário acerca do que o usuário espera da aplicação.

O resultado do questionário, junto com um processo de introspecção no sistema, serviram com um *feedback* da aplicação, aprovando ou não as funcionalidades que foram entregues.

Cada etapa de desenvolvimento passou por uma inspeção do tipo revisão por pares, desde o documento de visão até o desenvolvimento do código. Como o produto já foi desenvolvido e entregue, o resultado será aplicado apenas na avaliação do produto completo, já que não é possível modificar o código ou a documentação.

3. Objetivo

O objetivo da validação é garantir se as funcionalidades presentes no software estejam de acordo com as necessidades descritas pelo cliente, logo o cliente é o validador final da funcionalidade. Se usada a validação como uma ferramenta de garantia da funcionalidade, é possível validar se a funcionalidade está se comportando de forma desejada.

4. Verificação e Validação de Software

4.1. Conceitos

Verificação e validação são duas formas de se avaliar um produto com requisitos de aprovação diferentes. Enquanto a verificação busca avaliar se uma determinada atividade ou funcionalidade foi desenvolvida corretamente, a validação é foca em analisar se essa funcionalidade está de acordo com a expectativa do cliente.

“Verificação se refere a um conjunto de atividades que garantem que o software implementa corretamente uma função específica e a Validação se refere a um conjunto de atividades diferentes que garante que o software construído corresponde aos requisitos do cliente”

(Pressman, 2006 apud Ferreira, 2008, p. 49).

Quanto a metodologia para se executar uma validação, Pressman (2006) diz que “[...] há uma forte divergência de opinião sobre que tipos de teste constituem validação. Algumas pessoas acreditam que todo teste é verificação e que validação é conduzida quando requisitos são revisados e aprovados, e posteriormente, pelo usuário, quando o sistema estiver operacional. Outras pessoas consideram teste unitário e de integração como verificação e testes de alta ordem como validação”.

4.2. Técnicas de Verificação e Validação

As técnicas de Verificação e Validação são utilizadas a fim de estimar a qualidade de um software de maneira padronizada, além de buscar erros e incoerências no projeto, evitando que seja desenvolvida uma reação em cadeia em todo o projeto a partir do momento de sua implementação.

Apesar das divergências de opiniões sobre os conceitos de Verificação e Validação, há uma concordância quanto às técnicas utilizadas para testá-los. A tabela a seguir mostra as técnicas de Validação e Verificação e suas respectivas classificações.

Tabela 4.1 – Taxonomia de Técnicas de Verificação & Validação (Fonte: Ferreira, 2008 apud Balci, 1995).

Técnicas de Validação e Verificação	
Classificação	Técnicas
Informal	Auditoria, Validação de Face, Inspeções, Revisões, Teste de Turing, Walkthroughs
Estática	Checagem de Inconsistência, Análise de Fluxo de Dados, Análise Baseada em Grafos, Análise Semântica, Análise Estrutural, Análise Sintática
Dinâmica	Teste de Caixa Branca, Teste Button-Up, Monitoração de Execução, Execução de Perfil, Execução de Traço, Testes de Campo, Comparação Gráfica, Validação Preditiva, Teste de Regressão, Análise Sensitiva, Técnicas Estatísticas, Teste de estresse, Testes de Sub-Modelo, Debug Simbólico, Teste Top-down, Visualização, Teste de Caixa Branca
Simbólica	Gráfico de Causa Efeito, Análise de Partição, Análise de Caminhos, Execução Simbólica
Restrições	Checagem de Suposições, Análise de Limites, Suposição Indutiva
Formal	Indução, Inferência, Dedução Lógica, Cálculos Preditivos, Transformação Preditiva, Prova de Exatidão

Técnicas informais são técnicas que não se baseiam em números, cálculos ou contas para se determinar o resultado, mas sim na subjetividade e no raciocínio de quem efetua a avaliação do Software (Ferreira, 2008).

As técnicas estáticas são testes aplicados ao código-fonte do programa, sem que seja necessária a execução do código, diferente das técnicas dinâmicas, onde a avaliação se dá após a execução do código, através da análise da saída do programa (Ferreira, 2008).

Assim como as técnicas dinâmicas, as técnicas simbólicas avaliam o programa dinamicamente. Nela, são fornecidos dados simbólicos (exemplos), e examinadas as modificações desses dados ao longo da execução do programa

(Ferreira, 2008).

Por último, as técnicas formais utilizam de fórmulas e valores matemáticos e exatos para a análise da qualidade do software (Ferreira, 2008). Apesar de parecer o melhor método para muitos, as técnicas formais costumam ser mais caras e com metas mais difíceis de serem alcançadas.

A técnica utilizada neste trabalho, com o fim de avaliar o projeto já citado, será uma técnica informal de Revisão nomeada de *Peer-Review* (Revisão por Pares).

5. Revisão

As revisões são algumas das muitas práticas que possuem o objetivo de assegurar a qualidade de um software ao longo de seu ciclo de vida.

Um processo de revisão busca encontrar falhas em cada etapa do processo de desenvolvimento de um software, evitando que erros permaneçam de uma etapa para outra.

“ Revisões fazem parte do conjunto de atividades de garantia de qualidade de software. Essas atividades constituem um padrão sistemático e planejado de ações que são exigidas para garantia de qualidade do software e que devem ser aplicadas ao longo de todo processo de engenharia de software. “

(Melo, apud Pressman, 2000)

As técnicas de revisão podem ser classificadas em Discussão Informal, Apresentação e Revisões Técnicas Formais. As técnicas categorizadas como de Discussão Informal possuem o objetivo de solucionar erros e problemas técnicos e são executadas pelo grupo de desenvolvedores. A apresentação constitui uma exposição do projeto por parte do autor para os interessados, como o pessoal técnico, os administradores e para o cliente. E as Revisões Técnicas Formais são avaliações de base técnica com pequenos grupos que fornecem informações confiáveis sobre o desenvolvimento do projeto. (Melo)

De acordo com Melo, as Revisões Técnicas Formais (RTF) têm como principais objetivos descobrir erros de função, lógica ou implementação, em qualquer produto de software, verificar se o software que se encontra sob revisão atende a seus requisitos, garantir que o software tenha sido representado de acordo com padrões predefinidos, obter um software que seja desenvolvido uniformemente e tornar os projetos mais administráveis.

Nesse trabalho utilizaremos um método classificado como uma Revisão Técnica Formal nomeado *Peer-Review* (Revisão por Pares).

5.1. Revisão por Pares

A revisão por pares é uma técnica de revisão onde um grupo se segmenta em duplas que possuem conhecimentos equiparáveis, com objetivo de obter diferentes resultados através dos diferentes pontos de vista. Após isso, todas as duplas devem revisar o projeto até a fase corrente em busca de erros e problemas não vistos anteriormente. Os erros encontrados deverão ser dispostos em um relatório, que, se considerado pertinente, será formalizado e inserido no relatório final (Melo).

É importante ressaltar que a revisão deve ser feita em pequenas partes do código ou funcionalidade por ciclo de revisão.

5.2. Outros métodos de RTF

Além da Revisão por Pares, a Revisão Técnica Formal possui outros dois principais métodos de revisão: *Walkthrough* e Inspeção

O *Walkthrough* é uma técnica na qual um grupo pequeno, de 3 (três) a 6 (seis) pessoas, executam um código passo-a-passo no papel (Melo). Cada revisor deverá percorrer o código e executar a técnica, sendo que, previamente, deverá ser feita uma reunião e estabelecidos os casos de testes utilizados.

A Inspeção, considerada uma das mais relevantes para a avaliação da qualidade de software de um produto de software, é um processo no qual se busca erros no software através da análise de seus documentos, possibilitando o diagnóstico prévio desses problemas. A Inspeção possui seis fases: Planejamento, Apresentação, Preparação, Reunião de Inspeção, Retrabalho e Acompanhamento.

6. Objeto de análise

6.1. Escopo

O projeto apresenta formas de auxiliar a interação entre os jogadores de tênis através do gerenciamento de rankings e partidas.

6.2. Público-alvo

O aplicativo Nexte pretende atingir as organizações amadoras de tênis, assim como seus sócios. Atualmente, no Brasil, existem cerca de [300 clubes](#) que poderiam se beneficiar do aplicativo.

7. Avaliação

O projeto tem como principais usuários jogadores de tênis que se localizam principalmente na região sul do Brasil, contudo, o grupo buscou validar o documento com um jogador de tênis amador de Brasília. A avaliação também foi feita com um dos desenvolvedores do projeto.

Através de um Checklist realizado no Planilhas da *Google*, foi feito uma revisão em pares através do documento de visão pelos itens de Capacidades, Recursos e Restrições do Aplicativo de acordo com o Documento de Visão do grupo.

Foram feitos duas revisões em pares do projeto. O aluno João Lucas fez a revisão em par com o desenvolvedor do Nexte Alexandre Miguel. Larissa Sales fez o checklist com o jogador amador Guilherme Baldissera.

Com as validações realizadas com ambos entrevistados, foi feito um checklist contendo ambos resultados.

Através das validações, a maioria dos ideais propostos pelo documento de visão do Nexte descrito como capacidade do aplicativo foi devidamente validado com a versão final do software.

Foi adicionada uma observação ao item Gerenciamento de Clubes, explicando uma restrição da aplicação, mas que era parte dos requisitos elicitados no desenvolvimento do aplicativo.

Tabela 1: Tabela de Validação feita com os desenvolvedores.

Validação	Capacidades do Nexte	Descrição	Observações
<input checked="" type="checkbox"/>	Gerenciamento de clubes	Engajamento dos clubes já existentes de tênis, permitindo cadastro de seus membros.	Usuários só podem se cadastrar através de um clube, não podendo efetuar cadastro por conta própria
<input checked="" type="checkbox"/>	Sistema de rankings	Colocação dos membros do clube com base em seu desempenho nas partidas.	
<input checked="" type="checkbox"/>	Gerenciamento de partidas	Marcação da partida por meio de desafios com regras pré-definidas e controle de resultados para o sistema de rankings	
<input checked="" type="checkbox"/>	Sistema de gamificação	Incentivo à competitividade sadia entre jogadores do clube	

Tabela 2: Tabela de Validação feita com os desenvolvedores.

Validação	Recursos do Nexte	Descrição
<input type="checkbox"/>	Sistema de autenticação de clubes	O aplicativo fornece um sistema de gerenciamento de clubes, permitindo cadastramento, atualização, login ou exclusão da conta.
<input checked="" type="checkbox"/>	Sistema de rankings	O aplicativo segue regras pré-definidas para classificar o usuário
<input checked="" type="checkbox"/>	Sistema de desafios	O aplicativo torna possível o usuário desafiar outros jogadores seguindo regras pré-definidas.
<input checked="" type="checkbox"/>	Gerenciamento de partidas	O aplicativo gerencia as partidas realizadas, desde o desafio até a coleta do resultado.

Tabela 3: Tabela de Validação feita com os desenvolvedores.

Validação	Restrições do Nexte	Descrição
✓	Restrições de design	O design do aplicativo é limpo e de uso "user-friendly", de modo a tornar a navegação o mais natural possível
✓	Restrições de plataforma	O aplicativo foi desenvolvido para smartphones com sistema android
✓	Restrições de implementação	Utilizado no desenvolvimento a linguagem Kotlin.
✓	Restrições de uso	Para a utilização do aplicativo é necessário que o usuário possua um dispositivo android com conexão de internet.

8. Conclusão

A documentação é uma forma de garantir à equipe que haja todo alinhamento e todos os envolvidos possam ter o mesmo entendimento do que está sendo executado em todo o desenvolvimento do software.

Este trabalho só foi possível ser executado porque a aplicação escolhida para ser revisada e validada tinha toda a documentação disponível para ser avaliada. Um produto de software que não tem atualização em sua documentação é passível de falhas, pois dificulta a elaboração dos testes e de sua manutenibilidade.

Um dos pontos que trouxe maiores impedimentos foi não poder contatar o usuário real do aplicativo. Foi decidido então buscar o cliente e elaborar o *checklist* avaliando as capacidades, os recursos e as restrições do aplicativo de acordo com o Documento de Visão.

A técnica utilizada para estimar o grau de harmonia entre as funcionalidades e a aspiração do cliente foi a *checklist*, onde foram marcados os requisitos que foram aplicados ao projeto.

Considerando o alto nível de concordância entre a aplicação e a expectativa do cliente, o aplicativo foi considerado validado pela equipe.

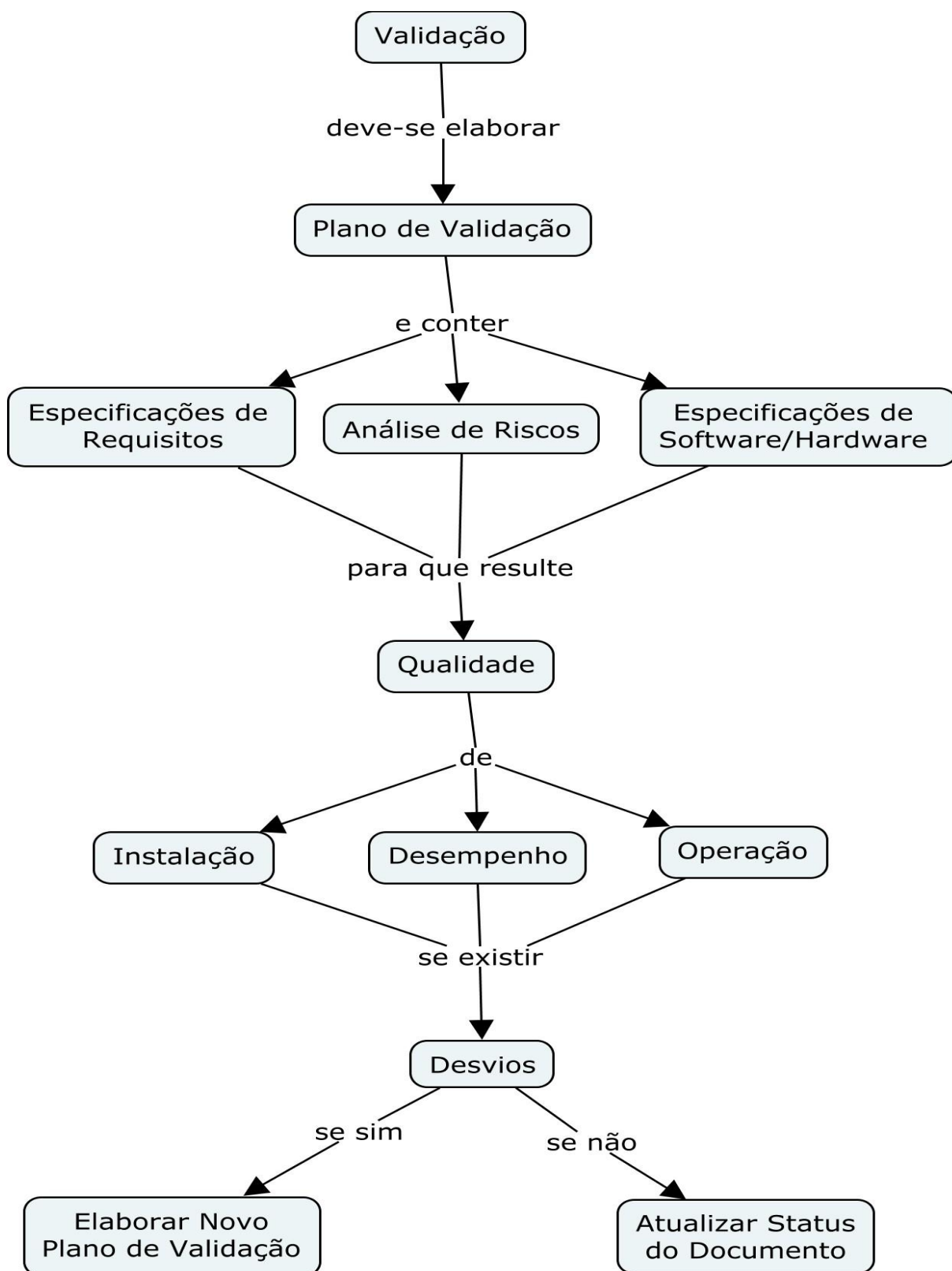
9. Referências bibliográficas

FERREIRA, Bruno. *UMA TÉCNICA PARA VALIDAÇÃO DE PROCESSOS DE DESENVOLVIMENTO DE SOFTWARE*. 2008. 148F. *Dissertação de Mestrado* – Centro Federal de Educação Tecnológica de Minas Gerais, Minas Gerais, 2008.

MELO, Silvana. *INSPEÇÃO DE SOFTWARE*. 10F. Universidade de São Paulo, São Paulo.

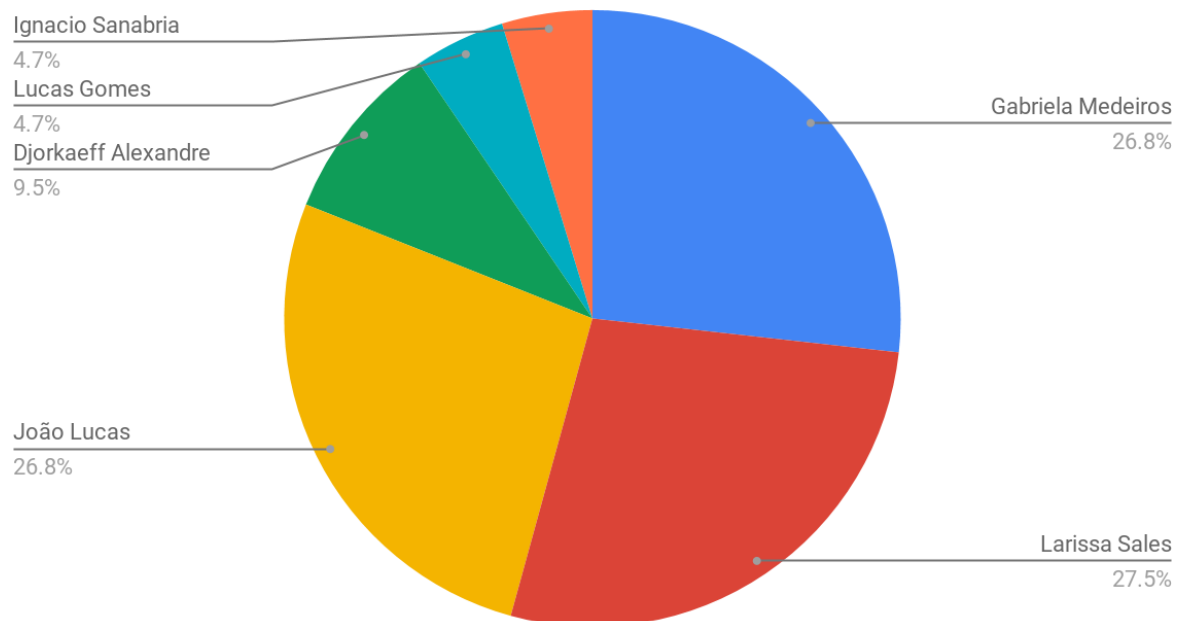
ALBINO, G. *et al.* *GITHUB: DOCUMENTO DE VISÃO DO PROJETO 2018.1.NEXTE DE MDS/EPS*. Disponível em: <https://github.com/fga-eps-mds/2018.1_Nexte/blob/dev/docs/MDS/DocVisao.md>. Acesso 14 abr. 2019.

APÊNDICE I - Mapa Conceitual



APÊNDICE II - Diagrama de Contribuição

Contribuição Individual



APÊNDICE III - Diagrama de Contribuição por Tópico

Tópico	Autor
1 - Introdução	Larissa Sales
2 - Metodologia	Larissa Sales
3 - Objetivo	Lucas Gomes
4 - Verificação e Validação	Gabriela Medeiros
5 - Revisão	Gabriela Medeiros
6 - Objeto de Análise	Djorkaeff Alexandre
7 - Avaliação	João Lucas e Larissa Sales
8 - Conclusão	Larissa Sales, João Lucas e Gabriela Medeiros
Mapa Conceitual	Djorkaeff Alexandre, Ignacio Sanabria e João Lucas