**DevWorthy: Soluções para uma fábrica de software**

# Andrey Vieira Lopes,Breno Henrique Martins Xavier, Eduardo Araujo Junqueira, Victor Augusto dos Santos;

1Instituto de Informática e Ciências Exatas– Pontifícia Universidade de Minas Gerais (PUC MINAS)

Belo Horizonte – MG – Brasil

[andrey.lopes.1247463@sga.pucminas.br](mailto:andrey.lopes.1247463@sga.pucminas.br)   
[brenoxavier8.bx@gmail.com](mailto:brenoxavier8.bx@gmail.com)   
[eajunqueira@sga.pucminas.br](mailto:eajunqueira@sga.pucminas.br) [victor.santos.892238@sga.pucminas.br](mailto:victor.santos.892238@sga.pucminas.br)

***Resumo.*** *Esse trabalho tem o objetivo de automatizar os processos de uma fábrica de software para que isto aconteça o grupo utilizara de técnicas de modelagem e tecnologias de programação aprendidas no curso de engenharia de software para construir essa automação. Com o objetivo de ajudar e atender os envolvidos na operação da fábrica de software.*

# Introdução

O seguinte trabalho tem como objetivo criar uma solução que contempla as técnicas e tecnologias apreendidas no primeiro e segundo semestre do curso de engenharia de software para automatizar os processos de uma fábrica de software. Para o entendimento e automatização desses processos utilizaremos as técnicas da matéria de Modelagem de Processos de Negócios e as tecnologias de Programação Modular e Desenvolvimento de Interfaces Web.

A automatização desses processos numa fábrica de software tem como objetivo melhorar e torná-los mais eficientes para que uma empresa de fabricação de software possa se concentrar na sua principal tarefa.

Um dos principais problemas de uma fábrica de software é a parte administrativa do negócio pois os profissionais da área são formados para ser desenvolvedores de software nem sempre tendo conhecimento para administrar um negócio de software por isso que a automatização dos processos que envolvem a parte administrativa do negócio podem ser de grande ajuda numa empresa desse tipo.

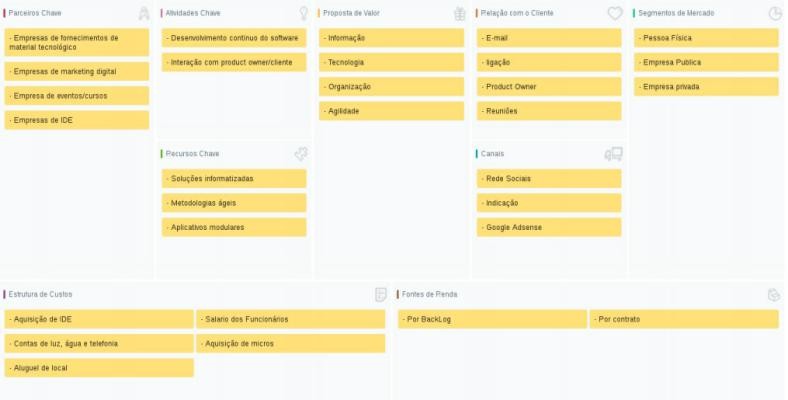
O objetivo geral deste trabalho é entender do negócio de fabricação de software para propor uma automatização do mesmo.

Os objetivos específicos são fazer automatizações da parte administrativa do desenvolvimento de um software e sua parte de elaboração.

Os motivos para fazer esse trabalho são tornar uma fábrica de software mais eficiente e eficaz para que ela seja bem competitiva onde atua.

# Modelo de negócios

**Figura 1: Modelo de negócio de uma fábrica de software feita na plataforma** [**www.sebraecanvas.com**](http://www.sebraecanvas.com/)



**Fonte: Elaborado pelo autor.**

# Participantes do processo

Os stakeholders de uma fábrica de software para automatizar outras fábricas são outras empresas de tecnologia que fornecem a tecnologia para a fábrica poder produzir, outras empresas de tecnologia que necessitam do serviço e do produto que a fábrica presta e oferta, as instituições que querem começar a automatizar seus processos para oferecerem melhores funções na comunidade onde estão inseridas.

Todos esses stakeholders possuem características que os fazem ter um entendimento da importância da tecnologia e seus benefícios tais como alto nível de educação, cultura aberta a novas mudanças de paradigmas e de diversas idades fazem parte das suas características.

# 4. Modelagem do processo de negócio

## 4.1. Análise da situação atual

Os problemas encontrados na fábrica de software que viabilizam a proposta de automatizar seus processos foram de ordem gerencial e técnica. No caso da parte gerencial os problemas encontrados foram a criação de projetos, o gerenciamento destes e suas atribuições de responsabilidades. Já na parte técnica a criação e gerenciamento das tarefas para resolver os problemas do projeto e a sua atribuição para saber quem é responsável pela mesma, bem como alteração das tarefas e sua correção.

**4.2. Descrição Geral da proposta**

A proposta encontrada para resolver os problemas da fábrica de software foi criar um software de gerenciamento de projetos para atender as necessidades da fábrica. Desta forma a proposta foi planejada para que tenha funcionalidades que automatizem e resolvam os processos modelados assim estando alinhado com os interesses do negócio. Os limites que surgem com a solução são o quanto a solução se encaixa nas necessidades da fábrica pois a abrangência e complexidade dos processos impõe dificuldades de encontrar uma resposta correta.

## 4.3. Modelagem dos processos

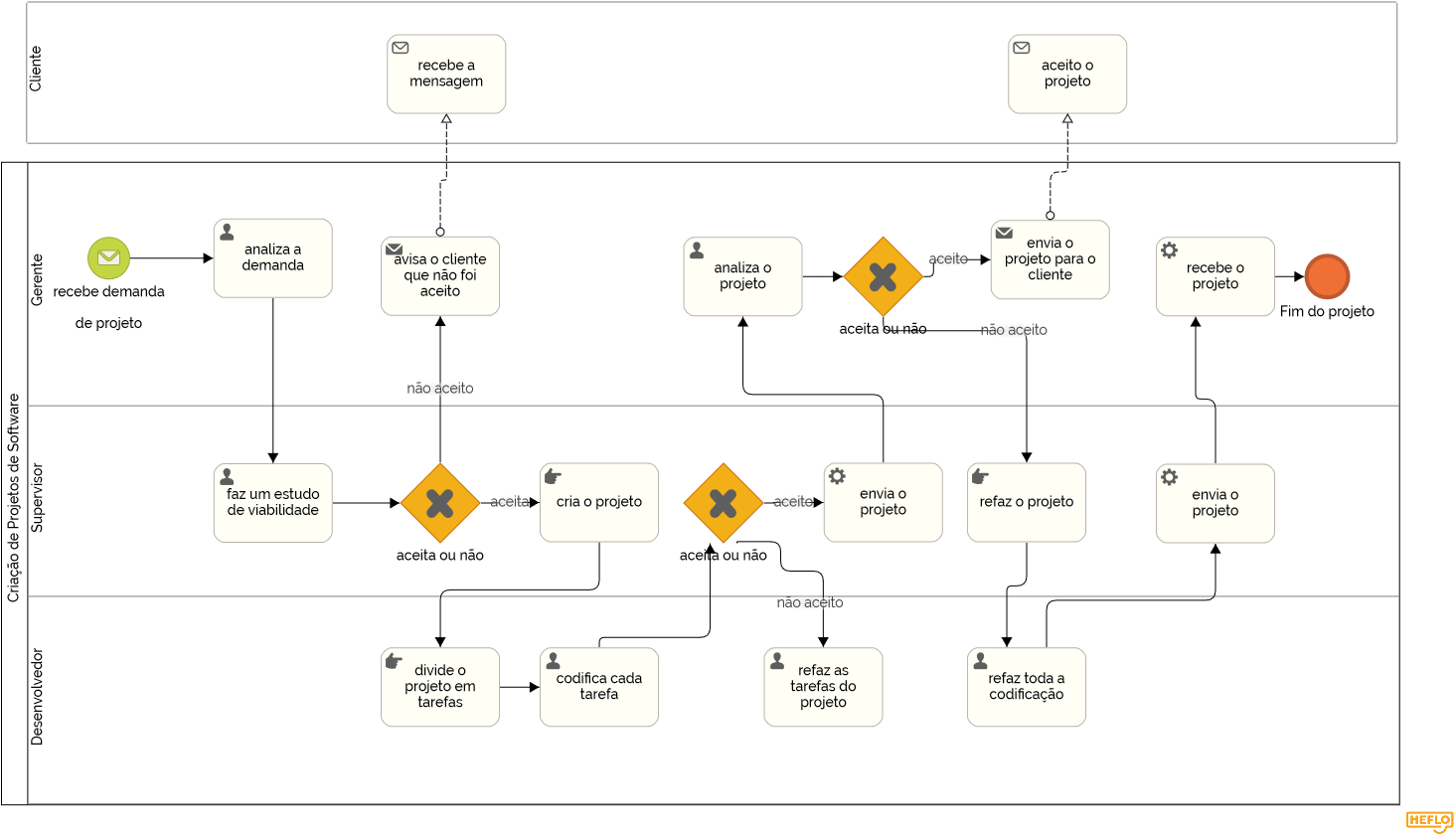
As melhorias que podem ser feitas nos processos são maior detalhamento, simplificações quando houver necessidade e mudanças para que facilite a sua automação. Tudo isso para que a modelagem dos processos de negócios reflita melhor a realidade e possa ser transformada em um software.

## 4.3.1 Processo 1 – Criação de projeto de software

## 

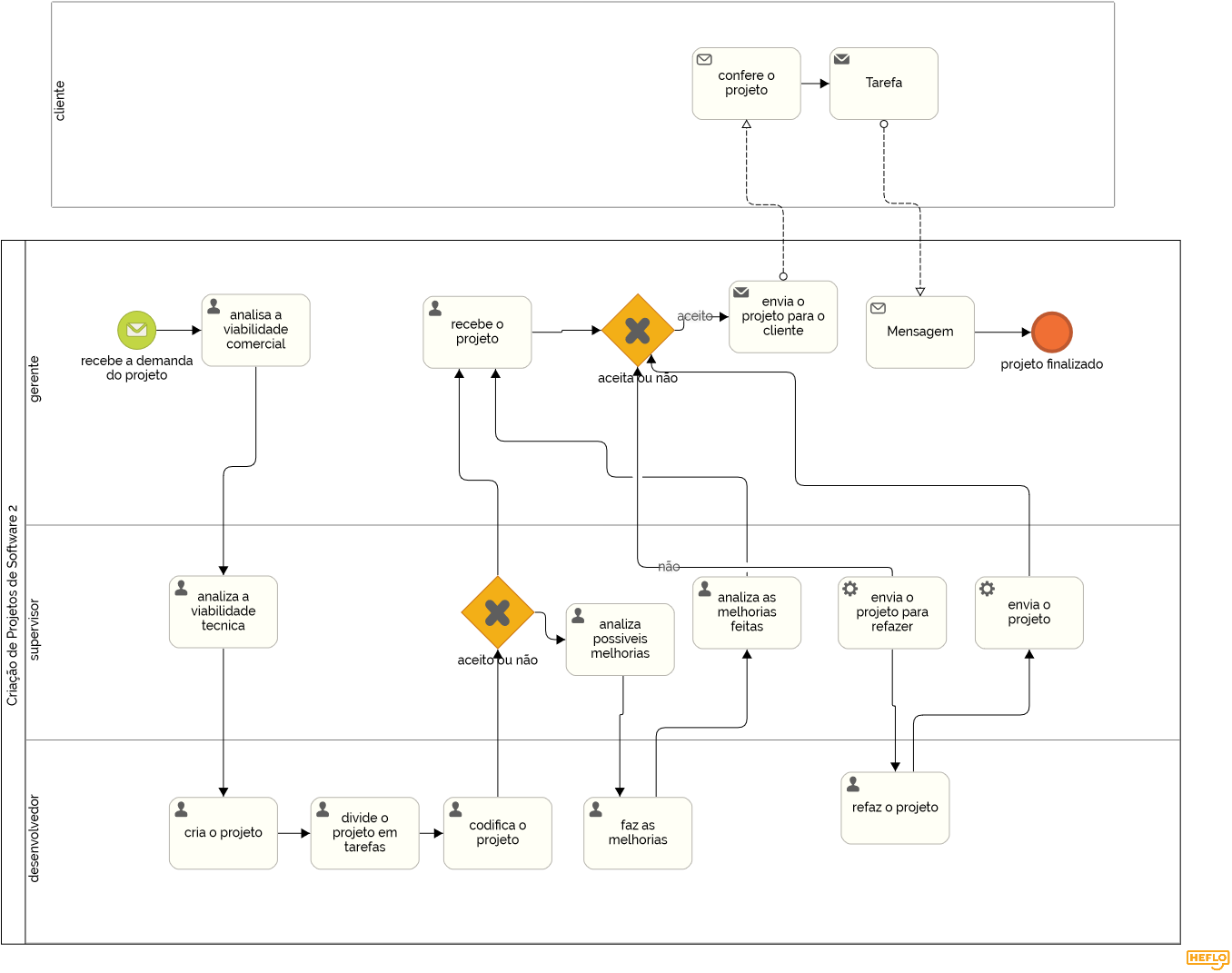
O seguinte processo demonstra a criação de um projeto de software para que possa ser executado.

**Figura 2: Modelagem do processo de criação de projeto de software AS IS utilizando HEFLO**

**Fonte: Elaborado pelo autor.**

**4.3.2 Processo 1 – modelo TO BE**

**Figura 3: Modelagem do processo de criação de projeto de software TO BE utilizando HEFLO**



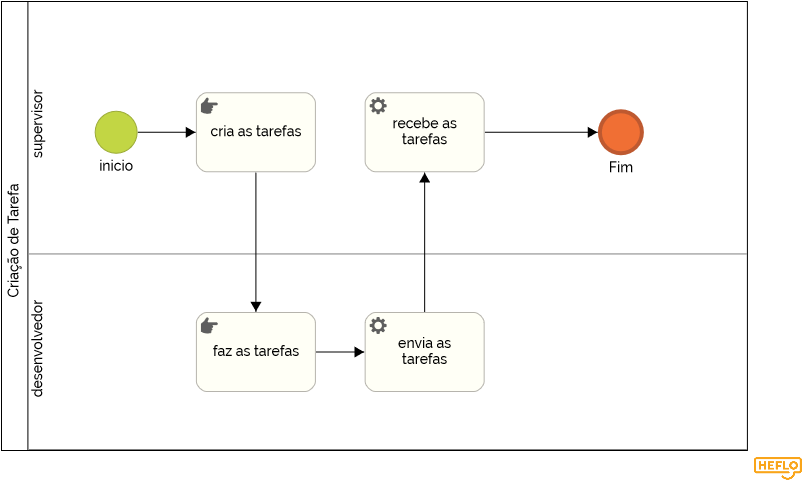
**Fonte: Elaborado pelo autor.**

## 4.3.3 Processo 2 – Criação de tarefas

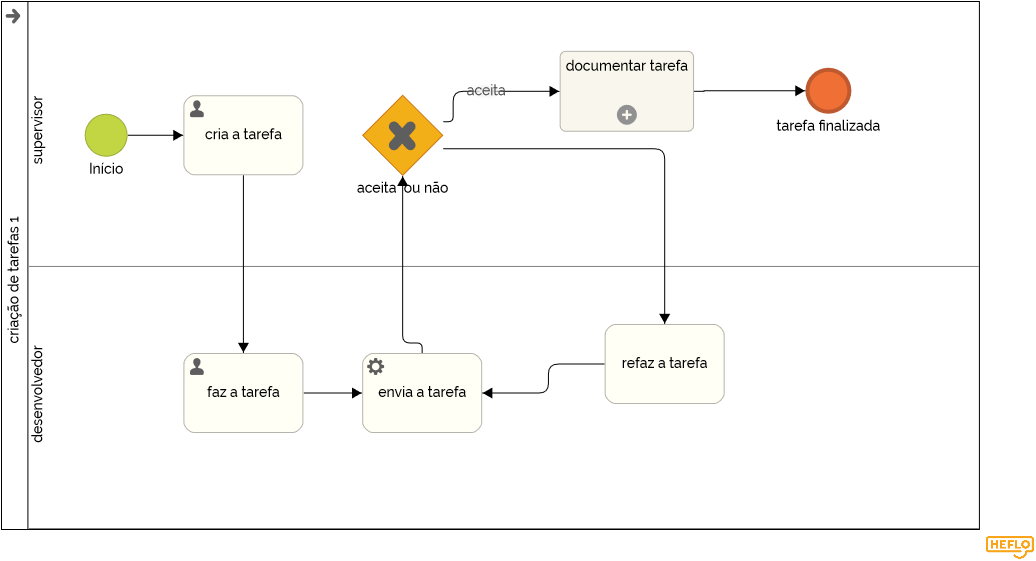
## 

## O seguinte processo de software mostra a criação de tarefas.

**Figura 4: Modelagem do processo de criação de tarefas AS IS utilizando HEFLO**

  
**Fonte: Elaborado pelo autor.**

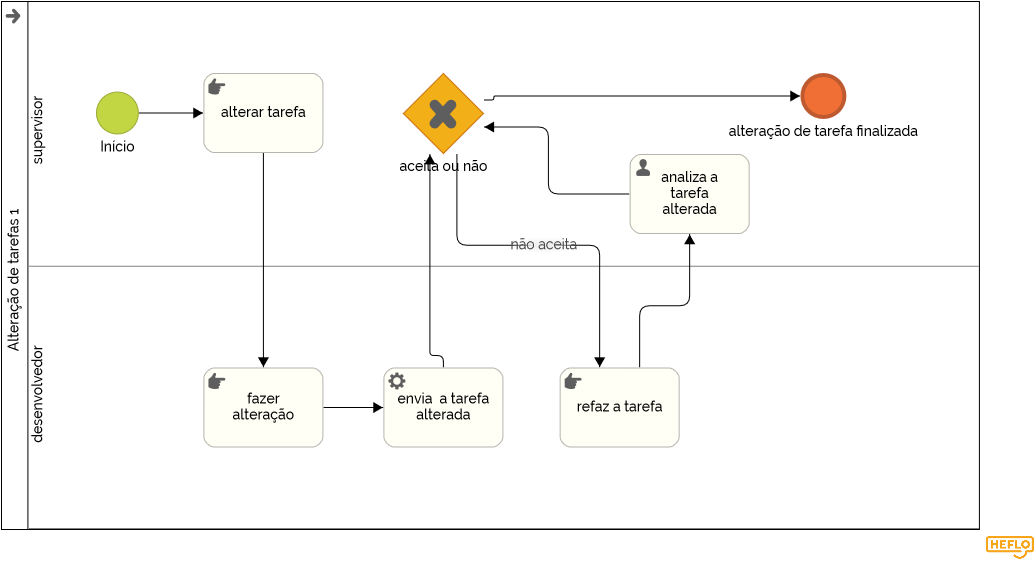
**4.3.4 Processo 2 – modelo TO BE**

**Figura 5: Modelagem do processo de criação de tarefas TO BE utilizando HEFLO**

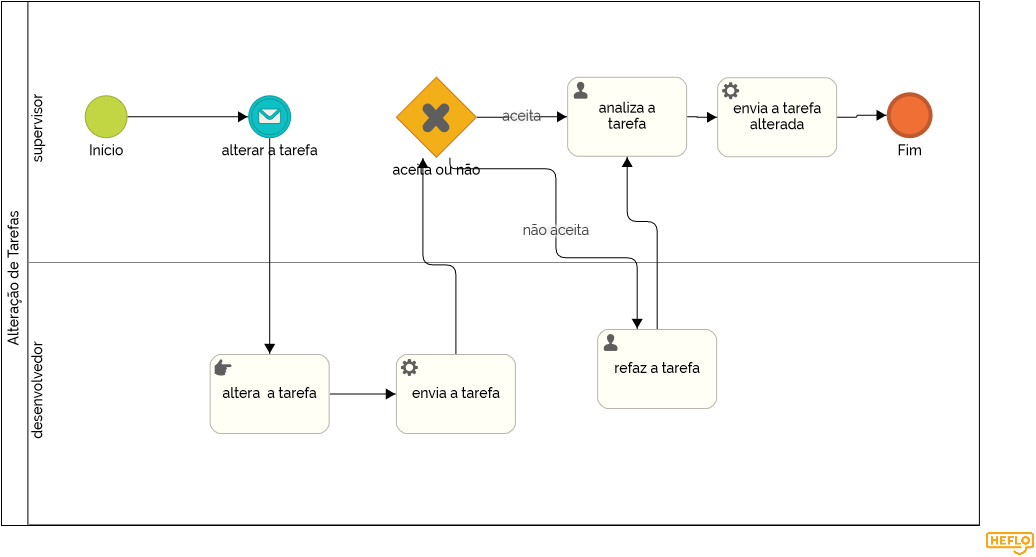
**Fonte: Elaborado pelo autor.**

**4.3.4 Processo 3 – Alteração de tarefa**

O seguinte processo mostra a alteração de tarefas.

**Figura 6: Modelagem do processo de alteração de tarefa AS IS utilizando HEFLO****Fonte: Elaborado pelo autor.**

**4.3.5 Processo 3 – modelo TO BE**

**Figura 7: Modelagem do processo de alteração de tarefa TO BE utilizando HEFLO** **Fonte: Elaborado pelo autor.**

## 4.4. Indicadores de desempenho

Segue abaixo propostas de métricas que serão utilizadas para medir a eficiência do projeto proposto.

**Tabela 1: Indicadores de desempenho DevWorthy**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Indicador** | **Objetivos** | **Descrição** | **Cálculo** | **Fonte dados** | **Perspectiva** |
| Porcentagem de tarefas reabertas | Avaliar quantitativamente a quantidade de retrabalho. | Percentual de tarefas reabertas em relação ao total de tarefas fechadas. | Tarefas reabertas / (tarefas fechadas \* 100) | Tabelas tarefas e tarefas\_status. | Processos internos |
| Média de tempo por tarefa | Avaliar melhor a velocidade de trabalho da equipe. | Tempo médio de uma tarefa desde sua criação até quando foi finalizada. | Somatória de data final – data inicial divido pelo número de tarefas | Tabelas tarefas e tarefas\_status. | Processos internos |
| Média de horas de trabalho por tarefa | Avaliar melhor a velocidade de trabalho da equipe. | Tempo médio de horas usadas numa tarefa desde sua criação até quando foi finalizada. | Somatória da quantidade de horas gastas nas tarefas divido pelo total de tarefas. | Tabelas tarefas, tarefas\_status e tabela tarefas\_horas. | Processos internos |
| Porcentagem de tarefas recusadas | Avaliar a taxa de aprovação de tarefas. | Percentual de tarefas que foram recusadas. | Quantidade de tarefas recusadas / total de tarefas | Tabelas tarefas e tarefas\_status. | Processos internos |
| Porcentagem de tarefas que voltou para o cliente | Avaliar o entendimento inicial da descrição das tarefas. | Percentual de tarefas que após aprovação retornou para o cliente com questionamentos. | Quantidade de tarefas retornadas / total de tarefas | Tabelas tarefas e tarefas\_status. | Processos internos |

**Fonte: Elaborado pelo autor.**

**5****. Projeto da Solução**

## 5.1. Requisitos funcionais

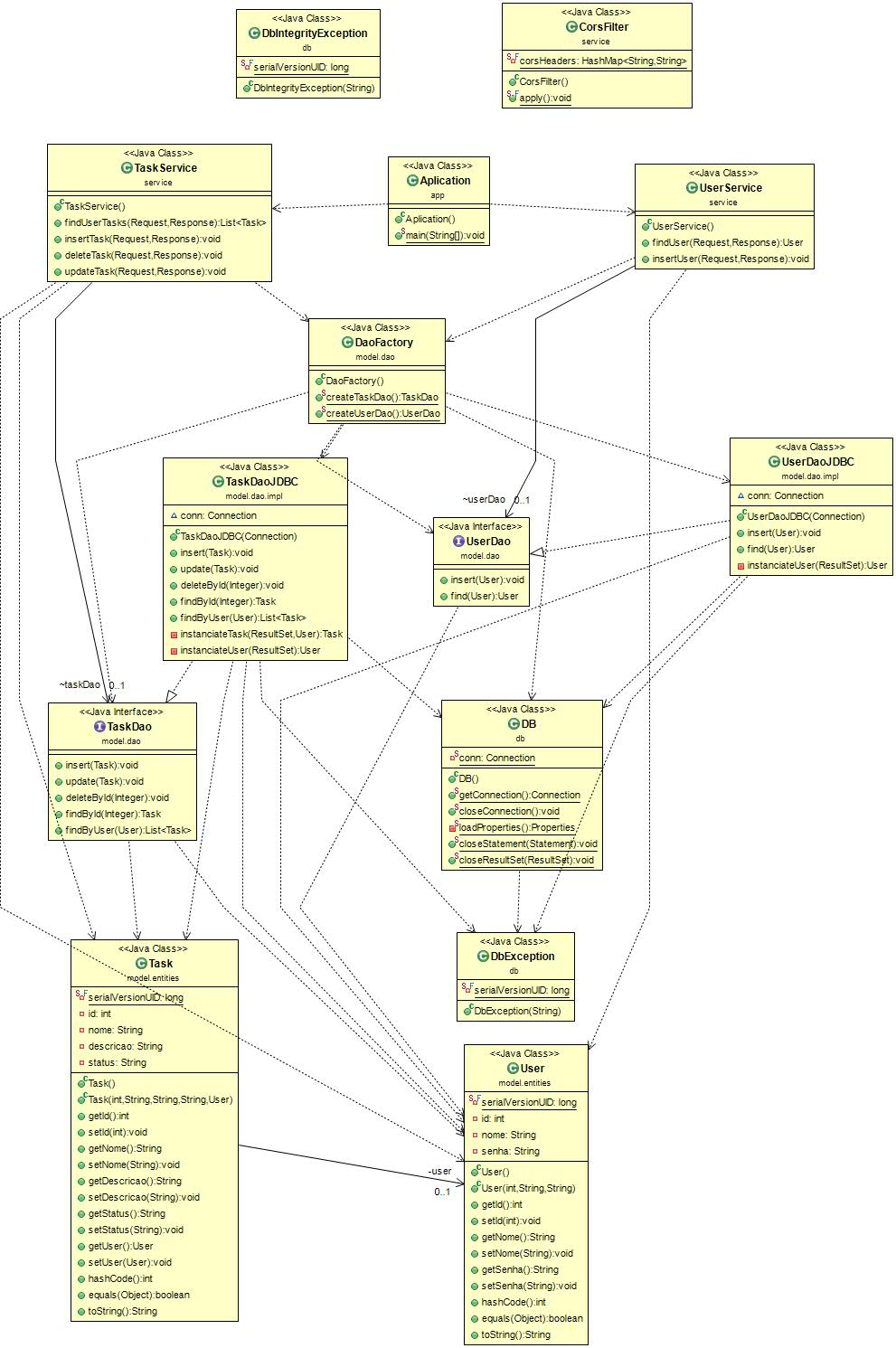
**Tabela 2: Requisitos Funcionais da DevWorthy**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Processo/tarefa** | **Descrição** | **Prioridade** |
| **RF001** | Identificação de usuário | O usuário faz um cadastro e em seguida efetua o login para entrar no seu perfil em que tem os projetos e tarefas. | Alta |
| **RF002** | Criação de projetos | O usuário cria um projeto e pode incluir várias tarefas no mesmo projeto | Alta |
| **RF003** | Criação de tarefas | O usuário cria uma tarefa e pode atribuir a tarefa a mais usuários | Alta. |
| **RF004** | Status da tarefa | De acordo com o andamento da tarefa o status pode estar como em andamento, em atraso, concluído | Média |
| **RF005** | Pesquisa de tarefas | Pesquisar tarefas a partir das tarefas criadas | Média. |
| **RF006** | Comentário de tarefas | Pode comentar quem está atrelado a tarefa | Alta |
| **RF007** | Atribuição de tarefas aos usuários | Atribuir tarefa a usuários respectivos | Alta |

## Fonte: Elaborado pelo autor.

## 5.2. Diagrama de Classe

O objetivo deste diagrama é representar a estrutura que vai estar presente no sistema. Assim entender melhor em uma visão geral a aplicação e expressar visualmente as necessidade específicas de nossa aplicação.

**Figura 8: Diagrama de classe DevWorthy**

**Fonte: Elaborado pelo autor.**

# 5.3. Metodologia

# 

Foi utilizado a metodologia ágil Scrum para o gerenciamento de equipe e projeto, também fizemos uso da ferramentas de apoio Trello. As técnicas utilizadas para coletas de informação foram:

1. Primeiramente foi realizado uma observação direta nos executores do processo, no caso os desenvolvedores. Foi observado que o desenvolvedor executa as demandas enviadas pelo supervisor ponderando se é possível, realiza testes convencionais e caso tenha encontrado possível solução ele envia para o supervisor.
2. Foi realizado também observação direta com supervisor das demandas do processo. Foi observado que o supervisor é responsável por identificar se uma demanda do cliente vai ser feita, se o resultado de uma demanda enviada pelo desenvolvedor é satisfatória e ele mesmo deve notificar o cliente sobre o projeto.
3. Também foi realizado uma pesquisa sobre qualquer documentação ou notas existentes do processo. Foi encontrado descrições escritas do processo pelo supervisor sobre as demandas consideradas importantes.
4. Então foi realizado entrevista com o usuário identificado como usuário chave pela equipe de análise. As perguntas foram:

- Como é decidido qual demanda realizar atualmente?

-> Orientação da supervisão ou decisão do desenvolvimento.

- Como é solicitado a alteração de um componente no software?

-> O supervisor envia o email ou solicita verbalmente a demanda ao desenvolvimento.

- Existe alguma forma do supervisor ou desenvolvedor identificar em qual status está a demanda de um software?

-> Somente se o supervisor questionar o andamento da demanda.

- Como é identificado se o resultado é satisfatório pelo cliente?

-> Não havendo retorno da demanda já solucionada.

- Quem realiza o testes da demanda?

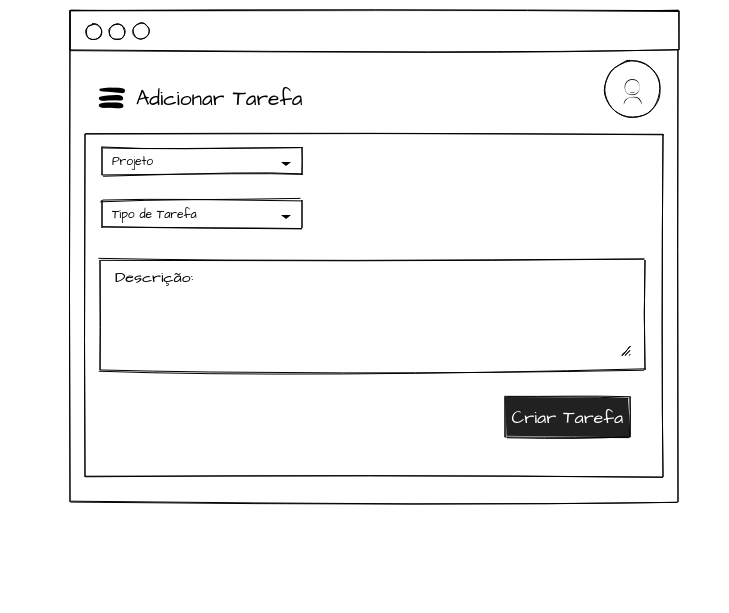
-> O próprio desenvolvimento.

Então com as informações coletadas foi possível modelar os diagramas de processos de negócios (bpmn) e a execução do diagrama de classes. Para a modelagem BPMN foi utilizado o Bizaggi e para o diagrama de classes um plugin da IDE eclipse e o Astah UML. Já no software em si, no back-end foi utilizado a IDE eclipse, linguagem Java, serviços web Java, framework Spark e gerenciador de projetos Maven. Para o Front-end foi utilizado o Javascript, VueJs, HTML e CSS.

## 5.4 Layout da tela

O objetivo do wireframe é auxiliar o desenvolvedor no entendimento dos requisitos que foram recolhidos junto ao cliente com relação às funções e objetos que um sistema deverá conter. Com esta motivação, realizamos os wireframes necessários para elucidar as possíveis telas de nossa aplicação.

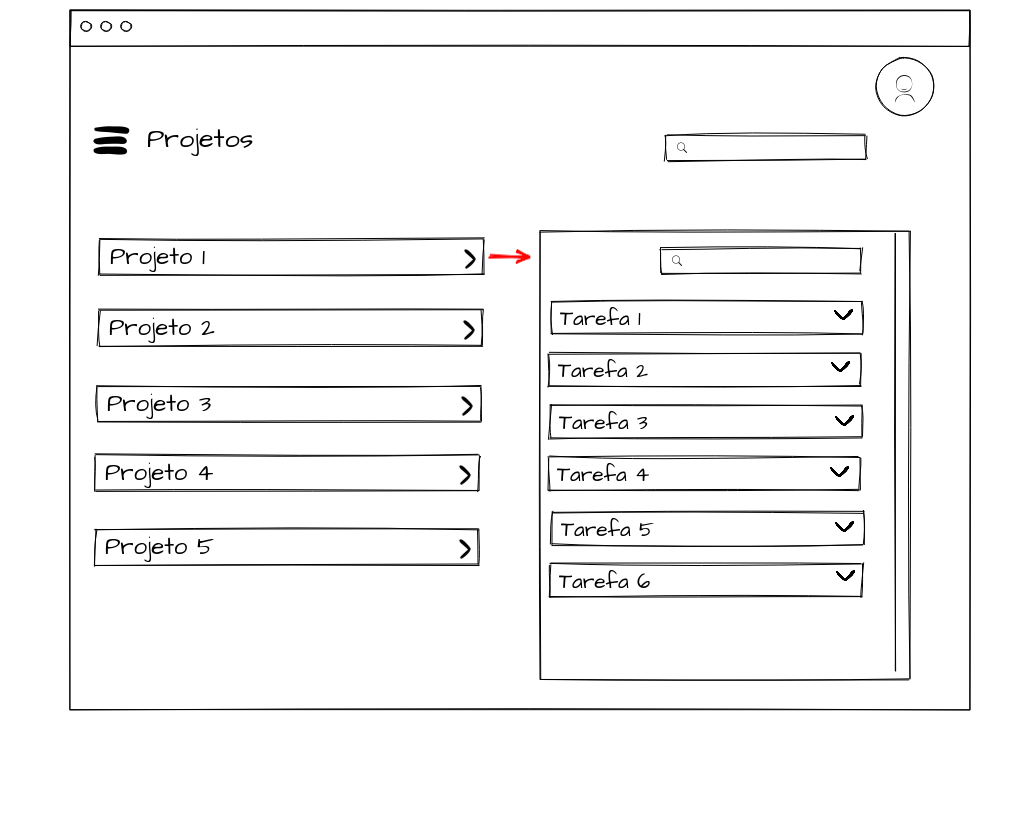
## 5.4.1 Adicionar Tarefa



## 5.4.2 Adicionar Usuário Adicionar_usuários

## 5.4.3 Editar Descrição da Tarefa Editar descrição tarefa 5.4.4 Editar Comentário da Tarefa Editar_comentários5.4.5 Editar Usuário Editar_usuário 5.4.6 Login

## login 5.4.7 Projetos



## 5.4.8 Titulos Tarefas tarefa 5.4.9 Tarefas

## tarefas 5.4.10 Listar Usuários usuarios

# 

## 6 USO SOFTWARE

Faça aqui uma breve descrição do software e coloque as principais telas com uma explicação de como usar cada uma

## 7 AVALIAÇÃO

Faça aqui sobre a avaliação do software. Indique se ele atendeu as expectativas e ele é viável. Para não ficar subjetivo, o ideal é fazer um questionário e pedir ao usuário do processo que faça a avaliação

## 8. CONCLUSÃO

Apresente aqui a conclusão do seu trabalho. Discussão dos resultados obtidos no trabalho, onde se verifica as observações pessoais de cada aluno. Poderá também apresentar sugestões de novas linhas de estudo.

# REFERÊNCIAS

Como um projeto de software não requer revisão bibliográfica, a inclusão das referências não é obrigatória. No entanto, caso você deseje incluir referências relacionadas às tecnologias, padrões, ou metodologias que serão usadas no seu trabalho, relacione-as de acordo com a ABNT.

Verifique no link abaixo como devem ser as referencias no padrão ABNT:

http://www.pucminas.br/imagedb/documento/DOC\_DSC\_NOME\_ARQUI20160217102425.pdf

# APÊNDICES

**Colocar link:**

Do código;

Dos artefatos;

Da apresentação final;

Do vídeo de apresentação.