TaskBerry: Soluções para uma fábrica de software

## Sarah Julia Campos de Freitas Lara

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

MINAS)

Belo Horizonte – MG – Brasil

sarah.lara@sga.pucminas.br

***Resumo.*** *Esse trabalho tem o objetivo de automatizar os processos de uma fabrica*

*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.*

## 1. 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

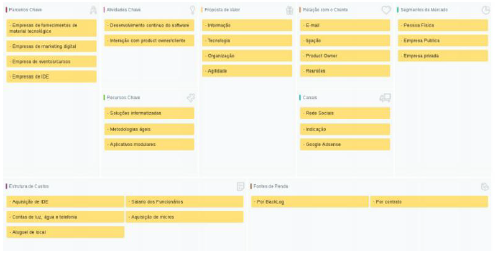
uma automatização dele.

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.

## 2. Modelo de negócios



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

## 2.1. Participantes do processo

Os stakeholders de uma fabrica 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.

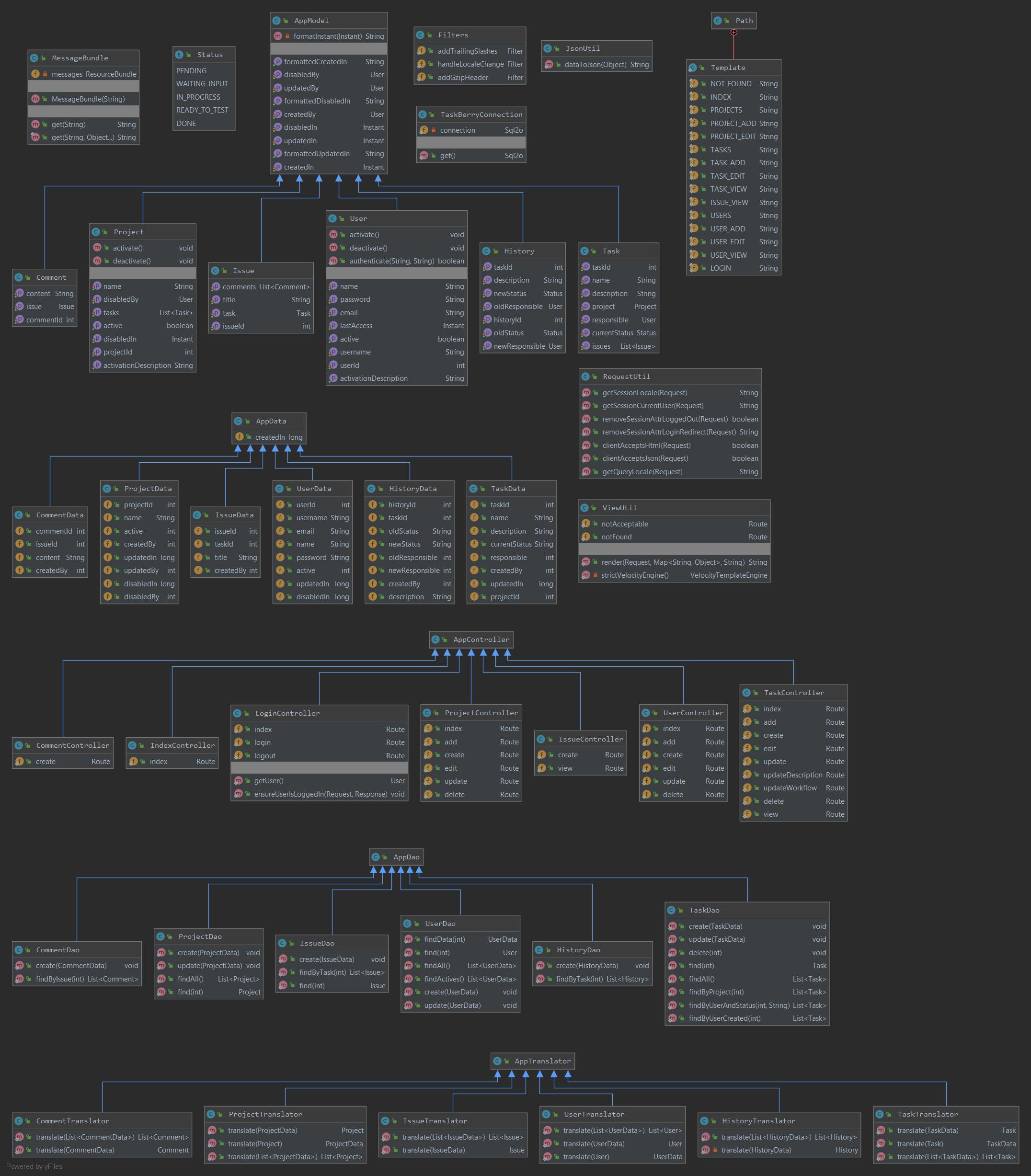
## 3. Projeto da Solução

## 3.1. Requisitos funcionais

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Processo/tarefa** | **Descrição** | **Prioridade** |
| **RF001** | Criação de tarefas. | O usuário pode cadastrar novas tarefas. | Alta |
| **RF002** | Criação de projetos. | O usuário pode cadastrar novos projetos. | Alta |
| **RF003** | Mudança de status. | O usuário pode mudar o status das tarefas. | Alta. |
| **RF004** | Descrever tarefa | O usuário pode adicionar descrição para uma tarefa. | Média |
| **RF005** | Listar tarefas | O usuário pode visualizar uma lista de tarefas filtrando por status e atribuição | Média. |
| **RF006** | Atribuição de tarefas | O usuário pode atribuir um usuário como responsável por uma tarefa. | Média |
| **RF007** | Login | O usuário deve executar login para acessar o sistema. | Média |
| **RF008** | Logout | O usuário pode encerrar sua sessão para bloquear o acesso ao sistema. | Média |
| **RF009** | Cadastro de usuários | O usuário administrador pode cadastrar novos usuários e editar os existentes. | Média |
| **RF010** | Desativar projeto | O usuário administrador pode desativar projetos, bloqueando a criação de novas tarefas e encerrando todas as existentes. | Baixa |

## 3.2. Diagrama de Classes

Observação: O diagrama de classes também está disponível no seguinte link para visualização em maior resolução: <https://github.com/sarahjfreitas/taskBerry/blob/master/docs/Diagrama%20de%20Classes.jpg>



# 

## 3.3. Metodologia

Para coleta de informação foi usado como base principalmente o estudo de soluções similares que são concorrentes no mercado.

O software será disponibilizado na plataforma web sendo publicado em um servidor que será definido no futuro.

O código será feito principalmente em Java com o auxílio do framework Spark. Será usado também componentes do Bootstrap no front end e JavaScript com JQuery. Além disso SQLite será usado como gerenciador de banco de dados para armazenamento de informações.

Para produzir o código será utilizada a IDE IntelliJ IDEA e a ferramente SQLite Studio será usada para facilitar o gerenciamento do banco de dados. Além disso o versionamento do código será feito usando o GIT e será armazenado em um repositório no GitHub.

## 3.4. Layout da tela

Observação: Todos os layouts de tela podem ser acessados no seguinte link para visualização em maior resolução: <https://github.com/sarahjfreitas/tis2/tree/master/docs/Templates%20de%20Telas>

## 3.4.1 Adicionar Tarefa



## 3.4.2 Adicionar Usuário

## 

## 3.4.3 Editar Descrição da Tarefa

## 

## 3.4.4 Editar Comentário da Tarefa

## 

## 3.4.5 Editar Usuário

## 

## 3.4.6 Login

## 

## 3.4.7 Tarefa

## 

## 3.4.6 Listar Tarefas

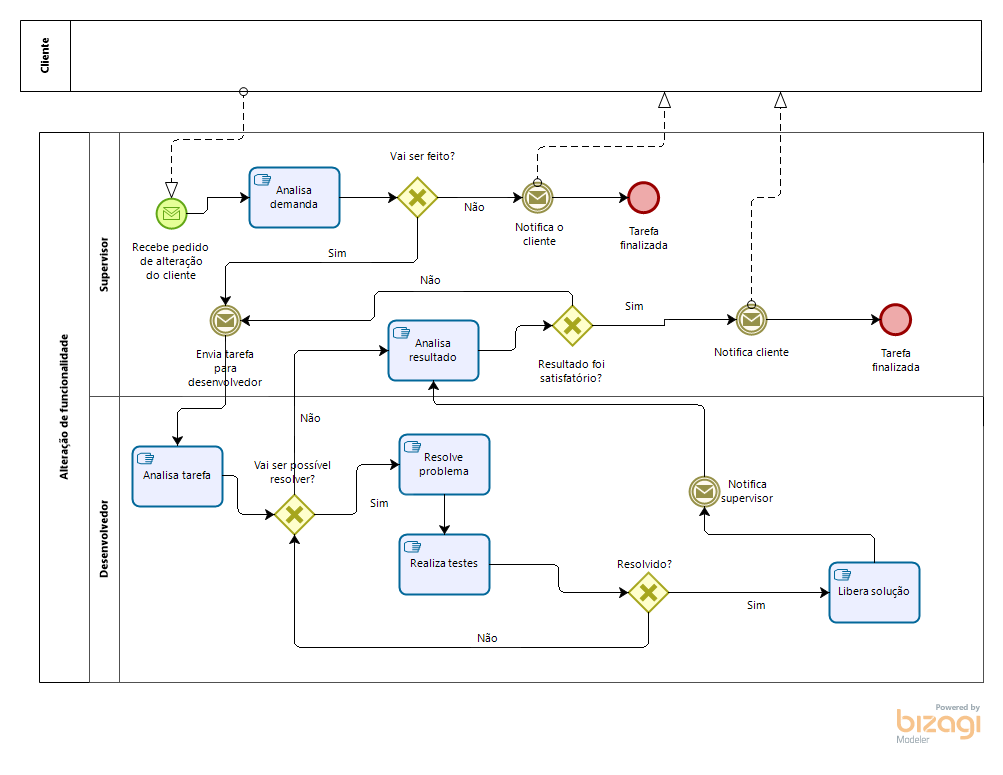
## 

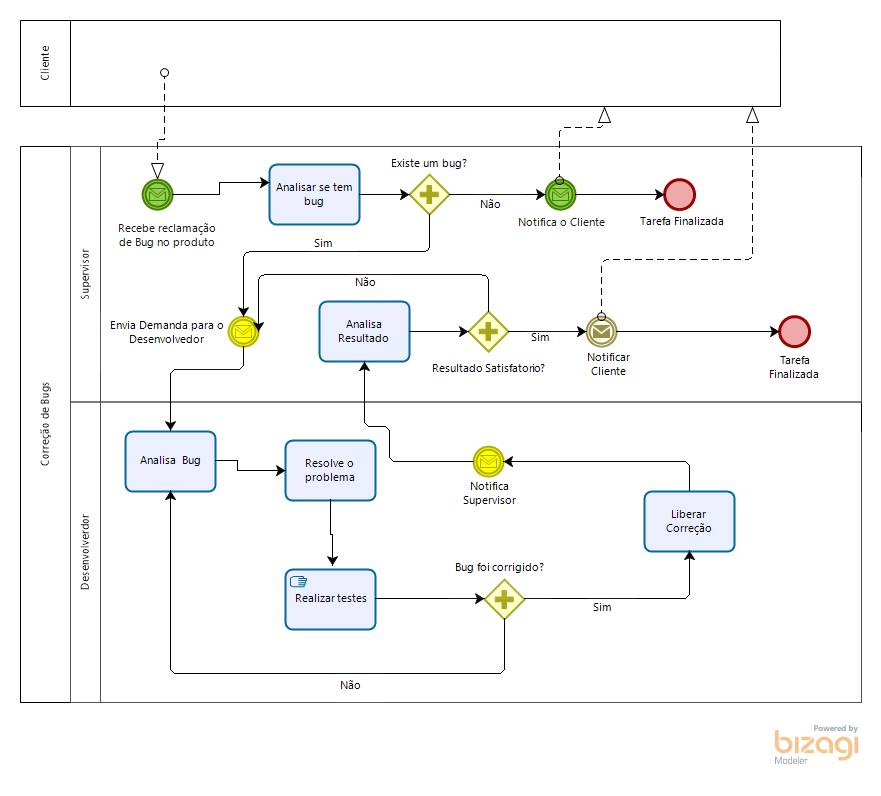
## 3.4.6 Listar Usuários

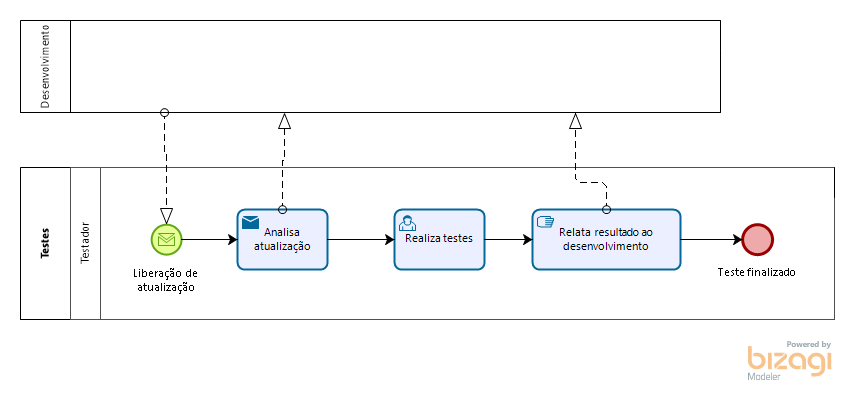
# 4. Modelagem do processo de negócio

Observação: Os modelos demonstrados abaixo podem ser acessos no seguinte link para visualização em maior resolução: <https://github.com/sarahjfreitas/taskBerry/tree/master/docs/Modelagem%20de%20Processos>

## 4.1. Análise da situação atual

O foco desta análise é uma empresa de desenvolvimento de software que não possui um sistema para organizar suas pendências. Analisamos a situação atuação de seus processos e verificados diversos problemas que poderiam facilmente serem resolvidos caso a empresa tivesse maior organização em suas tarefas. Segue abaixo alguns processos que foram analisados:  
  
**Alteração de funcionalidade**:  
  
No modelo assim é possível perceber as falhas do processo analisado. A resolução depende muito do conhecimento e memória do supervisor e do desenvolvedor. Não há nenhuma forma de documentação e a análise da demanda é superficial. O principal problema do processo atual é que a tarefa pode ir e voltar entre o supervisor e o desenvolvedor várias vezes até que uma solução seja encontrada, gerando muito retrabalho e prejuízo para a empresa.

**Correção de bugs:**Na correção de bugs vemos que as opções são superficiais, dando muitas brechas para a não correção do bug todo em si. Primeiramente a análise do bug é feita pelo supervisor, que provavelmente já fica sobrecarregado com outras obrigações. O desenvolvedor fica preso tentando resolver o problema e não possui nenhuma ajuda externa caso encontre obstáculos. Posteriormente, o supervisor novamente fica responsável por encaminhar o resultado para o cliente, causando ainda mais sobrecarga em seu trabalho.

**Testes:**

Nesse processo é evidente notar que os testes são rasos, sem opções de maiores questionamentos ou novos destinos da tarefa. Com esse modelo simplificado de testes a probabilidade de realmente encontrar erros é muito baixa.

## 4.2. Descrição Geral da proposta

Nossa proposta é um software capaz de gerenciar as tarefas relacionadas com desenvolvimento de software, garantindo mais precisão em todos esses processos e, consequentemente, melhor qualidade, menos erros e mais satisfação ao cliente. Queremos um ambiente onde os setores de desenvolvimento, testes e supervisão possam colaborar uns com os outros em prol da melhoria e da qualidade, com uma maior facilidade de troca de informações.

Foi acrescentado o setor de atendimento, que será responsável pelo contato direto com o cliente, garantindo diminuição no trabalho do supervisor e garantindo uma melhor comunicação com o cliente.

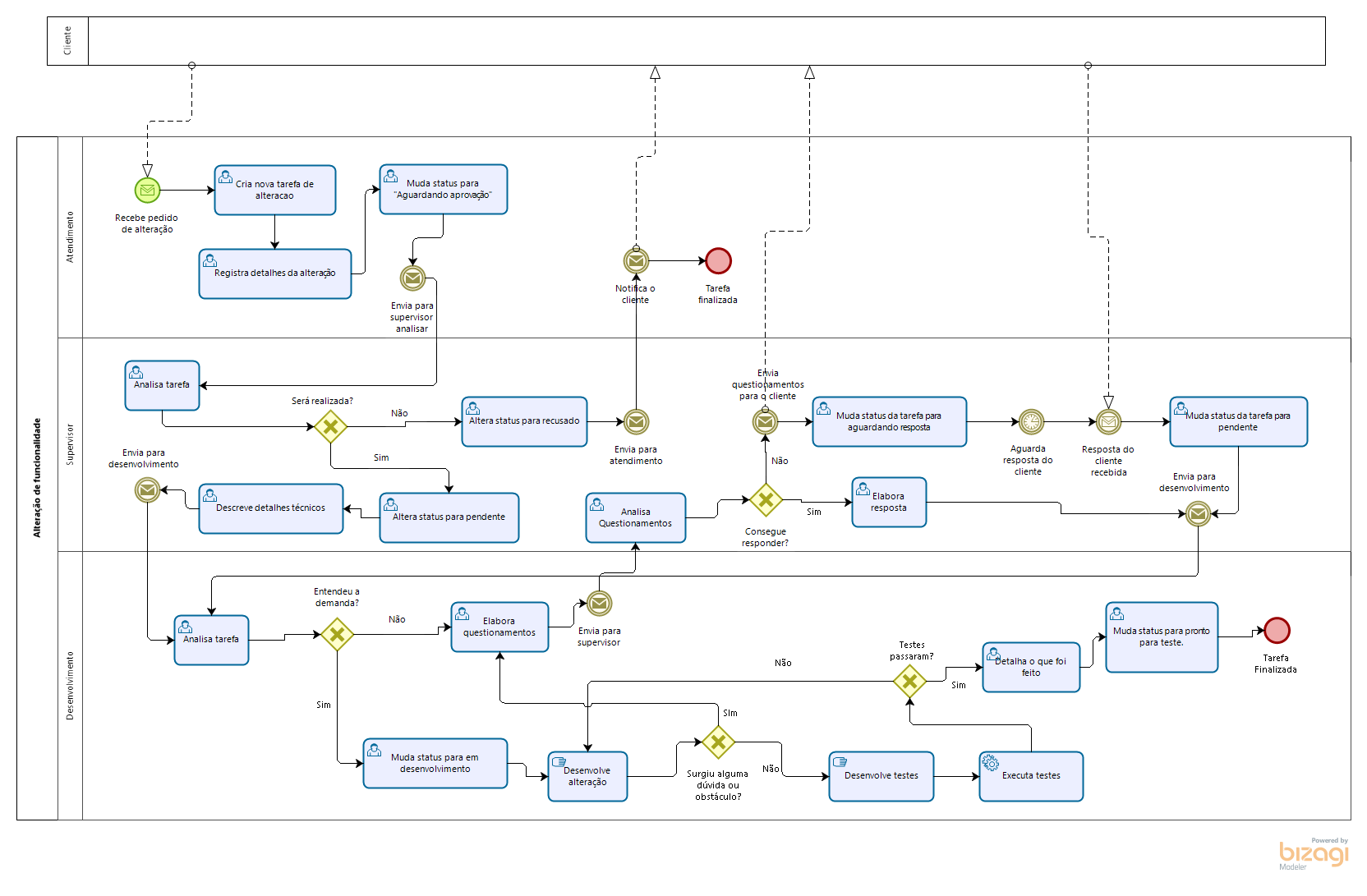
Todas as tarefas serão adicionadas no sistema, assim mantendo organização das tarefas atuais e mantendo um histórico das alterações que foram realizadas no passado. As tarefas vão seguir um fluxo de acordo com sua situação (status) atual. Exemplo: pendente, concluída, aguardando resposta.

## 4.3. Modelagem dos processos

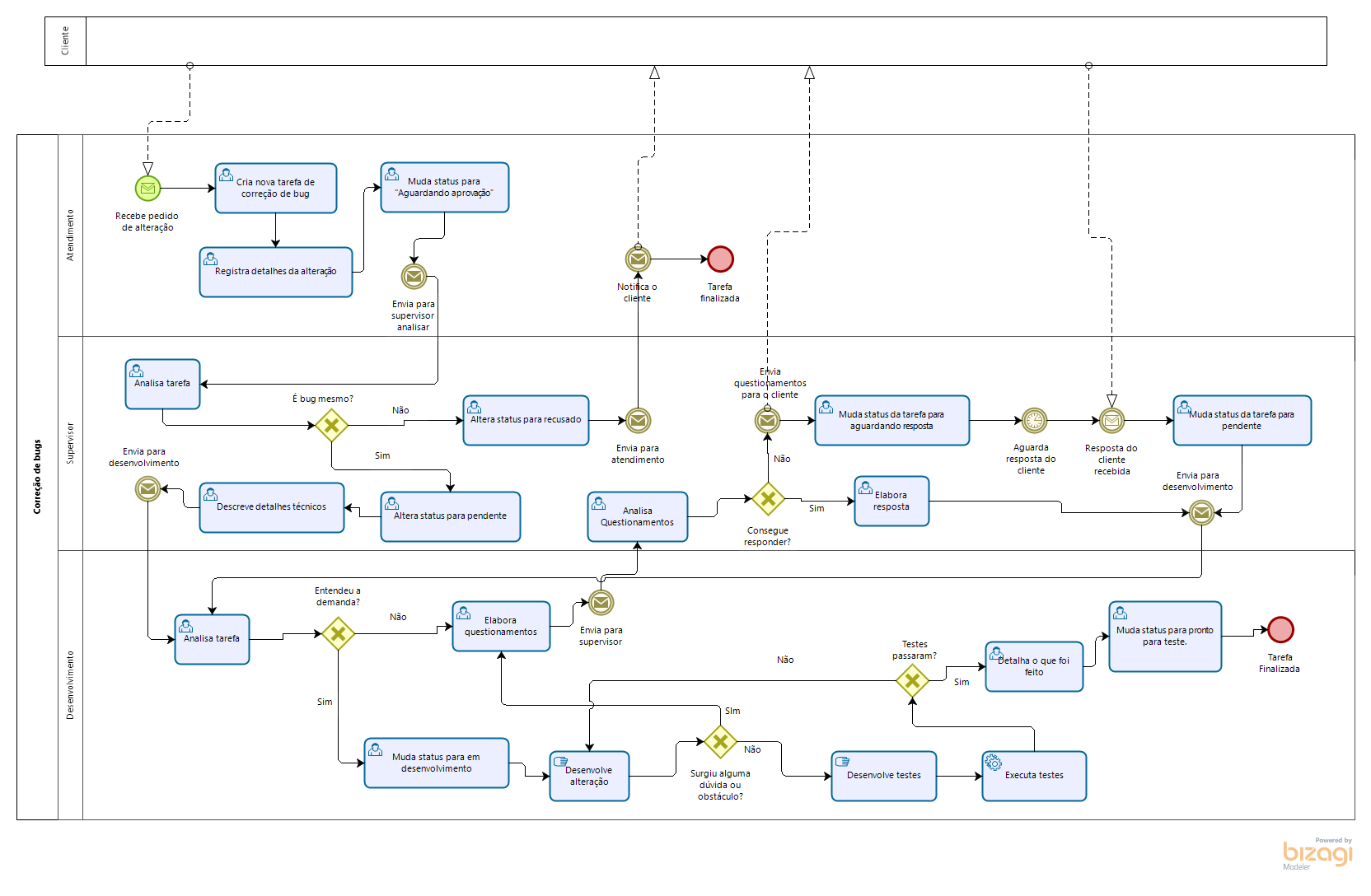
Segue abaixo a proposta de melhoria nos processos mostrados acima.

## 4.3.1 Processo 1 – Alteração de Funcionalidade

Como descrito acima, as tarefas serão registradas no sistema, garantindo que todas as partes fiquem cientes de seu andamento, fazendo com que não dependa mais da memória dos envolvidos. A tarefa terá uma análise muito mais detalhada, e em caso de questionamentos não previstos o cliente pode ser contactado.

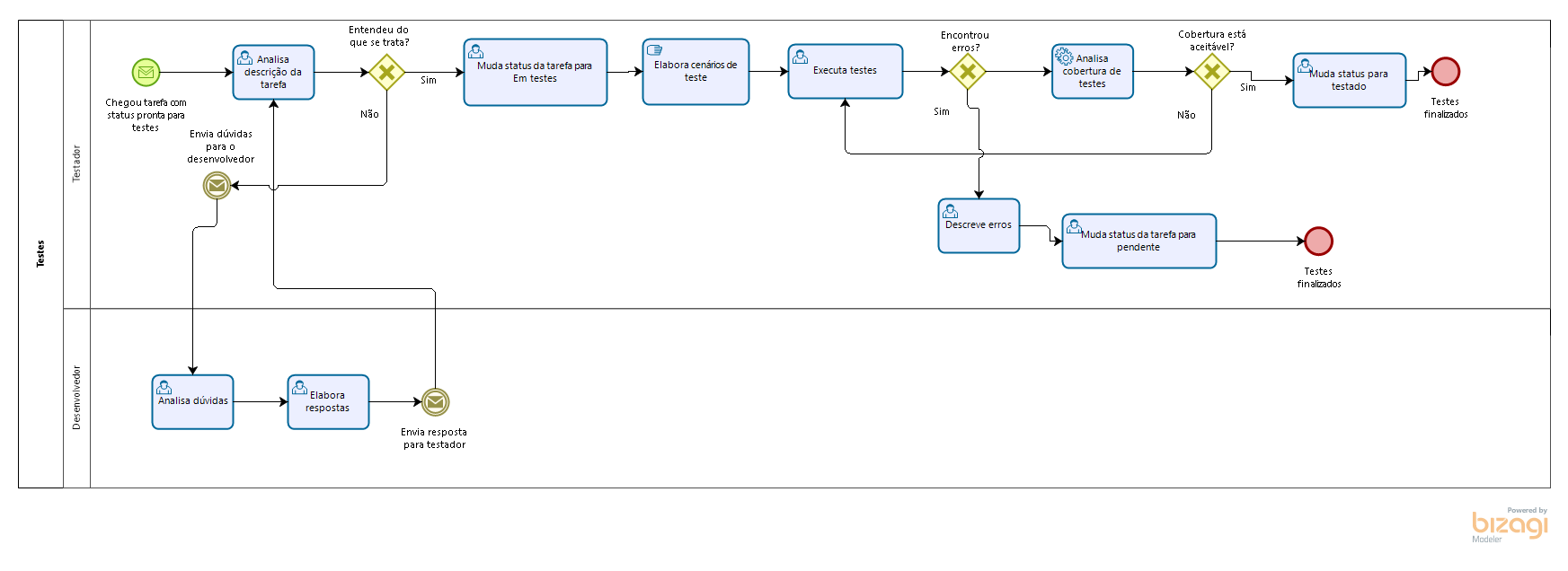


## 4.3.2 Processo 2 – Correção de Bugs Uma breve visualização dos modelos entre AS IS e TO BE, já podemos notar uma drástica mudança no processo de correção de bugs. A análise é muito mais detalhada, gerando uma maior chance de entendimento do problema. O cliente pode ser questionado durante o processo caso necessário. Os testes são bem mais rígidos e o desenvolvedor tem a possibilidade de pedir ajuda caso tenha dificuldades.



## 4.3.3 Processo 3 – Testes

## O processo proposto para testes é quase completamente diferente da situação atual. Primeiramente o setor de testes tem acesso a tarefa registrada no sistema, podendo verificar com facilidade o detalhamento de tudo que foi feito. Caso mesmo com esse registro, tenha dificuldades em entender os detalhes da tarefa que deve testes, existe a opção de enviar dúvidas para o desenvolvedor responsável pela tarefa. Além disso, os cenários de testes são elaborados com cuidado, os erros são detalhados na tarefa em questão e ainda há uma análise para verificar se a cobertura dos testes foi adequada. Com essas mudanças há chance de um erro passar despercebido diminui drasticamente.

****

**4.4. Indicadores de desempenho**

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **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 |

## 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

Avaliando os resultados é possível concluir que o projeto foi um sucesso. As principais funcionalidades desejadas foram implementadas com sucesso e até mesmo aprimoradas durante o desenvolvimento gerando um sistema em funcionamento com uso viável pelo mercado de trabalho.

## 8. Conclusão

Os resultados do projeto foram satisfatórios pois o sistema proposto foi concluído e sua usabilidade ficou melhor do que o esperado no início do projeto. Além disso durante o desenvolvimento foram percebidos diversos pontos de melhoria para que o projeto seja lapidado e distribuído como uma ferramenta para o público.

# APÊNDICES

Repositório do projeto: <https://github.com/sarahjfreitas/taskBerry/>

Código fonte: <https://github.com/sarahjfreitas/taskBerry/tree/master/src/main>

Documentação e artefatos diversos: <https://github.com/sarahjfreitas/taskBerry/tree/master/docs>

Apresentação final:

Vídeo de apresentação: