



*Tecnologia em Análise e Desenvolvimento de Sistemas
4º Período*

Oficina de Integração
AN34E / N14

PROFESSOR

Diogo Cezar Teixeira Batista

DOCUMENTO DE PROJETO DE SOFTWARE

Task Manager X

Carlos R. F. Sanches

N. Boralli Neto

Rodrigo Miyashiro Mucci

Thiago Oliveira de Souza

SUMÁRIO

INTRODUÇÃO	3
CONTEXTO	3
JUSTIFICATIVA	3
PROPOSTA	3
ORGANIZAÇÃO DO DOCUMENTO	4
DESCRIÇÃO GERAL DO SISTEMA	5
OBJETIVOS	5
LIMITES E RESTRIÇÕES	5
DESCRIÇÃO DOS USUÁRIOS DO SISTEMA	6
DESENVOLVIMENTO DO PROJETO	7
TECNOLOGIAS E FERRAMENTAS	7
METODOLOGIA DE DESENVOLVIMENTO	8
CRONOGRAMA PREVISTO	8
REQUISITOS DO SISTEMA	10
REQUISITOS FUNCIONAIS	10
REQUISITOS NÃO-FUNCIONAIS	11
DIAGRAMAS DE CASOS DE USO	12
ANÁLISE DO SISTEMA	13
MODELO DO BANCO DE DADOS	13
MODELO CONCEITUAL	13
MODELO LÓGICO	14
DICIONÁRIO DE DADOS	15
DIAGRAMA DE CLASSES	17
DIAGRAMA DE ATIVIDADES	18
BIBLIOGRAFIA	28

1 INTRODUÇÃO

1.1 CONTEXTO

Com a evolução do modo de desenvolver produtos e o aumento de desenvolvedores trabalhando em um ou mais projetos, se mostra necessário uma maior interação e compreensão geral do projeto como um todo.

Diante do exposto acima, fica claro a indispensabilidade de um *software* que vá suprir as carências expostas no atual cenário de desenvolvimento de produtos.

1.2 JUSTIFICATIVA

A falta de organização no desenvolvimento de produtos, torna inviável o bom gerenciamento no seu desenvolvimento, afetando tanto desenvolvedores quanto clientes. A falta de um bom gerenciamento dificulta o cumprimento de prazos e entendimento de ambas as partes interessadas no produto, comprometendo assim o acordo entre os envolvidos.

Nosso *software* propõe tornar a comunicação mais clara e precisa entre os envolvidos no desenvolvimento do projeto. Apresentando a eles, através de uma interface simples e intuitiva, funcionalidades que auxiliarão os envolvidos durante a fase de desenvolvimento do projeto.

1.3 PROPOSTA

Visto que as alternativas disponíveis para *softwares* de gerenciamento de projetos no mercado, por vezes, apresentam uma interface de interação com o usuário de difícil entendimento. Nosso *software* propõe um sistema com uma

interface simples e intuitiva, que facilite a compreensão do andamento do projeto, tornando mais claro o processo de desenvolvimento, facilitando a visualização das tarefas a serem cumpridas e suas prioridades.

Diferente da grande maioria dos *softwares* disponíveis no mercado, o mesmo oferece total acesso à todas as suas funcionalidades de forma gratuita. Visando também a evolução do cenário atual, nosso *software* será *open source* (código livre para a acesso por todos).

1.4 ORGANIZAÇÃO DO DOCUMENTO

Este documento está organizado em 5 capítulos, cada um dividido nos seguintes tópicos:

1. Introdução, contextualizando e justificando o projeto.
2. Descrição geral do sistema, apresentando seus objetivos e restrições.
3. Desenvolvimento do sistema, descrevendo as ferramentas manipuladas, metodologias utilizadas e o cronograma.
4. Requisitos do sistema contíguo ao diagrama de casos de uso.
5. Referências bibliográficas consultadas que auxiliaram na criação deste documento.

2 DESCRIÇÃO GERAL DO SISTEMA

2.1 OBJETIVOS

Com o intuito de oferecer ao usuário uma plataforma, onde ele consiga visualizar e compreender todo o processo de desenvolvimento do seu projeto, nosso *software* tem como o objetivo principal, facilitar o processo de gerenciamento de projetos, tornando o mais simples e de fácil entendimento.

O *software* tem em vista alcançar os seguintes objetivos:

- Proporcionar ao usuário condições de definir tarefas, que visam fragmentar as fases do projeto que está sendo desenvolvido.
- Implementar uma interface de interação simples e intuitiva, que evidencie as tarefas presentes no projeto.
- Fornecer ao usuário um mecanismo de interação que manipule as tarefas, bem como os projetos.
- Tornar a compreensão geral da equipe envolvida no desenvolvimento do projeto, mais clara, facilitando assim, a comunicação e interação entre os usuários envolvidos.

2.2 LIMITES E RESTRIÇÕES

Este projeto está limitado à gerenciar a organização do desenvolvimento de um produto, mostrando todas as fases que o transpõe, entretanto não é responsável por quaisquer casos além do gerenciamento.

Para o funcionamento íntegro do sistema é necessário uma conexão estável com a internet e possuir um navegador atualizado.

O prazo máximo para a entrega deste projeto será até a data limite: 01 de Dezembro de 2017.

Para a utilização do sistema, faz se necessário o cadastro do usuário.

2.3 DESCRIÇÃO DOS USUÁRIOS DO SISTEMA

Este sistema se destina a qualquer empresa e/ou usuário que apresente ser necessário um sistema gerenciador de projetos. Os usuários podem ser classificados como: Gerente de projeto tem como propósito definir as metas, vidas úteis, prazos e prioridades do projeto; Desenvolvedor é designado a cumprir os critérios definidos pelo Gerente.

3 DESENVOLVIMENTO DO PROJETO

3.1 TECNOLOGIAS E FERRAMENTAS

- Banco de dados
 - *mongoDB* -> Foi escolhido para este projeto devido a boa interação com outras ferramentas utilizadas neste projeto, que serão apresentadas a seguir, e por se tratar de uma ferramenta livre que possui um bom suporte da comunidade. O *mongoDB* se trata de um banco de dados não relacional, de fácil usabilidade e capaz de um alto armazenamento de dados.
- Linguagens de programação
 - *JavaScript* -> Por ser uma linguagem em constante desenvolvimento, com alto apoio da comunidade e por ser extremamente adaptável às necessidades do projeto.
- Criação de layout
 - *HTML5* -> Por ser a linguagem de marcação mais encontrada na web, seu uso é inquestionável.
 - *CSS3* -> É a folha de estilo com melhor integração com o *HTML5*.
- Controle de versão
 - *GitHub* -> Por se tratar da ferramenta mais utilizada pela comunidade.
- Diagramação
 - *Astah* -> Melhor opção gratuita disponível no mercado.
- *Frameworks*
 - *VueJs* -> Para tornar mais dinâmica interação do usuário com o sistema, e por possuir uma curva de aprendizado melhor que as outras alternativas.
 - *ExpressJs* -> Para facilitar as operações de manipulação de dados e por ser uma ferramenta de fácil uso e aprendizagem.

3.2 METODOLOGIA DE DESENVOLVIMENTO

O modelo escolhido para o desenvolvimento desta aplicação foi o *BASE*, por ser um método extremamente flexível, ele se encaixa adequadamente às necessidades do projeto.

Como um modelo flexível, o desenvolvimento do projeto ocorre de uma maneira fluida e livre, apenas seguindo quatro pilares do *BASE*: comunicação, simplicidade de projeto e documentação, planejamento e trabalho contínuo.

BASE é entendido por um conjunto de pequenos processos, permitindo uma implementação mais ágil e segura, além de garantir a manutenibilidade do sistema. Não existem regras no *BASE* apenas dá o alvitre na forma pela qual o processo possa ser desenvolvido de forma simples e prática.

3.3 CRONOGRAMA PREVISTO

Nome	Data Inicial	Data Final	Desenvolvedor
Elaboração da descrição do sistema	18/08/17	01/09/17	Neto
Elaboração dos requisitos e caso de uso	04/09/17	08/09/17	Carlos
Modelagem do banco de dados	11/09/17	22/09/17	Rodrigo e Thiago
Criação dos diagramas de classe e atividades	25/09/17	06/10/17	Carlos
Desenvolvimento <i>back-end</i>	09/10/17	20/10/17	Neto e Carlos

Desenvolvimento front-end	23/10/17	03/11/17	Thiago e Rodrigo
Fase de testes	06/11/17	17/11/17	Todos
Resoluções finais	19/11/17	01/12/17	Todos

Esta tabela define como será procedido o desenvolvimento do projeto, bem como seus prazos e por quem serão executados.

4 REQUISITOS DO SISTEMA

4.1 REQUISITOS FUNCIONAIS

ID	FUNCIONALIDADE	PRIORIDADE
RF01	O sistema deve permitir a criação de um novo projeto ou tarefa.	Essencial
RF02	O sistema deve permitir a edição dos projetos ou tarefas já existentes.	Essencial
RF03	O sistema deve permitir o cadastro de usuários com login e senha.	Essencial
RF04	O sistema deve permitir a associação de prioridades às tarefas existentes.	Importante
RF05	O sistema deve verificar a autenticidade dos usuários.	Importante
RF06	O sistema pode gerar relatórios gerais e/ou específicos.	Desejável

Esta tabela representa os requisitos funcionais, ou seja, os requisitos que se referem as funcionalidades do sistema, e também apresentando seu nível de prioridade, sendo elas: Essencial, importante e desejável.

Descrição dos requisitos funcionais :

RF01: O sistema deve permitir, através de uma interface gráfica, que o usuário crie projetos definindo nome, equipe de desenvolvimento e prazos. Bem como pode adicionar tarefas as esses projetos definindo nome, descrição, prazo e prioridade.

RF02: O sistema deve permitir, através de uma interface gráfica, que o usuário altere projetos modificando seu nome e adicionando ou removendo membros da equipe de desenvolvimento. Bem como a alteração de tarefas, modificando seu nome, descrição, prazo e prioridade.

RF03: O sistema deve permitir, através de uma interface gráfica, que o usuário realize o seu cadastro e login, mediante apresentação de um *e-mail* válido. Também permite a alteração da senha de usuário, mediante a ultima senha correta.

RF04: O sistema deve permitir, através de uma interface gráfica, que o usuário possa adicionar prioridades as tarefas previamente criadas. As prioridades deverão ser alteradas conforme o prazo de entrega da tarefa se aproxima do prazo final.

RF05: O sistema deve verificar, por meio de uma requisição via *e-mail*, a autenticidade do usuário que solicitou o cadastro.

RF06: O sistema pode gerar, através de solicitação do usuário, relatórios gerais e/ou específicos sobre o andamento do projeto. Os relatórios podem ser apresentados por meio de gráficos de coluna, barra, setores e linha.

4.2 REQUISITOS NÃO-FUNCIONAIS

ID	FUNCIONALIDADE	PRIORIDADE
RNF01	O sistema necessita de uma conexão estável com a <i>internet</i> para funcionar integralmente.	Usabilidade
RNF02	O sistema deve funcionar na maior parte dos navegadores modernos.	Compatibilidade

Está tabela representa os requisitos não funcionais, ou seja, os requisitos que se referem em termos de usabilidade, desempenho, compatibilidade, segurança e confiabilidade.

Descrição dos requisitos não funcionais:

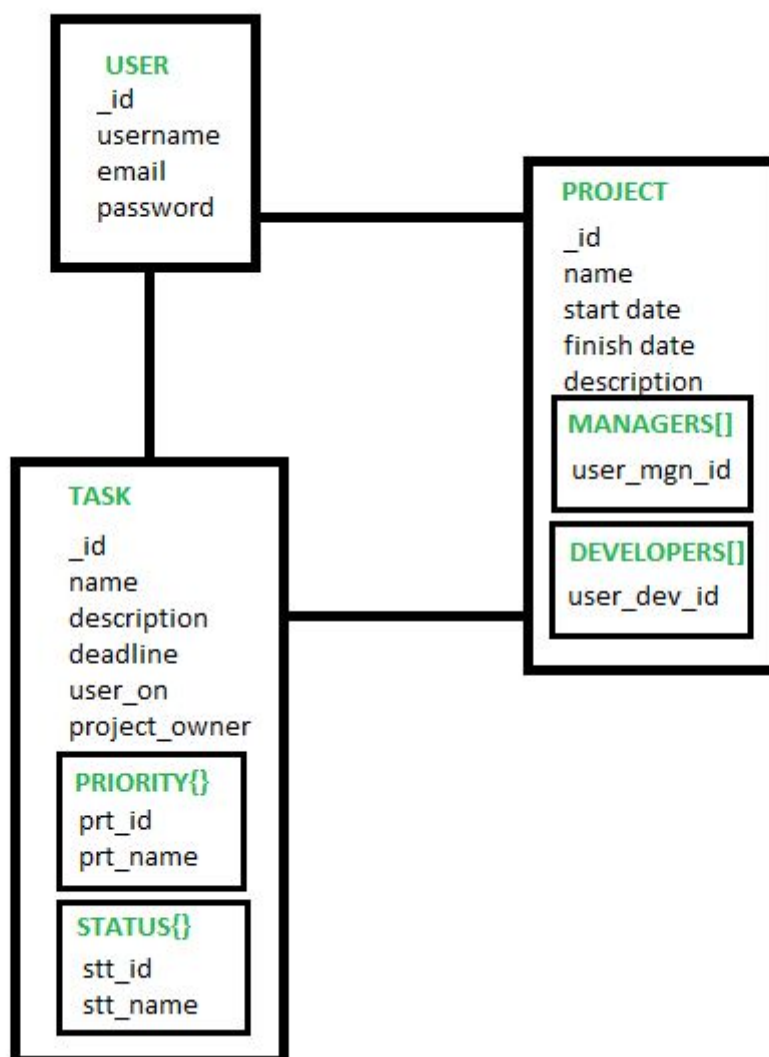
RNF01: Por se tratar de um sistema unicamente web, ele necessita de uma conexão integral com a *internet*. Na ausência de conexão com a *internet* o usuário não poderá acessar nenhuma das funcionalidades do sistema.

RNF02: O sistema deve funcionar nas versões atualizadas dos seguintes navegadores: *Mozilla Firefox*, *Google Chrome*, *Opera*, *Safari* e *Microsoft Edge*.

5 ANÁLISE DO SISTEMA

5.1 MODELO DO BANCO DE DADOS

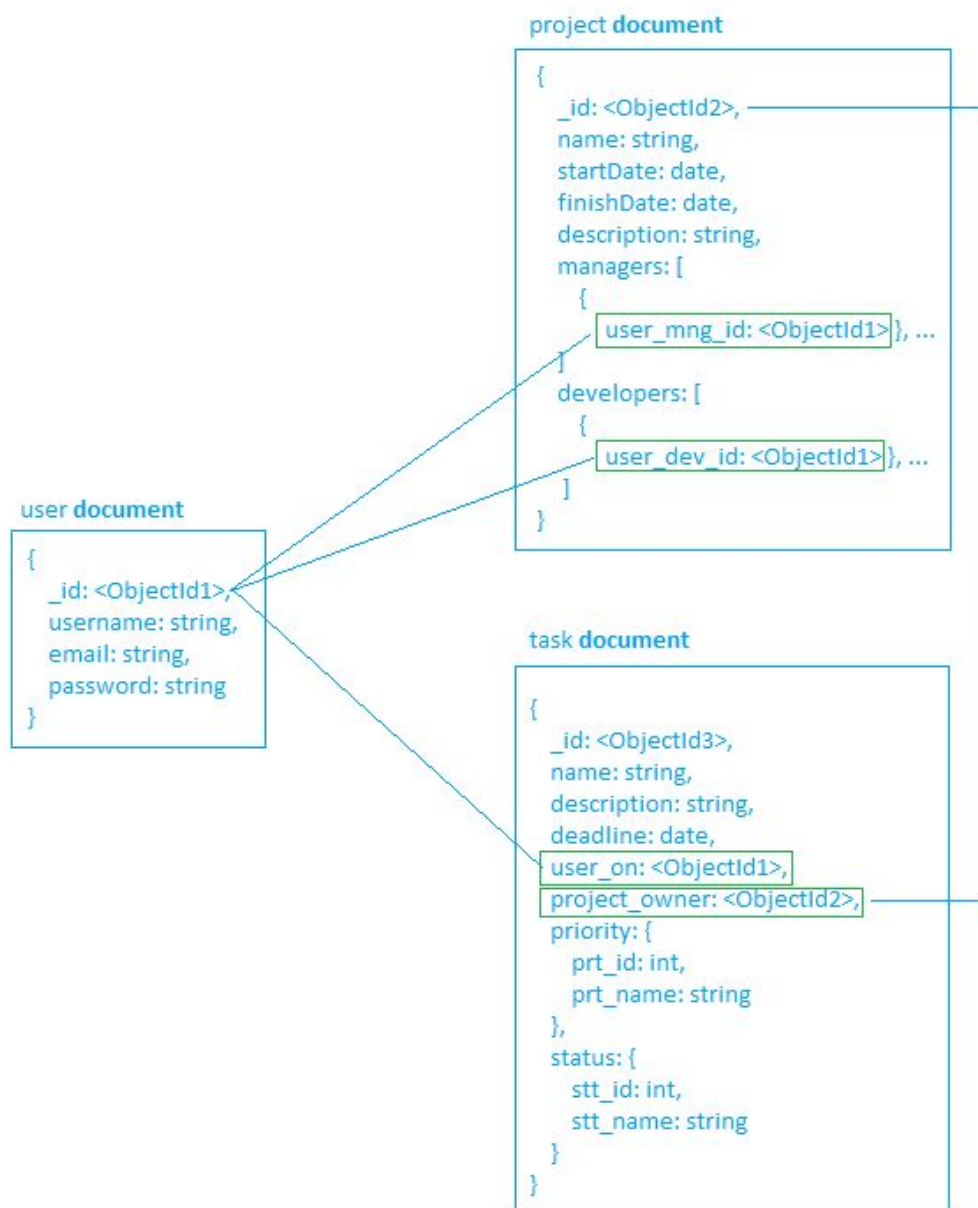
5.1.1 MODELO CONCEITUAL



A imagem acima representa a modelagem conceitual do banco de dados. Por se tratar de um banco não relacional (*noSQL*), não apresenta um padrão de modelagem UML. O que seria uma relação “muitos para muitos” em um esquema relacional, neste tipo de modelagem se torna um *array*, como no caso dos sub-documentos “*managers*” e “*developers*” no documento *project*. No caso dos

sub-documentos “*priority*” e “*status*”, que em um modelo relacional, seriam duas tabelas separadas, foram transformados em dois objetos dentro do documento *task*.

5.1.2 MODELO LÓGICO



A imagem acima representa o modelo lógico do banco de dados. Novamente, por se tratar de um banco de dados não relacional, onde o conceito de tabelas, colunas

e linhas não está presente, sua modelagem se torna muito parecida com um arquivo do tipo *JSON*. A imagem também apresenta os tipos de dados que cada atributo receberá. Os tipos “<ObjectId1>”, “<ObjectId2>” e “<ObjectId3>” são valores gerados automaticamente pelo banco, uma relação entre documentos é criada quando um atributo de um documento X recebe um valor do tipo “ObjectIdY”. No caso acima, existem 4 relações entre documentos, duas entre *user* e *project* (*user_mng_id* e *user_dev_id*), uma entre *user* e *task* (*user_on*) e uma entre *project* e *task* (*project_owner*).

5.1.3 DICIONÁRIO DE DADOS

user **document:**

ATRIBUTO	TIPO	OBSERVAÇÃO
<i>_id</i>	<ObjectId1>	É um objeto único, gerado pelo banco, contendo números e letras.
<i>username</i>	<i>string</i>	
<i>email</i>	<i>string</i>	Deve ser um e-mail existente e válido.
<i>password</i>	<i>string</i>	A senha armazenada no banco será previamente criptografada.

Tabela que representa os nome, seus tipos e algumas observações, dos atributos do documento *user*.

project **document:**

ATRIBUTO	TIPO	OBSERVAÇÃO
<i>_id</i>	<ObjectId2>	É um objeto único, gerado pelo banco, contendo números e letras.
<i>name</i>	<i>string</i>	
<i>startDate</i>	<i>date</i>	Deve ser gerada no momento da criação de um novo projeto.
<i>finishDate</i>	<i>date</i>	Não pode ser menor que a data de início.
<i>description</i>	<i>string</i>	
<i>managers</i>	<i>object array</i>	Array de objetos que armazena os <i>ids</i> do documento <i>user</i> .
<i>user_mng_id</i>	<ObjectId1>	Objeto <i>id</i> gerado no documento <i>user</i> .
<i>developers</i>	<i>object array</i>	Array de objetos que armazena os <i>ids</i> do documento <i>user</i> .
<i>user_dev_id</i>	<ObjectId1>	Objeto <i>id</i> gerado no documento <i>user</i> .

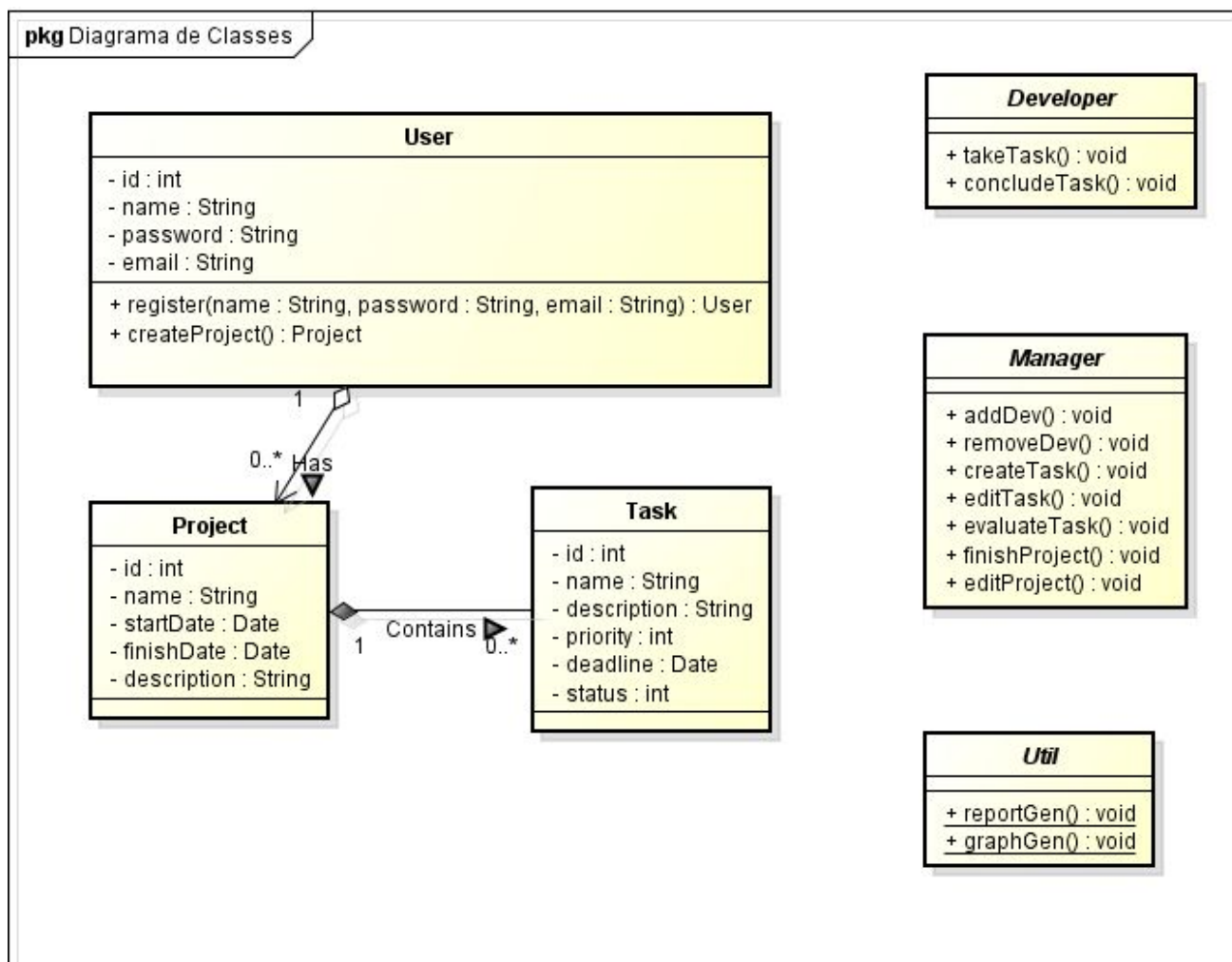
Tabela que representa os nome, seus tipos e algumas observações, dos atributos do documento *project*.

task document:

ATRIBUTO	TIPO	OBSERVAÇÃO
<i>_id</i>	<ObjectId3>	É um objeto único, gerado pelo banco, contendo números e letras.
<i>name</i>	<i>string</i>	
<i>description</i>	<i>string</i>	
<i>deadline</i>	<i>date</i>	Não pode ser menor que a data de início do projeto, e nem maior que a data de término.
<i>user_on</i>	<ObjectId1>	Objeto <i>id</i> gerado no documento <i>user</i> . Representa o último usuário a ter modificado a <i>task</i> ..
<i>project_owner</i>	<ObjectId2>	Objeto <i>id</i> gerado no documento <i>project</i> . Representa a qual <i>project</i> a <i>task</i> pertence.
<i>priority</i>	<i>object</i>	
<i>prt_id</i>	<i>int</i>	Terá os valores 1, 2 e 3, cada um tendo um nome associado a ele.
<i>prt_name</i>	<i>string</i>	Terá os valores “Baixa”, “Média” e “Alta”, cada uma tendo um <i>id</i> associado a ele.
<i>status</i>	<i>object</i>	
<i>stt_id</i>	<i>int</i>	Terá os valores 1, 2 e 3, cada um tendo um nome associado a ele.
<i>stt_name</i>	<i>string</i>	Terá os valores “A fazer”, “Fazendo” e “Feito”, cada um tendo um <i>id</i> associado a ele.

Tabela que representa os nome, seus tipos e algumas observações, dos atributos do documento *task*.

5.2 DIAGRAMA DE CLASSES

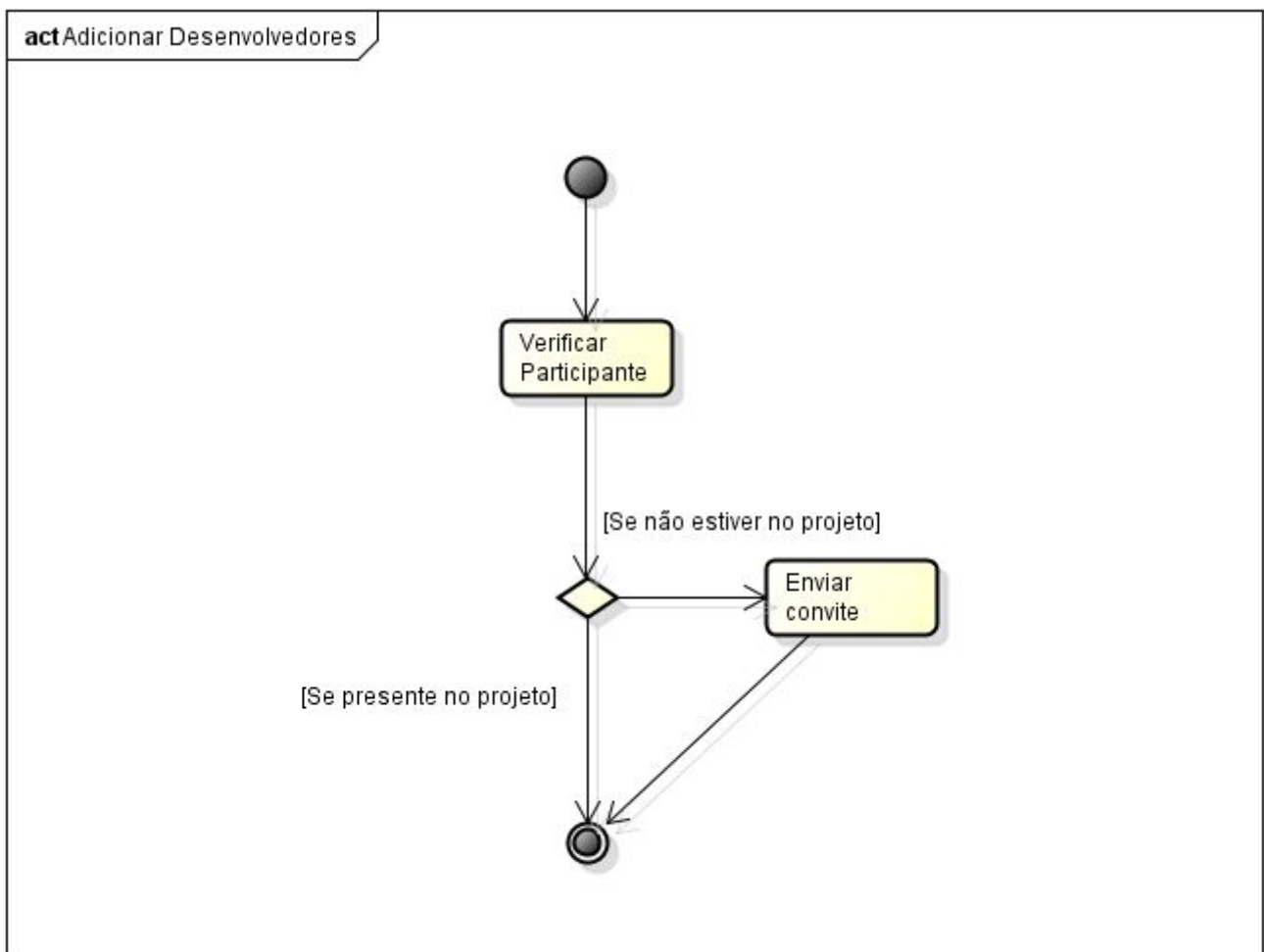


Este diagrama representa as Classes do sistema (objetos que compõem o sistema), as classes são representadas como retângulos, e as que possuem o nome em itálico são interfaces de interação. O diagrama também apresenta os atributos de cada classe (conjunto de dados que compõem a classe), atributos são representados na segunda seção da classe (sendo a primeira o nome da classe). O diagrama também aponta os métodos das classes (funções que a classe deve cumprir), os métodos são representados na terceira seção da classe. As ligações entre as classes determina como será o seu relacionamento.

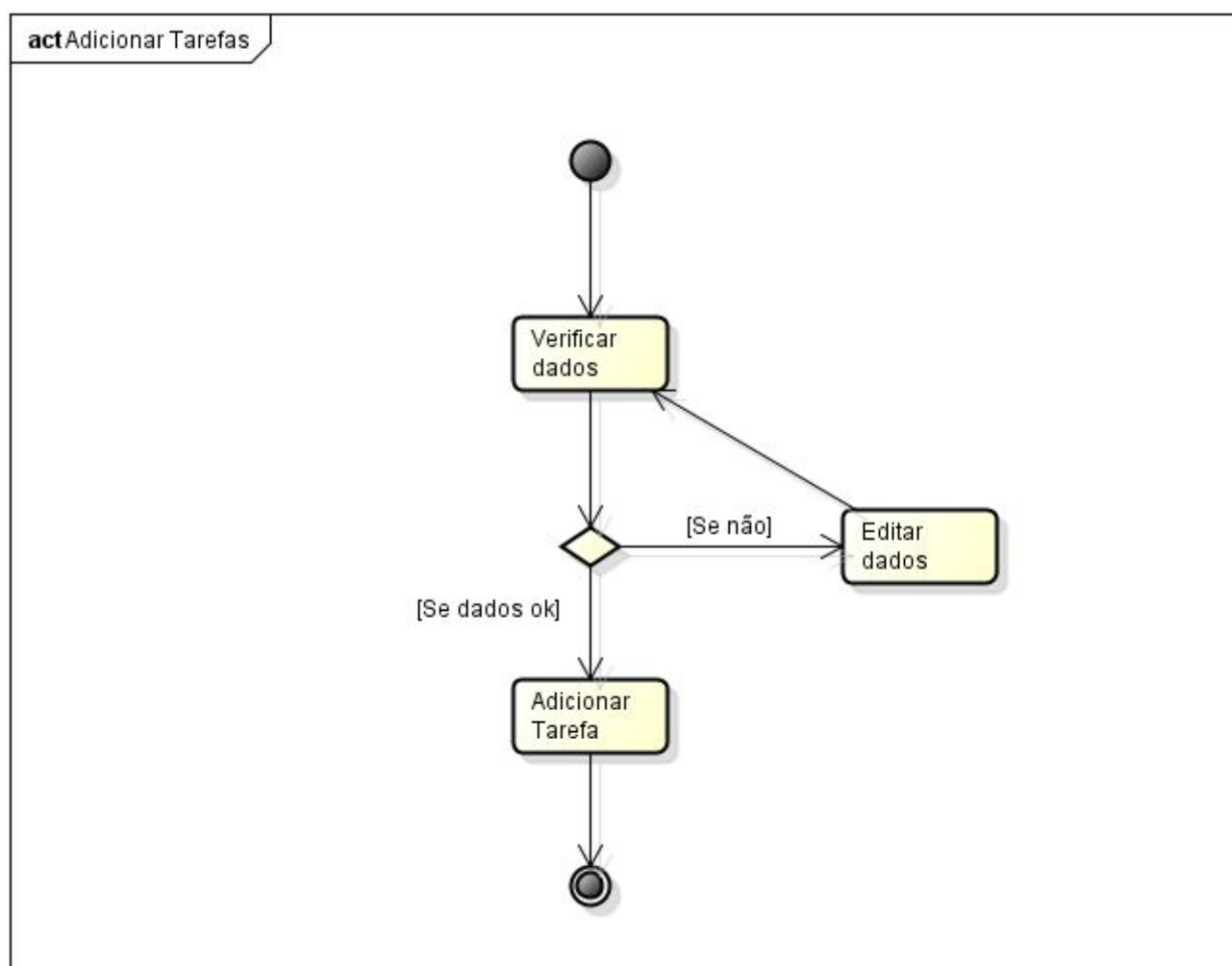
5.3 DIAGRAMA DE ATIVIDADES

O diagrama de atividades detalha as atividades (balões) do diagrama do caso de uso. O diagrama de atividades descreve como segue o fluxo das atividades.

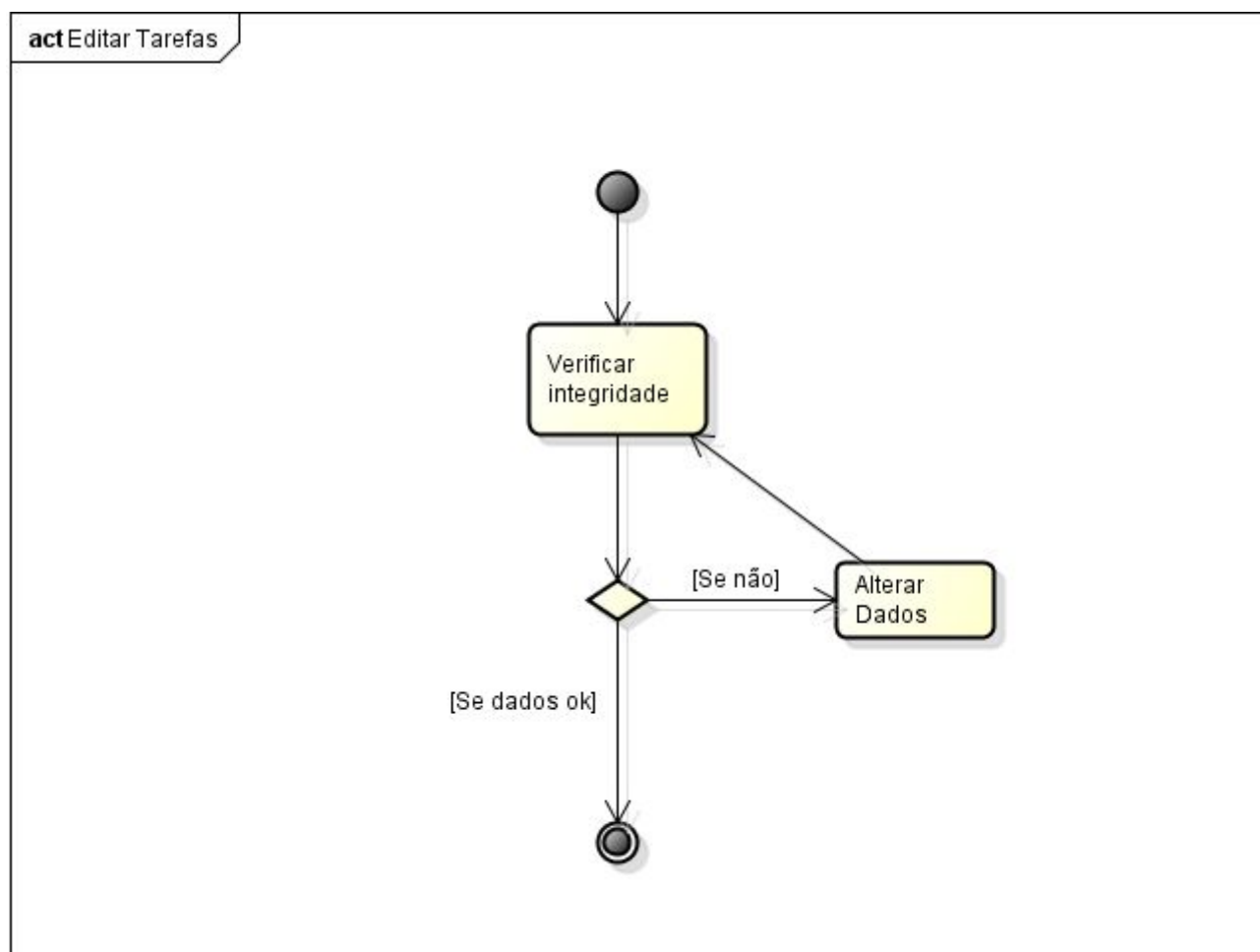
Adicionar desenvolvedores:



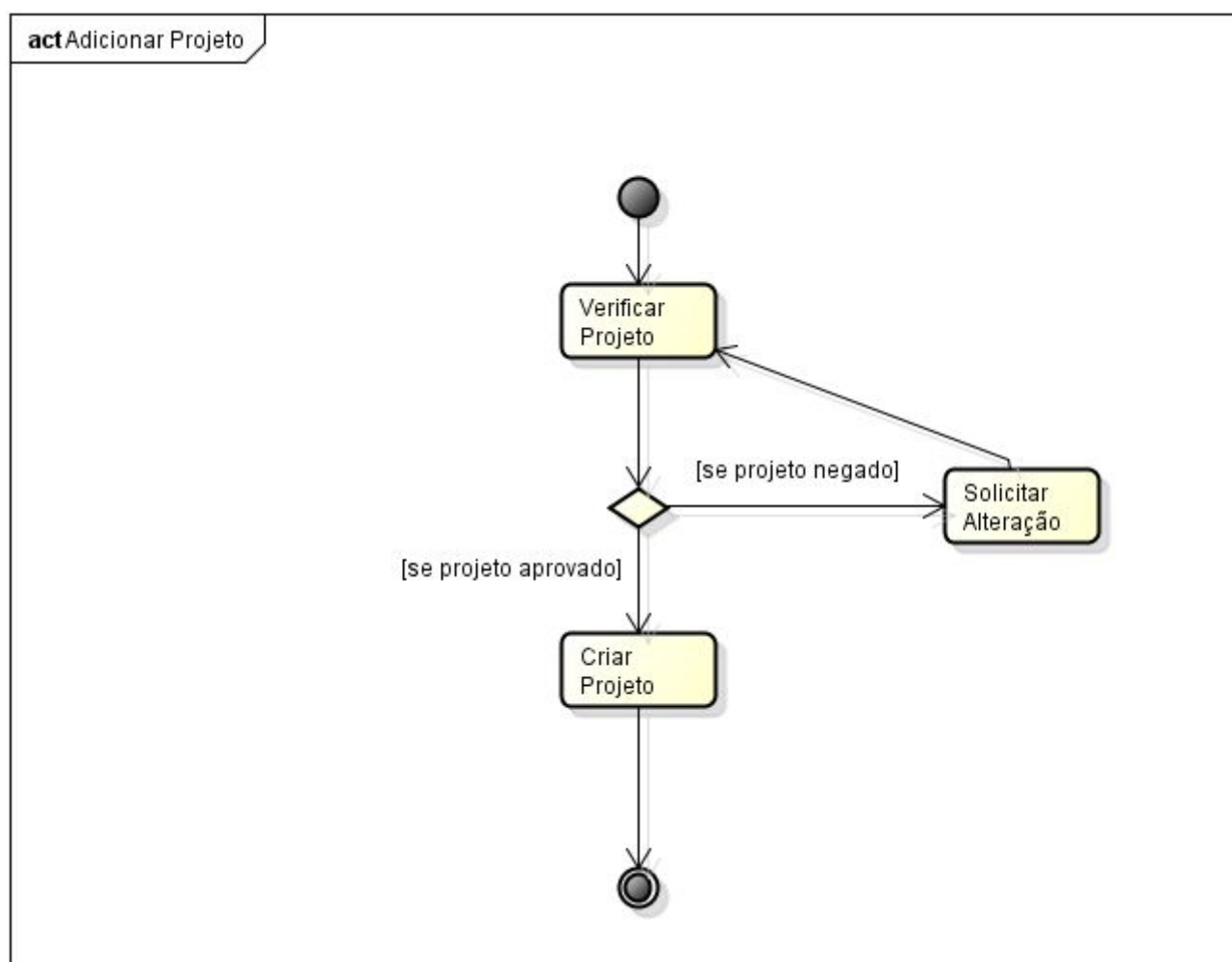
Este diagrama de atividades detalha a ação adicionar desenvolvedores.

Adicionar tarefas:

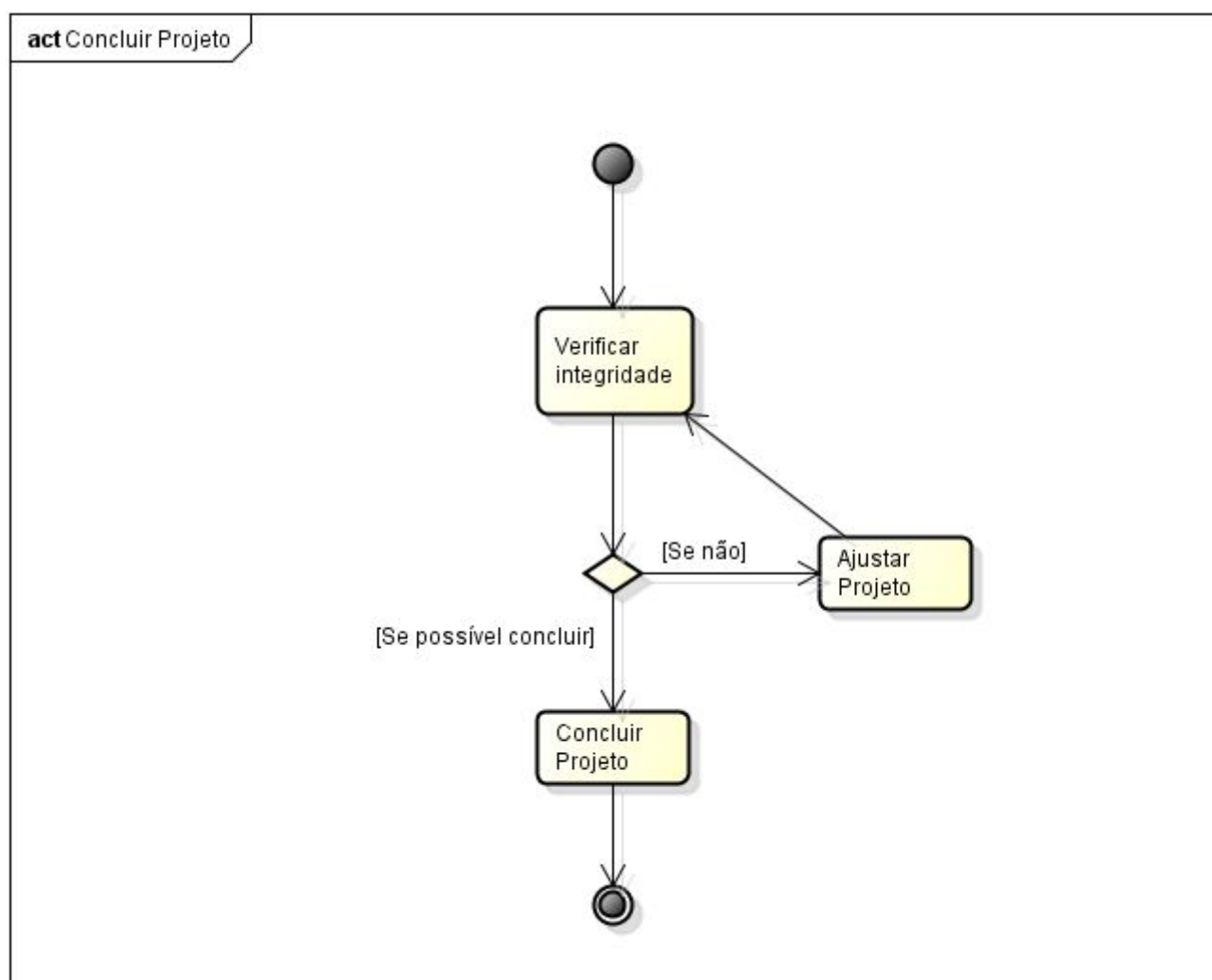
Este diagrama de atividades detalha a ação adicionar tarefas.

Editar tarefas:

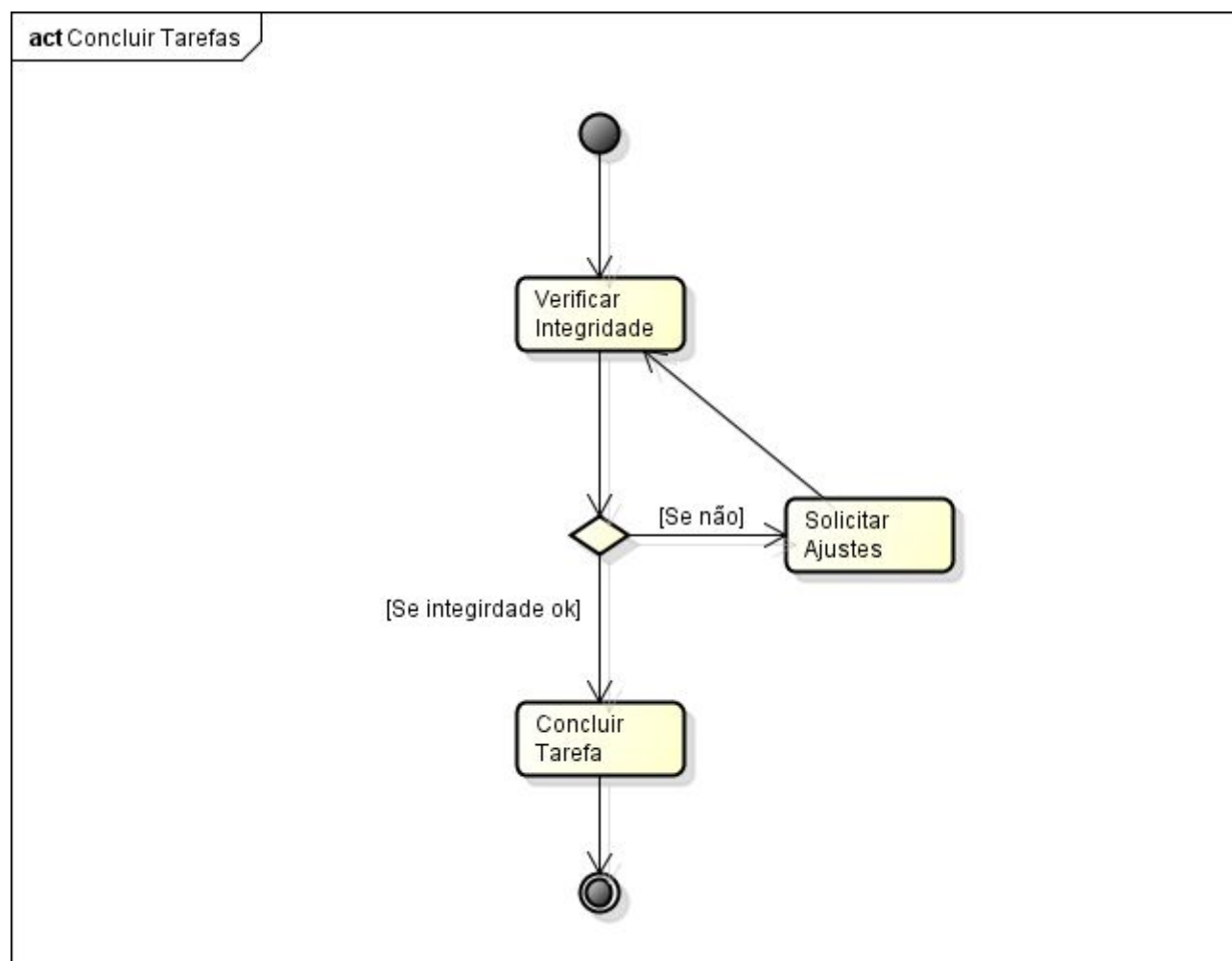
Este diagrama de atividades detalha a ação editar tarefas.

Adicionar projeto:

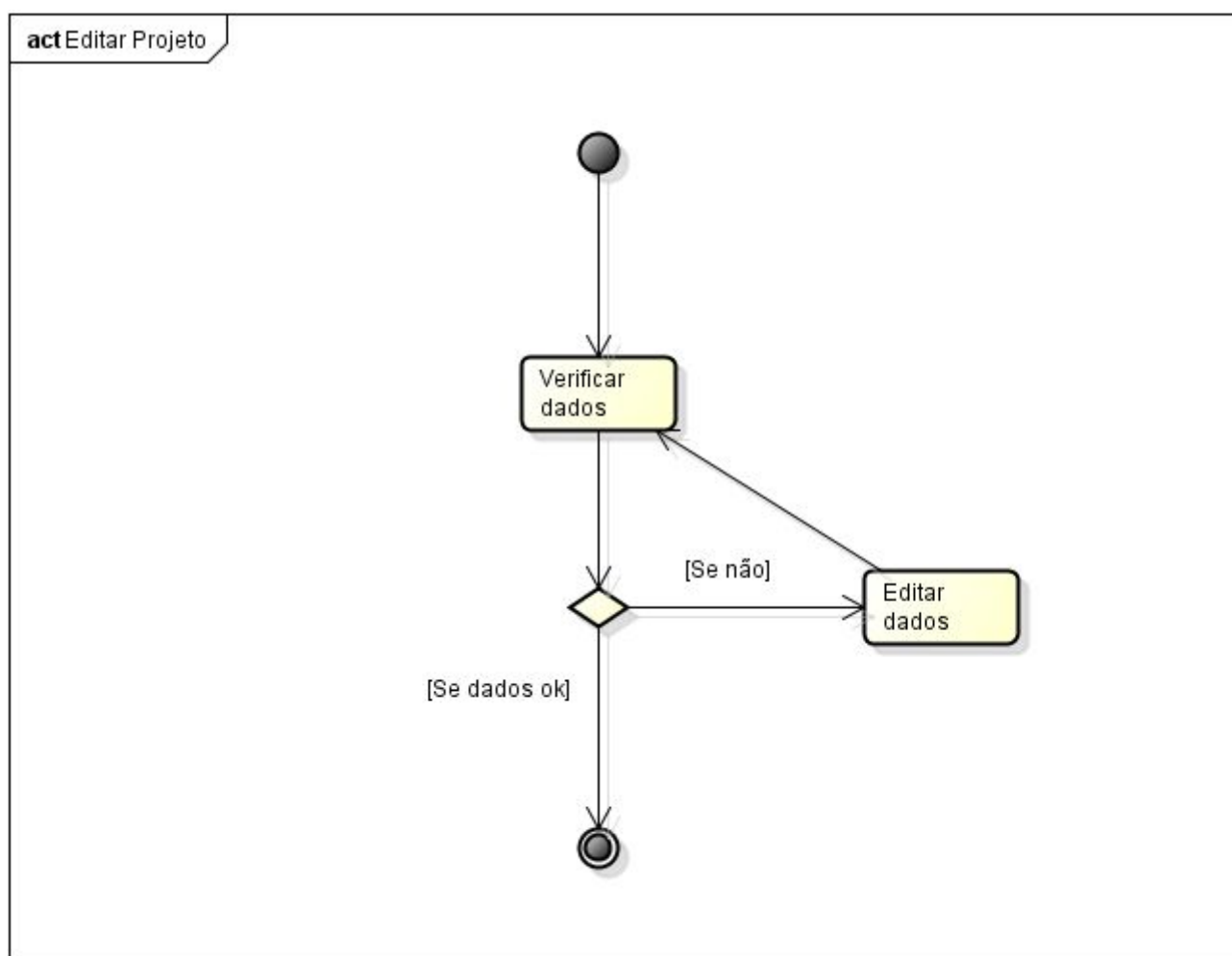
Este diagrama de atividades detalha a ação adicionar projeto.

Concluir projeto:

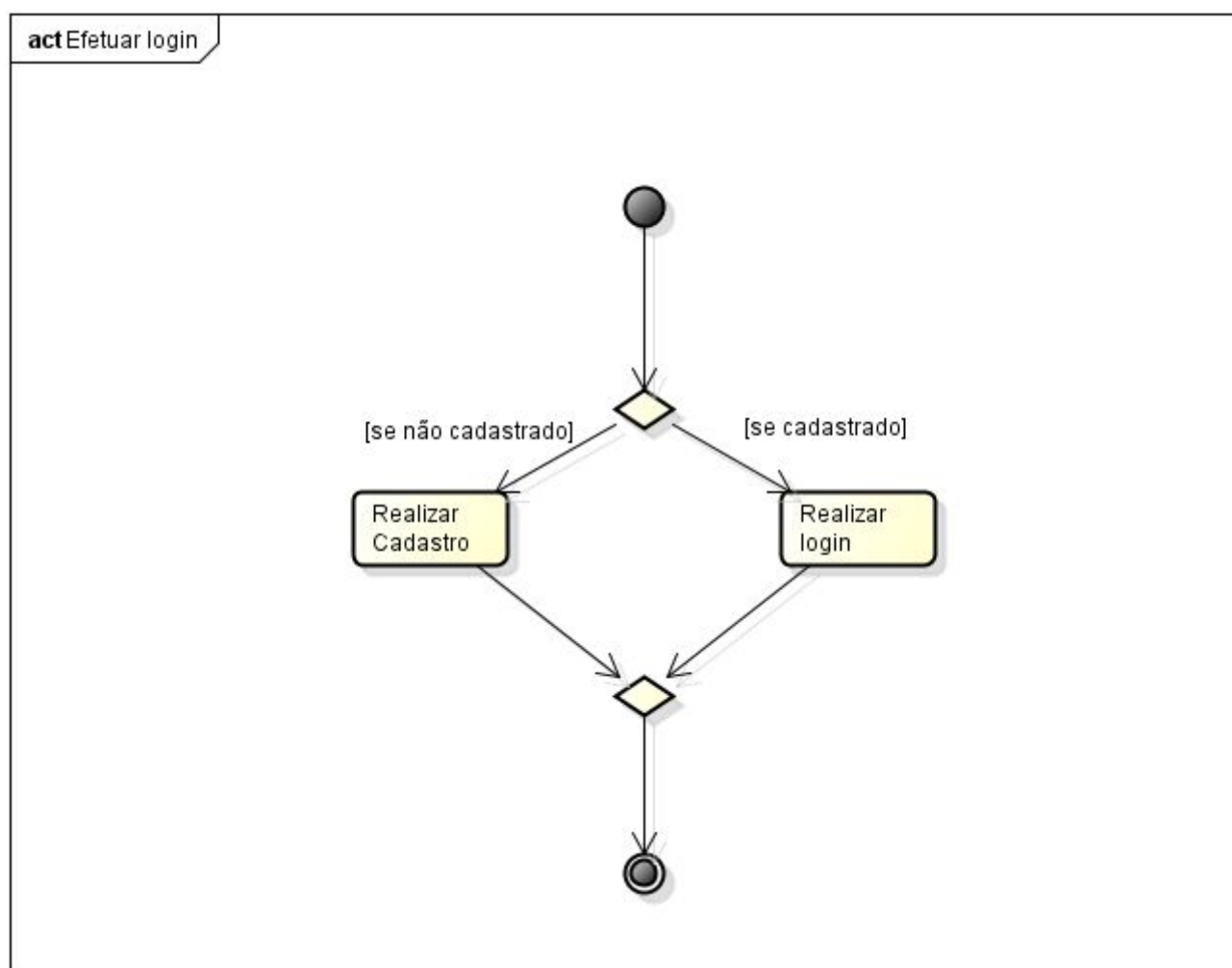
Este diagrama de atividades detalha a ação concluir projeto.

Concluir tarefas:

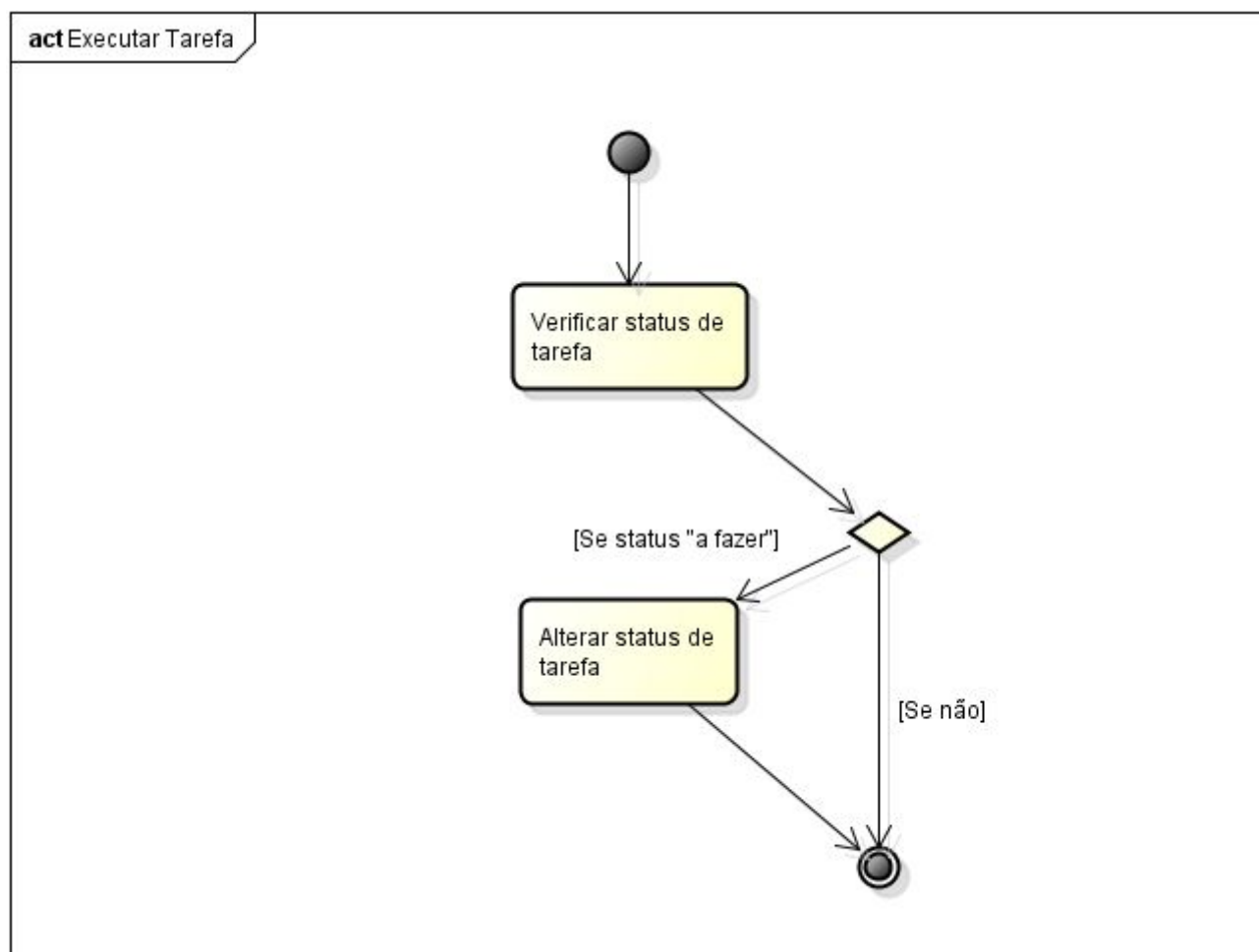
Este diagrama de atividades detalha a ação concluir tarefas.

Editar projeto:

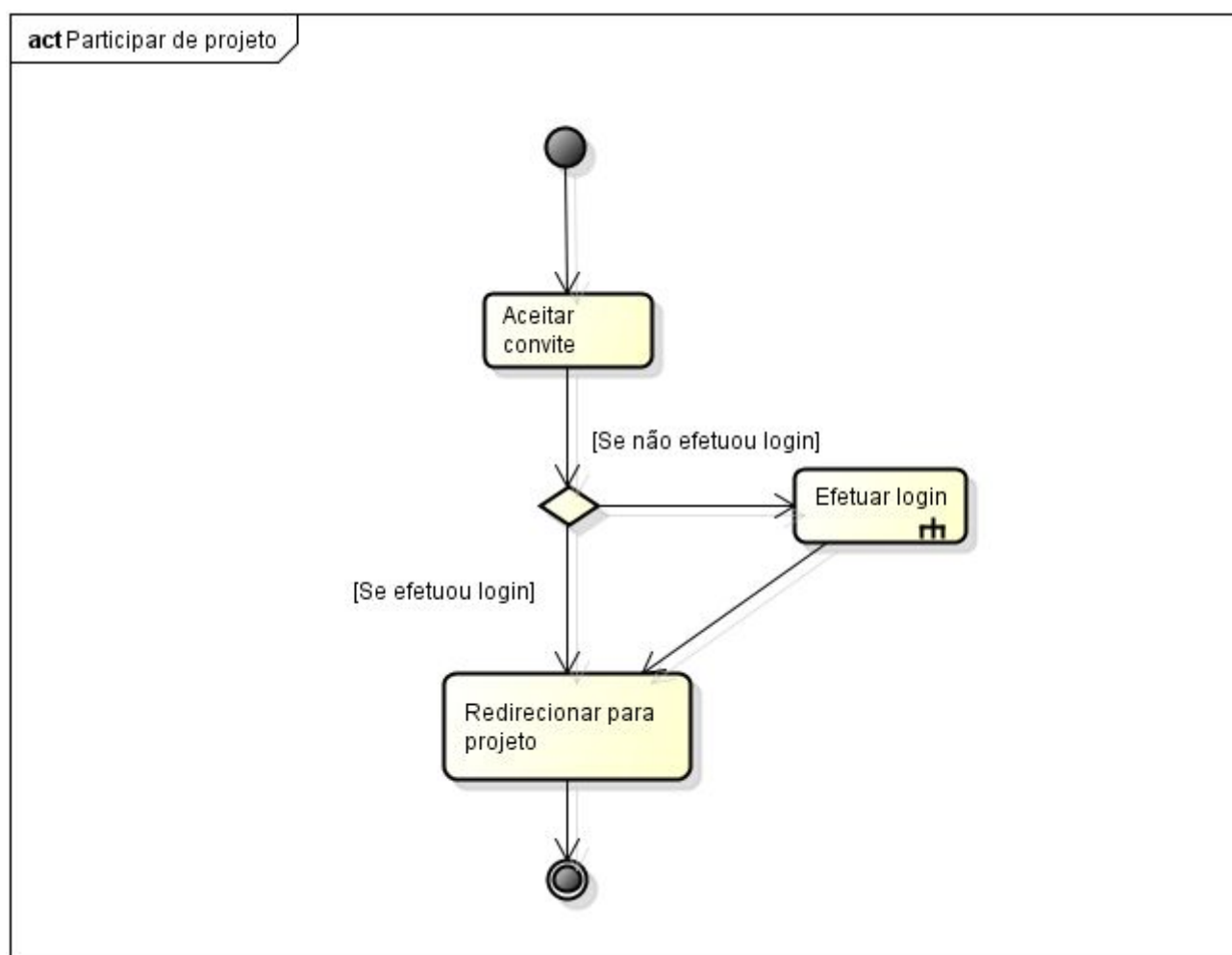
Este diagrama de atividades detalha a ação editar projeto.

Efetuar login:

Este diagrama de atividades detalha a ação efetuar login.

Executar tarefa:

Este diagrama de atividades detalha a ação executar tarefa.

Participar de projeto:

Este diagrama de atividades detalha a ação participar de projeto.

6 BIBLIOGRAFIA

BARCAUI, André B. (2008). **Por que gerenciar projetos?**,
<https://exame.abril.com.br/carreira/por-que-gerenciar-projetos-m0042508/#>

LEITE, Jair C. (2000). **Notas de aula de Engenharia de Software**,
<https://www.dimap.ufrn.br/~jair/ES/c1.html>

VASCONCELOS, Eduardo M. (). **BASE, uma metodologia ágil voltada para pequenos projetos**,
<http://docplayer.com.br/17052951-Base-uma-metodologia-agil-voltada-para-pequenos-projetos.html>

(2016). **Primeiros passos com Express em Node.js**,
<http://nodebr.com/primeiros-passos-com-express-em-node-js/>

(2016). **O que é Node.js?**, <http://nodebr.com/o-que-e-node-js/>

(2017). **Data model design**,
<https://docs.mongodb.com/manual/core/data-model-design/#data-modeling-referencing>