

Universidade Federal do Rio de Janeiro

Escola Politécnica

Departamento de Eletrônica e de Computação

**Sistema de Gerenciamento de Tarefas para usuários de**  
***Scrum***

Autor:

---

Andreza Cristina da Silva

Orientador:

---

Prof. Antônio Cláudio Gómez Sousa

Examinador:

---

Prof Sergio Palma da Justa Medeiros, D. Sc.

Examinador:

---

Prof. Marcelo Luiz Drumond Lanza

DEL

Fevereiro de 2011

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

Escola Politécnica - Departamento de Eletrônica e de Computação  
Centro de Tecnologia, bloco H, sala H-217, Cidade Universitária  
Rio de Janeiro - RJ CEP 21949-900

Este exemplar é de propriedade da Universidade Federal do Rio de Janeiro, que poderá incluí-lo em base de dados, armazenar em computador, microfilmear ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do(s) autor(es) e do(s) orientador(es).

## DEDICATÓRIA

Dedico este Projeto de Graduação às duas pessoas mais importantes da minha vida: meu pai e minha mãe, que fizeram sacrifícios durante os 3 anos do meu ensino médio para que eu pudesse ter a opção de estudar, segundo a minha opinião, na melhor universidade do Brasil.

## AGRADECIMENTO

Agradeço em primeiro lugar à minha maravilhosa família, que sempre me apoiou em quaisquer decisões que fiz, mesmo achando que eu pudesse estar errada. A liberdade de escolha que vocês me deram foi fundamental pra eu me tornar a pessoa que hoje eu sou. Quanto ao amor que vocês me deram nesses 24 anos, eu não tenho nem palavras. Mãe e Pai, obrigada por todo o apoio que vocês me deram. Kinha, obrigada por me aturar mesmo nos meus momentos de crise e estresse extremos, que não foram poucos nesses 5 anos Engenharia Eletrônica!

Impossível deixar de agradecer à família que eu escolhi pra mim: meus amigos. Sem eles eu não seria nada! Amigos, obrigada por tornarem o meu tempo muito mais divertido ao lado de vocês, por todas as vezes que eu precisei de vocês do meu lado e vocês estavam lá para não me deixar cair.

Não querendo ser redundante, mas já sendo, gostaria de agradecer a companhia dos meus queridos amigos-colegas de profissão, alguns que ficaram comigo desde o primeiro período até o final, outros que ficaram pelo caminho ou os que chegaram depois. Eu tenho quase certeza que se não fossem vocês, a missão de me tornar Engenheira Eletrônica e de Computação seria incrivelmente mais difícil.

Obrigada ao meu orientador António Cláudio por aceitar a tarefa de me orientar neste projeto e que foi sempre tão paciente, me dando a luz necessária sempre que tudo parecia escuro.

Não poderia também deixar de agradecer ao Professor José Manoel de Seixas e ao Laboratório de Processamento de Sinais, onde fui aluna de Iniciação Científica por quase 2 anos. Sem a experiência adquirida no LPS e, principalmente, no CERN, eu não seria nem de longe a pessoa e profissional que sou.

Ao coordenador do curso de Engenharia Eletrônica e de Computação, Professor Carlos José Ribas D'Avila, vai meu agradecimento por sempre estar disposto

a resolver os eventuais problemas, fossem eles causados pelo SIGA ou por maus entendidos meus.

Aos Professores Sergio Palma da Justa Medeiros e Marcelo Luiz Drumond Lanza, obrigada pelos conhecimentos passados em sala de aula (sem dúvida foram alguns dos melhores cursos que fiz ao longo desses 5 anos) e por fazerem parte da minha banca de Projeto de Graduação.

Ao Professor Edson Watanabe, do Programa de Engenharia Elétrica, pelo apoio que me foi dado recentemente.

Agradeço também a todos os professores da Escola Politécnica, em especial aos do DEL, que nós alunos, no desespero nosso de cada dia, nem sempre reconhecemos o valor que eles têm. Já estou com saudades de conviver com esses gênios diariamente.

Não poderia também deixar de agradecer aos meus queridos companheiros de trabalho na empresa Chemtech que, desde o primeiro dia de estágio, estiveram prontos para tirar quaisquer dúvidas que eu pudesse ter, muitas vezes me oferecendo ajuda antes mesmo de eu descobrir que precisava. E também por que não agradecer por fazer meu dia bem mais engraçados?

Por fim, gostaria de agradecer a incrível dificuldade do nosso curso de Engenharia Eletrônica e de Computação, que nos prepara para o mundo, pois depois de passar por isso, qualquer coisa fica fácil!

## RESUMO

Este trabalho apresenta a idealização, projeto e implementação do iPlan, um sistema voltado para usuários da metodologia ágil *Scrum*, cujo principal objetivo é auxiliar no gerenciamento de tarefas necessárias ao completo desenvolvimento de um ou mais projetos.

Através do iPlan é possível cadastrar usuários, equipes, projetos, tarefas, além de planejar quando e por quem estas serão executadas, tudo isto de acordo com a framework do *Scrum*.

Palavras-Chave: *Scrum*, iPlan, gerenciamento de tarefas, Engenharia de Software, Metodologia Ágil

## ABSTRACT

This paper presents the idealization, project and implementation of iPlan, a software especially developed for users of the agile methodology Scrum, which main goal is to help managing the demanded tasks to accomplish the development of one or more projects.

The software iPlan allows the register of users, teams, projects, tasks, besides planning when and by whom these will be executed, all according to the Scrum framework.

Key-words: Scrum, iPlan, task management, Software Engineering, Agile Methodology

## SIGLAS

UFRJ - Universidade Federal do Rio de Janeiro

PGPS - Plano Para o Gerenciamento de Projeto de Software

ERS - Especificação de Requisitos de Software

C#: Linguagem de programação orientada a objeto da Microsoft

BD: Banco de dados

SGBD: Sistema de Gerenciamento de Banco de Dados

L<sup>A</sup>T<sub>E</sub>X: Ferramenta de tipografia muito apropriada para textos técnicos e científicos;

MVC - *Model-View-Control*

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Tema . . . . .	1
1.2	Delimitação . . . . .	1
1.3	Justificativa . . . . .	2
1.4	Objetivos . . . . .	2
1.5	Metodologia . . . . .	2
1.6	Descrição . . . . .	3
<b>2</b>	<b>Scrum</b>	<b>5</b>
2.1	Introdução . . . . .	5
2.1.1	Origem . . . . .	6
2.1.2	Principais Características . . . . .	6
2.1.3	Os Valores . . . . .	7
2.2	Funcionamento do <i>Scrum</i> . . . . .	7
2.3	A Framework do Scrum . . . . .	9
2.3.1	Papéis . . . . .	9
2.3.2	Artefatos . . . . .	10
2.3.3	Reuniões . . . . .	15
<b>3</b>	<b>Planejamento</b>	<b>19</b>
3.1	Apresentação . . . . .	19
3.1.1	Sumário do Projeto . . . . .	19
3.1.2	Evolução do Plano . . . . .	20
3.2	Referências . . . . .	21
3.3	Definições . . . . .	21

3.4	Organização do Projeto . . . . .	22
3.4.1	Interfaces Externas . . . . .	22
3.4.2	Estrutura Interna . . . . .	22
3.4.3	Papéis e Responsabilidades . . . . .	22
3.5	Processos de Gerenciamento . . . . .	22
3.5.1	Partida do Projeto . . . . .	22
3.5.2	Plano de Trabalho . . . . .	24
3.5.3	Planos de Controle . . . . .	25
3.5.4	Plano de Gerenciamento de Riscos . . . . .	26
3.5.5	Plano de Encerramento . . . . .	28
3.6	Processos Técnicos . . . . .	28
3.6.1	Modelo dos Processos . . . . .	28
3.6.2	Métodos, Ferramentas e Técnicas . . . . .	28
3.6.3	Infra-estrutura . . . . .	29
3.6.4	Plano para Aceitação do Produto . . . . .	29
3.7	Planos para os processos de Suporte . . . . .	29
3.7.1	Gerenciamento de Configuração . . . . .	29
3.7.2	Plano de Verificação e de Validação . . . . .	29
3.7.3	Documentação . . . . .	29
3.7.4	Plano para Assegurar a Qualidade . . . . .	30
3.7.5	Revisões e Auditorias . . . . .	30
3.7.6	Plano para a Resolução de Problemas . . . . .	30
3.7.7	Gerenciamento de Subcontratações . . . . .	31
3.7.8	Plano de Aperfeiçoamento . . . . .	31
3.8	Planos Adicionais . . . . .	31
<b>4</b>	<b>Especificação de Requisitos de Software</b>	<b>32</b>
4.1	Introdução . . . . .	32
4.1.1	Finalidade . . . . .	32
4.1.2	Escopo . . . . .	32
4.1.3	Definições, Acrônimos e Abreviaturas . . . . .	33
4.1.4	Referências . . . . .	33
4.1.5	Resumo . . . . .	33

4.2	Descrição Geral . . . . .	33
4.2.1	Perspectiva do Produto . . . . .	33
4.2.2	Funções do Produto . . . . .	33
4.2.3	Características do Usuário . . . . .	35
4.2.4	Restrições . . . . .	35
4.2.5	Pressupostos e Dependências . . . . .	35
4.2.6	Postergar Requisitos . . . . .	35
4.3	Requisitos Específicos . . . . .	36
4.3.1	Interfaces Externas . . . . .	36
4.3.2	Requisitos Funcionais . . . . .	36
4.3.3	Requisitos de Desempenho . . . . .	53
4.3.4	Restrições de Projeto . . . . .	53
4.3.5	Atributos . . . . .	53
4.3.6	Outros Requisitos . . . . .	54
<b>5</b>	<b>Descrição de Projeto de Software</b>	<b>55</b>
5.1	Introdução . . . . .	55
5.1.1	Finalidade . . . . .	55
5.1.2	Escopo . . . . .	55
5.1.3	Definições, Acrônimos e Abreviaturas . . . . .	55
5.2	Referências . . . . .	56
5.3	Decomposição . . . . .	56
5.3.1	Decomposição em Módulos . . . . .	56
5.3.2	Decomposição em Processos Concorrentes . . . . .	61
5.3.3	Decomposição de Dados . . . . .	61
5.4	Descrição das Dependências . . . . .	66
5.4.1	Dependências entre módulos . . . . .	66
5.4.2	Dependências entre processos . . . . .	68
5.4.3	Dependências entre dados . . . . .	68
5.5	Descrição das Interfaces . . . . .	68
5.6	Projeto Detalhado . . . . .	68
5.6.1	Projeto Detalhado dos Módulos . . . . .	68
5.6.2	Projeto Detalhado das Entidades de Dados . . . . .	68

<b>6</b>	<b>Plano de Testes</b>	<b>74</b>
6.1	Introdução . . . . .	74
6.1.1	Identificador . . . . .	74
6.1.2	Finalidade . . . . .	74
6.1.3	Referências . . . . .	74
6.2	Descrição Geral . . . . .	74
6.2.1	Itens . . . . .	74
6.2.2	Requisitos . . . . .	75
6.2.3	Visão Geral . . . . .	75
6.2.4	Suspensão ou Conclusão . . . . .	75
6.2.5	Ambiente . . . . .	75
6.2.6	Tarefas e Cronograma . . . . .	76
6.2.7	Riscos e Gerenciamento . . . . .	76
6.3	Especificação dos Testes . . . . .	76
6.3.1	Especificação 1 . . . . .	76
6.3.2	Especificação 2 . . . . .	80
6.3.3	Especificação 3 . . . . .	81
6.4	Casos de Teste . . . . .	84
6.4.1	Caso de teste 1 . . . . .	84
6.4.2	Caso de teste 2 . . . . .	84
6.4.3	Caso de teste 3 . . . . .	88
6.5	Procedimentos de Teste . . . . .	88
6.5.1	Procedimento 1 . . . . .	88
6.5.2	Procedimento 2 . . . . .	89
6.5.3	Procedimento 3 . . . . .	90
<b>7</b>	<b>Manual Do Usuário</b>	<b>95</b>
7.1	Introdução . . . . .	95
7.2	Descrição . . . . .	95
7.3	Utilização do iPlan . . . . .	95
7.3.1	Acesso . . . . .	95
7.3.2	Página Principal . . . . .	96
7.3.3	Usuários . . . . .	96

7.3.4	Projetos . . . . .	100
7.3.5	Grupos . . . . .	103
7.3.6	<i>Sprints</i> . . . . .	106
7.3.7	Tarefas . . . . .	106
7.3.8	Planejar <i>Sprint</i> . . . . .	109
<b>8</b>	<b>Conclusão</b>	<b>112</b>
	<b>Bibliografia</b>	<b>114</b>

# Lista de Figuras

1.1	Modelo de Processos: Cascata - Linear Sequencial . . . . .	3
2.1	Formação Scrum no Rugby. . . . .	6
2.2	Funcionamento Básico do <i>Scrum</i> , por Mountain Goat Software. . . . .	8
2.3	Amostra de <i>backlog</i> do projeto. . . . .	11
2.4	<i>Burndown Chart</i> de <i>sprint</i> bem planejada. . . . .	12
2.5	<i>Burndown Chart</i> de <i>sprint</i> subestimada. . . . .	13
2.6	<i>Burndown Chart</i> de <i>sprint</i> superestimada. . . . .	13
2.7	Exemplo de um Quadro de Tarefas. . . . .	14
2.8	Algumas cartas do baralho utilizado no Planning Poker. . . . .	16
2.9	Equipe no momento da reunião diária. . . . .	17
3.1	Modelo Cascata Linear-Sequencial . . . . .	20
3.2	Cronograma de atividades do projeto . . . . .	21
3.3	Backlog do projeto. . . . .	23
4.1	Diagrama de Casos de Uso para Administradores do Sistema. . . . .	37
4.2	Diagrama de Casos de Uso para <i>Product Owners</i> . . . . .	38
4.3	Diagrama de Casos de Uso para <i>Scrum Masters</i> . . . . .	39
4.4	Diagrama de Casos de Uso para membros da equipe desenvolvedora. . . . .	40
4.5	Diagrama de Classes. . . . .	53
5.1	Estrutura de <i>Forms</i> e <i>UserControls</i> . . . . .	57
5.2	Arquitetura do Módulo de Acesso. . . . .	58
5.3	Arquitetura do Módulo de Equipe. . . . .	59
5.4	Arquitetura do Módulo de Projeto. . . . .	60
5.5	Arquitetura do Módulo de <i>Backlog</i> . . . . .	61

5.6	Diagrama de classes. . . . .	62
5.7	Diagrama de tarefa para incluir uma nova tarefa. . . . .	69
5.8	Diagrama de sequência para incluir uma nova tarefa. . . . .	70
5.9	Modelo Lógico do Banco de Dados para o iPlan. . . . .	71
5.10	Modelo Físico do Banco de Dados para o iPlan. . . . .	71
7.1	Página principal do iPlan. . . . .	96
7.2	Modulo de Usuários. . . . .	97
7.3	Usuários listados. . . . .	98
7.4	Formulário para a adição de novo usuário. . . . .	98
7.5	Formulário para a edição de usuário cadastrado. . . . .	99
7.6	Procedimento necessário para desfazer exclusão de usuário. . . . .	100
7.7	Módulo de Projetos. . . . .	101
7.8	Busca de Projetos. . . . .	102
7.9	Formulário a ser preenchido para adicionar um projeto. . . . .	102
7.10	Módulo de grupos. . . . .	103
7.11	Exemplo de Resultado de Busca de Grupos. . . . .	104
7.12	Linha em branco para adição de novo grupo. . . . .	105
7.13	Módulo de <i>Sprints</i> . . . . .	106
7.14	Módulo de <i>Tarefas</i> . . . . .	107
7.15	Exemplo de busca de tarefas com respectivos vínculos. . . . .	108
7.16	Adicionar Tarefa a <i>Sprints</i> . . . . .	109
7.17	Exemplo de tarefas planejadas para grupo e <i>sprint</i> exibidos. . . . .	110
7.18	Opção de prioridades para que o usuário selecione a desejada. . . . .	111

# Lista de Tabelas

3.1	Datas de início de fim de cada <i>sprint</i> . . . . .	24
3.2	Riscos previstos para o projeto. . . . .	27
5.1	Atributos da Classe <i>User</i> . . . . .	63
5.2	Atributos da Classe <i>Group</i> . . . . .	63
5.3	Atributos da Classe <i>Position</i> . . . . .	64
5.4	Atributos da Classe <i>Group Member</i> . . . . .	64
5.5	Atributos da Classe <i>Sprint</i> . . . . .	64
5.6	Atributos da Classe <i>Project</i> . . . . .	65
5.7	Atributos da Classe <i>Status</i> . . . . .	65
5.8	Atributos da Classe <i>Priority</i> . . . . .	66
5.9	Atributos da Classe <i>Effort</i> . . . . .	66
5.10	Atributos da Classe <i>Task</i> . . . . .	67
5.11	Atributos da Classe <i>Sprint Task</i> . . . . .	67
6.1	Cronograma. . . . .	76
6.2	Entradas e Saídas Esperadas do Caso de Teste 1. . . . .	85
6.3	Entradas e Saídas Esperadas do Caso de Teste 1. . . . .	86
6.4	Entradas e Saídas Esperadas do Caso de Teste 1. . . . .	87
6.5	Entradas e Saídas Esperadas do Caso de Teste 3. . . . .	91
6.6	Entradas e Saídas Esperadas do Caso de Teste 3. . . . .	92
6.7	Entradas e Saídas Esperadas do Caso de Teste 3. . . . .	93
6.8	Entradas e Saídas Esperadas do Caso de Teste 3. . . . .	94

# Capítulo 1

## Introdução

### 1.1 Tema

O tema do trabalho é o desenvolvimento de um sistema gerenciador de tarefas do *Scrum*, uma metodologia ágil para o gerenciamento de projetos. Pretende-se desenvolver um sistema que auxilie os membros de uma equipe a planejar e acompanhar as tarefas a serem desenvolvidas, o momento em que começarão a ser executadas, o prazo em que elas devem ser implementadas e o esforço utilizado por cada membro da equipe em sua execução.

### 1.2 Delimitação

O objeto de estudo é o gerenciamento das atividades a serem realizadas no desenvolvimento de um *software* quando há utilização do *Scrum*.

Neste sentido o sistema desenvolvido deverá permitir ao usuário cadastrar tarefas a serem realizadas durante a elaboração de um produto, associá-las ao esforço gasto para sua implementação, ao ciclo em que deverá ser realizada, sua atual situação, ou seja, deve fornecer ao primeiro um meio de gerenciar o desenvolvimento de um produto.

## 1.3 Justificativa

Por ser fácil de ser compreendido, não necessitar de treinamentos extensos e apresentar produtividade maior, quando se comparado a métodos tradicionais, o uso do *Scrum* vem crescendo bastante como metodologia empregada no desenvolvimento de software dentro de empresas. Além disso o emprego do *Scrum* acaba por unir a equipe em torno de um objetivo comum, no caso o desenvolvimento de um produto.

Contudo, para ser realmente eficiente, é preciso que as atividades que serão realizadas a cada *sprint* ao longo do desenvolvimento do produto final, sejam bem planejadas, que as devidas prioridades sejam atribuídas, e respeitadas, de forma que haja um desenvolvimento gradual coerente, para que ao fim de cada *sprint* um novo produto (parcial) possa ser lançado.

Atualmente as tarefas de um projeto que utiliza *Scrum* são geralmente gerenciadas utilizando planilhas de Excel ou similar, o que dificulta o compartilhamento de informações pela equipe, além de tais planilhas não serem otimizadas para o uso do *Scrum*. Nesse sentido, o presente projeto representa um *software* livre para gerenciamento de tarefas para projetos que utilizam esta metodologia.

## 1.4 Objetivos

O objetivo geral é desenvolver um sistema que possa armazenar as atividades a serem realizadas ao longo de um projeto, relacioná-las às *sprints*, para assim auxiliar o *scrum master* a planejar, acompanhar, analisar as tarefas realizadas no desenvolvimento de um sistema.

## 1.5 Metodologia

No desenvolvimento do *software*, será utilizado o modelo de processos em cascata, como mostrado na figura 1.1. A cada etapa do modelo serão geradas documentações tais como planejamento, especificação de requisitos, diagrama de casos de uso, diagrama de classe e plano de testes.

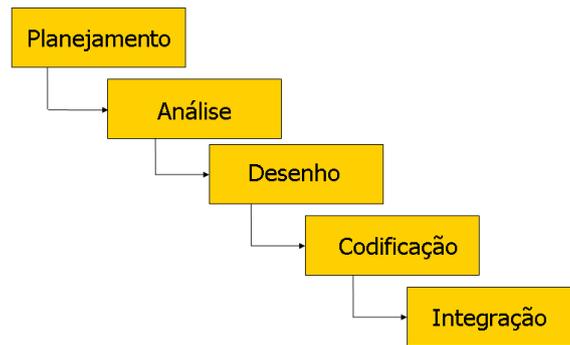


Figura 1.1: Modelo de Processos: Cascata - Linear Sequencial

A linguagem utilizada na codificação será C# e a programação, orientada a objeto. Devido a isso, a análise utilizada será a Análise Orientada a Objetos.

Já para o armazenamento de dados, será feito em um banco de dados relacional, e será utilizado *SQL Server*.

## 1.6 Descrição

No capítulo 2 será feita uma breve introdução sobre a metodologia ágil *Scrum*, onde serão apresentados ao leitor suas principais características, artefatos, origem e foco principal.

Em seguida, no capítulo 3, será apresentado o planejamento realizado para que o sistema pudesse ser desenvolvido. Neste, serão apresentados o cronograma detalhado do projeto, medidas de gerenciamento de risco, entre outros.

Os requisitos definidos para o software são apresentados no capítulo 4. Nele constarão os diagramas de casos de uso para cada grupo de usuário, bem como os casos de uso detalhados e o diagrama de classes.

Um maior detalhamento do projeto será especificado no capítulo 5, onde constarão seu modelo de banco de dados, arquitetura de cada módulo definido, as classes reais e seus principais métodos.

No capítulo seguinte, serão abordados os testes planejados para validar o que foi desenvolvido.

Em seguida estão detalhadas todas as funcionalidades e sua respectivas interfaces, no manual do usuário, para auxiliar o usuário na utilização do sistema.

Por fim, o trabalho é concluído no capítulo 8, onde são abordados os resultados atingidos com seu desenvolvimento, sua importância na utilização do *Scrum* e os planos para o futuro.

# Capítulo 2

## Scrum

### 2.1 Introdução

*Scrum* é uma metodologia ágil geralmente, porém não exclusivamente, utilizada para o desenvolvimento de software.

Devido à atual frequente mudança de requisitos de projetos e exigência de entrega em curtos prazos, faz-se cada vez mais necessária a utilização de processos ágeis de desenvolvimento. Neste sentido, o *Scrum* tem como princípios [1]:

- Criar um ambiente favorável a mudanças de requisitos, mesmo durante o desenvolvimento do produto e, conseqüentemente proporcionar entregas mais rápidas do produto ao cliente
- Reforçar o planejamento constante do projeto, evitando portanto riscos
- Priorizar ao máximo a satisfação do cliente

Conforme já citado, apesar de ser bastante utilizado no desenvolvimento de software, o *Scrum* pode ser utilizado em qualquer situação em que pessoas necessitem trabalhar em equipe, desde grandes projetos de engenharia à organização de pequenos eventos.

### 2.1.1 Origem

Originalmente, o *Scrum* foi concebido como um estilo de gerenciamento de projetos em empresas de fabricação de automóveis e produtos de consumos, por Takeuchi e Nonaka [2], que notaram que melhores resultados eram obtidos ao utilizar equipes menores e multidisciplinares.

Desta percepção surgiu o nome *Scrum*, uma analogia com a formação *Scrum* do jogo de Rugby<sup>1</sup>, quando os jogadores de ambos os times de uma partida se unem para recolocar a bola em jogo. Nesse momento, oito jogadores de cada equipe posicionam-se uns contra os outros, conforme mostra a figura 2.1. Após um jogador da equipe que não cometeu a penalidade lançar a bola no espaço entre os jogadores, estes devem tentar, com os pés, recuperar a bola tendo, assim, que trabalhar em equipe para alcançar um objetivo comum [3].



Figura 2.1: Formação Scrum no Rugby.

### 2.1.2 Principais Características

Mais do que uma metodologia, *Scrum* é uma framework, o que faz com que boa parte das decisões de como executar tarefas ou resolver problemas fique a critério

---

<sup>1</sup>Esporte coletivo de intenso contato físico semelhante ao futebol, praticado com uma bola oval, cujo objetivo é marcar o maior número de pontos

da equipe desenvolvedora, já que geralmente esta sabe melhor como lidar com os problemas que está enfrentando.

Com isso, é possível notar uma das características mais fortes no uso do *Scrum*, que é a existência de uma equipe autogerenciável. Como veremos adiante, apesar de existirem papéis pre-definidos na equipe, esta deve ser disciplinada e multidisciplinar, não necessitando de um líder.

Um bom diferencial em relação ao método tradicional de desenvolvimento é o rápido progresso do produto gerado, pois uma nova versão, contendo novas funcionalidades a cada ciclo.

Ainda, deve ser destacada a quantidade de reuniões no *Scrum*. Há quatro tipos de reuniões que devem acontecer de tempos em tempos, sendo elas: reunião diária, reunião de planejamento de *sprint*, reunião de revisão de *sprint* e reunião de *sprint*. Cada uma dessas será abordada mais adiante.

### 2.1.3 Os Valores

Segundo [4], os principais valores das metodologias ágeis devem ser:

- Indivíduos e interações ao invés de processos e ferramentas
- Software funcional ao invés de uma documentação abrangente
- Participação do cliente ao invés de uma negociação por contrato
- Adaptação a mudanças ao invés de seguir um plano

## 2.2 Funcionamento do *Scrum*

Antes de entrar em detalhes sobre o funcionamento do *Scrum* é necessário definir o que é uma *sprint*.

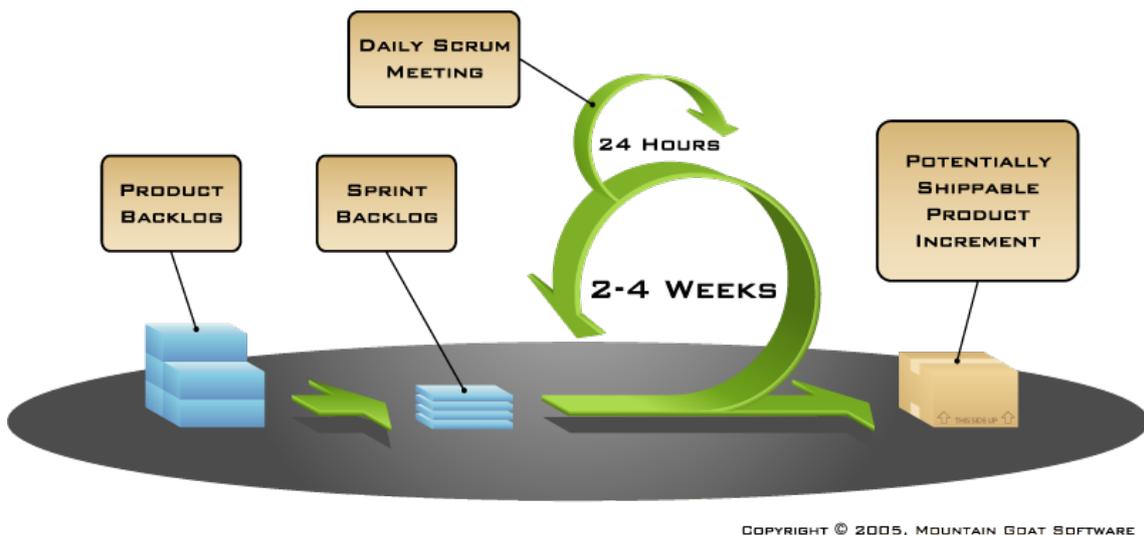
*Sprints* são iterações de períodos de tempo, com duração aproximada de duas a, no máximo, quatro semanas. Durante esse tempo, uma nova versão do produto desenvolvido é idealizada, implementada e testada.

Para um bom funcionamento do *Scrum*, conhecimento da capacidade da equipe e melhor planejamento de tarefas, é aconselhável que as *sprints* tenham duração fixa para cada equipe.

Tendo isso definido, podemos então dizer que, basicamente, o *Scrum* funciona da seguinte forma:

- todas as tarefas necessárias para o desenvolvimento de um produto ficam armazenadas e devidamente priorizadas num repositório chamado *Product Backlog*, ou Backlog do Produto.
- no início de cada *sprint* é realizada uma reunião, na qual a equipe seleciona as tarefas do Backlog do Produto que serão executadas da *sprint* que se inicia, formando o *Sprint Backlog*, ou Backlog da Sprint.
- todos os dias os membros da equipe devem atualizar uns aos outros sobre as tarefas em execução, de modo que todos estejam sempre cientes do que cada um está fazendo
- ao final de cada *sprint* uma nova versão do produto é lançada e enviada ao cliente.

A figura 2.2 ilustra o que foi descrito acima.



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

Figura 2.2: Funcionamento Básico do *Scrum*, por Mountain Goat Software.

## 2.3 A Framework do Scrum

De acordo com [5], a framework do *Scrum* é composta essencialmente por papéis, reuniões e artefatos que serão definidos e detalhados a seguir.

### 2.3.1 Papéis

Apesar de não ter funções específicas para cada membro integrante da equipe, há três papéis pre-definidos.

#### 2.3.1.1 *Product Owner*

O *Product Owner*, ou Dono do Produto, é o responsável por definir o produto a ser desenvolvido e a data em que deve estar pronto, além de cuidar da lucratividade daquele.

O *Product Owner* pode ser o cliente ou um membro da equipe com amplo conhecimento do que é desejado pelo primeiro. A ele cabe a tarefa de priorizar todas tarefas de acordo com seu valor de mercado, bem como alterar requisitos e prioridades, caso necessário, a cada *sprint*. Por fim, é o *Product Owner* que aceita ou rejeita o que foi produzido.

#### 2.3.1.2 ScrumMaster

O *ScrumMaster* é a figura que mais se aproxima do líder em uma equipe de *Scrum*. É dele a tarefa de garantir que os princípios e valores do *Scrum* sejam seguidos, além de eliminar possíveis impedimentos e proteger a equipe de interferências externas, lidando ele próprio com os problemas, garantindo assim que a equipe desenvolvedora esteja totalmente funcional e produtiva.

#### 2.3.1.3 Equipe Desenvolvedora

É a responsável por desenvolver o produto a ser entregue. É aconselhável que as equipes desenvolvedoras não sejam muito grandes, contendo geralmente de cinco a nove pessoas, de diferentes áreas (programadores, testadores, designers, ...).

Como citado anteriormente, deve ser autogerenciável, o que, neste caso, significa que os próprios membros selecionam as tarefas a serem executadas durante uma *sprint* e cada um tem a liberdade de escolher aquilo que vai fazer, dentre as tarefas selecionadas.

Com raras exceções (no caso de administradores de banco de dados, por exemplo), os membros desta equipe não devem dividir seu tempo com outras equipes e, a fim de evitar complicações no decorrer das *sprints*, é desejável que não haja mudança de membros no decorrer das mesmas.

## 2.3.2 Artefatos

### 2.3.2.1 Backlog do Produto

O *Backlog* do Produto, ou *Product Backlog* é essencialmente um repositório onde todas as tarefas necessárias ao completo desenvolvimento do produto ficam armazenadas.

Diferentemente do método tradicional, no *Scrum* geralmente todos os pré-requisitos do produto não são definidos quando o projeto se inicia. O que costuma acontecer é o armazenamento do maior número de tarefas possíveis para que o projeto possa ao menos ser iniciado e no decorrer do projeto, mais tarefas vão sendo adicionadas ou até mesmo removidas, conforme a demanda.

As tarefas armazenadas no *backlog* podem ser de diferentes naturezas, tanto técnicas, quanto relativas à coordenação, suporte, etc. A maneira com que aquele vai ser organizado cabe exclusivamente ao seu criador.

As tarefas contidas no *backlog* do produto devem estar priorizadas de acordo com seus respectivos valores de mercado ou sua necessidade ao bom desenvolvimento do projeto. Com isso, as tarefas mais importantes são realizadas antes das demais tarefas. Devido a isso, é natural que com o passar do tempo, algumas tarefas, antes menos prioritárias, tornem-se desnecessárias, e não há nenhum mal em retirá-las do *backlog*, ao passo que novas tarefas podem se fazer urgentes, sendo executadas antes de muitas tarefas já constantes no *backlog*.

A figura 2.3 representa uma amostra de *backlog* do produto, retirada do site *Mountain Goat Software* [7].

	Item #	Description	Est	By
<b>Very High</b>				
	1	<b>Finish database versioning</b>	16	KH
	2	<b>Get rid of unneeded shared Java in database</b>	8	KH
		<b>- Add licensing</b>	-	-
	3	Concurrent user licensing	16	TG
	4	Demo / Eval licensing	16	TG
		<b>Analysis Manager</b>		
	5	File formats we support are out of date	160	TG
	6	Round-trip Analyses	250	MC
<b>High</b>				
		<b>- Enforce unique names</b>	-	-
	7	In main application	24	KH
	8	In import	24	AM
		<b>- Admin Program</b>	-	-
	9	Delete users	4	JM
		<b>- Analysis Manager</b>	-	-
	10	When items are removed from an analysis, they should show up again in the pick list in lower 1/2 of the analysis tab	8	TG
		<b>- Query</b>	-	-
	11	Support for wildcards when searching	16	T&A
	12	Sorting of number attributes to handle negative numbers	16	T&A
	13	Horizontal scrolling	12	T&A
		<b>- Population Genetics</b>	-	-
	14	Frequency Manager	400	T&M
	15	Query Tool	400	T&M
	16	Additional Editors (which ones)	240	T&M
	17	Study Variable Manager	240	T&M
	18	Haplotypes	320	T&M
	19	<b>Add icons for v1.1 or 2.0</b>	-	-
		<b>- Pedigree Manager</b>	-	-
	20	Validate Derived kindred	4	KH
<b>Medium</b>				
		<b>- Explorer</b>	-	-
	21	Launch tab synchronization (only show queries/analyses for logged in users)	8	T&A
	22	Delete settings (?)	4	T&A

Figura 2.3: Amostra de *backlog* do projeto.

### 2.3.2.2 Backlog da Sprint

O *backlog* da *sprint* nada mais é que as tarefas do *backlog* do produto selecionadas para serem executadas em uma *sprint*.

Como será mais detalhado mais adiante, a cada *sprint* são selecionadas as tarefas mais importantes para serem executadas, de modo que a equipe consiga finalizar sua implementação ao final do tempo determinado.

Como mencionado em [1] e em [5], é importante que as tarefas e cada *sprint* não sejam atribuídas aos integrantes da equipe, e sim que cada um escolha aquelas que deseja executar, facilitando assim a motivação da equipe por desenvolver o trabalho.

No decorrer da *sprint*, o backlog da mesma vai sendo atualizado pelo *ScrumMaster*, através da sinalização, de algum modo de que a tarefa foi concluída e de quanto falta para que isto ocorra. Para auxiliar esse acompanhamento, podem ser usados os artefatos abaixo descritos: o *Burndown Chart* e o *Task Board*.

### 2.3.2.3 *Burndown Chart*

O *burndown chart* é simplesmente um gráfico, gerado manualmente ou através de algum *software* que pode ser utilizado tanto para acompanhamento de uma *sprint* (*Sprint Burndown Chart*) quanto para o acompanhamento do desenvolvimento do produto final (*Release Burndown Chart*).

No *Burndown Chart* é representada a uma linha de referência e a uma outra, que representa a realidade. Na figura 2.4 temos o exemplo de um *burndown chart* de uma *sprint* de 10 dias bem planejada, já que as linhas seguem bem próximas uma da outra.

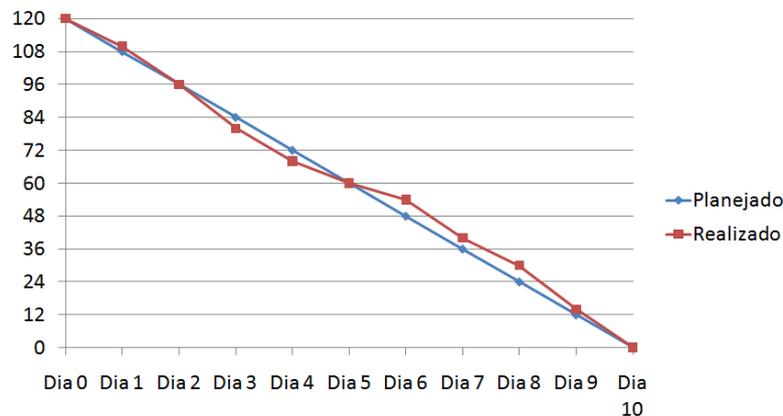


Figura 2.4: *Burndown Chart* de *sprint* bem planejada.

Já as figuras 2.5 e 2.6 representam gráficos de *sprints*, também de 10 dias, mal planejadas.

Como é possível observar, no gráfico da figura 2.5, as tarefas do *sprint backlog* foram subestimadas, fazendo com que seus executores levassem um tempo superior para desenvolvê-las do que aquele que foi planejado. O resultado é o término da *sprint* sem o término do desenvolvimento das tarefas planejadas.

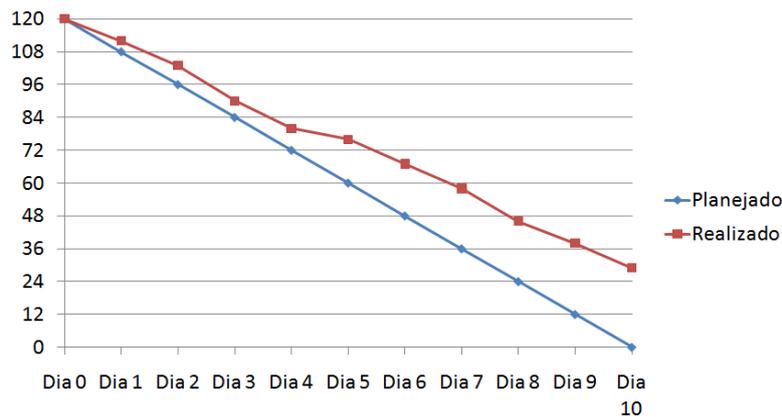


Figura 2.5: *Burndown Chart* de *sprint* subestimada.

Já o gráfico da figura 2.6, mostra o acompanhamento de uma *sprint* cujas tarefas foram superestimadas. Neste caso, os membros da equipe levaram um tempo bem inferior no desenvolvimento das tarefas, fazendo com que estas fossem finalizadas antes do término da *sprint*.

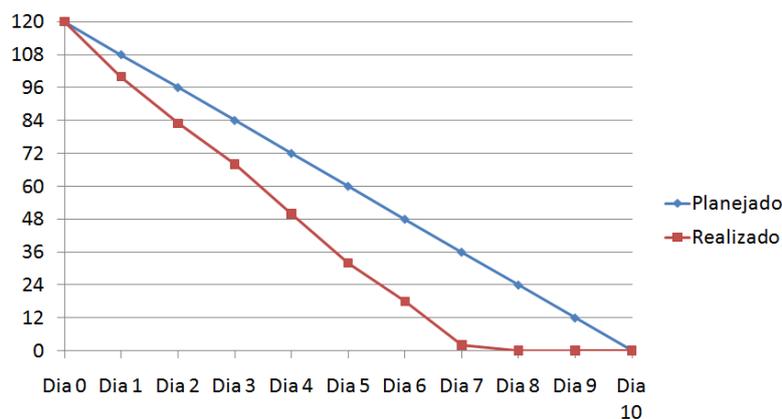


Figura 2.6: *Burndown Chart* de *sprint* superestimada.

#### 2.3.2.4 *Task Board*

O *Task Board*, ou Quadro de Tarefas é um quadro onde são representadas as tarefas de uma *sprint* e sua situação no momento, conforme representado na figura 2.7. É bastante útil para ter uma clara e rápida ideia das tarefas que devem ser desenvolvidas e das que já foram terminadas.

<i>Story</i>	<i>To Do</i>	<i>Doing</i>	<i>Done</i>
De acordo com o Product Owner...	Implement funcion. A 5  Verificar Impacto funcion. D 2  Implement melhoria funcion. E ED 5	Testar funcion. A 3  Testar funcion. B 5  Implement funcion. B AS 13  Levantam. Requisitos funcion. A DG 5	Testar funcio... BC  Levantam. Requisitos funcion. B FL 5
De acordo com o usuário...	Desenvol. novo módulo F 13  Implement funcion. G 2  Testar funcion. G 5		Levantar Requisitos funcion. G FL 3
De acordo com o Product Owner...	Corrigir Erro X AS 1	Implement filtro dados Y BC 5	Integrar funcion... ED 5

Figura 2.7: Exemplo de um Quadro de Tarefas.

Cada equipe deve montar o seu quadro de tarefas com as colunas que achar necessárias, mas em geral são representadas as seguintes situações:

- **Story** - Breve descrição da necessidade do usuário do sistema
- **To Do** - Breve descrição das tarefas que ainda não começaram a ser executadas, contendo também o esforço estimado para executá-las e, possivelmente, um identificador do integrante que as desenvolverão
- **Doing** - Breve descrição das tarefas que estão sendo desenvolvidas no momento, também contendo o esforço restante para finalizá-las e um identificador do integrante que as estão executando
- **Done** - Breve descrição das tarefas que já foram executadas, contendo o esforço total utilizado para finalizá-las e um identificador do membro que as executou.

O quadro de tarefas é utilizado através da transferência das tarefas através das colunas, conforme a atual situação de cada uma delas. Em geral, essa transferência é feita durante a reunião diária, que será melhor descrita mais adiante.

Os quadros de tarefas podem ser feitos também da maneira que a equipe julgar conveniente, sendo bastante comum sua construção utilizando um cartaz e *post-its* contendo as *stories* e as descrições das tarefas.

### 2.3.3 Reuniões

#### 2.3.3.1 Reunião de Planejamento de *Sprint*

A reunião de planejamento de *sprint* deve acontecer a todo início de *sprint* e é dividida em duas etapas.

Na primeira parte da reunião, devem estar presentes a equipe do *Scrum*, o *Product Owner* e quem mais for pertinente, como o cliente com alto conhecimento dos requisitos e suas prioridades. Durante esta etapa, as tarefas do *Product Backlog* são brevemente descritas e o cliente ou *Product Owner* prioriza cada uma delas.

Dependendo do tamanho do *Product Backlog*, não é necessário descrever cada uma das atividades, mas apenas as mais prioritárias e que possivelmente preenchem uma *sprint*.

Ao passar pelas tarefas, os membros da equipe podem interagir com o *Product Owner* e dar opiniões sobre as dependências técnicas entre as tarefas e possibilidade de algumas delas serem desenvolvidas em alguma outra *sprint*, ao invés de na *sprint* que se inicia.

Todos os presentes, então definem um objetivo para ser realizado no próximo ciclo, como o fechamento de um conjunto de novas funcionalidades.

Após essa parte, todos que não fazem parte da equipe do *Scrum*, inclusive o *Product Owner* se retiram e a primeira decide que tarefas realmente selecionar para o *Sprint Backlog*, com base naquilo que foi discutido anteriormente e na dificuldades das tarefas a serem executadas.

Para auxiliar no bom planejamento *sprint*, principalmente no que diz respeito à quantidade de tarefas a serem selecionadas, existe uma técnica chamada *Planning*

*Poker* [9]. O *Planning Poker* consiste no uso de cartas de baralho especiais, conforme ilustrado na figura 2.8, para estimar o esforço gasto para realizar cada tarefa.

O funcionamento é bem simples e intuitivo. Após serem definidas todas as atividades, os membros da equipe as selecionam, uma a uma, e cada membro escolhe, secretamente, uma carta que representa o esforço que ele acha que aquela tarefa demanda. Havendo discordância entre as pessoas, cada um levanta seus argumentos, é chegado a um consenso e definido o esforço para realizar cada uma das tarefas.



Figura 2.8: Algumas cartas do baralho utilizado no Planning Poker.

É válido aqui comentar que o modo mais comum de representar o esforço de uma tarefa é através de números, como os da sequência de Fibonacci, ou a mesma adaptada, cujos valores são 1, 2, 3, 5, 8, 13, 20, 40, 60 e 100. Entretanto, há também equipes que usam outros meios para quantizar suas tarefas, como animais, sendo uma formiga, por exemplo atribuído a tarefas mais fáceis, enquanto o touro, às mais difíceis.

Neste projeto consideraremos a sequência de Fibonacci adaptada para o *Scrum*.

Ainda, é desejável que as tarefas não sejam muito grandes, ou seja, não exijam mais do que 16 horas para serem terminadas. Caso haja tarefas grandes, o melhor a fazer é dividi-las, até que existam tarefas, cujo tempo de desenvolvimento seja menor que o previamente dito.

Tendo então definidos os esforços para cada tarefa, e feita uma estimativa do esforço total que a equipe pode gastar durante toda a *sprint*. Este valor só pode ser estimado pelos próprios integrantes da equipe, levando em consideração as competências de cada um do grupo, o tempo de trabalho, a concentração de cada um. Este valor total do esforço de todos da equipe é chamado de *Sprint Velocity*. Dizemos que o *Sprint Backlog* está planejado e a reunião encerrada quando a soma das tarefas selecionadas é igual à *Sprint Velocity*.

É importante ressaltar que é fundamental que os integrantes da equipe sejam realistas ao planejarem cada *Sprint*, não selecionando tarefas a mais ou a menos que a equipe é capaz de executar. Um bom planejamento, é essencial para o bom funcionamento do *Scrum*.

### 2.3.3.2 Reunião Diária

Para acompanhar o andamento da *sprint*, todos os dias deve ser feita um reunião diária, também conhecida como *Daily Scrum*, ou *Scrum Diário*, onde os integrantes da equipe se reúnem por no máximo 15 minutos e falam o que já fizeram, o que farão e se houve ou se há a possibilidade de haver algum impedimento.



Figura 2.9: Equipe no momento da reunião diária.

Quaisquer problemas levantados nesse momento devem ser discutidos posteriormente, de preferência com o *ScrumMaster*, já que ele é o encarregado de eliminar os obstáculos existentes.

### 2.3.3.3 Reunião de Revisão de *Sprint*

Ao final de cada *sprint* é feita uma reunião entre os membros da equipe, *Product Owner*, clientes e quem mais for convidado. Esta reunião deve ser bastante informal, não deve usar slides e nem demorar mais do que duas horas para ser preparada. Deve ser uma simples exibição dos resultados obtidos no ciclo que se encerrou, basicamente através da apresentação de uma demonstração do produto ou nova versão deste lançada.

### 2.3.3.4 Reunião de Restrospectiva de *Sprint*

Também ao final da *sprint* é feita uma breve reunião, que deve demorar em torno de 15 a 30 minutos para avaliar o que foi bom e o que foi ruim, o que funcionou para o grupo no ciclo encerrado e o que pode ser melhorado no futuro.

Deste reunião participam toda a equipe, incluindo o *Product Owner* e, talvez os clientes.

É essencial que não haja tarefas executadas em uma *sprint* além das que foram planejadas, porém na prática, observamos que no decorrer de um ciclo, certas tarefas podem perder ou ganhar prioridade. Portanto, é fundamental um bom acompanhamento das tarefas por parte do *ScrumMaster* para que tais mudanças sejam evitadas ao máximo, porém caso sejam inevitáveis, a atitude certa seja tomada.

# Capítulo 3

## Planejamento

### 3.1 Apresentação

#### 3.1.1 Sumário do Projeto

##### 3.1.1.1 Finalidades, Escopo e Objetivos

O presente projeto consiste de um *software* que irá permitir o registro de atividades, sua visualização e manipulação, de forma a permitir um eficiente gerenciamento de *sprints* por parte de *Scrum Masters*.

##### 3.1.1.2 Postulados e Restrições

O *software* em desenvolvimento não conta com restrições de orçamento ou tecnologia, visto que as ferramentas que serão utilizadas já foram adquiridas e suas licenças foram obtidas gratuitamente através do Programa *MSDN Academic Alliance* [11]. Contudo, existe restrição de tempo de desenvolvimento, já que o projeto deve ser finalizado até Dezembro de 2010.

##### 3.1.1.3 Liberações Parciais

Neste projeto, será seguido o modelo cascata linear-sequencial, ilustrado na figura 3.1, no qual primeiramente é feito o planejamento e em seguida, são abordados análise, desenho, codificação e integração do projeto.

Os produtos liberados seguirão o seguinte planejamento:

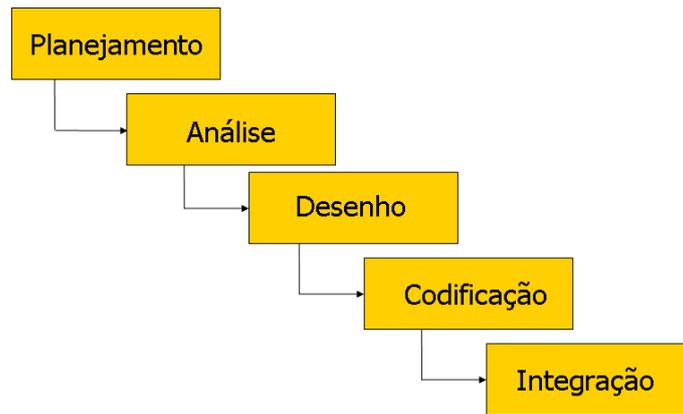


Figura 3.1: Modelo Cascata Linear-Sequencial

- Planejamento para o Gerenciamento de Projeto de Software
- Especificações de Requisitos de Software
- Descrição de Projeto de Software
- Plano de Testes
- Manual de Usuário
- Versão alfa do sistema
- Documentação Total

#### 3.1.1.4 Sumário de Cronograma e Orçamento

A figura 3.2 representa o cronograma a ser seguido para o desenvolvimento do projeto.

Como já mencionado anteriormente, não há orçamento programado para este projeto.

#### 3.1.2 Evolução do Plano

O PGPS será analisado pelo orientador deste projeto e caso seja necessária alguma alteração, o primeiro será refeito e posteriormente entregue ao professor, como uma nova versão. Ainda, se, durante o processo de desenvolvimento for notada alguma necessidade de alteração do produto ou de alguma especificação, o PGPS também

	Maio/10	Junho/10	Julho/10	Agosto/10	Setembro/10	Outubro/10	Novembro/10	Dezembro/10
Estudo do problema	■			■				
Escolha das ferramentas utilizadas	■							
Aquisição das ferramentas utilizadas		■						
Análise de riscos		■						
Levantamento de requisitos			■					
Elaboração dos casos de uso			■					
Modelagem das classes e definição do Banco de Dados			■	■				
Definição de interfaces				■				
Definição da arquitetura do software				■				
Elaboração do Plano de Testes				■	■			
Elaboração das interfaces da aplicação				■				
Codificação da aplicação					■	■	■	
Elaboração de procedures para aquisição de dados					■	■	■	
Testes relativos à aquisição de dados					■	■	■	
Testes do funcionamento da aplicação					■	■	■	
Testes finais						■	■	
Elaboração da documentação		■	■	■			■	■

Figura 3.2: Cronograma de atividades do projeto

será feito. Esse procedimento será repetido quantas vezes forem necessárias, até que o PGPS seja totalmente aprovado.

## 3.2 Referências

As referências bibliográficas encontram-se no final deste documento.

## 3.3 Definições

- Visual Studio: Pacote de programas da Microsoft para desenvolvimento de softwares;
- SQL Server: SGBD da Microsoft, que será utilizado no presente projeto;

## **3.4 Organização do Projeto**

### **3.4.1 Interfaces Externas**

O projeto a ser desenvolvido foi proposto como Projeto de Graduação do curso de Engenharia Eletrônica e de Computação da UFRJ e será orientado pelo professor Antônio Cláudio.

A comunicação entre a aluna e o orientador será feita em reuniões a cada 15 dias e via email sempre que necessário.

### **3.4.2 Estrutura Interna**

O projeto será desenvolvido pela aluna Andreza Cristina da Silva, autora deste documento, que será responsável por todas as tarefas propostas.

O controle de qualidade será feito por seu orientador, Antônio Cláudio, através das reuniões realizadas no decorrer da execução do projeto.

### **3.4.3 Papéis e Responsabilidades**

Como dito anteriormente, a responsável pela execução de todas as tarefas necessárias para o desenvolvimento do projeto proposto é a aluna Andreza Cristina da Silva.

## **3.5 Processos de Gerenciamento**

### **3.5.1 Partida do Projeto**

#### **3.5.1.1 Previsões**

No desenvolvimento do projeto será utilizada a metodologia ágil *Scrum*, portanto foi feito um *backlog* com as atividades a serem executadas.

Caso, durante a execução do projeto, surjam novas atividades, essas serão adicionadas ao *backlog* referido e será realizado um novo planejamento, levando-se em

consideração a prioridade, o esforço das atividades pendentes e o prazo final de entrega.

O *backlog* inicial do projeto pode ser visualizado na figura 3.3 e as datas em que serão iniciadas e finalizadas as *sprints* podem ser vistas na tabela 3.1.

Tarefa	Sprint	Esforço Planejado	Esforço Real	Obs
<b>Download de Ferramentas</b>	1			
Visual Studio	1	1		
SQL Server	1	1		
<b>Elaboração PGPS</b>	2,3			
Apresentação e Organização do Projeto	2	3		
Análise de Riscos	2	3		
Processos Técnicos e Planos para os processos de Suporte	3	2		
<b>Elaboração ERS</b>	4			
Levantamento de Requisitos	4	5		
Elaboração do casos de uso e Diagrama de classes	4	3		
Projeto	5	5		
Definição do Banco de Dados	6	3		
Desenho das Interfaces	6	2		
Definição da interface com o Banco de Dados	6	2		
<b>Elaboração do Plano de Testes</b>	6,7			
Parte 1	6	3		
Parte 2	7	5		
Elaboração das interfaces	7	5		
Estudo Tree Grid View	8	8		
<b>Codificação (aplicação e banco)</b>	9,11			
Parte 1	9	8		
Parte 2	10	8		
Parte 3	11	8		
<b>Testes</b>	12,13			
Parte 1	12	5		
Parte 2	13	5		
<b>Elaboração da documentação complementar</b>	14,16			
Elaboração Documentação Scrum - Parte 1	14	5		
Parte 2	15	5		
Parte 3	16	5		

Figura 3.3: Backlog do projeto.

### 3.5.1.2 Equipe

Como citado no item "Interfaces Internas", a equipe é composta apenas por uma pessoa, que possui conhecimento avançado da linguagem de programação C#, utilização do *software Visual Studio* e *SQL Server*, ferramentas essas que serão utilizadas no desenvolvimento do projeto.

### 3.5.1.3 Plano para a Aquisição de Recursos

Como todos os recursos necessários já estão disponíveis, este item não se aplica a esse projeto.

<i>Sprint</i>	<b>Início</b>	<b>Fim</b>
1	19/Maio	25/Maio
2	26/Maio	8/Junho
3	9/Junho	22/Junho
4	23/Junho	6/Julho
5	7/Julho	20/Julho
6	21/Julho	3/Agosto
7	4/Agosto	17Agosto
8	18/Agosto	31/Agosto
9	1/Setembro	14/Setembro
10	15/Setembro	28/Setembro
11	29/Setembro	12/Outubro
12	13/Outubro	26/Outubro
13	27/Outubro	9/Novembro
14	10/Novembro	23/Novembro
15	24/Novembro	7/Dezembro
16	8/Dezembro	22/Dezembro

Tabela 3.1: Datas de início de fim de cada *sprint*.

#### **3.5.1.4 Plano de Treinamento da Equipe**

Como mencionado no item "Equipe", as ferramentas que serão utilizadas já são familiares à equipe, portanto, não será necessário treiná-la. Quaisquer dificuldades não planejadas serão solucionadas através de consultas à internet e da ajuda do orientador.

### **3.5.2 Plano de Trabalho**

#### **3.5.2.1 Atividades**

As atividades desenvolvidas nesse projeto estão especificadas na figura 3.2.

### **3.5.2.2 Prazos**

Os prazos para a realização das tarefas do projeto encontram-se na figura 3.3 e na tabela 3.1.

### **3.5.2.3 Alocação de Recursos**

Todos os recursos necessários para o desenvolvimento do projeto já foram devidamente adquiridos, portanto não se faz necessária a alocação de recursos.

### **3.5.2.4 Alocação de Orçamento**

Como não há necessidade de compra de material ou pagamento de mão-de-obra, esse item não se aplica a este projeto.

## **3.5.3 Planos de Controle**

### **3.5.3.1 Controle de Requisitos**

Ao dar início ao desenvolvimento do projeto, caso haja requisitos que sejam inviáveis à realização da proposta, os requisitos inicialmente propostos, especificados no documento ERS serão revistos e alterados de forma que viabilizem a realização do projeto.

Ao ser feito isso, uma nova versão da ERS será feita e as atividades do *backlog* do projeto, planejadas novamente.

### **3.5.3.2 Controle de Prazos**

Como já mencionado, será utilizada a metodologia *Scrum* no desenvolvimento do sistema. Com isso, ao final de cada *sprint* serão vistas as atividades que foram concluídas e, caso haja atividades não finalizadas no tempo previsto, será realizado um novo planejamento das *sprints* seguintes, para que o prazo final seja atendido.

### **3.5.3.3 Controle de Orçamento**

Como já citado anteriormente, não há custo para o desenvolvimento da proposta feita, portanto não é necessário o controle de orçamento.

#### 3.5.3.4 Controle de Qualidade

O controle de qualidade será feito através das reuniões com o orientador do projeto, realizadas a cada duas semanas. Caso este julgue necessário, serão realizadas alterações no material já elaborado.

#### 3.5.3.5 Plano de Relatórios

Caso se faça necessário, serão emitidos relatórios para serem apresentados nas reuniões quinzenais entre a equipe e o orientador do projeto.

#### 3.5.3.6 Plano de Medidas

A principal medida tomada será o esforço para cada tarefa ser realizada.

O esforço estimado para a realização de cada tarefa está representado na figura 3.3.

A cada tarefa concluída, será adicionado ao *backlog* do projeto o esforço real para realizá-la e, caso necessário, serão analisadas as causas entre a possível diferença entre os esforços planejado e real.

### 3.5.4 Plano de Gerenciamento de Riscos

Os riscos previstos estão relacionados na tabela 3.2.

- RMMM1

Monitoramento: Acompanhamento das atividades desenvolvidas através do *burndown chart* e avanço no *backlog*

Política Pró-ativa: Planejamento de cada *sprint* de forma que a equipe não fique sobrecarregada nem com menos atividades que o necessário

Política Reativa: Novo planejamento das *sprints* seguintes

- RMMM2

Monitoramento: Manter *backups* virtuais e em disco.

Risco	Categoria	Probabilidade	Impacto	RMMM
Entrega fora do prazo	Equipe	20%	Catastrófico	RMMM1
Falta de capacitação ou experiência da equipe	Equipe	10%	Crítico	
Perda de documentos	Tecnológica	20%	Catastrófico	RMMM2
Mudança de requisitos	Cliente/Equipe	40%	Marginal	
Uso de ferramentas inadequadas	Equipe	10%	Marginal	
Falta de experiência da equipe	Equipe	10%	Crítico	
Complexidade do sistema	Tecnológica	30%	Crítico	RMMM3

Tabela 3.2: Riscos previstos para o projeto.

Política Pró-ativa: Manter *backups* virtuais e em disco a cada nova versão feita de arquivos fonte e de documentos gerados.

Política Reativa: Resgatar a última versão em backup, antes de ocorrer o problema de perdas.

- RMMM3

Monitoramento: Verificação de complexidade do sistema ao realizar análise e também projeto de software.

Política Pró-ativa: Tentar utilizar o máximo de funcionalidades já existentes e utilizar os conhecimentos de Engenharia de Software.

Política Reativa: Reduzir os requisitos inicialmente propostos, para ade-

quar a complexidade do projeto ao tempo disponível para sua realização.

### **3.5.5 Plano de Encerramento**

A maior parte da documentação do projeto será elaborada durante o período de desenvolvimento, sendo esta apresentada ao orientador conforme planejado. A documentação restante, será elaborada após a elaboração do sistema.

Após o completo desenvolvimento do sistema e sua documentação, esta será entregue à banca avaliadora, que assistirá uma apresentação sobre o projeto e o julgará.

## **3.6 Processos Técnicos**

### **3.6.1 Modelo dos Processos**

Como dito anteriormente, neste projeto será usado o Modelo Cascata - Linear Sequencial, representado na figura 3.1.

As liberações parciais foram mencionadas no item "Liberações Parciais" e o marcos estão estabelecidos como o fim de cada *sprint*, cujas datas podem ser observadas na tabela 3.1.

### **3.6.2 Métodos, Ferramentas e Técnicas**

Para gerar toda a documentação será utilizado  $\text{\LaTeX}$ .

As linguagens de programação utilizadas serão *C#*, para a codificação da aplicação e *SQL* para o banco de dados. Para tais, será feito o uso dos *softwares Microsoft Visual Studio* e *SQL Server*, respectivamente.

Como a linguagem escolhida é orientada a objeto, será feita a Análise Orientada a Objeto.

Caso faça-se necessário o uso de mais ferramentas estas serão escolhidas ao decorrer do desenvolvimento do projeto.

### **3.6.3 Infra-estrutura**

O ambiente de desenvolvimento será Windows Vista e os computadores utilizados no processo serão de uso pessoal.

As licenças eventuais necessárias, serão adquiridas através do programa *MSDN Academic Alliance*, como anteriormente mencionado.

Toda a documentação gerada estará no padrão ABNT [12].

### **3.6.4 Plano para Aceitação do Produto**

Para avaliar se as funcionalidades do sistema estão de acordo com seus requisitos, serão realizados os testes previstos no Plano de Testes. Além disso, serão feitas reuniões periódicas com o orientador do projeto, que homologará o funcionamento do sistema.

## **3.7 Planos para os processos de Suporte**

### **3.7.1 Gerenciamento de Configuração**

Como, nesse projeto, a equipe é formada por apenas 1 (uma) pessoa, o controle de versões não se faz tão necessário. Ao invés de utilizar *softwares* para controlar versões, serão feitos *backups* em disco e de forma eletrônica a cada documento gerado e/ou funcionalidade implementada.

### **3.7.2 Plano de Verificação e de Validação**

Como já mencionado, para verificar o funcionamento do projeto, serão realizados os testes contidos no Plano de Testes, que será entregue ao orientador antes de o início da codificação do sistema.

### **3.7.3 Documentação**

Como documentação não liberável, estão as mensagens de *e-mail* trocadas entre a aluna e seu orientador, assim como as versões parciais dos documentos a serem

entregues.

Como documentação liberável estão o presente documento, a ERS, o Plano de Testes, a Descrição de Projeto de Software e a documentação exigida para a avaliação do projeto.

A elaboração dos documentos será de responsabilidade da aluna, enquanto a avaliação destes será de responsabilidade de seu orientador durante o desenvolvimento do projeto e da banca avaliadora, no momento de sua defesa.

### **3.7.4 Plano para Assegurar a Qualidade**

Durante a elaboração dos documentos serão feitas revisões por parte da aluna antes de sua entrega ao professor. Este último fará a verificação final, assegurando a qualidade dos produtos desenvolvidos.

Analogamente, no processo de desenvolvimento serão feitas avaliações e testes das funcionalidades até então terminadas, verificando se estão de acordo com o previsto. Durante as reuniões quinzenais com o professor, este também verificará o funcionamento do sistema.

### **3.7.5 Revisões e Auditorias**

O revisor final de toda a documentação elaborada e o projeto desenvolvido será o orientador do projeto. O produto lhe será entregue (em forma de papel no caso da documentação e por meio eletrônico, do sistema) nas reuniões quinzenais e eventualmente por correio eletrônico, quando for necessário.

Caso novas versões do produto desenvolvido forem necessárias, o processo se repetirá até que aquele seja aceito.

### **3.7.6 Plano para a Resolução de Problemas**

Caso algum problema previsto no item "Plano de Gerenciamento de Riscos" ocorra, serão tomadas as medidas previstas para tentar solucioná-lo.

Para problemas de dificuldade relevante não previstos, serão discutidas algumas possíveis soluções nas reuniões realizadas entre a aluna e seu orientador.

### **3.7.7 Gerenciamento de Subcontratações**

Como nesse projeto não será realizado nenhum tipo de contratação, esse item não se aplica.

### **3.7.8 Plano de Aperfeiçoamento**

Durante as reuniões de início/fim de *sprint*, será avaliado se as ferramentas utilizadas no desenvolvimento do projeto são realmente as mais indicadas, já que o tempo disponível para a realização deste é um fator crítico. Caso necessário e viável, será realizada nova escolha de ferramentas.

## **3.8 Planos Adicionais**

Ainda não existe nenhum plano adicional para esse projeto.

# Capítulo 4

## Especificação de Requisitos de Software

### 4.1 Introdução

#### 4.1.1 Finalidade

Esta ERS tem como finalidade permitir o entendimento de como será feito o sistema, além de estruturar o projeto para facilitar o desenvolvimento do mesmo.

Este documento é dirigido ao orientador Antônio Cláudio Gómez Sousa e à banca examinadora do projeto.

#### 4.1.2 Escopo

O iPlan é um sistema de gerenciamento de tarefas para Usuários de *Scrum* e tem por objetivo facilitar o planejamento de tarefas de um projeto a serem executadas por usuários da metodologia ágil *Scrum*.

Através do iPlan, *Product Owners* poderão gerenciar diferentes projetos e os grupos que os executarão, *Scrum Masters* serão capazes adicionar tarefas a serem desenvolvidas em seus respectivos projetos, além de planejar quando estas deverão ser executadas.

Ainda, membros da equipe desenvolvedora poderão observar o planejamento feito pelos *Scrum Masters*.

O sistema será desenvolvido de forma a ser acessado via web, porém sua execução se dará localmente (através do *download* da aplicação) pelos usuários, que compartilharão um mesmo servidor de banco de dados.

### **4.1.3 Definições, Acrônimos e Abreviaturas**

As definições necessárias encontram-se no início desta documentação.

### **4.1.4 Referências**

As referências bibliográficas encontram-se no final deste documento.

### **4.1.5 Resumo**

Esta ERS mostra com detalhes o funcionamento do iPlan. Suas dependências e requisitos serão especificadas, assim como suas funcionalidades, esclarecidas através dos diagramas de casos de uso e das especificações de cada caso. Nesse documento também é apresentado o diagrama de classes.

Informações adicionais sobre o funcionamento do iPlan constarão no Manual do Usuário e na Descrição de Projeto de Software.

## **4.2 Descrição Geral**

### **4.2.1 Perspectiva do Produto**

O sistema a ser desenvolvido é independente de qualquer outro sistema.

### **4.2.2 Funções do Produto**

- Usuários
  - Inclusão

- Edição
- Exclusão
- Visualização
- Grupos de usuários
  - Inclusão
  - Edição
  - Exclusão
  - Visualização
  - Inserção de integrantes
- Projetos
  - Inclusão
  - Edição
  - Exclusão
  - Visualização
- *Sprints*
  - Inclusão
  - Edição
  - Exclusão
  - Visualização
- Tarefas
  - Inclusão
  - Inclusão de tarefa filha
  - Edição
  - Exclusão
  - Visualização

– Filtro por diferentes parâmetros

- *Login* e *Logout* de usuários

### 4.2.3 Características do Usuário

O iPlan é destinado a equipes que utilizam o *Scrum* como forma de organização e gerenciamento do projeto, principalmente aos *Scrum Masters*. Portanto, os usuários previstos são pessoas com idade entre 18 a 65 anos, não importando o gênero.

O tempo de duração de uso do *software* depende da duração do projeto desenvolvido pelos usuários.

### 4.2.4 Restrições

A única restrição ao uso do sistema é a utilização de Windows como sistema operacional.

### 4.2.5 Pressupostos e Dependências

O *software* desenvolvido será compatível apenas com sistema operacional Windows. Para que dados sejam salvos e acessados será criado um banco de dados no momento da instalação do sistema e acessado sempre que necessário.

Tendo em vista que o sistema desenvolvido terá interface amigável e utilização intuitiva, os demais pressupostos desse projeto são a utilização de um banco de dados único através de uma rede por parte dos usuários e a leitura do Manual do Usuário, que será liberado juntamente com a versão alfa do sistema.

### 4.2.6 Postergar Requisitos

Duas desejáveis funcionalidades futuras do sistema são a possibilidade de o usuário adicionar mais dados em "tarefas" e a exportação dos dados observados para arquivos de formato .xls ou .pdf, por exemplo.

## 4.3 Requisitos Específicos

### 4.3.1 Interfaces Externas

#### 4.3.1.1 Interfaces dos Usuários

A interface com os usuários se dará através de uma aplicação executada localmente, porém cujo acesso será feito através da WEB.

Um maior detalhamento da interface com usuários pode ser encontrado no Manual do Usuário.

#### 4.3.1.2 Interfaces de Hardware

Como o sistema desenvolvido não se comunica com *hardwares*, não haverá interfaces de *hardware*.

#### 4.3.1.3 Interfaces de Software

Como esse sistema não se comunica com nenhum outro *software*, não haverá interfaces de *software*.

#### 4.3.1.4 Interfaces de Comunicação

Como será utilizado um banco de dados concentrado, o *software* terá como interface de comunicação a WEB.

### 4.3.2 Requisitos Funcionais

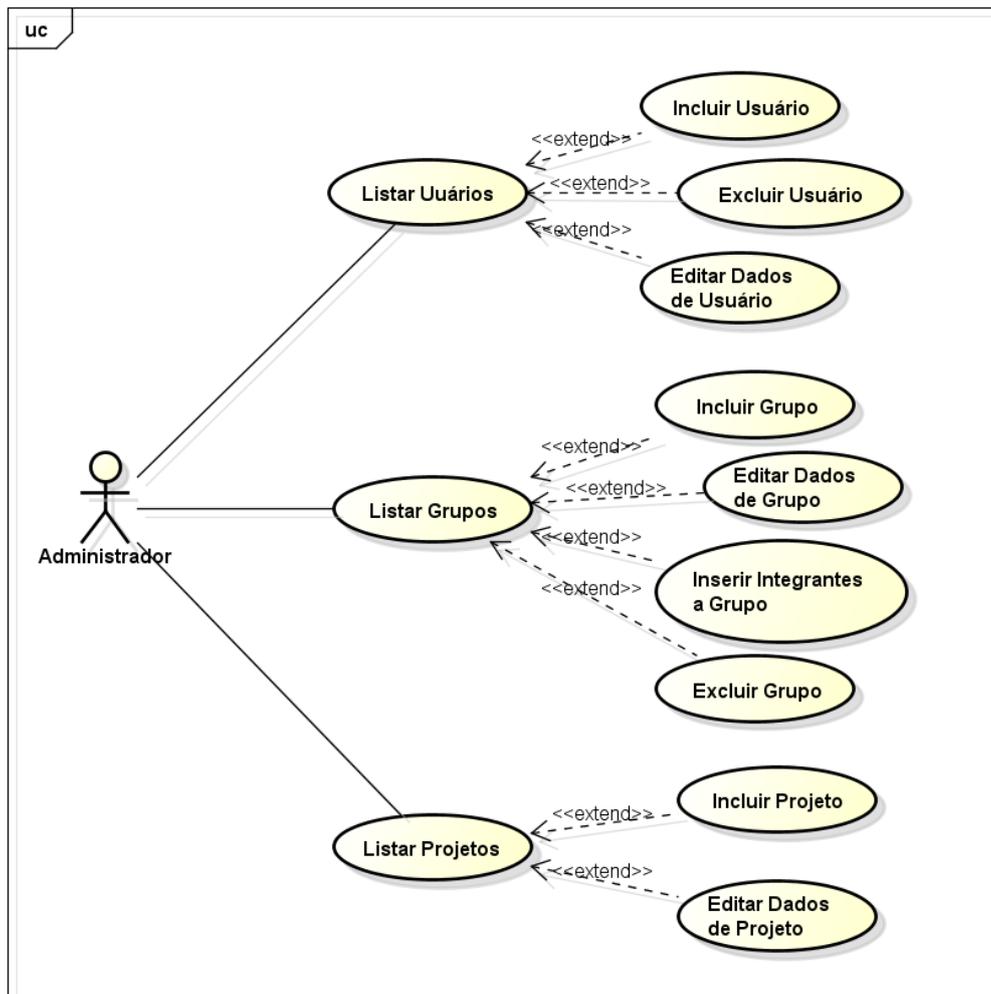
#### Casos de Uso

Os diagramas de casos de uso são ilustrados nas figuras 4.1, 4.2, 4.3 e 4.4, de acordo com o grupo de usuário.

A especificação de cada caso é feita a seguir:

- **Listar Usuários**

Ator: Administrador, *Product Owner*, *Scrum Master*, Membro Equipe



powered by astah

Figura 4.1: Diagrama de Casos de Uso para Administradores do Sistema.

Pré-condições: Estar autenticado pelo sistema

Casos de Uso associados: Incluir Usuário, Editar Dados de Usuário Cadastrado, Excluir Usuário

Pós-condições: NA

Fluxo Principal:

1. Usuário acessa a tela relativa a usuários
2. Sistema lista na tela os usuários cadastrados

Fluxo Alternativo:

- 3a. Usuário ordena usuários listados por nome, nome de usuário ou email

- **Incluir Usuário**

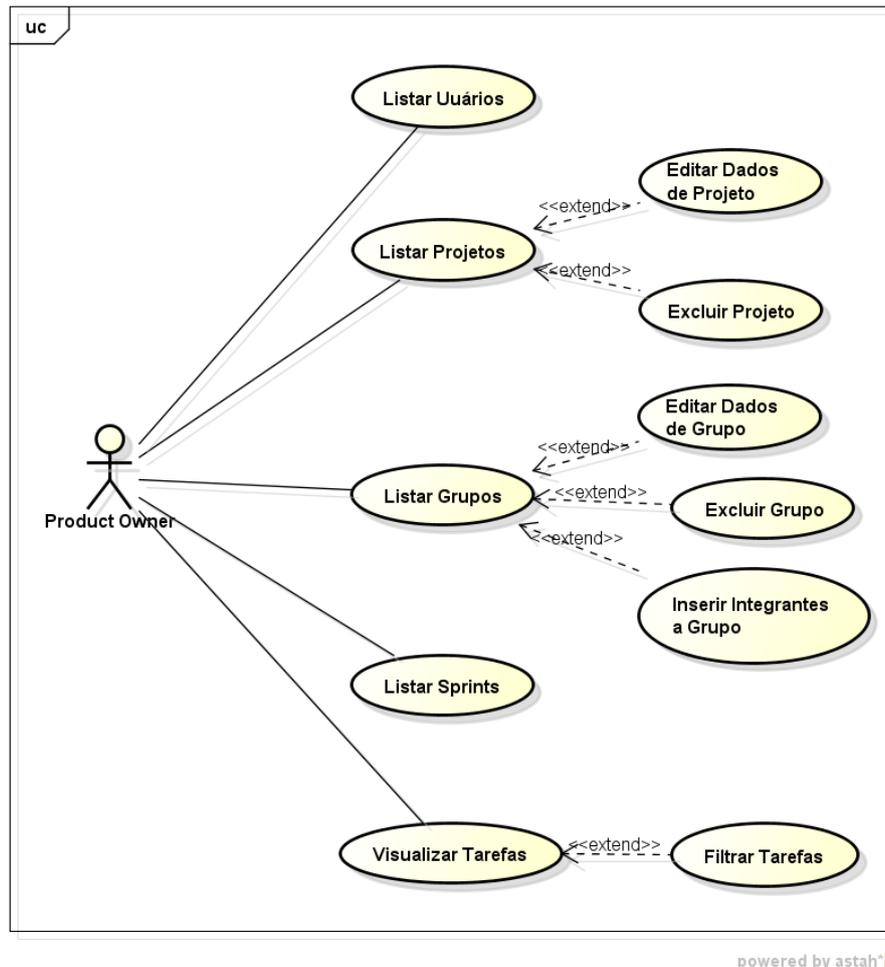


Figura 4.2: Diagrama de Casos de Uso para *Product Owners*.

Ator: Administrador

Pré-condições: Estar autenticado pelo sistema e ter listado usuários

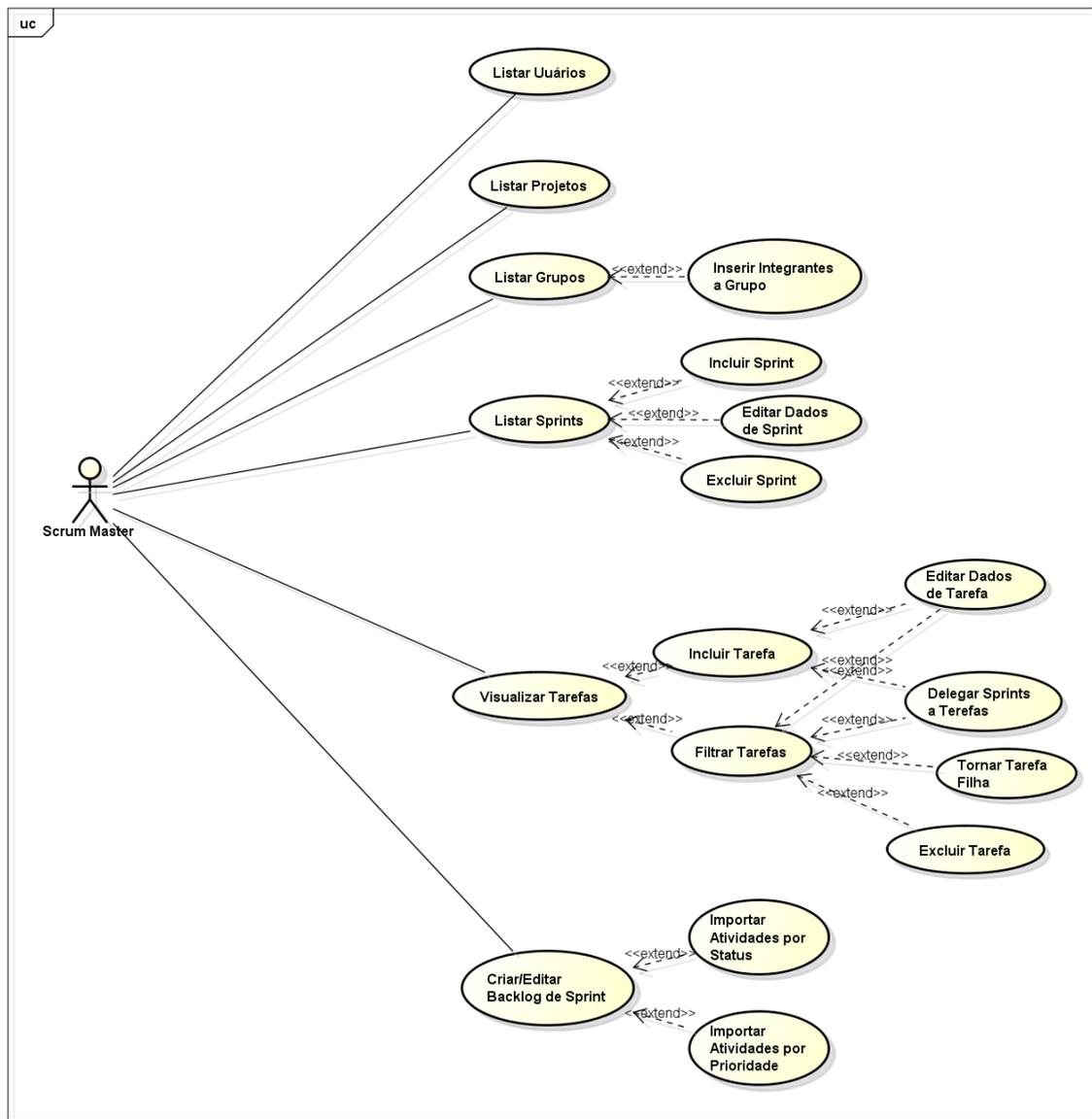
Casos de Uso associados: Listar Usuários

Pós-condições: Haverá mais um usuário cadastrado no sistema

Fluxo Principal:

1. Usuário seleciona opção de incluir usuário
2. Sistema exibe um formulário
3. Usuário preenche as informações do novo usuário e solicita o armazenamento dos dados
4. Sistema exibe informação confirmando inclusão do usuário

Fluxo Alternativo:



powered by astah

Figura 4.3: Diagrama de Casos de Uso para *Scrum Masters*.

4a. Sistema exibe mensagem de erro caso alguma informação não esteja correta ou haja campos obrigatórios não preenchidos

5a. Retorna ao Fluxo Principal

- **Editar Informações de Usuário Cadastrado**

Ator: Administrador

Pré-condições: Estar autenticado pelo sistema, ter listado usuários e haver pelo menos um usuário cadastrado no sistema

Casos de Uso associados: Listar Usuários

Pós-condições: Os dados de um usuário cadastrado no sistema serão alterados

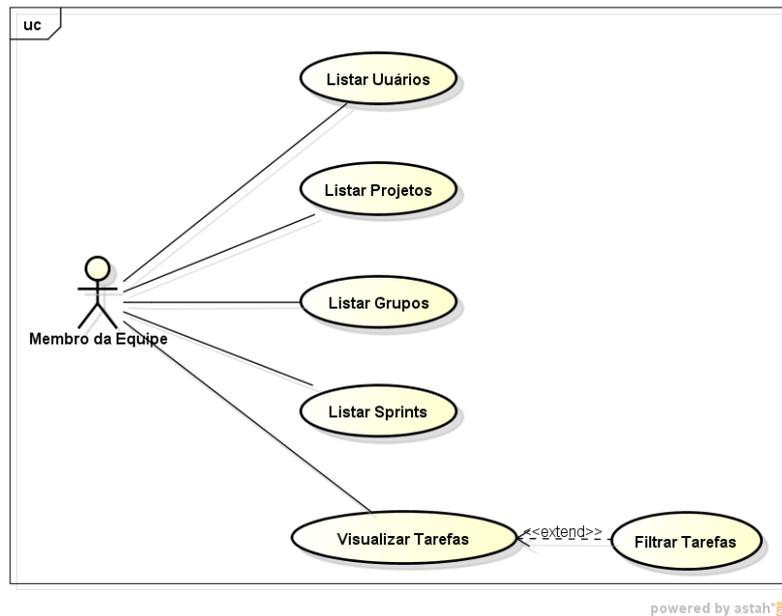


Figura 4.4: Diagrama de Casos de Uso para membros da equipe desenvolvedora.

Fluxo Principal:

1. Usuário seleciona um usuário previamente cadastrado no sistema
2. Usuário seleciona opção de edição de dados
3. Sistema exibe um formulário com informações cadastradas preenchidas
4. Usuário altera as informações desejadas
5. Sistema exibe informação confirmando edição de dados do usuário

Fluxo Alternativo:

- 5a. Sistema exibe mensagem de erro caso alguma informação não esteja correta
- 6a. Retorna ao Fluxo Principal
- 5b. Sistema exibe mensagem informando que não há dados para serem alterados
- 6b. Retorna ao Fluxo Principal

- **Excluir Usuários**

Ator: Administrador

Pré-condições: Estar autenticado pelo sistema, ter listado usuários e existir ao

menos 1 (um) usuário, além do próprio, cadastrado no sistema

Casos de Uso associados: Listar Usuários

Pós-condições: Os usuários selecionados serão excluídos logicamente do sistema

Fluxo Principal:

1. Usuário seleciona um ou mais usuários dentre os listados
2. Usuário seleciona opção para excluir o(s) usuário(s) selecionado(s)
3. Sistema exibe informação confirmando a exclusão do usuário

### ● **Listar Projetos**

Ator: Administrador, *Product Owner*, *Scrum Master*, Membro Equipe

Pré-condições: Estar autenticado pelo sistema

Casos de Uso associados: Incluir Projeto, Editar Dados de Projeto Cadastrado, Excluir Projeto

Pós-condições: NA

Fluxo Principal:

1. Usuário seleciona a tela relativa a projetos
2. Sistema lista os projetos cadastrados relacionados aos grupos do usuário logado e os projetos sem nenhum grupo relacionado, caso haja

### ● **Incluir Projeto**

Ator: Administrador

Pré-condições: Estar autenticado pelo sistema e ter acessado a tela relativa a projetos

Casos de Uso associados: Listar Projetos

Pós-condições: Haverá mais um projeto cadastrado no sistema

Fluxo Principal:

1. Usuário seleciona opção de incluir novo projeto
2. Sistema exibe formulário
3. Usuário preenche informações relativas ao projeto
4. Sistema exibe informação confirmando a inclusão do projeto

Fluxo Alternativo:

- 4a. Sistema exibe mensagem de erro caso alguma informação digitada pelo usuário não seja aceita
- 5a. Retorna ao Fluxo Principal

- **Editar Informações de Projeto Cadastrado**

Ator: *Product Owner*

Pré-condições: Estar autenticado pelo sistema, ter listado projetos e haver pelo menos 1 (um) projeto cadastrado no sistema

Casos de Uso associados: Listar Projetos

Pós-condições: Os dados de um projeto cadastrado no sistema serão alterados

Fluxo Principal:

1. Usuário seleciona um projeto previamente cadastrado no sistema
2. Usuário seleciona opção de edição de dados do projeto
3. Sistema exibe um formulário com informações cadastradas preenchidas
4. Usuário altera as informações desejadas
5. Sistema exibe informação confirmando edição de dados do projeto

Fluxo Alternativo:

- 5a. Sistema exibe mensagem de erro caso alguma informação não esteja correta
- 6a. Retorna ao Fluxo Principal
- 5b. Sistema exibe mensagem informando que não há dados para serem alterados
- 6b. Retorna ao Fluxo Principal

- **Excluir Projetos**

Ator: *Product Owner*

Pré-condições: Estar autenticado pelo sistema, ter acesso a tela relativa a projetos e existir ao menos 1 (um) projeto cadastrado no sistema

Casos de Uso associados: Listar Projetos

Pós-condições: Os projetos selecionados serão excluídos logicamente do sistema

Fluxo Principal:

1. Usuário seleciona um ou mais projetos a serem excluídos
2. Usuário seleciona opção de exclusão de projeto
3. Sistema exibe informação confirmando a exclusão do(s) projeto(s)

Fluxo Alternativo:

- 3a. Caso haja tarefas não finalizadas para o projeto, usuário poderá escolher entre cancelar as mesmas e proceder com a exclusão ou cancelar a exclusão do projeto
- 4a. Retorna ao Fluxo Principal

#### ● **Listar Grupos**

Ator: Administrador, *Product Owner*, *Scrum Master*, Membro Equipe

Pré-condições: Estar autenticado pelo sistema

Casos de Uso associados: Incluir Grupo, Inserir Integrantes ao Grupo, Editar Informações de Grupo Cadastrado, Excluir Grupo

Pós-condições: NA

Fluxo Principal:

1. Usuário seleciona a tela relativa a grupos
2. Sistema lista os grupos cadastrados dos quais o usuário logado faz parte (no caso do administrador, o sistema lista todos os grupos existentes)

Fluxo Alternativo:

- 3a. Usuário ordena grupos por descrição ou projeto

#### ● **Incluir Grupo**

Ator: Administrador

Pré-condições: Estar autenticado pelo sistema e ter listado grupos

Casos de Uso associados: Listar Grupos, Inserir Integrantes ao Grupo, Excluir Grupos

Pós-condições: Haverá mais um grupo cadastrado no sistema

Fluxo Principal:

1. Usuário seleciona a opção para incluir um grupo
2. Sistema exibe uma linha nova onde estão listados os grupos
3. Usuário preenche informações relativas ao grupo
4. Sistema exibe uma mensagem confirmando a inclusão do grupo

Fluxo Alternativo:

- 4a. Sistema exibe mensagem de erro caso alguma informação esteja incorreta
- 5a. Retorno ao Fluxo Principal
- 5b. Usuário insere integrantes ao grupo criado

#### ● **Editar Dados de Grupo Cadastrado**

Ator: Administrador, *Product Owner*

Pré-condições: Estar autenticado pelo sistema, existir ao menos 1 (um) grupo cadastrado no sistema e ter selecionado grupo

Casos de Uso associados: Listar Grupos

Pós-condições: Dados de um grupo cadastrado no sistema serão alterados

Fluxo Principal:

1. Usuário seleciona opção de edição de dados de grupo
2. Usuário altera dados desejados do grupo
3. Sistema exibe mensagem confirmando a edição de dados da grupo

Fluxo Alternativo:

- 3a. Sistema exibe mensagem de erro, cada alguma informação não esteja correta
- 4a. Retorno ao Fluxo Principal

#### ● **Inserir Integrantes a Grupo**

Ator: Administrador, *Product Owner*, *Scrum Master*

Pré-condições: Estar autenticado pelo sistema, ter listado grupos e existir ao

menos 1 (um) grupo cadastrado no sistema

Casos de Uso associados: Listar Grupos

Pós-condições: Haverá um ou mais integrantes cadastrados em um grupo

Fluxo Principal:

1. Usuário seleciona grupo dentre os listados na tela
2. Usuário seleciona opção de incluir integrantes no grupo selecionado
3. Sistema exibe tela com os usuários existentes e os usuários já relacionados ao grupo, caso existam
4. Usuário adiciona ou exclui usuários do grupo, nas posições desejadas
5. Sistema exibe mensagem confirmando a alteração nos usuários do grupo

#### ● **Excluir Grupos**

Ator: Administrador, *Product Owner*

Pré-condições: Estar autenticado pelo sistema, ter listado os grupos e existir ao menos 1 (um) grupo cadastrado no sistema

Casos de Uso associados: Listar Grupos

Pós-condições: Os grupos selecionados serão excluídos logicamente do sistema

Fluxo Principal:

1. Usuário seleciona um ou mais grupos dentre os listados na tela
2. Usuário seleciona a opção que exclui os grupos selecionados
3. Sistema exibe mensagem informando que grupos foram excluído com sucesso

Fluxo Alternativo:

- 3a. Caso haja sprints e/ou tarefas planejadas para o grupo, o usuário poderá escolher entre excluir a(s) sprint(s) e cancelar as tarefas para proceder com a exclusão ou cancelar a última.
- 4a. Retorna ao Fluxo Principal

#### ● **Listar *Sprints***

Ator: *Product Owner*, *Scrum Master*, Membro Equipe

Pré-condições: Estar autenticado pelo sistema

Casos de Uso associados: Incluir *Sprint*, Editar Dados de *Sprint* Cadastrada, Excluir *Sprints*

Pós-condições: NA

Fluxo Principal:

1. Usuário seleciona tela relativa a *sprints*
2. Sistema exibe *sprints* cadastradas para os grupos dos quais o usuário logado faz parte

Fluxo Alternativo:

- 3a. Usuário ordena *sprints* por Descrição, Data de Início ou Data de Fim

- **Incluir *Sprint***

Ator: *Scrum Master*

Pré-condições: Estar autenticado pelo sistema e ter listado *sprints*

Casos de Uso associados: Listar *Sprints*

Pós-condições: Haverá mais uma *sprint* cadastrada no sistema

Fluxo Principal:

1. Usuário seleciona opção de inclusão de nova *sprint*
2. Sistema exibe uma linha em branco onde as *sprints* estão listadas
3. Usuário preenche dados da *sprint* adicionada
4. Sistema exibe mensagem confirmando a inclusão da nova *sprint*

Fluxo Alternativo:

- 4a. Sistema exibe mensagem de erro, caso alguma informação não esteja correta
- 5a. Retorno ao Fluxo Principal

- **Editar Dados de *Sprint* Cadastrada**

Ator: *Scrum Master*

Pré-condições: Estar autenticado pelo sistema, existir ao menos 1 (uma) *sprint* cadastrada no sistema e ter selecionado *sprint*

Casos de Uso associados: Listar *Sprints*

Pós-condições: Dados de uma *sprint* cadastrada no sistema serão alterados

Fluxo Principal:

1. Usuário seleciona opção de edição de dados de *sprint*
2. Usuário altera dados desejados da *sprint*
3. Sistema exhibe mensagem confirmando a edição de dados da *sprint*

Fluxo Alternativo:

- 3a. Sistema exhibe mensagem de erro, caso alguma informação não esteja correta
- 4a. Retorno ao Fluxo Principal

- **Excluir *Sprints***

Ator: *Scrum Master*

Pré-condições: Estar autenticado pelo sistema, ter listado *sprints* e existir ao menos 1 (uma) *sprints* cadastrada no sistema

Casos de Uso associados: Listar *sprints*

Pós-condições: As *sprints* selecionadas serão excluídas logicamente do sistema

Fluxo Principal:

1. Usuário seleciona uma ou mais *sprints* dentre as listadas na tela
2. Usuário seleciona a opção de exclusão das *sprints* selecionadas
3. Sistema exhibe mensagem confirmando a exclusão das *sprints*

Fluxo Alternativo:

- 3a. Caso haja tarefas planejadas para a *sprint*, o usuário deverá escolher entre cancelar as tarefas planejadas para proceder com a exclusão da *sprint* ou cancelar a exclusão da última
- 4a. Retorna ao Fluxo Principal

- **Visualizar Tarefas**

Ator: *Product Owner, Scrum Master, Membro Equipe*

Pré-condições: Estar autenticado pelo sistema

Casos de Uso associados: Filtrar Tarefas, Incluir Tarefa, Editar Dados de Tarefa Cadastrada, Excluir Tarefas, Exibir Tarefas Não Concluídas

Pós-condições: NA

Fluxo Principal:

1. Usuário seleciona tela relativa a tarefas
2. Sistema exibe tela relativa a tarefas, com todos os filtros que podem ser utilizados pelo usuário

#### ● **Filtrar Tarefas**

Ator: *Product Owner, Scrum Master*, Membro Equipe

Pré-condições: Estar autenticado pelo sistema e ter acessado tela de tarefas

Casos de Uso associados: Incluir Tarefa, Editar Dados de Tarefa Cadastrada, Excluir Tarefas, Exibir Tarefas Não Concluídas

Pós-condições: NA

Fluxo Principal:

1. Usuário seleciona valores a serem levados em consideração nos filtros disponíveis, que serão: Título, Prioridade, Grupo, Status, Responsável, *Sprint*, Esforço e Observação
2. Sistema exibe as tarefa de acordo com os filtros realizados pelo usuário

Fluxo Alternativo:

- 3a. Usuário ordena tarefas pelo título

#### ● **Incluir Tarefa**

Ator: *Scrum Master*

Pré-condições: Estar autenticado pelo sistema e ter listado tarefas

Casos de Uso associados: Filtrar Tarefas

Pós-condições: Haverá mais uma tarefa cadastrada no sistema

Fluxo Principal:

1. Usuário seleciona opção de inclusão de nova tarefa
2. Sistema exibe uma nova linha onde o usuário desejar
3. Usuário preenche formulário com informações da tarefa

4. Sistema exibe mensagem confirmando inclusão da nova tarefa

Fluxo Alternativo:

- 4a. Sistema exibe mensagem de erro, caso alguma informação dada pelo usuário esteja incorreta
- 5a. Retorno ao Fluxo Principal

- **Delegar *Sprints* a Tarefa**

Ator: *Scrum Master*

Pré-condições: Estar autenticado pelo sistema, ter listado tarefas e haver ao menos uma tarefa cadastrada

Casos de Uso associados: Filtrar Tarefas

Pós-condições: Haverá uma ou mais *sprints* em que a tarefa deve ser executada

Fluxo Principal:

1. Usuário seleciona uma tarefa listada na tela
2. Usuário seleciona opção de delegar *sprints* a tarefa
3. Sistema exibe formulário contendo as *sprints* existentes para o grupo e possíveis esforços planejado e real
4. Usuário seleciona *sprints* em que a tarefa deve ser executada, assim como seus respectivos esforços planejados (ou real também, caso tarefa já tenha sido executada) e solicita o armazenamento dos dados
5. Sistema exibe mensagem confirmando a persistência dos dados

- **Tornar Tarefa(s) Filha(s) de Outra Tarefa**

Ator: *Scrum Master*

Pré-condições: Estar autenticado pelo sistema, haver pelo menos 2 (duas) tarefas cadastradas no sistema e ter filtrado tarefas

Casos de Uso associados: Filtrar tarefas

Pós-condições: Haverá uma tarefa relacionada como filha de outra tarefa

Fluxo Principal:

1. Usuário seleciona tarefa(s) que deve(m) se tornar filha(s) da tarefa imediatamente acima

2. Usuário seleciona opção de tornar tarefa(s) filha(s) da tarefa acima
3. Sistema exibe mensagem confirmando a relação entre as tarefas

- **Desfazer Vínculo entre Tarefas**

Ator: *Scrum Master*

Pré-condições: Estar autenticado pelo sistema, haver pelo menos 2 (duas) tarefas cadastradas no sistema e haver relação de pai e filha entre as tarefas

Casos de Uso associados: Filtrar tarefas

Pós-condições: Tarefa antes relacionada como filha, será irmã de outra tarefa

Fluxo Principal:

1. Usuário seleciona tarefa(s) filha(s) que deve(m) ter o vínculo desfeito
2. Usuário seleciona opção de desfazer vínculo entre tarefas pai e filha
3. Sistema exibe mensagem confirmando a operação

- **Editar Dados de Tarefa Cadastrada**

Ator: *Scrum Master*

Pré-condições: Estar autenticado pelo sistema, existir ao menos 1 (uma) tarefa cadastrada no sistema e ter selecionado tarefa

Casos de Uso associados: Listar tarefas

Pós-condições: Dados de uma tarefa cadastrada no sistema serão alterado

Fluxo Principal:

1. Usuário seleciona opção de edição de dados de tarefa
2. Usuário edita informações desejadas da tarefa
3. Sistema exibe mensagem confirmando edição de dados da tarefa

Fluxo Alternativo:

- 3a. Sistema exibe mensagem de erro, caso alguma informação dada pelo usuário esteja incorreta
- 4a. Retorno ao Fluxo Principal

- **Excluir Tarefas**

Ator: *Scrum Master*

Pré-condições: Estar autenticado pelo sistema, ter listado tarefas e existir ao menos 1 (uma) tarefa cadastrada no sistema

Casos de Uso associados: Listar tarefas

Pós-condições: As tarefas selecionadas serão excluídas logicamente do sistema

Fluxo Principal:

1. Usuário seleciona uma ou mais tarefas dentre as exibidas na tela
2. Usuário seleciona a opção de exclusão das tarefas selecionadas
3. Sistema exibe mensagem confirmando a exclusão das tarefas selecionadas

### ● **Criar ou Editar *Backlog* da *Sprint***

Ator: *Scrum Master*

Pré-condições: Estar autenticado pelo sistema, existir pelo menos 1 *sprint* e uma tarefa cadastradas no sistema

Casos de Uso associados: Importar tarefas por status, Importar tarefa por prioridade, Filtrar tarefas, Editar Dados de tarefa Cadastrada

Pós-condições: Haverá mais uma *sprint* planejada no sistema

Fluxo Principal:

1. Usuário acessa tela relativa ao planejamento da *sprint*
2. Usuário preenche o grupo e a *sprint* que deseja planejar
3. Sistema exibe, caso existam, as tarefas já planejadas para a *sprint* selecionada e seu esforço total planejado
4. Usuário salva informações desejadas

Fluxo Alternativo:

- 4a. Usuário importa tarefas através de status e prioridade
- 5a. Usuário planeja tarefas filtradas para a *sprint* selecionada
- 6a. Retorno ao Fluxo Principal

### ● **Importar Tarefas por Status**

Ator: *Scrum Master*

Pré-condições: Estar autenticado pelo sistema e ter acessado criação ou edição

de *backlog* de *sprint*

Casos de Uso associados: Criar ou Editar *Backlog* de *Sprint* tarefas

Pós-condições: NA

Fluxo Principal:

1. Usuário seleciona opção de importação de tarefas por status
2. Sistema exibe tarefas cadastradas no sistema, planejadas para *sprints* anteriores que ainda não foram concluídas

Fluxo Alternativo:

- 3a. Usuário planeja atividades importadas para *sprint* selecionada

#### ● **Importar Tarefas por Prioridade**

Ator: *Scrum Master*

Pré-condições: Estar autenticado pelo sistema e ter acessado criação ou edição de *backlog* de *sprint*

Casos de Uso associados: Criar ou Editar *Backlog* de *Sprint* tarefas

Pós-condições: NA

Fluxo Principal:

1. Usuário seleciona opção de importação de tarefas por prioridade
2. Sistema exibe tela para que usuário preencha o valor de prioridade mínimo desejado
3. Sistema exibe tarefas cadastradas no sistema ainda não concluídas com os prioridade igual ou maior que o selecionado pelo usuário

### **Diagrama de Classes**

O diagrama de classes pode ser visualizado através da figura 4.5.

Um maior detalhamento das classes e seus respectivos dicionários de dados serão apresentados na Descrição de Projeto de Software.

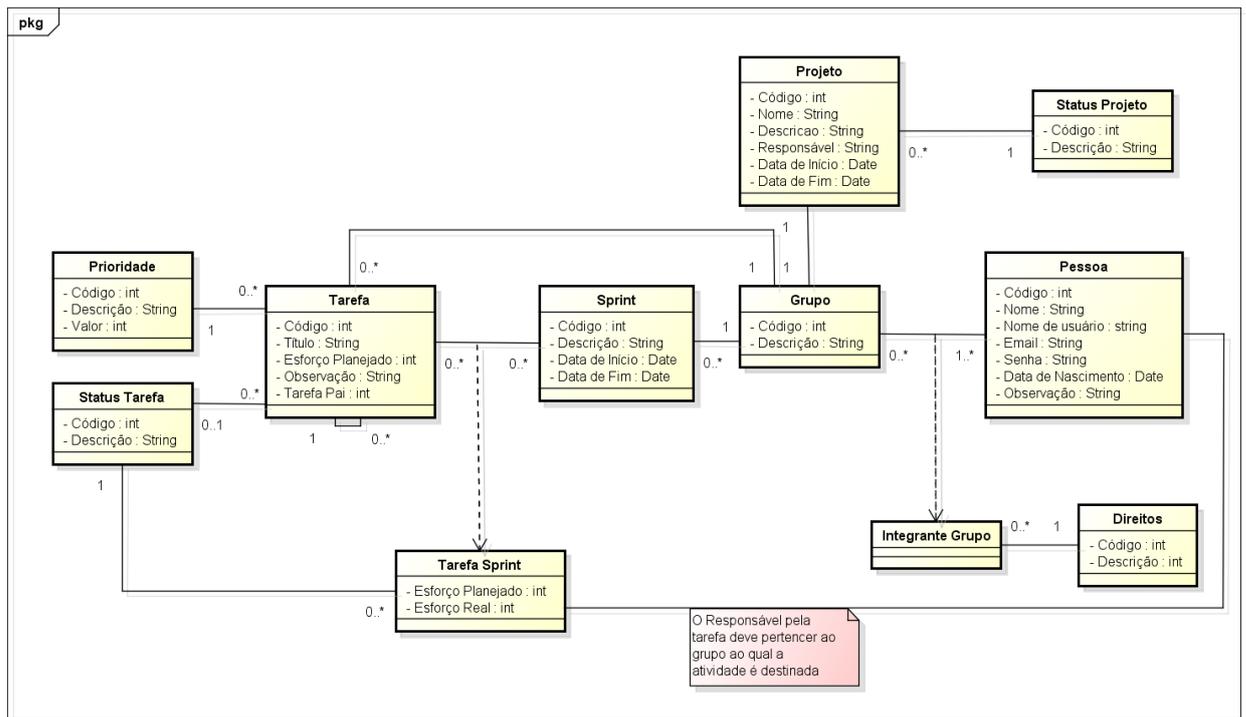


Figura 4.5: Diagrama de Classes.

### 4.3.3 Requisitos de Desempenho

Visto que a aplicação desenvolvida será rodada localmente, não é necessário o suporte de vários usuários simultaneamente, entretando, o servidor de banco de dados usado deverá permitir acesso de diferentes usuários.

Além disso, fica estabelecido o tempo máximo para consultas e persistência no banco de dados de 1 segundo.

### 4.3.4 Restrições de Projeto

Devido à escolha das ferramentas *Microsoft Visual Studio* e *Microsoft SQL Server*, o projeto será compatível apenas com sistema operacional Windows.

### 4.3.5 Atributos

Os atributos principais do sistemas são:

- Funcionabilidade

O iPlan será capaz de realizar com sucesso, precisão e segurança todas as funções propostas neste documento.

- Confiabilidade

Após o desenvolvimento da aplicação serão feitos repetidos testes (previstos do Plano de Testes) para que o sistema não apresente erros.

- Usabilidade

Para o público para o qual foi idealizado, a utilização do sistema a ser desenvolvido será facilmente compreendida.

Ainda, o *software* será implementado de forma que o aprendizado do usuário e sua operabilidade sejam facilitadas.

Como todo sistema amigável, é esperado que o usuário possa usar o iPlan tendo apenas lido o Manual do Usuário.

- Manutenibilidade

Durante o desenvolvimento do sistema, serão tomadas medidas para que este seja facilmente analisado, testado e modificado, caso necessário.

- Segurança

Dado que os usuários do sistema devem ser autenticados para desfrutar de suas funcionalidades, os dados salvos apenas serão acessados por pessoas autorizadas.

### **4.3.6 Outros Requisitos**

Para acessar o sistema, os usuários deverão estar cadastrados no mesmo e ser autenticados através de nome de usuário e senha.

# Capítulo 5

## Descrição de Projeto de Software

### 5.1 Introdução

#### 5.1.1 Finalidade

O presente capítulo tem como finalidade especificar a arquitetura do software desenvolvido, além de estruturar o projeto para facilitar o desenvolvimento do mesmo.

#### 5.1.2 Escopo

Através deste ficarão definidas as estruturas de cada módulo do sistema desenvolvido, além de ser apresentado o modelo de banco de dados a ser usado e os principais métodos utilizados para realizar a persistência de informações no mesmo, proporcionando, assim, um melhor entendimento do projeto ao leitor.

Vela ressaltar que as classes, seus atributos e métodos, assim como as correspondentes tabelas no banco de dados foram projetados em inglês por uma questão de universalização do código fonte.

#### 5.1.3 Definições, Acrônimos e Abreviaturas

As definições necessárias foram feitas no início desta documentação.

## 5.2 Referências

As referências bibliográficas encontram-se no final deste documento.

## 5.3 Decomposição

### 5.3.1 Decomposição em Módulos

Para elaborar a arquitetura do projeto desenvolvido foi usada a metodologia MVC (*Model-View-Control*).

De uma maneira geral, as camadas *Model*, *View* e *Control* serão utilizadas da seguinte forma:

- ***View***

Representa a interface do usuário com o sistema. Esta camada é codificada inteiramente em C#, utilizando os componentes *Form* e *UserControl*.

Os componentes nomeados da forma [nome]Form, [nome]RegisterForm e [nome]UserControl obedecem à arquitetura representada na figura 5.1.

Os componentes genéricos *BaseUserControl*, *BaseForm* e *BaseGridForm* foram criados a fim de evitar repetições desnecessárias no código, agregando, assim, as funcionalidades básicas comuns a todos os componentes que destes derivam.

- ***Control***

Contém as regras de negócios a serem utilizadas, além de manipulação de dados enviados ao banco de dados e recebidos do mesmo. Assim como a camada anterior, é codificada em C#.

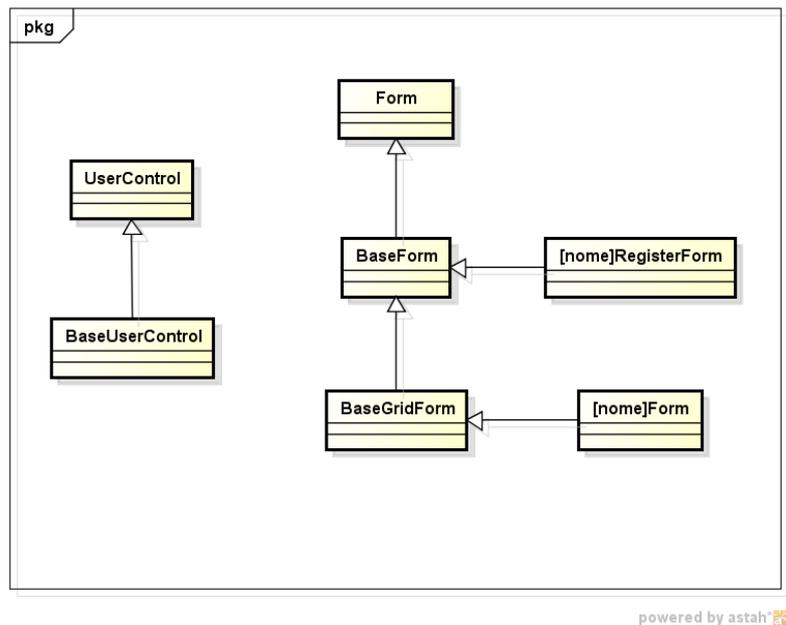


Figura 5.1: Estrutura de *Forms* e *UserControls*.

Pelo fato de o iPlan ser um cliente-servidor, parte das regras de negócio ficará no banco de dados, em *stored procedures*, para viabilizar uma boa performance.

- **Model**

Contém a classe que realiza a conexão com o banco de dados, assim como as classes de entidade. Igualmente às demais camadas, é codificada em C#.

De forma semelhante ao módulo de interface com o usuário, há uma classe base denominada *BaseEntity* que agrega as funcionalidades úteis e comuns a todas as classes de entidade.

A descrição dos módulos que compõem o sistema segue nos itens abaixo.

### 5.3.1.1 Módulo de Acesso

Este módulo é relativo à funcionalidade de acesso dos usuários ao sistema.

Para garantir a segurança do acesso e dos dados dos usuários, a senha dos mesmos é armazenada no banco de dados criptografada através de SHA (*Secure Hash Algorithm*).

A arquitetura deste módulo está representada pela figura 5.2.

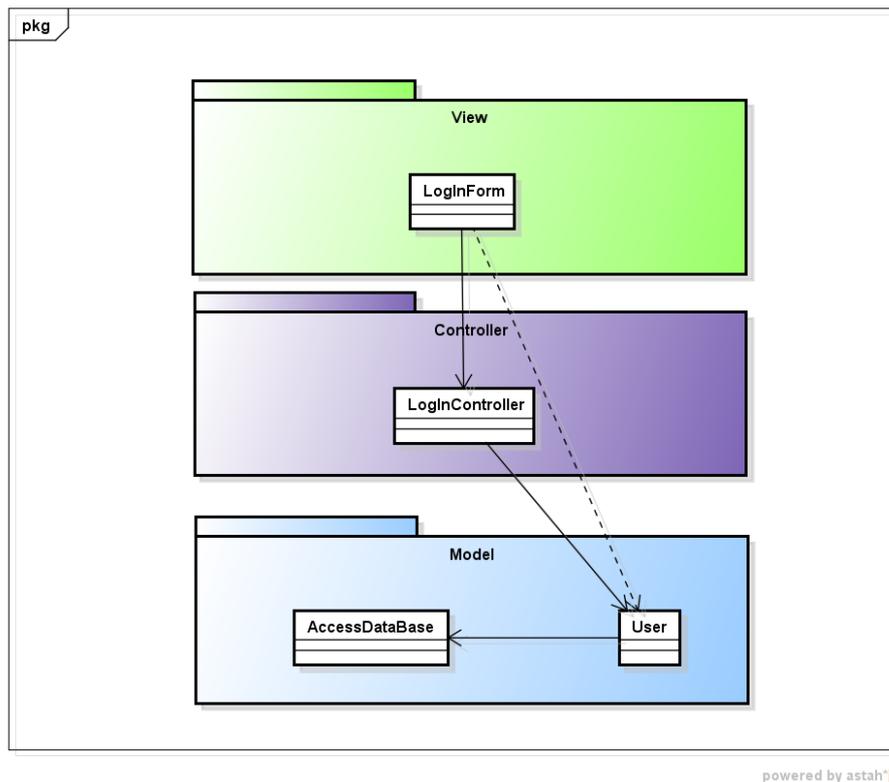


Figura 5.2: Arquitetura do Módulo de Acesso.

### 5.3.1.2 Módulo de Equipe

Este módulo agrega todas as funcionalidades relativas a usuários e grupos, sendo essas:

- Visualização de Usuários
- Cadastro de Usuários
- Edição de Dados Usuários
- Exclusão de Usuários
- Visualização de Grupos
- Cadastro de Grupos
- Edição de Dados Grupos

- Adição de Integrantes a Grupo
- Exclusão de Grupos

A arquitetura do módulo de equipe está apresentado na figura 5.3.

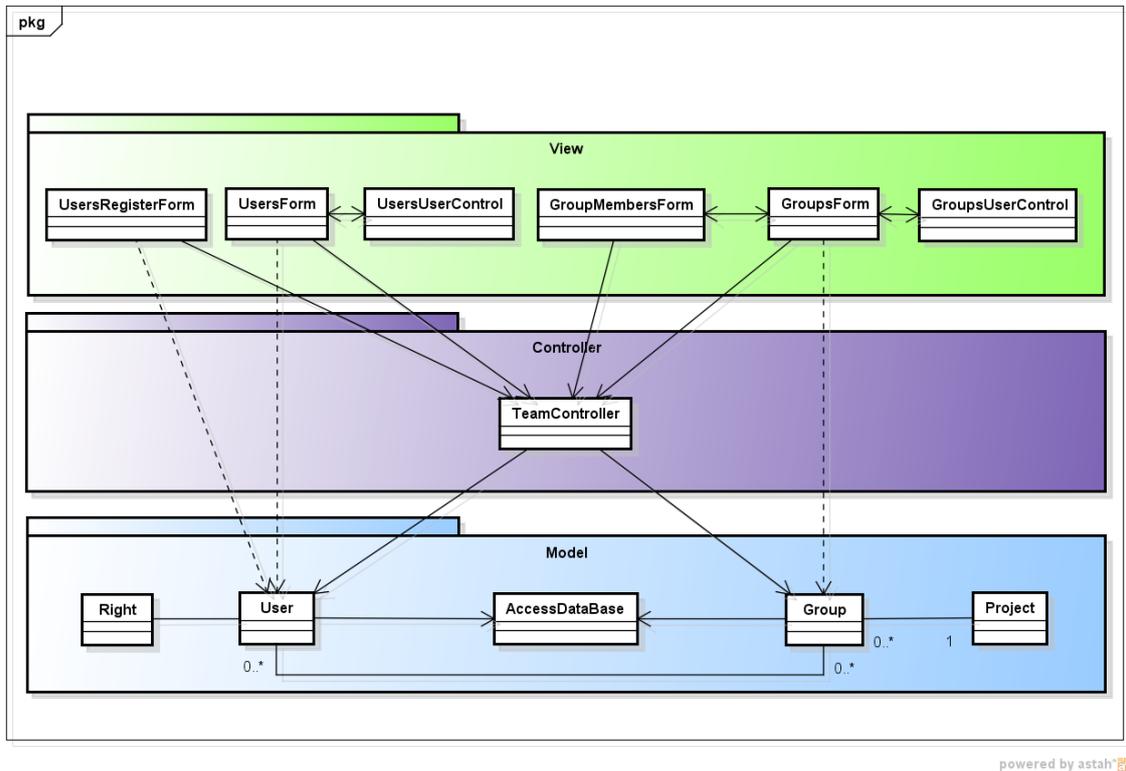


Figura 5.3: Arquitetura do Módulo de Equipe.

### 5.3.1.3 Módulo de Projeto

Este módulo agrega as funcionalidades relativas a projetos, sendo estas:

- Visualização de Projetos
- Inclusão de Projetos
- Edição de Dados de Projetos
- Exclusão de Projetos

A arquitetura do módulo de projeto está apresentado na figura 5.4.

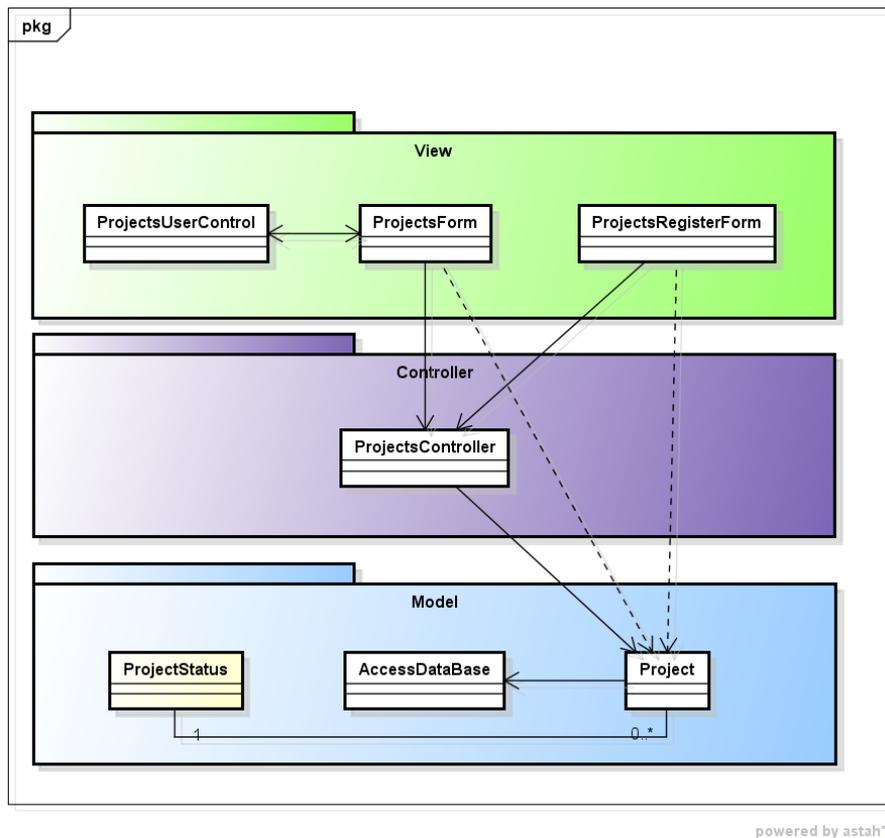
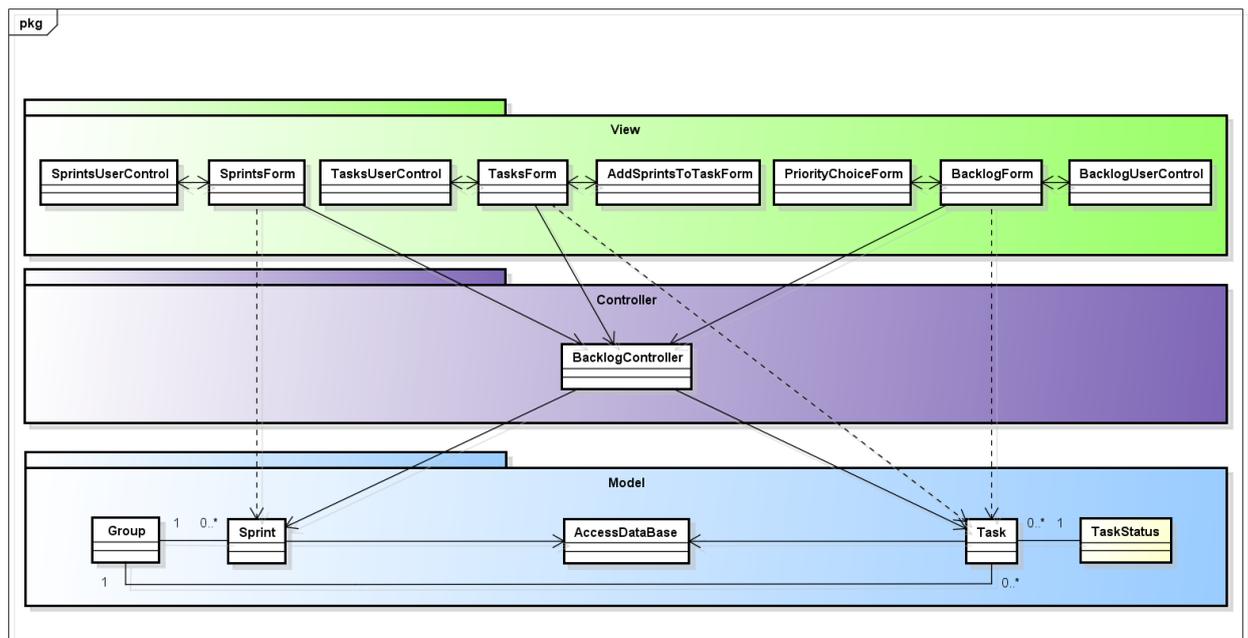


Figura 5.4: Arquitetura do Módulo de Projeto.

#### 5.3.1.4 Módulo de Backlog

Este módulo agrega todas a funcionalidades relativas a tarefas, sprints e planejamento de sprints, sendo estas;

- Visualização de *Sprint*
- Inclusão de *Sprint*
- Edição de Dados de *Sprint*
- Exclusão de *Sprint*
- Visualização de Tarefas
- Inclusão de Tarefas
- Atribuição de *Sprints* a Tarefa
- Edição de Dados de Tarefas



powered by astah

Figura 5.5: Arquitetura do Módulo de *Backlog*.

- Exclusão de Tarefas
- Criar *Backlog* de *Sprint*
- Importação de Tarefas por Status
- Importação de Tarefas por Prioridade

A arquitetura do presente módulo está apresentada na figura 5.5.

### 5.3.2 Decomposição em Processos Concorrentes

O único caso onde há processos concorrentes é o de múltiplos usuários acessando o banco de dados. Os múltiplos acessos são controlados pelo próprio SGBD.

### 5.3.3 Decomposição de Dados

Na figura 5.6 está apresentado diagrama de classes do iPlan.

As classes e seus atributos estão especificados nos itens a seguir.

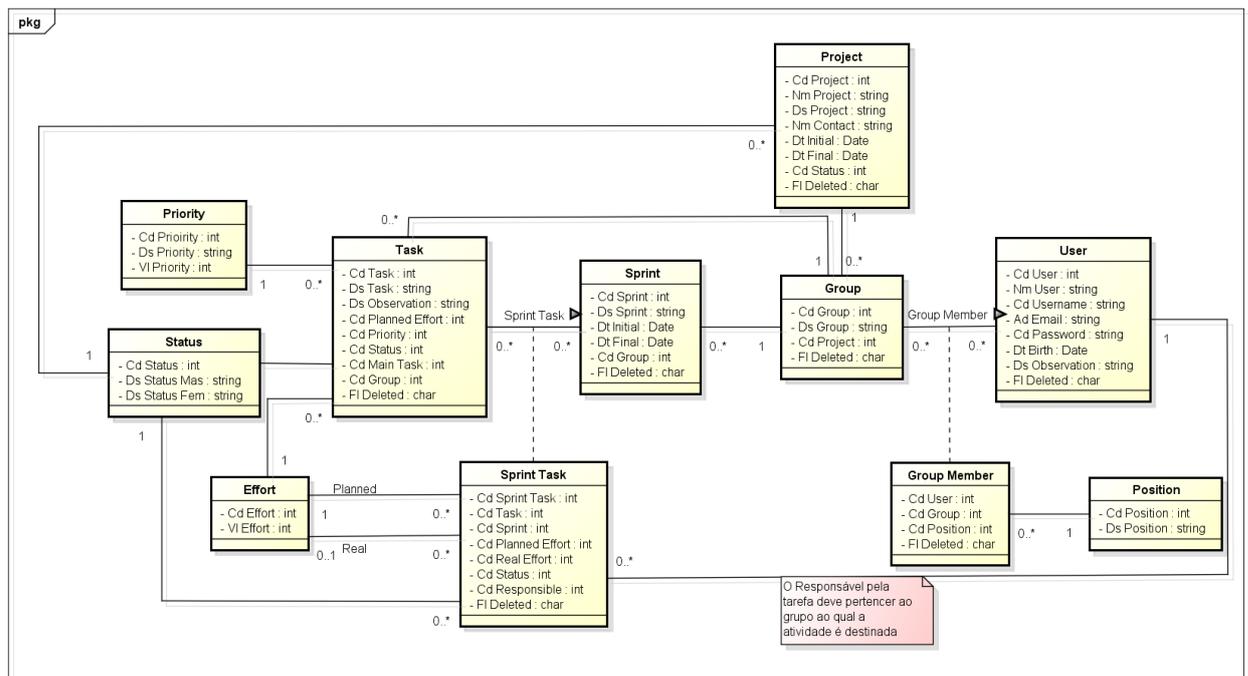


Figura 5.6: Diagrama de classes.

Com exceção das classes "Priority", "Effort", "Status" e "Position", todas as classes possuem o atributo "Fl Deleted", que consiste de um indicador de exclusão, já que, quando excluídos, os dados serão excluídos do sistema logicamente, e não fisicamente.

### 5.3.3.1 Classe User

Representa uma pessoa cadastrada no sistema.

Como a autenticação é necessária para acessar o iPlan, somente as pessoas cadastradas no sistema podem desfrutar de suas funcionalidades.

### 5.3.3.2 Classe Group

Representa uma pequena associação de pessoas que formam uma equipe necessária para o desenvolvimento de projetos ou partes destes utilizando *Scrum*.

<b><i>Atributo</i></b>	<b>Descrição</b>
<i>Cd User</i>	Código numérico único identificador de pessoa
<i>Nm User</i>	Denominação que identifica a pessoa em seu documento de identidade
<i>Cd User-name</i>	Combinação de caracteres única que representa o nome pelo qual o usuário será reconhecido no iPlan
<i>Ad Email</i>	Indicação de correio eletrônico
<i>Cd Pass-word</i>	Combinação de símbolos que contém a senha criptografada do usuário
<i>Dt Birth</i>	Indicação de dia, mês e ano do nascimento da pessoa
<i>Ds Observation</i>	Nota aditiva sobre a pessoa

Tabela 5.1: Atributos da Classe *User*

<b><i>Atributo</i></b>	<b>Descrição</b>
<i>Cd Group</i>	Código numérico único identificador do grupo de pessoas
<i>Ds Group</i>	Narração pormenorizada do grupo
<i>Cd Project</i>	Código relativo ao projeto ao qual o grupo está relacionado

Tabela 5.2: Atributos da Classe *Group*

### 5.3.3.3 Classe *Position*

Representa o papel que uma pessoa cadastrada exerce dentro de um grupo do sistema. Como certas funcionalidades estão disponíveis apenas para certos usuários (por exemplo: Planejamento de *Sprint* disponível apenas para *Scrum Masters*), é necessário verificar a função do usuário no grupo.

Os valores possíveis para esta classe são "Administrador", "Product Owner", "Scrum Master" e "Membro da Equipe".

<b><i>Atributo</i></b>	<b>Descrição</b>
<i>Cd Position</i>	Código numérico único identificador do papel do integrante de um grupo no sistema
<i>Ds Position</i>	Narrativa pormenorizada do papel do integrante de um grupo no sistema

Tabela 5.3: Atributos da Classe *Position*

#### 5.3.3.4 Classe *Group Member*

Representa a associação entre usuários do sistema em um grupo e a função exercida no último.

<b><i>Atributo</i></b>	<b>Descrição</b>
<i>Cd User</i>	Código referente ao usuário na associação
<i>Cd Group</i>	Código referente ao grupo na associação
<i>Cd Position</i>	Código referente à posição do usuário no grupo

Tabela 5.4: Atributos da Classe *Group Member*

#### 5.3.3.5 Classe *Sprint*

Representa o período de tempo básico de desenvolvimento e entrega de um novo produto no *Scrum*.

<b><i>Atributo</i></b>	<b>Descrição</b>
<i>Cd Sprint</i>	Código numérico único identificador da <i>sprint</i>
<i>Ds Sprint</i>	Narração pormenorizada da <i>sprint</i>
<i>Dt Initial</i>	Indicação de dia, mês e ano que marca o início da <i>sprint</i> cadastrada
<i>Dt Final</i>	Indicação de dia, mês e ano que marca o fim da <i>sprint</i> cadastrada
<i>Cd Group</i>	Código referente ao grupo para o qual a <i>sprint</i> foi planejada

Tabela 5.5: Atributos da Classe *Sprint*

### 5.3.3.6 Classe *Project*

Representa o projeto cadastrado no sistema, a ser desenvolvido por um ou mais grupos cadastrados.

<b><i>Atributo</i></b>	<b><i>Descrição</i></b>
<i>Cd Project</i>	Código numérico único identificador do projeto
<i>Nm Project</i>	Denominação dada ao projeto
<i>Ds Project</i>	Narração pormenorizada do projeto
<i>Nm Contact</i>	Nome da pessoa que deve ser contactada para levantar requisitos ou para sanar quaisquer dúvidas
<i>Dt Initial</i>	Indicação de dia, mês e ano de início do projeto
<i>Dt Final</i>	Indicação estimada de dia, mês e ano para que o projeto acabe
<i>Cd Status</i>	Código referente ao estado atual do projeto

Tabela 5.6: Atributos da Classe *Project*

### 5.3.3.7 Classe *Status*

Representa o estado atual de um projeto ou tarefa, podendo assumir valores fixos entre eles, "Em Andamento", "Concluído", "Cancelado".

<b><i>Atributo</i></b>	<b><i>Descrição</i></b>
<i>Cd Status</i>	Código numérico único identificador do estado
<i>Ds Status</i>	Narração pormenorizada do estado

Tabela 5.7: Atributos da Classe *Status*

### 5.3.3.8 Classe *Priority*

Representa a prioridade atribuída a tarefas ao serem planejadas.

Prioridades podem assumir valores fixos múltiplos de 100, sendo o menor deles 100 e o maior, 1000.

<i>Atributo</i>	Descrição
<i>Cd Priority</i>	Código numérico único identificador da prioridade
<i>Ds Priority</i>	Narrativa pormenorizada da prioridade.
<i>Vl Priority</i>	Valor numérico assumido por uma prioridade

Tabela 5.8: Atributos da Classe *Priority*

### 5.3.3.9 Classe *Effort*

Representa a medida do esforço para realizar tarefas, levando-se em consideração sua dificuldade e o tempo estimado para realizá-las.

No caso iPlan os valores possíveis para esforço serão números pertencentes a uma série de *Fibonacci* adaptada, sendo eles: 0, 1, 2, 3, 5, 8, 13, 20, 40, 60 e 100.

<i>Atributo</i>	Descrição
<i>Cd Effort</i>	Código numérico único identificador do esforço
<i>Vl Effort</i>	Valor numérico assumido pelo esforço

Tabela 5.9: Atributos da Classe *Effort*

### 5.3.3.10 Classe *Task*

Representa uma tarefa de um projeto a ser desenvolvida em uma ou mais *sprints* por membros da equipe.

### 5.3.3.11 Classe *Sprint Task*

Representa uma tarefa a ser realizada em uma *sprint*.

## 5.4 Descrição das Dependências

### 5.4.1 Dependências entre módulos

O módulo de acesso, bem como o módulo de projeto são independentes dos demais.

O módulo de equipe depende da classe "Projeto", pertencente ao módulo de projeto. A dependência é justificada pelo fato de os grupos serem pertencentes a projetos.

<b><i>Atributo</i></b>	<b>Descrição</b>
<i>Cd Task</i>	Código numérico único identificador da tarefa
<i>Ds Task</i>	Narração pormenorizada da tarefa
<i>Ds Observation</i>	Nota aditiva sobre a tarefa
<i>Cd Planned Effort</i>	Código relativo ao esforço estimado necessário para completar uma tarefa
<i>Cd Priority</i>	Código relativo à prioridade da tarefa
<i>Cd Status</i>	Código relativo à situação da tarefa
<i>Cd Main Task</i>	Código relativo à tarefa pai da tarefa, se existir
<i>Cd Group</i>	Código relativo ao grupo para o qual a tarefa foi planejada

Tabela 5.10: Atributos da Classe *Task*

<b><i>Atributo</i></b>	<b>Descrição</b>
<i>Cd Sprint Task</i>	Código numérico único identificador da associação entre tarefa e <i>sprint</i>
<i>Cd Task</i>	Código relativo à tarefa
<i>Cd Sprint</i>	Código relativo à <i>sprint</i> para a qual a tarefa foi planejada
<i>Cd Planned Effort</i>	Código relativo ao esforço estimado para realizar a tarefa em dada <i>sprint</i>
<i>Cd Real Effort</i>	Código relativo ao esforço real utilizado para executar a tarefa em dada <i>sprint</i>
<i>Cd Status</i>	Código relativo à situação da tarefa na <i>sprint</i>
<i>Cd Responsible</i>	Código relativo ao usuário responsável por executar a tarefa em dada <i>sprint</i>

Tabela 5.11: Atributos da Classe *Sprint Task*

Por fim, o módulo de *backlog* depende do módulo de equipe, o que o torna dependente também do módulo de projeto.

As dependências acima explicitadas podem ser observadas através da figura 5.6.

### **5.4.2 Dependências entre processos**

Não se aplica.

### **5.4.3 Dependências entre dados**

Como foi escolhida uma linguagem orientada a objeto, as dependências entre dados estão também representadas no item "Dependências entre módulos".

## **5.5 Descrição das Interfaces**

As interfaces do sistema com o usuários serão representadas no Manual do Usuário. Já as interfaces entre os módulos está representada no item "Projeto Detalhados dos Módulos".

## **5.6 Projeto Detalhado**

### **5.6.1 Projeto Detalhado dos Módulos**

A arquitetura dos módulos foram especificados no item "Decomposição em Módulos" e pode ser melhor compreendida através do diagrama de atividade e do diagrama de sequência do caso de uso "Incluir Tarefa", ilustrados nas figuras 5.7 e 5.8, respectivamente.

### **5.6.2 Projeto Detalhado das Entidades de Dados**

O modelo de banco de dados criado para atender às necessidades do iPlan está ilustrado lógica e fisicamente nas figuras 5.9 e 5.10, respectivamente.

A seguir encontram-se descritos os métodos utilizados pelas classes reais.

Para todas as classes haverá um método para aquisição de dados relativos à classe da seguinte forma:

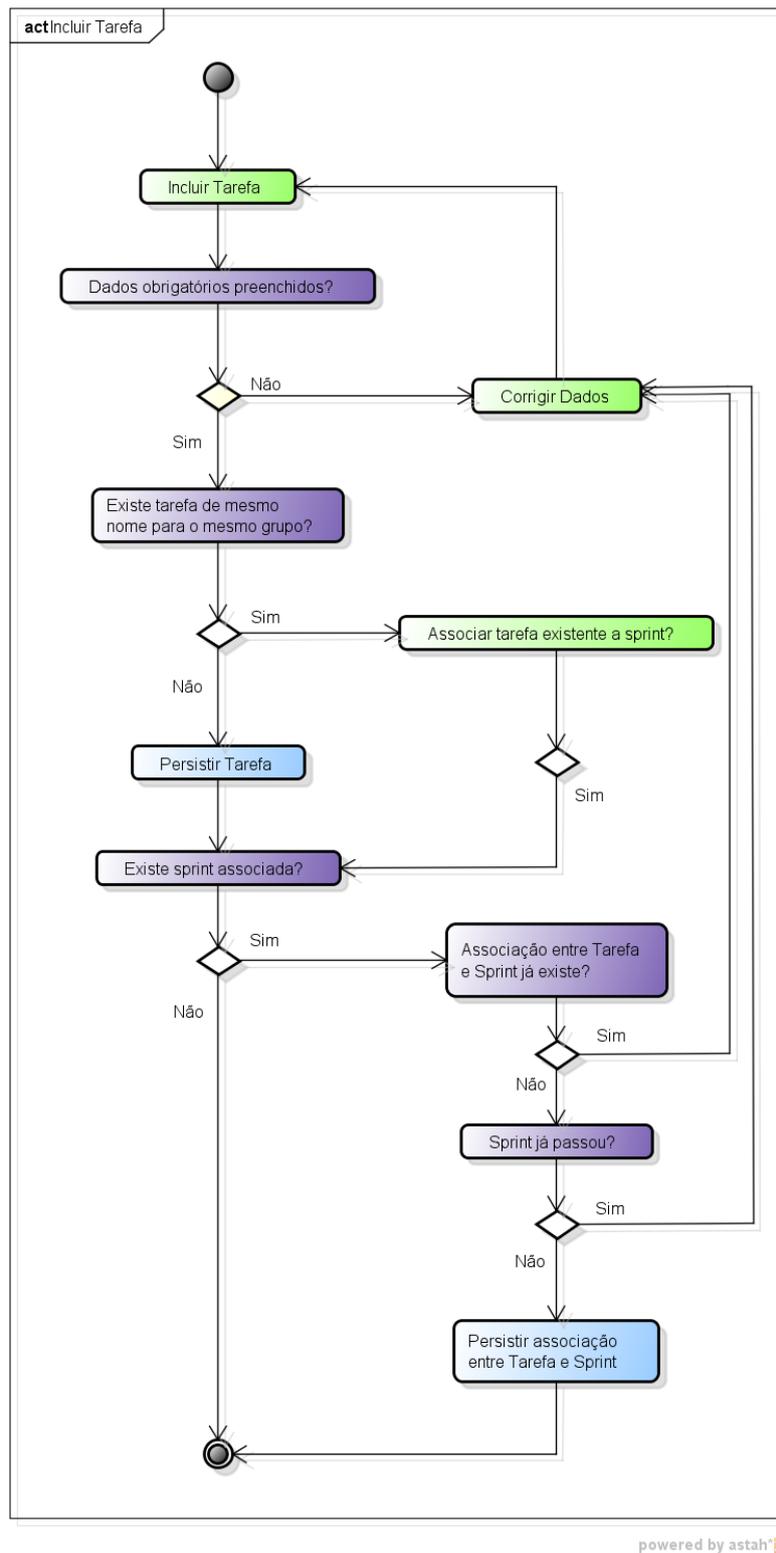
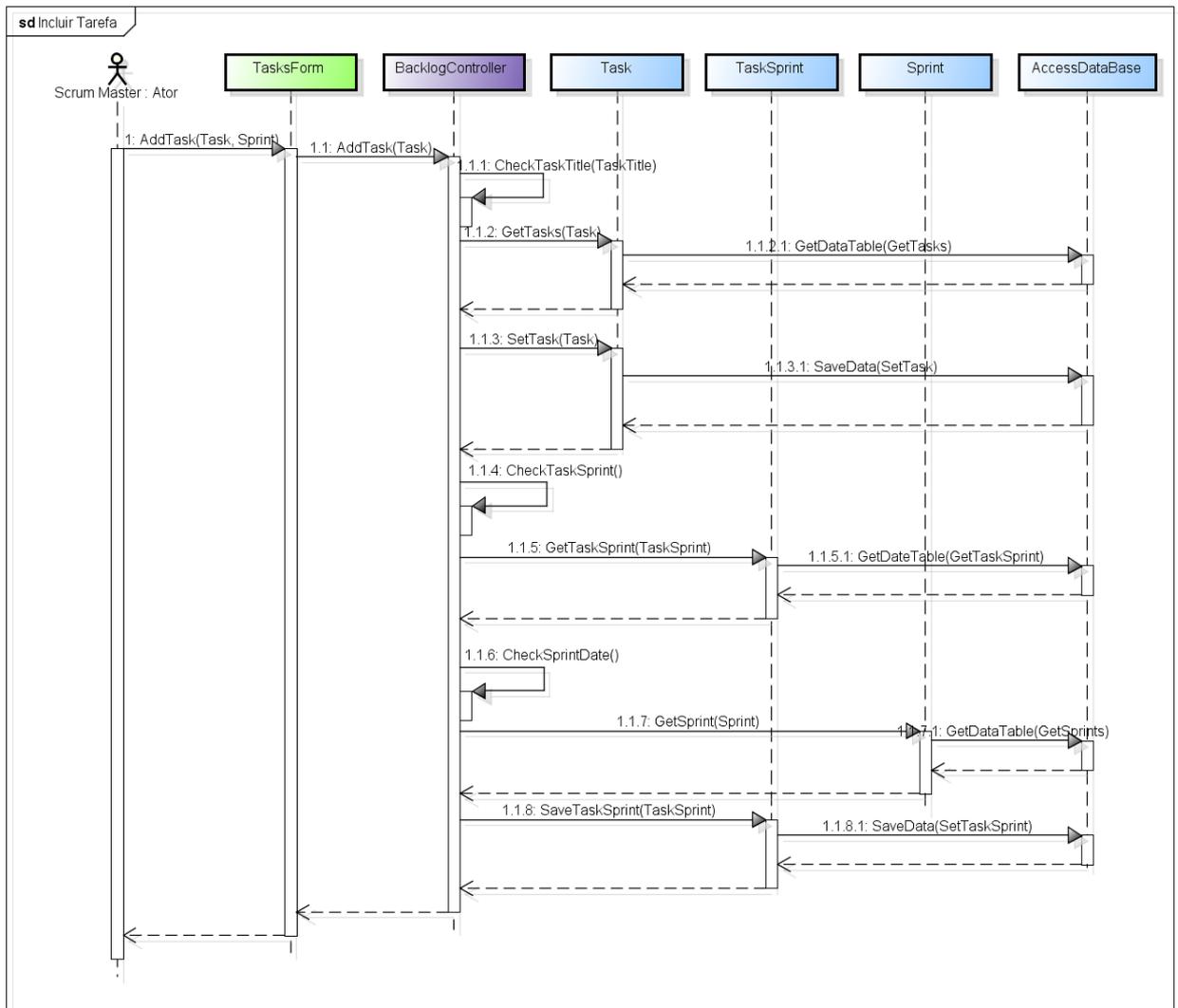


Figura 5.7: Diagrama de tarefa para incluir uma nova tarefa.

- **List<T> GetList(T object)**

onde  $T$  é a classe e *object* é o objeto da classe com os parâmetros a serem buscados, caso existam.



powered by astah

Figura 5.8: Diagrama de sequência para incluir uma nova tarefa.

Para as classes *Position*, *Priority*, *Effort* e *Status*, o método acima descrito será o único existente, já que novos valores não podem ser salvos para tais classes.

Para as demais classes, há ainda os métodos genéricos de persistência de dados descritos abaixo.

- **int SetObject(T object)**

onde *T* é a classe cujo objeto será persistido e *object* é o objeto em si. O valor retornado é usado para verificar se a persistência de dados foi realizada com sucesso.

- **List<int> SetListObjects(List<T> listObjects)**

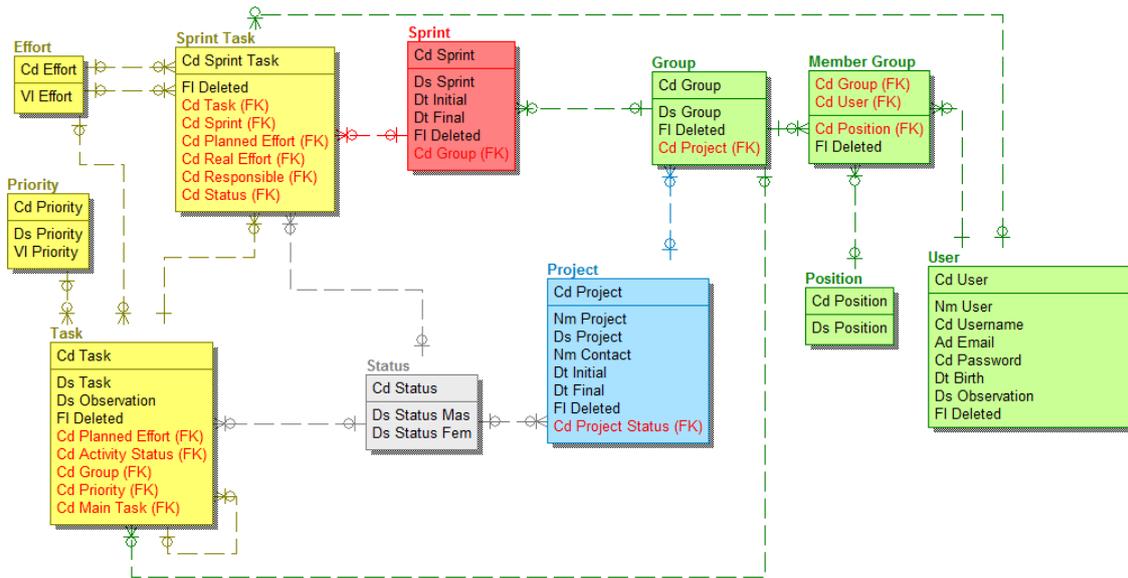


Figura 5.9: Modelo Lógico do Banco de Dados para o iPlan.

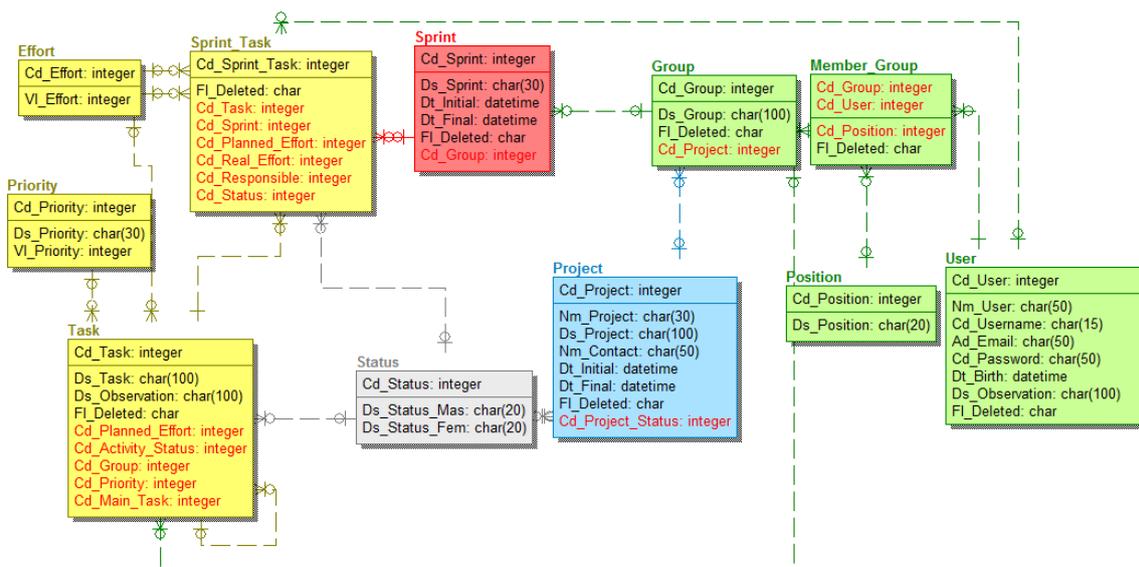


Figura 5.10: Modelo Físico do Banco de Dados para o iPlan.

Tem seu mecanismo similar ao método anterior, sendo a única diferença a persistência de múltiplos objetos e, portanto, o retorno também múltiplo.

### 5.6.2.1 Classe *User*

- `List<User> GetUsersInGroup(int groupCode, string position)`

Busca todos os usuários pertencentes ao grupo cujo identificador é passado como argumento e que ocupam a posição também passada. Caso seja desejado

buscar todos os usuários, independente da posição, basta passar uma string vazia para este campo.

- **DataSet GetUsersInGroups()**

Busca todos os usuários que são membros de quaisquer grupos. O DataSet é organizado agrupando-se usuários do mesmo grupo na mesma tabela.

#### 5.6.2.2 Classe *Group*

- **List<Group> GetUserGroups(int userCode)**

Busca todos os grupos dos quais um usuário faz parte, onde *userCode* é o identificador do usuário.

#### 5.6.2.3 Classe *GroupMember*

Esta classe possui somente os métodos genéricos.

#### 5.6.2.4 Classe *Sprint*

- **List<Sprint> GetSprints(Sprint searchSprint, DateTime beginInitialDate, DateTime endInitialDate)**

Busca todas as *sprints* de acordo com os dados preenchidos no objeto passado e cuja data de início está contida no intervalo passado como argumento.

- **string CheckSprintData(Sprint saveSprint)**

Verifica dados da *sprint* que será salva, tais como descrição da *sprint* e se a data de início ou fim está dentro de outra *sprint* do mesmo grupo. O parâmetro *saveSprint* contem os dados da *sprint* a ser verificada, enquanto o retorno é a mensagem de erro, caso exista.

#### 5.6.2.5 Classe *Project*

- **List<Project> GetProjects(Project searchProject, DateTime beginFinalDate, DateTime endFinalDate)**

Busca todos os projetos de acordo com os dados preenchidos no objeto passado como argumento e cuja data prevista de término está contida no intervalo referenciado.

- **List<Project> GetUserProjects(int userCode)**

Busca todos os projetos dos quais um usuário faz parte, sendo *userCode* o identificador do usuário.

#### 5.6.2.6 Classe *Task*

- **DataTable GetProductBacklogTasks(Task searchTask, SprintTask searchSprintTask)**

Busca todas as tarefas do *product backlog* cujos dados são iguais aos dados passados como parâmetros através dos objetos *searchTask* e *searchSprintTask*.

- **DataTable GetSprintBacklogTasks(int sprintCode)**

Busca todas as tarefas planejadas para dada *sprint*, onde *sprintCode* é o código da *sprint*.

- **List<Task> GetLateTasks(int sprintCode)**

Busca todas as tarefas que foram planejadas para *sprints* anteriores à *sprint* passada como parâmetro, mas não foram concluídas.

- **DataTable GetTasksByPriority(int groupCode, int priorityCode)**

Busca tarefas ainda não concluídas de determinado grupo cuja prioridade é superior à prioridade passada como parâmetro, onde *groupCode* é o código do grupo e *priorityCode* é o código da prioridade.

#### 5.6.2.7 Classe *Sprint Task*

Esta classe possui somente os métodos genéricos.

# Capítulo 6

## Plano de Testes

### 6.1 Introdução

#### 6.1.1 Identificador

PTiPlan

#### 6.1.2 Finalidade

Este plano de testes é dirigido ao professor Antônio Cláudio Gómez de Sousa, orientador deste Projeto de Graduação, e à banca examinadora do mesmo. O software a ser testado é denominado iPlan.

Os testes aqui propostos serão realizados para a apresentação do projeto desenvolvido à banca examinadora.

#### 6.1.3 Referências

As referências bibliográficas encontram-se no final deste documento.

### 6.2 Descrição Geral

#### 6.2.1 Itens

Os itens do projeto a serem verificados serão:

- Arquitetura. Será aceito se cada módulo da arquitetura estiver funcionando, independente dos outros e cada funcionalidade proposta nos módulos for executada com sucesso.
- Comunicação com o Banco de Dados. Será aceito se todos os dados forem armazenados satisfatoriamente e se as consultas corresponderem ao esperado.
- Validação dos campos digitados pelo usuário em formulários de cadastro. Será aceito se as mensagens exibidas pelo sistema após o preenchimento dos campos referidos for a esperada.

### **6.2.2 Requisitos**

Os requisitos estão previamente definidos no capítulo "Especificação de Requisitos de Software", além de os critérios estarem definidos na programação do software. Serão considerados aceitos os itens que corresponderem devidamente às expectativas.

### **6.2.3 Visão Geral**

Para testar os algoritmos desenvolvidos serão feitas funções testes, a fim de inserir dados conhecidos no sistema e serão observados os resultados obtidos através destas.

Os testes serão realizados e repetidos quantas vezes forem necessárias, fazendo com que todas as possibilidades de erro sejam testadas e aprovadas. O processo de testes será terminado quando os resultados obtidos foram satisfatórios para todos os testes feitos.

### **6.2.4 Suspensão ou Conclusão**

Não há previsão de suspensão dos testes a serem realizados. A conclusão destes se dará quando os resultados obtidos forem satisfatórios, portanto, quando todas as falhas que venham a ser encontradas, sejam devidamente corrigidas.

### **6.2.5 Ambiente**

Para a execução do iPlan, e portanto, para a realização de seus testes, são necessários:

- Sistema Operacional Windows
- Servidor de Banco de Dados SQL Server
- Microsoft Visual Studio

## 6.2.6 Tarefas e Cronograma

As tarefas e o cronograma dos testes realizados estão representados na tabela 6.1.

Item testado	Data
Arquitetura	05 - 07 de Fevereiro
Comunicação com o Banco de Dados	08 - 09 de Fevereiro
Validação dos Campos digitados	10 - 11 de Fevereiro

Tabela 6.1: Cronograma.

## 6.2.7 Riscos e Gerenciamento

Os riscos que serão considerados para os testes são:

- Imperícia ao realizar o teste
- Indisponibilidade de tempo

Para evitar os riscos acima citados, as possíveis falhas identificadas deverão ser investigadas no momento de sua ocorrência e corrigidas o mais rápido possível.

## 6.3 Especificação dos Testes

### 6.3.1 Especificação 1

#### 6.3.1.1 Identificador

Arquitetura

### 6.3.1.2 Caracaterísticas

Nesta especificação será testada a arquitetura do sistema. Os itens a serem testados são:

- Tela 'Usuários' Será verificado se as funcionalidades de "Filtrar", "Salvar", "Adicionar Usuário", "Remover Usuário", "Desfazer Remoção de Usuário" e "Editar Dados de Usuário" funcionam conforme abaixo especificado:
  - Filtrar: Todos os usuários que atendem aos filtros devem ser exibidos na tela.
  - Salvar: Quando não houver mudanças nos dados, deve ser enviada uma mensagem ao usuário indicando a ausência de dados a serem salvos e nada deve ser feito. Caso contrário, as alterações devem ser salvas no Banco de Dados.
  - Adicionar Usuário: Ao ser acionado o botão relacionado, o sistema deve exibir o formulário de cadastro de usuário em branco. Caso o usuário selecione a opção de salvar os dados preenchidos, o Banco de Dados deve ser atualizado, caso contrário, nada deve ser feito.
  - Remover Usuário: Caso o botão relacionado seja acionado sem que qualquer linha esteja selecionada, o sistema deve exibir uma mensagem de erro ao usuário. Caso contrário, o(s) usuário(s) selecionado(s) deve(m) ser excluído(s) logicamente do sistema.
  - Editar Dados de Usuário: Ao ser acionada a opção de edição de dados, o sistema deve exibir o formulário de cadastro de usuário, com os respectivos dados preenchidos e a opção "Alterar Senha" visível. A seguir, o procedimento deve ser o mesmo para o caso de "Adicionar Usuário".
  - Desfazer Remoção de Usuário: Ao ser acionada a opção de edição de dados de um usuário excluído, o formulário deve exibir o cadastro de usuário com os dados preenchidos, porém com os campos não editáveis e a opção "Desfazer Exclusão" deve estar visível. A seguir o procedimento deve ser o mesmo para o caso de "Adicionar Usuário".

- Tela 'Grupos' Será verificado se as funcionalidades "Filtrar", "Salvar", "Adicionar Grupo", "Remover Grupo" e "Atribuir Usuários a Grupo" funcionam conforme abaixo especificado:
  - Filtrar: Análogo à mesma funcionalidade da tela 'Usuários'.
  - Salvar: Análogo à mesma funcionalidade da tela 'Usuários', com a adição da verificação da existência de linhas ou campos e branco na tela.
  - Adicionar Grupo: Ao ser acionado o botão relacionado, o sistema deve exibir uma linha em branco na tela.
  - Remover Grupo: Análogo à mesma funcionalidade da tela 'Usuários'.
  - Atribuir Usuários a Grupo: Ao ser acionada a opção relacionada, o sistema deve exibir um formulário contendo todos os usuários cadastrado no sistema e os cargos possíveis para serem ocupados no grupo. Ao salvar dados, deve ser verificado se foram selecionados mais de um *Product Owner* ou *ScrumMaster*. Caso haja, deve ser exibida uma mensagem de erro.
  
- Tela 'Projetos' Será verificado se as funcionalidades "Filtrar", "Salvar", "Adicionar Projeto" e "Remover Projeto" funcionam conforme abaixo especificado:
  - Filtrar: Análogo à mesma funcionalidade da tela 'Usuários'.
  - Salvar: Análogo à mesma funcionalidade da tela 'Usuários'.
  - Adicionar Projeto: ao ser acionado o botão relacionado, o sistema deve exibir o formulário de cadastro de projeto em branco. Caso seja selecionada a opção de persistência, os dados digitados devem ser verificados e o Banco de Dados atualizado, caso não haja inconsistências. Caso haja, deve ser exibida mensagem de erro ao usuário indicando o erro.
  - Remover Projeto: Análogo à mesma funcionalidade da tela 'Usuários'.
  
- Tela 'Sprints' Será verificado se as funcionalidades "Filtrar", "Salvar", "Adicionar Sprint" e "Remover Sprint" funcionam conforme abaixo especificado:
  - Filtrar: Análogo à mesma funcionalidade da tela 'Usuários'.
  - Salvar: Análogo à mesma funcionalidade da tela 'Grupos'.

- Adicionar *Sprint*: Análogo à mesma funcionalidade da tela 'Grupos'.
  - Remover *Sprint*: Análogo à mesma funcionalidade da tela 'Usuários'.
- Tela 'Tarefas' Será verificado se as funcionalidades "Filtrar", "Salvar", "Adicionar Tarefa", "Remover Tarefa", "Vincular Tarefa a Outra", "Desfazer Vínculo entre Tarefas" e "Atribuir *Sprints* a Tarefa" funcionam conforme abaixo especificado:
    - Filtrar: Análogo à mesma funcionalidade da tela 'Usuários'.
    - Salvar: Análogo à mesma funcionalidade da tela 'Grupos'.
    - Adicionar Tarefa: Análogo à mesma funcionalidade da tela 'Grupos'.
    - Remover Tarefa: Análogo à mesma funcionalidade da tela 'Usuários'.
    - Vincular Tarefa a Outra: Ao ser acionado o botão correspondente deve ser verificado se há alguma tarefa selecionada e se há alguma tarefa acima da selecionada, para ser pai da primeira. A numeração das tarefas deve ser atualizada.
    - Desfazer Vínculo entre Tarefas: Ao ser acionado o botão correspondente, deve ser verificado se há alguma tarefa selecionada e se esta está vinculada como filha de outra. A numeração das tarefas deve ser atualizada.
    - Atribuir *Sprints* a Tarefa: Ao ser seleciona a opção correspondente deve ser exibido o formulário de atribuição de *sprints*. Ao salvar dados, deve ser verificado se os esforços selecionados foram salvos nas *sprints* corretas.
  - Tela 'Planejar *Sprint*' Será verificado se as funcionalidades "Exibir Tarefas Planejadas", "Salvar", "Remover Tarefa Planejada", "Importar Tarefas Atrasadas" e "Importar Tarefas por Prioridade" funcionam conforme abaixo especificado:
    - Exibir Tarefas Planejadas: Ao ser selecionada uma *sprint*, automaticamente as tarefas para aquela planejadas devem ser carregadas na tela.
    - Salvar: Análogo à mesma funcionalidade da tela 'Usuários'.
    - Remover Tarefa Planejada: Análogo à mesa funcionalidade da tela 'Usuários'.

- Importar Tarefas Atrasadas: Ao ser acionado o botão correspondente, deve ser verificado se todas as tarefas atrasadas (e não canceladas) foram carregadas na tela.
- Importar Tarefas por Prioridade: Ao ser acionado o botão correspondente, o sistema deve exibir uma tela com um valor de prioridade a ser selecionado. Ao acionar o botão 'Ok', deve ser verificado se todas as tarefas de prioridade igual ou superior à selecionada (e não canceladas) foram carregadas na tela.

### **6.3.1.3 Refinamento**

Para este teste, será verificado se o sistema responde de forma correta à entrada de dados inconsistentes, retornando mensagens de erro sempre que necessário. Já na exibição de dados, será verificado se o conteúdo exibido é o correspondente ao armazenado no Banco de Dados.

### **6.3.1.4 Identificador de Caso de Teste**

CT1

## **6.3.2 Especificação 2**

### **6.3.2.1 Identificador**

Comunicação com Banco de Dados

### **6.3.2.2 Características**

Nesta especificação será testada a comunicação com o Banco de Dados. Os itens a ser testados serão:

- Escrita no Banco de Dados

Será verificado se os dados estão sendo escritos no Banco de Dados de forma correta.

- Leitura do Banco de Dados

Será verificado se os dados existentes no Banco de Dados estão sendo lidos de forma correta.

### 6.3.2.3 Refinamento

Para este teste a verificação dos dados gravados no banco de dados se dará visualmente através da ferramenta Microsoft SQL Server. Caso seja encontrada alguma inconsistência no armazenamento ou leitura dos dados esperados e dos dados obtidos, o erro que gerou a falha deverá ser investigado, identificado e corrigido.

### 6.3.2.4 Identificador de Caso de Teste

CT2

## 6.3.3 Especificação 3

### 6.3.3.1 Identificador

Validação de Campos

### 6.3.3.2 Características

Serão verificados os campos de preenchimento pelo usuário dos campos de:

- Inclusão ou edição de dados de usuário
- Inclusão ou edição de dados de projeto
- Inclusão ou edição de dados de grupos
- Inclusão ou edição de dados de *sprints*
- Inclusão ou edição de dados de atividades
- Inclusão ou edição de dados de atividades em uma *sprint*

Os casos acima mencionados estão especificados individualmente a seguir:

#### **Inclusão ou Edição de Dados de Usuário**

Os campos a serem verificados são:

- Nome Deve conter entre 4 e 50 caracteres
- Nome de Usuário Deve conter entre 4 e 15 caracteres, e deve ser verificado se o nome de usuário escolhido já existe no banco de dados.
- Senha Deve conter entre 4 e 15 caracteres
- Confirmação de Senha Deve conter entre 4 e 15 caracteres e ser igual ao item *Senha*
- Data de Nascimento Os valores preenchidos deverão ser selecionados, e não digitados, portanto deve ser apenas verificado se o usuário selecionou valores para dia, mês e ano ou não
- Observação Campo opcional, deve conter no máximo 100 caracteres

### **Inclusão ou Edição de Dados de Projeto**

Os campos a serem verificados são:

- Nome Deve conter entre 4 e 20 caracteres, e deve ser verificado se já existe algum projeto cadastrado com o mesmo código
- Descrição Deve conter entre 4 e 100 caracteres
- Responsável Deve conter entre 4 e 50 caracteres
- Data de Início Os valores preenchidos deverão ser selecionados, e não digitados, portanto deve ser apenas verificado se o usuário selecionou valores para dia, mês e ano ou não
- Data de Fim Assim como no campo acima, os valores devem ser selecionados
- Status Deve ser verificado se o status do projeto foi selecionado

### **Inclusão ou Edição de Dados de Grupos**

Os campos a serem verificados são:

- Descrição Deve conter entre 4 e 50 caracteres
- Projeto Deve ser verificado se o projeto relacionado ao grupo foi selecionado

#### **Inclusão ou Edição de Dados de *Sprints***

Os campos a serem verificados são:

- Descrição Deve conter entre 4 e 20 caracteres e deve ser verificado se já existe alguma *sprint* cadastrada com a mesma descrição para o mesmo grupo
- Grupo Deve ser verificado se o grupo foi selecionado
- Data de Início Deve ser verificado se os valores preenchidos correspondem a uma data
- Data de Fim Assim como no campo acima, os valores preenchidos devem ser uma data

#### **Inclusão ou Edição de Dados de Tarefas**

Os campos a serem verificados são:

- Título Deve conter entre 4 e 100 caracteres e deve ser verificado se já existe alguma tarefa cadastrada para o mesmo projeto com o mesmo título
- Prioridade O valor deve ser selecionado
- Status O valor deve ser selecionado
- Esforço Planejado Campo opcional. Pode ser deixado em branco ou selecionado
- Grupo Deve ser verificado se o grupo foi selecionado
- Observação Campo opcional. Deve ter no máximo 100 caracteres

### **6.3.3.3 Refinamento**

Para testar os campos mencionados, serão preenchidos formulários com valores dentro, fora e nas margens de aceitação, caracterizando teste de caixa preta.

A técnica a ser utilizada para os testes de validação de campos consiste no preenchimento dos campos e envio dos mesmos. A análise deste teste é feita através da mensagem que será apresentada para o usuário.

#### **6.3.3.4 Identificador de Caso de Teste**

CT3

### **6.4 Casos de Teste**

#### **6.4.1 Caso de teste 1**

##### **6.4.1.1 Identificador**

CT1

##### **6.4.1.2 Itens**

Será verificado se os módulos da arquitetura funcionam independentemente um dos outros.

##### **6.4.1.3 Entrada e Saída**

As entradas e saídas esperadas relacionadas à inconsistências de dados de cada funcionalidade estão representadas nas tabelas 6.2, 6.3 e 6.4.

##### **6.4.1.4 Ambiente**

Para executar os testes acima descritos é somente necessário que o sistema esteja sendo executado.

#### **6.4.2 Caso de teste 2**

##### **6.4.2.1 Identificador**

CT2

##### **6.4.2.2 Itens**

Será verificado se a escrita e a leitura de dados no Banco de Dados está sendo feita de forma correta.

<b>Tela 'Usuários'</b>		
<b>Funcionalidade</b>	<b>Entrada</b>	<b>Saída Esperada</b>
Salvar	Nenhuma mudança realizada na tela Mudanças realizadas	Mensagem ausência de dados a serem salvos Dados salvos no sistema
Adicionar Usuário	Nome de usuário escolhido já existente Nome de usuário não cadastrado no sistema	Mensagem de nome de usuário inválido Dados de usuário salvos no sistema
Remover Usuário	Nenhum usuário selecionado  Um ou mais usuários selecionados	Mensagem de ausência de usuários selecionados Usuário removido logicamente do sistema
<b>Tela 'Grupos'</b>		
Salvar	Nenhuma mudança realizada na tela Mudanças realizadas e existência de linhas em branco Mudanças realizadas	Mensagem ausência de dados a serem salvos Mensagem indicando linhas em branco Dados salvos no sistema
Remover Grupo	Nenhum grupo selecionado  Um ou mais grupos selecionados	Mensagem de ausência de grupos selecionados Grupo removido logicamente do sistema
Atribuir Usuários a Grupo	Mais de um <i>Product Owner</i> selecionado Mais de um <i>ScrumMaster</i> selecionado Nenhum ou um <i>Product Owner</i> e <i>ScrumMaster</i> selecionado e nenhum ou mais membros selecionados	Mensagem de limite de um <i>Product Owner</i> Mensagem de limite de um <i>ScrumMaster</i> Usuários adicionados ao grupo

Tabela 6.2: Entradas e Saídas Esperadas do Caso de Teste 1.

#### 6.4.2.3 Entrada e Saída

As entradas e saídas para este caso de teste são livres, sendo a únicas condições de sucesso a leitura e escrita de dados nas tabelas e colunas corretas no banco de dados.

<b>Tela 'Projetos'</b>		
<b>Funcionalidade</b>	<b>Entrada</b>	<b>Saída Esperada</b>
Salvar	Nenhuma mudança realizada na tela Mudanças realizadas	Mensagem ausência de dados a serem salvos Dados salvos no sistema
Adicionar Projeto	Nome de projeto escolhido já existente Nome de projeto não cadastrado no sistema	Mensagem de nome de projeto inválido Dados de projeto salvos no sistema
Remover Projeto	Nenhum projeto selecionado Um ou mais projetos selecionados	Mensagem de ausência de projetos selecionados Projeto removido logicamente do sistema
<b>Tela '<i>Sprints</i>'</b>		
Salvar	Nenhuma mudança realizada na tela Mudanças realizadas e existência de linhas em branco Mudanças realizadas	Mensagem ausência de dados a serem salvos Mensagem indicando linhas em branco Dados salvos no sistema
Remover <i>Sprint</i>	Nenhuma <i>sprint</i> selecionada Uma ou mais <i>sprint</i> selecionadas	Mensagem de ausência de <i>sprints</i> selecionadas <i>Sprint</i> removida logicamente do sistema

Tabela 6.3: Entradas e Saídas Esperadas do Caso de Teste 1.

#### 6.4.2.4 Ambiente

A execução dos testes de verificação da comunicação com o Banco de Dados será feitas em duas etapas:

- Teste das procedures de leitura e escrita Nesse caso o ambiente utilizado será o próprio servido de banco de dados Microsoft SQL Server
- Teste dos valores observados e enviados pela aplicação Nesse caso o ambiente utilizado será o próprio sistema

<b>Tela 'Tarefas'</b>		
<b>Funcionalidade</b>	<b>Entrada</b>	<b>Saída Esperada</b>
Salvar	Nenhuma mudança realizada na tela Mudanças realizadas e existência de linhas em branco Mudanças realizadas	Mensagem ausência de dados a serem salvos Mensagem indicando linhas em branco Dados salvos no sistema
Remover Tarefa	Nenhuma tarefa selecionada  Uma ou mais tarefas selecionadas	Mensagem de ausência de tarefas selecionadas Tarefa removida logicamente do sistema
Vincular Tarefa	Nenhuma tarefa selecionada  Uma ou mais tarefas selecionadas, sem tarefas acima destas Uma ou mais tarefas selecionadas, com tarefas acima destas	Mensagem de ausência de tarefas a ser vinculada Mensagem de ausência de tarefa pai a ser vinculada  Tarefas vinculadas à tarefa acima
Desvincular Tarefas	Nenhuma tarefa selecionada  Uma ou mais tarefas selecionadas, sem tarefa vinculada Uma ou mais tarefas selecionadas, com tarefa vinculada	Mensagem de ausência de tarefas a desvincular Mensagem de ausência de tarefa pai a desvincular  Tarefas desvinculadas da tarefa pai
<b>Tela 'Planejar Sprints'</b>		
Salvar	Nenhuma mudança realizada na tela Mudanças realizadas	Mensagem ausência de dados a serem salvos Dados salvos no sistema
Remover Tarefa	Nenhuma tarefa selecionada  Uma ou mais tarefas selecionadas	Mensagem de ausência de tarefas selecionadas Tarefa removida logicamente da <i>sprint</i>

Tabela 6.4: Entradas e Saídas Esperadas do Caso de Teste 1.

### **6.4.3 Caso de teste 3**

#### **6.4.3.1 Identificador**

CT3

#### **6.4.3.2 Itens**

Será verificado se o software está validando corretamente os campos que devem ser digitados por usuários.

#### **6.4.3.3 Entrada e Saída**

As entradas e saídas esperadas estão representadas nas tabelas 6.5, 6.6, 6.7 e 6.8.

#### **6.4.3.4 Ambiente**

Para executar os testes acima descritos é somente necessário que o sistema esteja sendo executado.

## **6.5 Procedimentos de Teste**

### **6.5.1 Procedimento 1**

#### **6.5.1.1 Identificador**

PT1

#### **6.5.1.2 Finalidade**

Esse procedimento de testes diz respeito à especificação de teste 'Arquitetura' e descreve o procedimento de como devem ser efetuados os testes especificados no caso de teste CT1.

#### **6.5.1.3 Necessidades Especiais**

Não há necessidades especiais.

#### **6.5.1.4 Ações**

Para a correta execução do caso de teste CT1 o testador deverá ter acesso a todas as funcionalidades do sistema.

Deve-se executar cada uma das funcionalidades descritas no CT1 e deve ser observada a saída dada pelo sistema.

#### **6.5.1.5 Relatórios**

Não serão emitidos relatórios.

### **6.5.2 Procedimento 2**

#### **6.5.2.1 Identificador**

PT2

#### **6.5.2.2 Finalidade**

Esse procedimento de teste diz respeito à especificação de teste 'Comunicação com Banco de Dados' e descreve o procedimento de como devem ser efetuados os testes especificados no caso de teste CT2.

#### **6.5.2.3 Necessidades Especiais**

Não há necessidades especiais.

#### **6.5.2.4 Ações**

Para a correta execução do caso de teste CT2 o testador deverá ter acesso a modificar o banco de dados e visualizá-lo.

Deve-se executar funcionalidades que alterem dados no Banco de Dados, como a inclusão de novo usuário e também uma que precisem ler dados contidos no Banco de Dados, como a listagem de atividades cadastradas.

O testador deverá consultar o banco de dados via Microsoft SQL Server e verificar visualmente se os dados conferem.

#### **6.5.2.5 Relatórios**

Não serão emitidos relatórios.

### **6.5.3 Procedimento 3**

#### **6.5.3.1 Identificador**

PT3

#### **6.5.3.2 Finalidade**

Esse procedimento de teste diz respeito à especificação de teste 'Validação de Campos' e descreve o procedimento de como devem ser efetuados os testes especificados no caso de teste CT3.

#### **6.5.3.3 Necessidades Especiais**

Não há necessidades especiais.

#### **6.5.3.4 Ações**

Para a correta execução do caso de teste CT3 e evitar perdas de tempo desnecessariamente, o testador deverá isolar a camada de interface com o usuário, preencher os dados contidos na coluna "Entrada" das tabelas 6.5, 6.6, 6.7 e 6.8 e verificar as saídas obtidas.

Deve-se proceder preenchendo um ou mais campos incorretamente por funcionalidade, para testar a eficiência do sistema em detectar múltiplos erros.

#### **6.5.3.5 Relatórios**

Não serão emitidos relatórios.

Campos Relativos a Usuários		
Campo	Entrada Recebida	Saída Esperada
Nome	Campo em Branco String com menos de 3 caracteres String de comprimento permitido String com mais de 50 caracteres	Mensagem de campo em branco Mensagem de Nome muito pequeno Procedimento ok  Mensagem de Nome muito grande
Nome de Usuário	Campo em Branco String com menos de 3 caracteres String de comprimento permitido String com mais de 15 caracteres	Mensagem de campo em branco Mensagem de nome de usuário muito pequeno Procedimento ok  Mensagem de nome de usuário muito grande
Senha	Campo em Branco String com menos de 4 caracteres String de comprimento permitido String com mais de 15 caracteres	Mensagem de campo em branco Mensagem de senha muito curta  Procedimento ok  Mensagem de senha muito longa
Confirmação de Senha	Campo em Branco String diferente da existente no campo <i>Senha</i> String igual à existente no campo <i>Senha</i>	Mensagem de campo em branco Mensagem de <i>Senha</i> e <i>Confirmação de senha</i> diferentes Procedimento ok
Data de Nascimento	Valor igual à data de hoje  Valor diferente da data de hoje	Mensagem de Data de Nascimento não selecionada Procedimento ok
Observação	Campo em Branco String de comprimento permitido String contendo mais de 100 caracteres	Procedimento ok Procedimento ok  Mensagem de <i>Observação</i> muito longa

Tabela 6.5: Entradas e Saídas Esperadas do Caso de Teste 3.

<b>Campos Relativos a Projetos</b>		
<b>Campo</b>	<b>Entrada Recebida</b>	<b>Saída Esperada</b>
Nome	Campo em branco String com menos de 4 caracteres String de comprimento permitido String com mais de 20 caracteres	Mensagem de campo em branco Mensagem de <i>Nome</i> muito curto  Procedimento ok  Mensagem de <i>Nome</i> muito longo
Descrição	Campo em branco String com menos de 4 caracteres String de comprimento permitido String contendo mais de 100 caracteres	Mensagem de campo em branco Mensagem de <i>Descrição</i> muito curta Procedimento ok  Mensagem de <i>Descrição</i> muito longa
Data de Início	Qualquer valor selecionado	Procedimento ok
Data de Fim	Valor selecionado inferior à data de início Valor selecionado igual à data de início Valor selecionado superior à data de início	Mensagem de data de fim inferior à data de início Mensagem de data de início e fim iguais Procedimento ok
Status	Campo em Branco Status selecionado	Mensagem de campo em branco Procedimento ok
<b>Campos Relativos a Grupos de Usuários</b>		
Descrição	Campo em branco String com menos de 4 caracteres String de comprimento permitido String com mais de 50 caracteres	Mensagem de campo em branco Mensagem de <i>Descrição</i> muito curta Procedimento ok  Mensagem de <i>Descrição</i> muito longa
Projeto	Campo em Branco Projeto selecionado	Mensagem de campo em branco Procedimento ok

Tabela 6.6: Entradas e Saídas Esperadas do Caso de Teste 3.

<b>Campos Relativos a <i>Sprints</i></b>		
<b>Campo</b>	<b>Entrada Recebida</b>	<b>Saída Esperada</b>
Descrição	Campo em branco String com menos de 4 caracteres String de comprimento permitido String com mais de 20 caracteres	Mensagem de campo em branco Mensagem de <i>Descrição</i> muito curta Procedimento ok Mensagem de <i>Descrição</i> muito longa
Grupo	Campo em Branco Grupo selecionado	Mensagem de campo em branco Procedimento ok
Data de Início	Campo em Branco String qualquer  String no formato dd/mm/yyyy	Mensagem de campo em branco Mensagem de <i>Data de Início</i> fora do formato de data Procedimento ok
Data de Fim	Campo em Branco String qualquer  String no formato dd/mm/yyyy inferior à data de início String no formato dd/mm/yyyy igual à data de início String no formato dd/mm/yyyy superior à data de início	Mensagem de campo em branco Mensagem de <i>Data de Fim</i> fora do formato de data Mensagem de <i>Data de Fim</i> inferior à data de início Mensagem de <i>Data de Fim</i> igual à data de início Procedimento ok

Tabela 6.7: Entradas e Saídas Esperadas do Caso de Teste 3.

<b>Campos Relativos a Tarefas</b>		
<b>Campo</b>	<b>Entrada Recebida</b>	<b>Saída Esperada</b>
Título	Campo em branco String com menos de 4 caracteres String de comprimento permitido String contendo mais de 100 caracteres	Mensagem de campo em branco Mensagem de <i>Título</i> muito curto  Procedimento ok  Mensagem de <i>Título</i> muito longo
Prioridade	Campo em Branco Prioridade selecionada	Mensagem de campo em branco Procedimento ok
Status	Campo em Branco Status selecionado	Procedimento ok Procedimento ok
Esforço Planejado	Campo em Branco Esforço Planejado selecionado	Procedimento ok Procedimento ok
Observação	Campo em Branco String de comprimento permitido String contendo mais de 100 caracteres	Procedimento ok Procedimento ok  Mensagem de <i>Observação</i> muito grande
<b>Campos Relativos a Tarefas em <i>Sprint</i></b>		
<i>Sprint</i>	Nenhuma <i>sprint</i> selecionada <i>Sprint</i> selecionada	Procedimento ok Procedimento ok
Esforço Planejado	Campo em Branco Esforço Planejado selecionado	Mensagem de campo em branco Procedimento ok
Esforço Real	Campo em Branco Esforço Real selecionado	Procedimento ok Procedimento ok

Tabela 6.8: Entradas e Saídas Esperadas do Caso de Teste 3.

# Capítulo 7

## Manual Do Usuário

### 7.1 Introdução

Este manual foi elaborado com o propósito de orientar o usuário do iPlan na utilização do sistema. Serão abordados os modos de uso de cada funcionalidade existente.

### 7.2 Descrição

O iPlan visa auxiliar usuários da metodologia ágil *Scrum*, principalmente os *Scrum Masters*, a planejarem e acompanharem o andamento das tarefas necessárias ao desenvolvimento de um ou mais produtos.

Para ser utilizado por mais de uma pessoa, o iPlan deve ser acessado através da internet e o banco de dados deve ser compartilhado por todos os usuários.

### 7.3 Utilização do iPlan

#### 7.3.1 Acesso

No momento do primeiro acesso do usuário ao iPlan, será feito um *download* da aplicação para sua máquina, podendo, então, o acesso ser feito localmente. Porém, para consultar e salvar dados é necessário o acesso à internet ou à rede onde o sistema será disponibilizado.

Todos os usuários do sistema devem ter nome de usuário e senha próprios e, para tanto, devem ser cadastrados por um administrador.

## 7.3.2 Página Principal

Após fazer o *login* no sistema, o usuário visualizará a tela representada na figura 7.1.

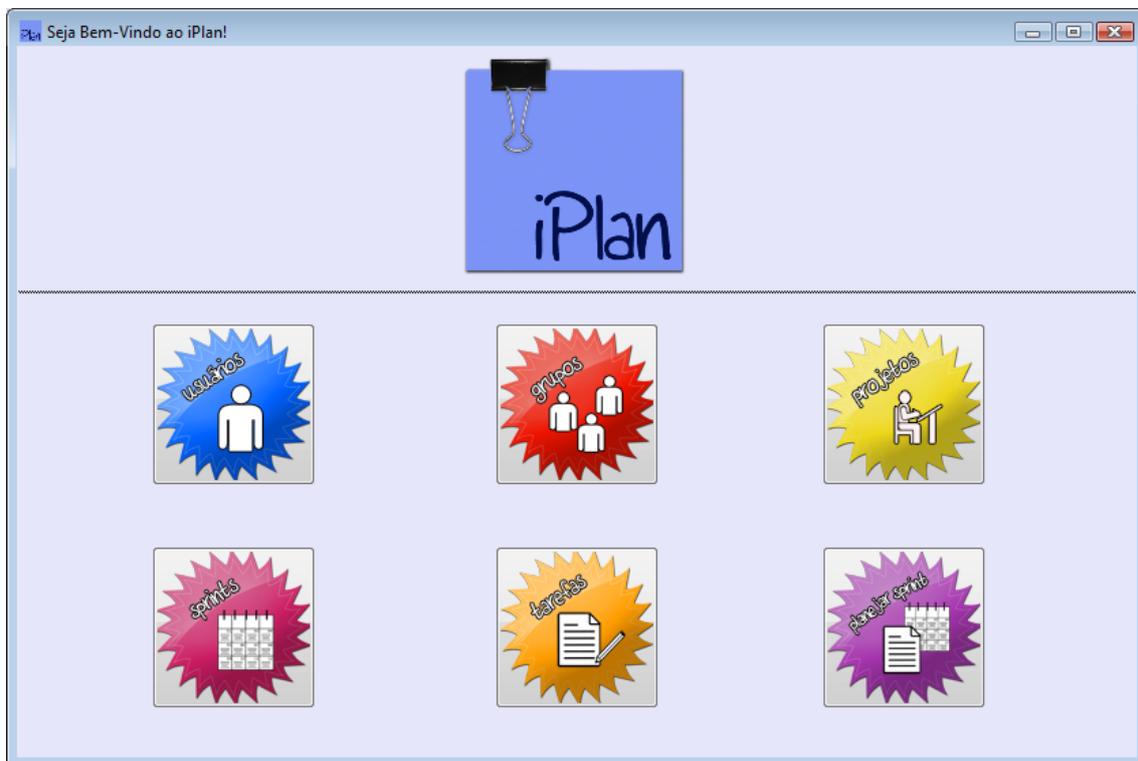


Figura 7.1: Página principal do iPlan.

As demais funcionalidades serão acessadas através dos botões contidos nesta página.

## 7.3.3 Usuários

Acessando o módulo de usuários, o usuário visualizará a tela representada na figura 7.2. Através desta tela, é possível realizar as seguintes operações:

- Listar ou Buscar Usuários
- Adicionar Novo Usuário
- Editar Dados de um Usuário Dadastrado
- Excluir ou Desfazer Exclusão de Usuário

Os botões destacados na figura 7.2 significam:

- 1 - Salvar

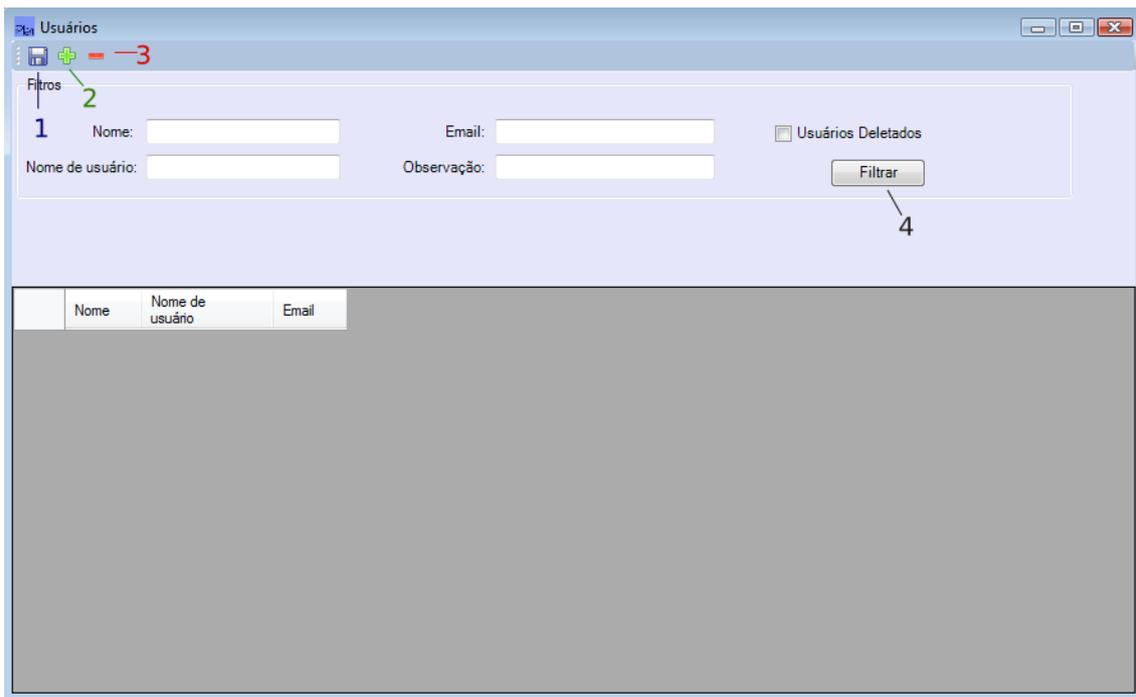


Figura 7.2: Modulo de Usuários.

- 2 - Adicionar
- 3 - Excluir
- 4 - Filtrar

Cada uma das operações citadas está detalhada a seguir.

### 7.3.3.1 Listar ou Buscar Usuários

Para listar todos os usuários do sistema ou buscar por alguns usuários especificamente, basta preencher os filtros desejados e pressionar o botão "Filtrar".

Após esse passo, os usuários cadastrados no sistema serão exibidos, conforme representado na figura 7.3.

Para procurar também pelos usuários que eventualmente foram excluídos do sistema, é preciso selecionar a opção "Usuários Deletados".

### 7.3.3.2 Adicionar Novo Usuário

Para adicionar novos usuários ao sistema, é necessário clicar no botão "Adicionar".

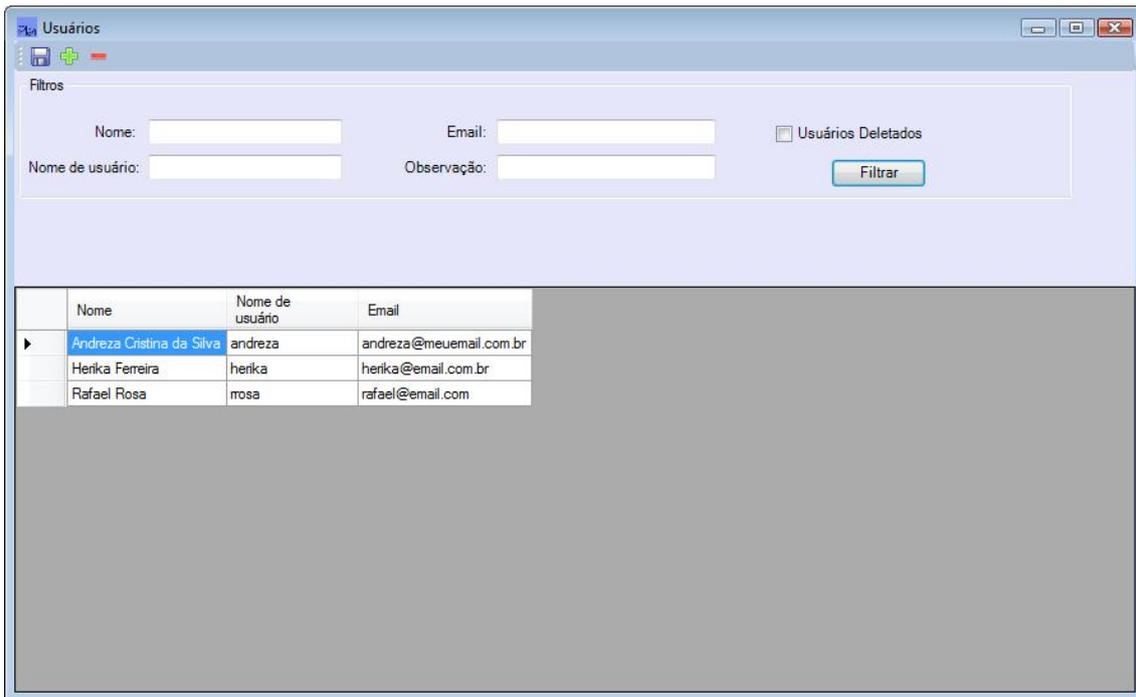


Figura 7.3: Usuários listados.

Ao solicitar a adição de um novo usuário, é exibido o formulário representado na figura 7.4.

The screenshot shows a web application window titled "Cadastro de Usuário". The form contains the following fields: "Nome:", "Nome de Usuário:", "Email:", "Data de Nascimento:" (with a date picker set to "Monday, February 07, 2011"), "Observação:", "Senha:", and "Confirmação de Senha:". There is a save icon in the top left corner of the window.

Figura 7.4: Formulário para a adição de novo usuário.

Após o preenchimento dos campos, os dados devem ser salvos, através do botão localizado no canto superior esquerdo da tela. Caso haja algum erro ou inconsistência nos

dados preenchidos, será exibida uma mensagem notificando o erro ocorrido, caso contrário, o usuário será avisado que os dados foram salvos com sucesso.

### 7.3.3.3 Editar Dados de um Usuário Cadastrado

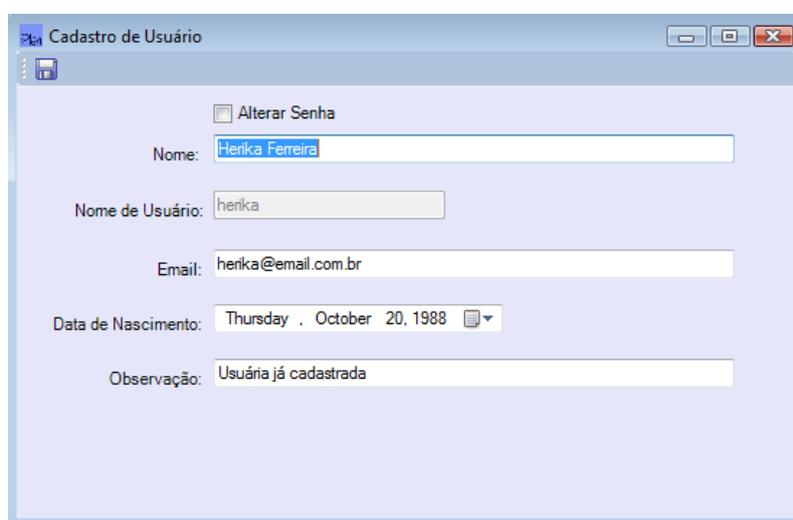
Para editar dados de um usuário, este deve ser listado na tela e, então, é necessário clicar duas vezes na coluna em branco que precede a coluna de "Nome", na linha correspondente ao usuário cujos dados serão alterados.

A seguir, o sistema exibe o formulário de cadastro, com os dados atuais preenchidos. Basta, então modificar os dados desejados e salvar as alterações, do mesmo modo que é feita a inclusão de um novo usuário.

Segue um exemplo ilustrativo:

#### Alteração de dados do usuário "herika"

Com os dados dos usuário exibidos na tela, conforme a figura 7.3, clica-se duas vezes na coluna em branco à esquerda, na segunda linha e o sistema exibe o formulário da figura 7.5.



A imagem mostra uma janela de software com o título "Cadastro de Usuário". No topo, há uma barra de ferramentas com ícones de minimizar, maximizar e fechar. Abaixo, há um formulário com os seguintes campos:

- Uma caixa de seleção desativada rotulada "Alterar Senha".
- Um campo de texto rotulado "Nome:" com o valor "Herika Ferreira".
- Um campo de texto rotulado "Nome de Usuário:" com o valor "herika".
- Um campo de texto rotulado "Email:" com o valor "herika@email.com.br".
- Um campo de texto rotulado "Data de Nascimento:" com o valor "Thursday, October 20, 1988" e um ícone de calendário.
- Um campo de texto rotulado "Observação:" com o valor "Usuária já cadastrada".

Figura 7.5: Formulário para a edição de usuário cadastrado.

Como é possível ver, o campo nome de usuário, que funciona como a identidade do usuário no iPlan, não pode ser alterado. Para alterar quaisquer outros dados, basta escrevê-los ou selecioná-los, conforme no caso de adição de usuário.

Para alterar a senha cadastrada, é necessário selecionar a opção "Alterar Senha", e os campos de "Senha" e "Confirmação de Senha" também serão exibidos.

#### 7.3.3.4 Excluir ou Desfazer Exclusão de Usuário

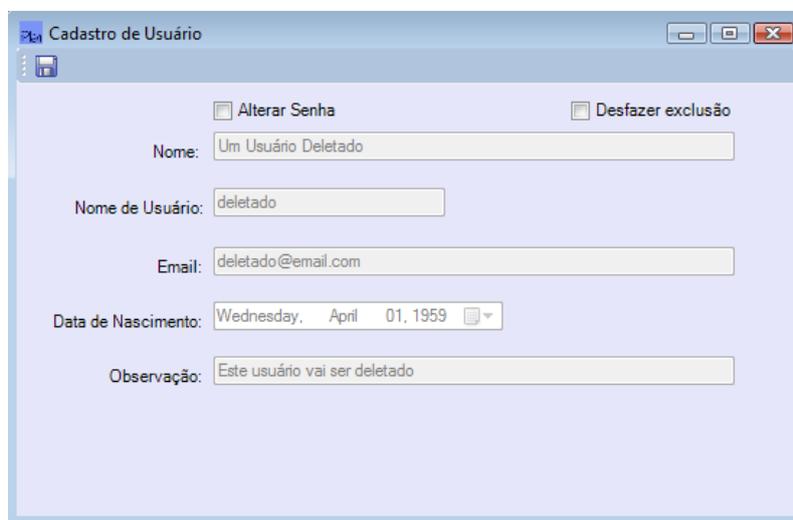
Para excluir um usuário do sistema, basta selecionar a linha em que seus dados estão exibidos na tela representada pela figura 7.3 e acionar o botão "Remover".

Vale lembrar que o usuário selecionado é aquele cuja linha possui uma seta preta. No caso da figura 7.3, o usuário selecionado é o de nome de usuário "andrezza".

Para confirmar a exclusão do usuário, basta acionar o botão "Salvar".

Já para desfazer a exclusão de um usuário, basta filtrar utilizando a opção "Usuários Deletados" e, a seguir, proceder como se fosse editar os dados do usuário excluído. A diferença consiste da opção "Desfazer Exclusão", exibida, conforme ilustrado na figura 7.6.

Para realizar o retorno do usuário ao sistema, basta selecionar a opção "Desfazer Exclusão" e acionar o botão "Salvar".



A imagem mostra uma janela de software intitulada "Cadastro de Usuário". No topo, há uma barra de ferramentas com ícones de minimizar, maximizar e fechar. Abaixo, há duas caixas de seleção: "Alterar Senha" e "Desfazer exclusão", ambas desativadas. O formulário contém os seguintes campos: "Nome:" com o texto "Um Usuário Deletado"; "Nome de Usuário:" com o texto "deletado"; "Email:" com o texto "deletado@email.com"; "Data de Nascimento:" com o texto "Wednesday, April 01, 1959" e um ícone de calendário; e "Observação:" com o texto "Este usuário vai ser deletado".

Figura 7.6: Procedimento necessário para desfazer exclusão de usuário.

#### 7.3.4 Projetos

Acessando o módulo de projetos, o usuário pode visualizar a tela representada pela figura 7.7. Através desta tela, o usuário pode:

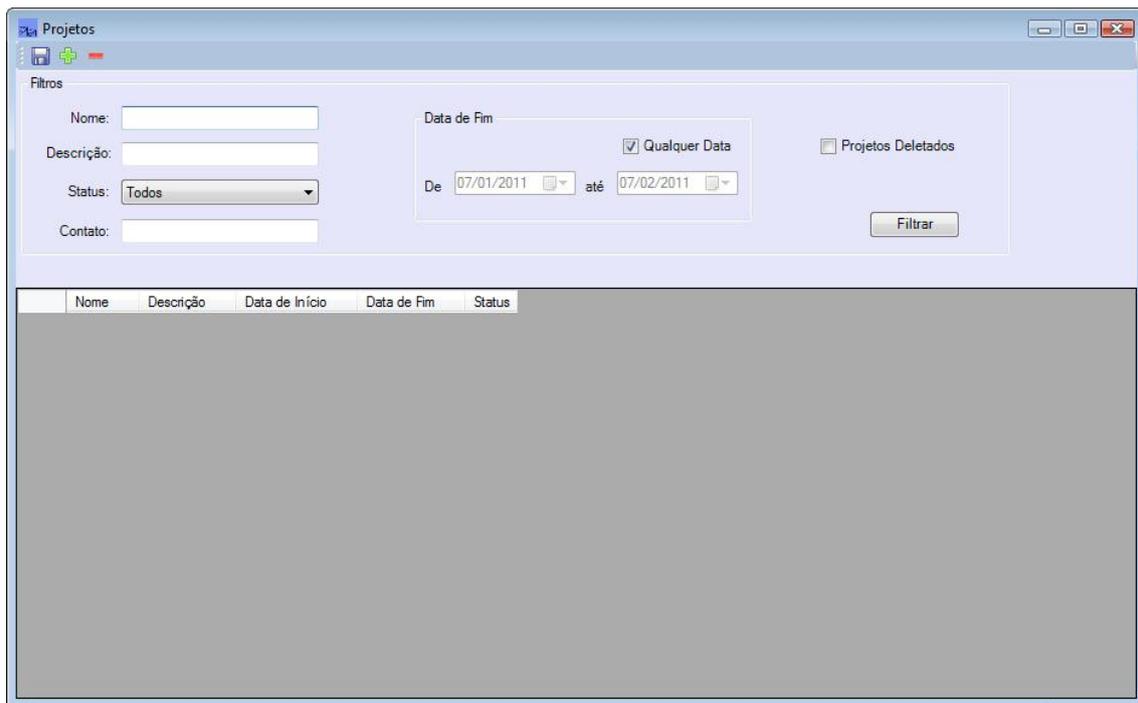


Figura 7.7: Módulo de Projetos.

- Listar ou Buscar Projetos
- Adicionar Projetos
- Editar Dados de Projetos
- Excluir ou Desfazer a Exclusão de Projetos

#### 7.3.4.1 Listar ou Buscar Projetos

De forma análoga ao módulo de usuários, para buscar projetos cadastrados, basta preencher os filtros conforme desejado e acionar o botão "Filtrar".

Como pode ser notado através da figura 7.7, há duas opções quanto ao filtro de data. O usuário pode decidir entre visualizar somente projetos cuja data de fim pertence ao intervalo selecionado, ou projetos independentemente da data de fim, através da marcação da opção "Qualquer Data".

A figura 7.8 representa um resultado obtido após a busca de projetos.

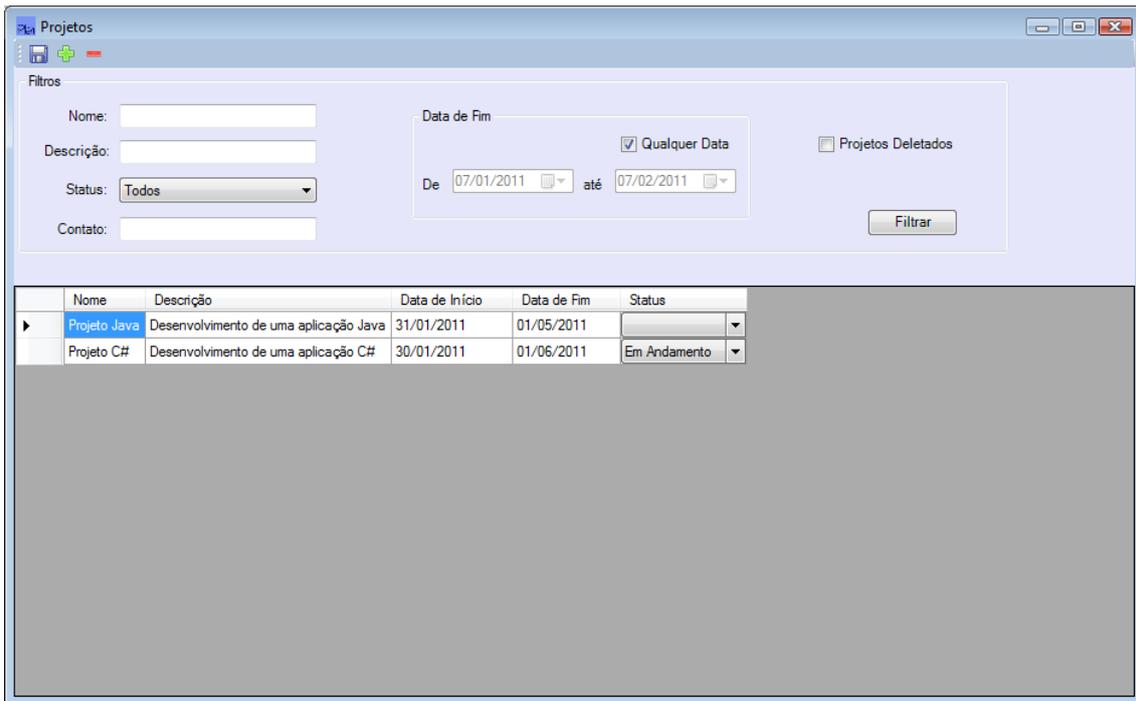


Figura 7.8: Busca de Projetos.

### 7.3.4.2 Adicionar Projetos

Para adicionar projetos, basta que o botão "Adicionar" seja acionado, o formulário exibido (representado pela figura 7.9) preenchido e os dados salvos, de forma semelhante ao procedimento de adição de usuários.

Figura 7.9: Formulário a ser preenchido para adicionar um projeto.

### 7.3.4.3 Editar Dados de Projetos

Para editar dados de um projeto, o procedimento a ser realizado é o mesmo do caso de edição de dados do usuário. Basta clicar duas vezes em um projeto listado na tela, editar os dados exibidos no formulário e salvar as alterações.

### 7.3.4.4 Excluir ou Desfazer a Exclusão de Projetos

Também de forma análoga ao módulo de usuários, para excluir um projeto é preciso acionar os botões "Remover" e, em seguida, "Salvar", enquanto para desfazer a exclusão de um projeto, basta, ao editar seus dados, selecionar a opção "Desfazer Exclusão" e confirmar a operação através do botão "Salvar".

## 7.3.5 Grupos

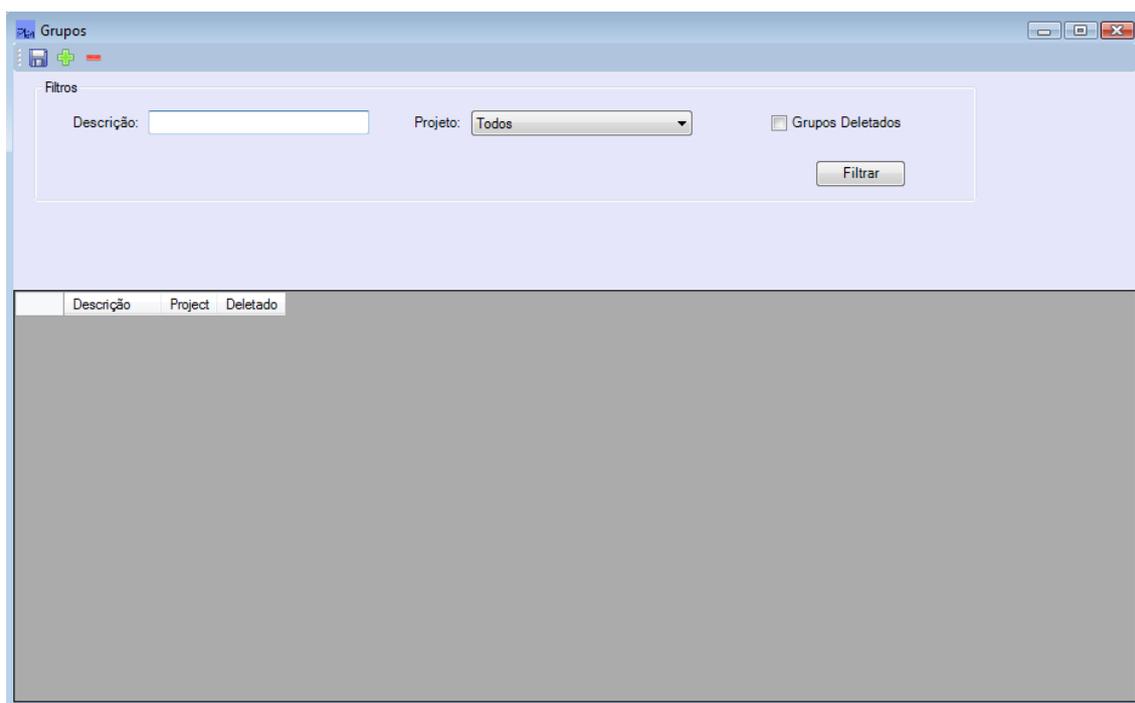


Figura 7.10: Módulo de grupos.

Ao acessar o módulo de grupos, o usuário visualizará a tela da figura 7.10.

Através desta tela é possível:

- Listar ou Buscar Grupos

- Adicionar Grupos
- Editar Dados de Grupo Cadastrado
- Excluir ou Desfazer Exclusão de Grupo

### 7.3.5.1 Listar ou Buscar Grupos

Da mesma forma dos módulos anteriores, para buscar grupos, basta preencher os parâmetros desejados e acionar o botão "Filtrar". A figura 7.11 representa resultados de uma busca de grupos.

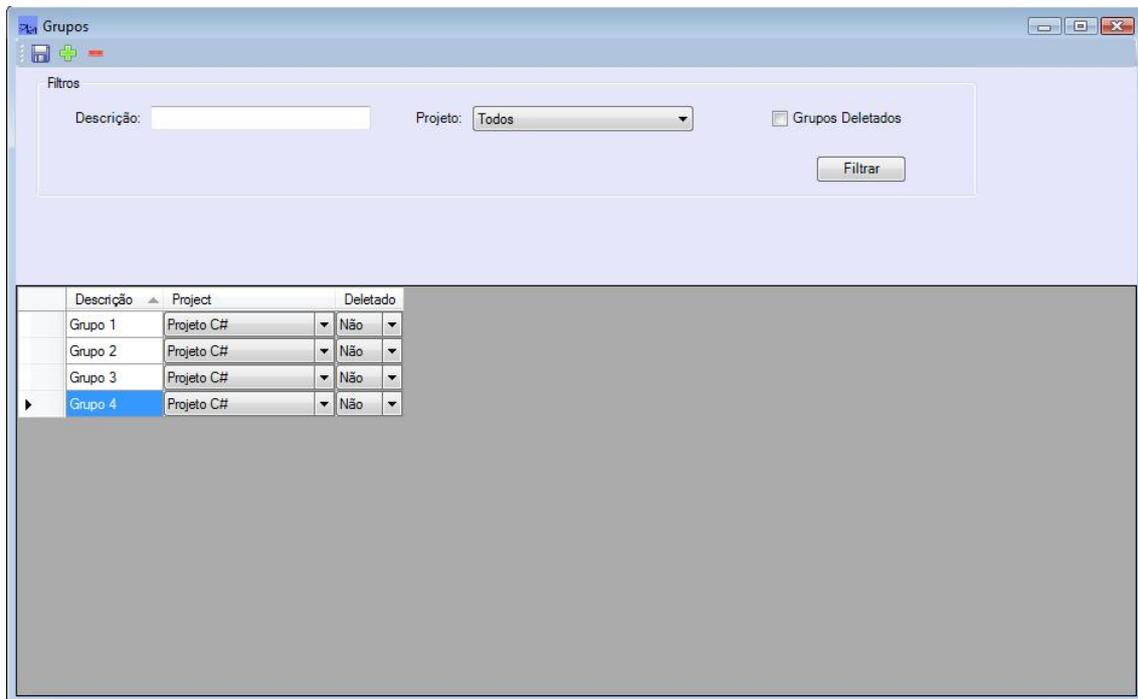


Figura 7.11: Exemplo de Resultado de Busca de Grupos.

### 7.3.5.2 Adicionar Grupos

Para adicionar grupos ao sistema, basta acionar o botão "Adicionar". O sistema então exibirá uma nova linha em branco abaixo da linha selecionada, caso haja, onde os dados do grupo devem ser preenchidos, conforme representado na figura 7.12.

A seguir, o usuário deve preencher os dados de grupo de acionar o botão "Salvar".

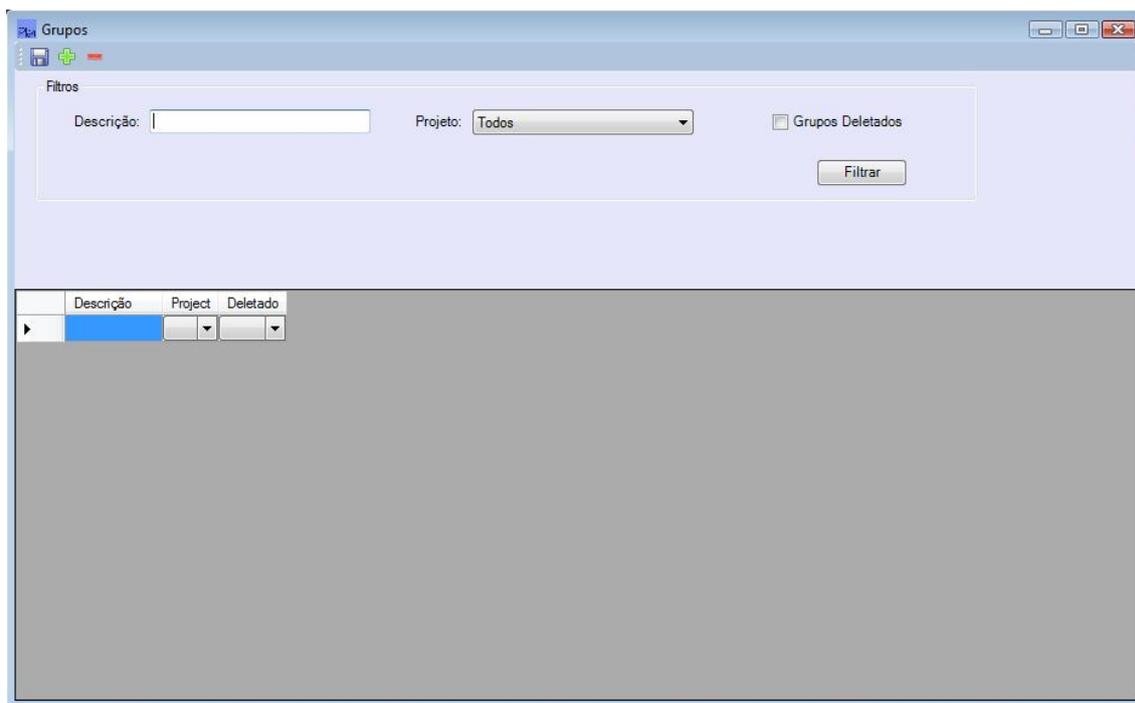


Figura 7.12: Linha em branco para adição de novo grupo.

Vale ressaltar que nesse caso, podem ser adicionados mais de um grupo ao mesmo tempo, se assim for desejado. Para isso, basta adicionar mais linhas em branco à tela e preencher o dados dos novos grupos.

### 7.3.5.3 Editar Dados de Grupo Cadastrado

Para editar dados de um grupo cadastrado, basta sobre-escrever os dados do grupo desejado, exibidos na tela e acionar o botão "Salvar".

### 7.3.5.4 Excluir ou Desfazer Exclusão de Grupo

Para excluir um grupo, há duas formas possíveis: alterar o valor da coluna "Deletado" para "Sim", ou utilizar o procedimento realizado para exclusão de usuários e projetos.

Para desfazer a exclusão de um grupo, basta alterar o valor da coluna "Deletado", de "Sim" para "Não" dos usuários excluídos.

### 7.3.6 *Sprints*

Ao acionar o botão relativo ao módulo de *Sprints*, o usuário tem acesso à tela ilustrada na figura 7.13.

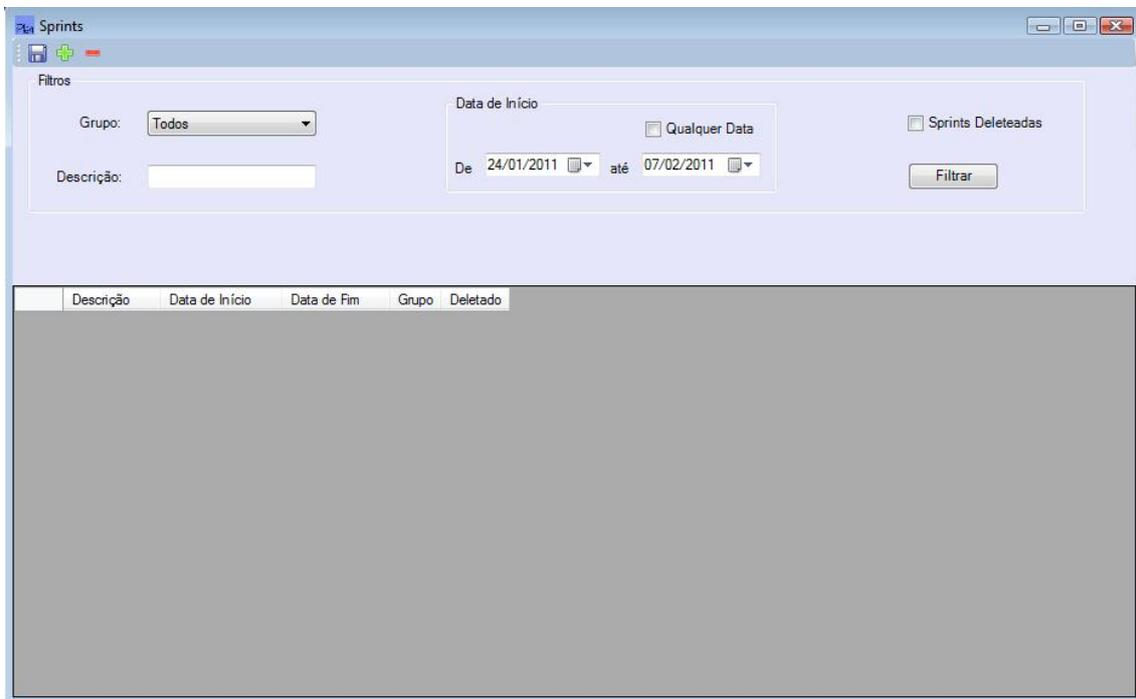


Figura 7.13: Módulo de *Sprints*.

Através desta tela é possível:

- Listar ou Buscar *Sprints*
- Adicionar *Sprints*
- Editar Dados de *Sprints*
- Excluir ou Desfazer Exclusão de *Sprints*

Este módulo funciona de forma semelhante ao módulo de Grupos, sendo a adição de novas *sprints* feita através do preenchimento de campos de novas linhas exibidas na tela, a deleção de usuários podendo ser feitas das duas formas já citadas e o retorno de *sprints* ao sistema através da alteração do valor da coluna "Deletada".

### 7.3.7 Tarefas

Ao acessar o módulo de Tarefas, o usuário visualiza a tela representada pela figura 7.14.

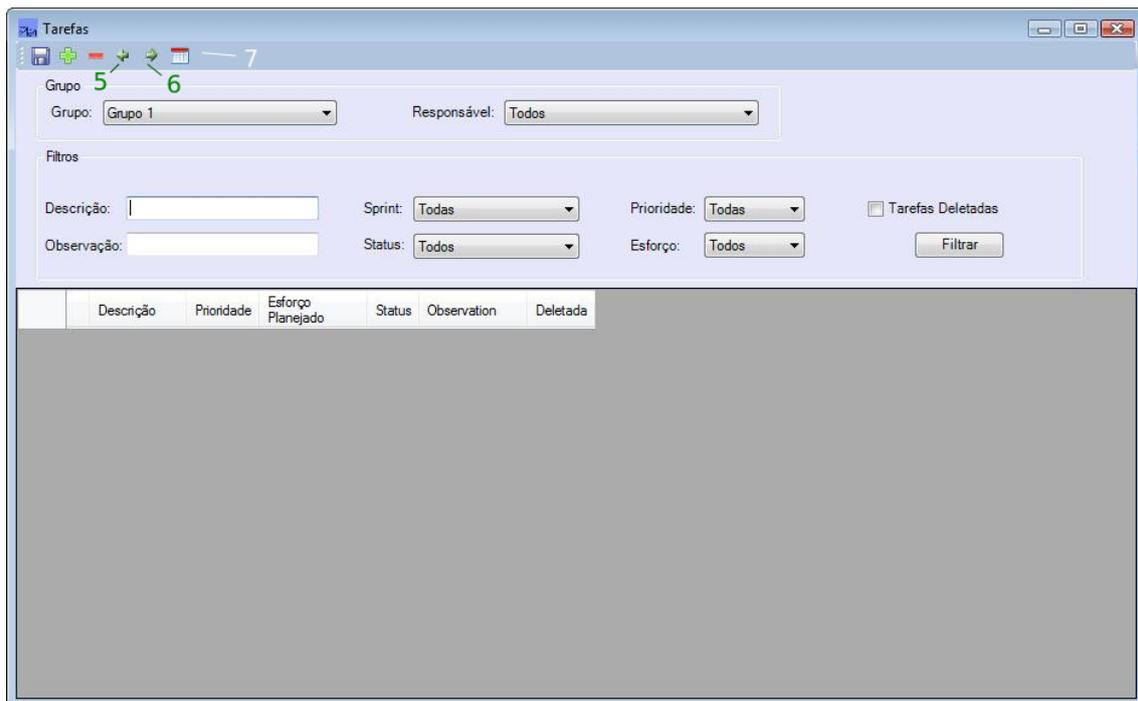


Figura 7.14: Módulo de *Tarefas*.

Através deste módulo é possível:

- Filtrar Tarefas
- Adicionar Tarefa
- Editar Dados de Tarefa
- Excluir ou Desfazer Exclusão de Tarefa
- Vincular Tarefa como Filha de Outra Tarefa
- Desfazer Vínculo entre Tarefas
- Adicionar Tarefa a *Sprints*

Os novos botões visualizados na tela têm as seguintes funções:

- **5** - Desvincular Tarefas
- **6** - Vincular Tarefas
- **7** - Adicionar Tarefa a *Sprints*

As funcionalidades "Filtrar Tarefas", "Adicionar Tarefas", "Editar Dados de Tarefas" e "Excluir ou Desfazer Exclusão de Tarefa" se dão de forma semelhante ao módulo de *Sprints*. As demais funcionalidades estão descritas a seguir.

### 7.3.7.1 Vincular Tarefa como Filha de Outra Tarefa

Para vincular uma tarefa como filha de uma outra, basta selecionar a tarefa filha e acionar o botão "Vincular Tarefas" e a tarefa acima da selecionada será vinculada como pai desta.

Para identificar tarefas filhas e tarefas pais há uma numeração na primeira coluna, sem título. Tarefas filhas em têm sua numeração a numeração da tarefa pai com a adição de um subíndice. Além disso, tarefas pais são destacadas pela fonte em negrito.

A figura 7.15 ilustra um exemplo de busca de tarefas com os respectivos vínculos.

	Descrição	Prioridade	Esforço Planejado	Status	Observation	Deletada
1	Uma tarefa	800	13	Em Andamento	Observação	Não
2	<b>Outra tarefa</b>	500	5	Concluída		Não
2.1	<b>Filha de uma tarefa</b>	700				Não
2.1.1	Neta de uma tarefa	500				Não
2.2	Outra Filha de uma tarefa	500				Não

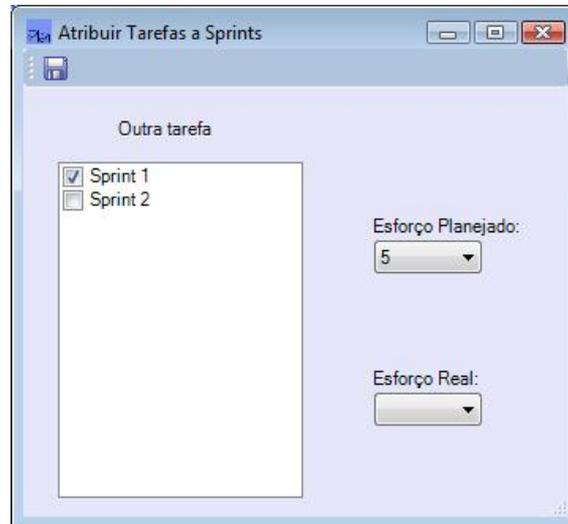
Figura 7.15: Exemplo de busca de tarefas com respectivos vínculos.

### 7.3.7.2 Desfazer Vínculo entre Tarefas

Para desfazer o vínculo entre uma tarefa e a respectiva tarefa pai, basta selecionar a primeira e acionar o botão "Desvincular Tarefas". A tarefa então, será "recuada" em um nível, caso possível.

### 7.3.7.3 Adicionar Tarefa a *Sprints*

Para planejar uma ou mais *sprints* em que certa tarefa deve ser executada, é necessário selecionar uma tarefa e acionar o botão "Adicionar Tarefa a *Sprints*". O sistema então exibe o formulário ilustrado na figura 7.16.



O formulário, intitulado "Atribuir Tarefas a Sprints", contém o seguinte conteúdo:

- Um grupo de seleção com o rótulo "Outra tarefa" contendo duas opções: "Sprint 1" (selecionada com um ícone de seta para cima) e "Sprint 2" (não selecionada).
- Um campo rotulado "Esforço Planejado:" com o valor "5" selecionado em um menu suspenso.
- Um campo rotulado "Esforço Real:" com um menu suspenso atualmente vazio.

Figura 7.16: Adicionar Tarefa a *Sprints*.

O procedimento a ser realizado, então, é a seleção de *sprints* em que a tarefa deve ser executada e a estimativa do esforço, através do preenchimento do campo "Esforço Planejado".

É possível ainda preencher o esforço de fato realizado para executar tal tarefa através do preenchimento do campo "Esforço Real".

### 7.3.8 Planejar *Sprint*

Ao acessar a tela "Planejar *Sprint*", o usuário deve selecionar o grupo e a respectiva *sprint* que deseja planejar e, automaticamente, as tarefas previamente planejadas para a última são carregadas na tela, conforme ilustrado na figura 7.17.

Como é possível notar, a presente tela faz o cálculo da *velocity* da *sprint* (localizado no canto superior direito da tela) automaticamente.

Os novos botões exibidos nesta tela possuem as seguintes finalidades:

- 8 - Importar Tarefas Atrasadas

	Descrição	Prioridade	Planned_Effort	Real_Effort	Status	Responsable	Observation	Deletada
1	Uma tarefa	800	3	8			Observação	Não
2	Outra tarefa	500	3	5				Não
▶ 2.1	Filho de uma tarefa	700	2					Não

Figura 7.17: Exemplo de tarefas planejadas para grupo e *sprint* exibidos.

- 9 - Importar Tarefas por Prioridade

A tela supracitada possui as funcionalidades abaixo, que serão detalhadas em seguida.

- Remover tarefas da *sprint*
- Importar tarefas de atrasadas
- Importar tarefas por prioridade

### 7.3.8.1 Remover Tarefas da *Sprint*

Para remover uma tarefa da *sprint* selecionada, basta selecionar a tarefa desejada e acionar o botão "Remover" ou alterar o valor da coluna "Deletada" para "Sim".

Vale ressaltar que tal tarefa não será removida do sistema, apenas não será planejada para ser executada na presente *sprint*.

### 7.3.8.2 Importar Tarefas Atrasadas

Para importar à *sprint* atual tarefas planejadas para *sprints* anteriores, porém que não foram concluídas, basta acionar o botão "Importar Tarefas Atrasadas".

### 7.3.8.3 Importar Tarefas por Prioridade

Para importar tarefas de determinada prioridade planejadas para *sprints* futuras, basta acionar o botão "Importar Tarefas por Prioridade".

O sistema, então, exibe ao usuário as possíveis prioridades de uma tarefa, conforme ilustrado na figura 7.18.

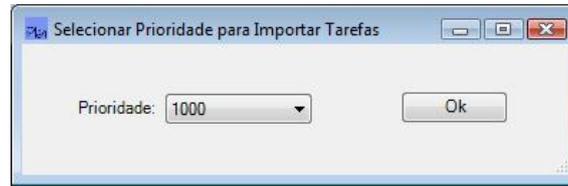


Figura 7.18: Opção de prioridades para que o usuário selecione a desejada.

Ao selecionar a prioridade desejada e clicar no botão "Ok", o sistema exibe tarefas cadastradas de acordo com a prioridade selecionada, somadas às tarefas previamente planejadas para a presente *sprint*.

# Capítulo 8

## Conclusão

O presente trabalho apresentou a idealização, projeto e implementação do iPlan, um sistema cujo objeto é facilitar a vida de usuários da metodologia ágil *Scrum*, principalmente *Product Owners* e *ScrumMasters* no que diz respeito ao gerenciamento de tarefas necessárias para a elaboração de um projeto.

Através do iPlan *Product Owners* e *ScrumMasters* poderão contar com um ambiente completamente adaptado à framework do *Scrum* para planejar e executar projetos, ao invés de utilizar planilhas, dificultando, portanto, a não conformidade de dados com o ambiente do *Scrum* e perda acidental daqueles.

Ainda, como o código-fonte será disponibilizado aos interessados e foi tomado o cuidado de deixá-lo claro, bem documentado e manutenível, o iPlan pode ser facilmente adaptado por outras pessoas às suas necessidades.

Todos os objetivos traçados para o sistema foram alcançados, entretanto, a funcionalidade de exibição de tarefas, no que se refere ao vínculo entre tarefas pais e filhas, teve sua complexidade alterada para que o projeto pudesse ser finalizado a tempo.

A metodologia adotada para o desenvolvimento do projeto se mostrou satisfatória, dado o baixo tempo de aprendizado necessário, já que a mesma já havia sido previamente utilizada pela aluna. O mesmo pode ser falado dos softwares escolhidos para o desenvolvimento do projeto, já que não são difíceis de serem manipulados, possuem extensa documentação e suas respectivas licenças estão disponíveis gratuitamente para alunos da UFRJ.

Apesar de possuir mais de um ano de experiência com *Scrum*, o desenvolvimento deste projeto me permitiu um conhecimento mais profundo das raízes e particularidades desta metodologia. Do meu ponto de vista, esta é bastante utilizável, não só em empresas, como em projetos acadêmicos, por facilitar o conhecimento e domínio de todos de uma equipe às várias nuances do objeto estudado ou desenvolvido, além de facilitar a substituição das pessoas de uma equipe a longo prazo, já que o conhecimento não fica restrito a uma pequena parte dos integrantes.

Por fim, durante a elaboração do projeto foram pensadas em maneiras de deixá-lo ainda mais robusto e adaptável a diferentes equipes, como uma maior flexibilidade quanto a quantidades de parâmetros de uma tarefa. Deste modo, como trabalho futuro, tem-se o desenvolvimento de novas funcionalidades a fim de tornar o iPlan adaptável a diferentes equipes sem que seja necessária manipulação de seu código-fonte.

# Referências Bibliográficas

- [1] FERREIRA, P., TORREÃO, P., MARÇAL, A. S., “Entendendo Scrum para Gerenciar Projetos de Forma Ágil”, *MundoPM*, v. 1, n. 17, 2007.
- [2] “Scrum”, <http://pt.wikipedia.org/wiki/Scrum>, (Acesso em 08 Fevereiro 2011).
- [3] “Scrum (rugby)”, [http://en.wikipedia.org/wiki/Scrum\\_\(rugby\)](http://en.wikipedia.org/wiki/Scrum_(rugby)), (Acesso em 09 Fevereiro 2011).
- [4] BECK, K., BEENDLE, M., BENNEKUM, A. V., *et al.*, “Manifesto for Agile Software Development”, <http://agilemanifesto.org>, 2001, (Acesso em 01 Fevereiro 2011).
- [5] COHN, M., “A Resusable Scrum Presentation. Introduce Scrum to your team, ScrumMaster and Product Owner”, <http://www.mountangoatsoftware.com/scrum-a-presentation>, 2002, Acesso em 31 Janeiro 2011.
- [6] “Learning Scrum - The Product Backlog”, <http://www.mountangoatsoftware.com/scrum/product-backlog>, Acesso em 09 Fevereiro 2011.
- [7] “Scrum Training and Agile Training from ScrumMaster Mike Cohn”, <http://www.mountangoatsoftware.com/>, Acesso em 09 Fevereiro 2011.
- [8] COHN, M., “Agile Training on Sprint Planning Meeting”, <http://www.mountangoatsoftware.com/scrum/sprint-planning-meeting>, Acesso em 31 Janeiro 2011.
- [9] “Planning Poker”, <http://www.planningpoker.com/>, Acesso em 09 Fevereiro 2011.

- [10] “Scrum Is an Innovative Approach to Getting Work Done”, [http://www.scrumalliance.org/pages/what\\_is\\_scrum](http://www.scrumalliance.org/pages/what_is_scrum), Acesso em 20 Agosto 2010.
- [11] “MSDN Academic Alliance”, <http://msdn.microsoft.com/en-us/academic/default.aspx>, Acesso em 23 Agosto 2010.
- [12] “Associação Brasileira de Normas Técnicas”, <http://www.abnt.org.br/>, Acesso em 25 Agosto 2010.