FACULDADES FACCAT

ELTON DIEGO GARBIN DO NASCIMENTO

GABRIEL FERNANDES DA SILVA

**GERENCIAMENTO ÁGIL DE STARTUPS**

TUPÃ

2016

ELTON DIEGO GARBIN DO NASCIMENTO

GABRIEL FERNANDES DA SILVA

GERENCIAMENTO ÁGIL DE STARTUPS

Trabalho de Conclusão de Curso apresentado a Banca Examinadora, como exigência parcial para a obtenção do título de Graduação do Curso de Sistemas de Informação, das Faculdades FACCAT sob a orientação do Prof. Guilherme Henrique dos Santos.

TUPÃ

2016

**RESUMO**

A área da tecnologia da informação e comunicação (T.I.C) vem cada vez mais ganhando espaço, influenciando nosso modo de viver ao ponto de ficarmos totalmente dependentes. Podemos atribuir essa tal evolução a era digital que está apenas começando, no entanto está conseguindo chamar a atenção do mundo. Desde a bolha da internet existe uma crescente corrida para o empreendedorismo, pois algumas pessoas enxergão quão promissor é o mercado e assim se aventuram a criar algo inovador que agregará a determinado negócio, ou que irá solucionar algo. Porém, para realmente se destacar ganhando conhecimento de um público, e subsequente obter o lucro desejado em cima do produto ou serviço oferecido é preciso sair do comum e trabalhar em um ambiente de extrema incerteza, onde será tudo ou nada. Por isso, muitas pessoas e empreendedores tradicionais enfrentam esse risco com base nos seus desejos e percepções ilusórias de mercado, e consequentemente acabam quebrando. Nesse cenário de extrema incerteza e tecnologia surgiu o termo Startup, que vem sendo muito discutido na atualidade. Nesse trabalho estaremos abordando esse termo e fazendo combinações com outros conceitos para criar uma ferramenta que possa gerenciar o processo de criação e ajudar essas startups a não desperdiçarem o tempo e esforços voltados ao projeto. Para isso, será baseado no conceito de startup enxuta e a metodologia ágil Scrum, combinando-os e tirando proveitos para obter os melhores resultados, garantindo assim a auto sustentabilidade do empreendimento. Com a ajuda de pesquisas realizadas através da internet e livros podemos construir esse trabalho.

**Palavras-chave:** Empreendedorismo; Startup; Scrum.

**AGRADECIMENTOS**

**AGRADECIMENTOS**

Agradeço primeiramente a minha família, minha mãe Ivanete e minhas irmãs Gabriela e Daniela pelo apoio e incentivo e apoio nesses quatro anos de estudos.

Ao meu companheiro neste trabalho Elton Garbin por aceitar realizar este trabalho comigo e se dedicar no desenvolvimento do mesmo.

Ao meu companheiro de trabalho Wilian Nascimento por me ajudar nos inúmeros problemas no decorrer deste trabalho.

Ao nosso orientador Guilherme Santos pelas dicas para que pudéssemos melhorar sempre nosso trabalho e pelo empenho em nos ajudar no pouco tempo que lhe coube.

Aos meus amigos Victor, Vinicius, Renan, Divaldo, Cid, Carol, Marcos Vinicius, Carlos e Jonas que sempre me apoiaram, mesmo na minha ausência.

E a todos os professores que me ensinaram, incentivaram e ajudaram, direta ou indiretamente à conclusão deste trabalho.

Gabriel Fernandes da Silva

**ABSTRACT**

The area of ​​information and communication technology (T.I.C) is increasingly gaining ground, influencing our way of life to the point of become very dependent. We can attribute this evolution to such digital era that has just begun, but is managing to draw the world's attention. Since the Internet bubble there is a growing race for entrepreneurship, because some people pallet how promising is the market and thus venture to create something new that will add a particular business, or that will solve something. However, to really stand out gaining knowledge of an audience, and further obtain the desired profit on the product or service offered is necessary to leave the common and work in an extremely uncertain environment, where it will be all or nothing. So many people and traditional entrepreneurs face this risk based on their desires and market illusory perceptions, and consequently end up breaking the person. In this scenario of extreme uncertainty and technology came the term Startup, which has been much discussed today. In this work, we will be addressing this term and making combinations with other concepts to create a tool that can manage the process of creation and help these startups not to waste time and efforts toward the project. For this, it will be based on the concept of lean startup and the Scrum agile methodology, combining them and taking income to get the best results, thus ensuring self-sustainability of the project. With the help of research conducted over the internet and books, we can make this work.

**Palavras-chave:** Entrepreneurship; Startup; Scrum.

**LISTA DE FIGURAS**

[**Figura 1** - Pilares do sistema Toyota de produção 18](#_Toc467002084)

[**Figura 2** - sistema de produção Tradicional x Nivelada (heijunka) 19](#_Toc467002085)

[**Figura 3** - Exemplo painel andon 21](#_Toc467002086)

[**Figura 4** - Just in Time Manufatura 22](#_Toc467002087)

[**Figura 5** - Agile Kanban Board 24](#_Toc467002088)

[**Figura 6** - Resumo da Metodologia Lean Startup 29](#_Toc467002089)

[**Figura 7** - Scrum Framework 33](#_Toc467002090)

[**Figura 8** - MER 42](#_Toc467002091)

[**Figura 9** - Diagrama de Classes - Configuração 51](#_Toc467002092)

[**Figura 10** - Diagrama de Classes - Modelos 52](#_Toc467002093)

[**Figura 11** - Diagrama de Classes - Estrutura 53](#_Toc467002094)

[**Figura 12** - Diagrama de Classes - Administrador 54](#_Toc467002095)

[**Figura 13** - Diagrama de Classes - Scrum Master 55](#_Toc467002096)

[**Figura 14** - Diagrama de Classes - Product Owner 56](#_Toc467002097)

[**Figura 15** - Diagrama de Classes - Membro Time 57](#_Toc467002098)

[**Figura 16** - Diagrama de sequência – Visitante 58](#_Toc467002099)

[**Figura 17** - Diagrama de sequência – Usuário 58](#_Toc467002100)

[**Figura 18** - Diagrama de sequência - Product Owner 59](#_Toc467002101)

[**Figura 19** - Diagrama de sequência - Scrum master 59](#_Toc467002102)

[**Figura 20** - Diagrama de sequência - Membro time 60](#_Toc467002103)

[**Figura 21** - Diagrama de sequência - Administrador 60](#_Toc467002104)

[**Figura 22** - Tela de cadastro de usuário 62](#_Toc467002105)

[**Figura 23** - Tela de alteração de função de usuário 63](#_Toc467002106)

[**Figura 24** - Tela de cadastro da sprint 64](#_Toc467002107)

[**Figura 25** - Tela de cadastro de reuniões 65](#_Toc467002108)

[**Figura 26** - Tela de inclusão do backlog do produto na sprint 67](#_Toc467002109)

[**Figura 27** - Tela de cancelamento da Sprint 68](#_Toc467002110)

[**Figura 28** - Tela de cadastro de backlogs do produto 69](#_Toc467002111)

[**Figura 29** - Tela de avaliação do backlog do produto 70](#_Toc467002112)

[**Figura 30** - Tela de cadastro de tarefa 71](#_Toc467002113)

[**Figura 31** - Tela de alteração de status da tarefa 72](#_Toc467002114)

[**Figura 32** - Tela de Backlog do Produto 77](#_Toc467002115)

[**Figura 33** - Tela de Tarefas 78](#_Toc467002116)

[**Figura 34** - Tela de sprint backlog 79](#_Toc467002117)

**LISTA DE TABELAS**

[**Tabela 1** - Onde estão as startups 25](#_Toc467002214)

[**Tabela 2** - Exemplo Scrum Product Backlog 39](#_Toc467002215)

[**Tabela 3** - Sprint Backlog 40](#_Toc467002216)

**LISTA DE GRÁFICOS**

[**Gráfico 1** - Quantidade de aporte recebido pelas startups 26](#_Toc467002242)

[**Gráfico 2** - Grau de satisfação do cliente antes do SCRUM 31](#_Toc467002243)

[**Gráfico 3** - Grau de satisfação do cliente depois do SCRUM 31](#_Toc467002244)

[**Gráfico 4** - Qual é o tamanho da sua equipe? 34](#_Toc467002245)

**LISTA DE QUADROS**

[**Quadro 1** - Autonomação x Automação 20](#_Toc467002254)

[**Quadro 2** - Controle de segurança 76](#_Toc467002255)

SUMÁRIO

[1 INTRODUÇÃO 13](#_Toc467002265)

[1.1 Definição do Problema 13](#_Toc467002266)

[1.2 Proposta da Pesquisa 13](#_Toc467002267)

[1.3 Objetivos 13](#_Toc467002268)

[1.4 Justificativa 14](#_Toc467002269)

[2 FUNDAMENTAÇÃO TEÓRICA 15](#_Toc467002270)

[2.1 Projeto 15](#_Toc467002271)

[2.2 Gerenciamento de projetos 15](#_Toc467002272)

[2.3 Padrões Tradicionais de Gerenciamento 16](#_Toc467002273)

[2.4 Produção Enxuta 17](#_Toc467002274)

[2.5 Startup 24](#_Toc467002275)

[2.6 Desenvolvimento Ágil 29](#_Toc467002276)

[2.7 Scrum 31](#_Toc467002277)

[3 DESENVOLVENDO O PROJETO PROPOSTO 42](#_Toc467002278)

[3.1 Diagramas UML 42](#_Toc467002279)

[3.2 Requisitos do sistema operacional 72](#_Toc467002280)

[3.3 Métodos para controle de segurança do sistema 74](#_Toc467002281)

[3.4 Layout dos relatórios 77](#_Toc467002282)

[4 IMPLEMENTAÇÕES FUTURAS 80](#_Toc467002283)

[5 CONCLUSÃO 80](#_Toc467002284)

[6 REFERÊNCIAS BIBLIOGRÁFICAS 80](#_Toc467002285)

# INTRODUÇÃO

## Definição do Problema

Segundo Ries (2011, p. 13), o termo startup pode ser interpretado como “[...] uma instituição humana projetada para criar novos produtos e serviços sob condições de extrema incerteza”. Portanto, podemos entender que é um grupo de pessoas com perfil de empreendedor à procura de um modelo de negócios repetível, escalável e inovador, que é normalmente apresentado em um cenário de incertezas.

Por ter essas características não é viável implantar as metodologias ou processos orientados a documentação em startups, pois são de certa forma, barreiras impostas ao desenvolvimento sendo que não possuem recursos para processos pesados de produção de software. Por esta razão, as instituições acabam por não usar nenhum processo. Isso pode trazer efeitos negativos no que diz respeito a qualidade do produto final, além de dificultar a entrega do software nos prazos, custos e funcionalidades previamente definidas.

## Proposta da Pesquisa

Auxiliar as startups no desenvolvimento de aplicativos, para que sejam seguidas as regras do modelo de desenvolvimento ágil implicando numa melhora de resultados no desenvolvimento e entrega do produto, sempre se atentando ao feedback do cliente para que correções e melhorias sejam realizadas de maneira mais rápida e eficiente.

## Objetivos

Visando contribuir com a área da Sistema de Informação, pretende-se atingir, com o presente trabalho, os seguintes objetivos:

### Objetivo Geral

Implantar a utilização da metodologia ágil de desenvolvimento scrum juntamente com os conceitos da startup enxuta para maximizar o aproveitamento dos recursos disponíveis, alcançar maior motivação e comprometimento da equipe, resultando em maior emprenho durantes os projetos assim como qualidade nas aplicações.

### Objetivos Específicos

Nesse sentido, os objetivos específicos a serem abordados nesta pesquisa são os seguintes:

* Definir o que é projeto.
* Introduzir o gerenciamento de projetos.
* Caracterizar a produção enxuta.
* Descrever o funcionamento e comportamento de uma startup.
  + Definir o que é uma startup.
  + Descrever as principais características de uma startup.
  + Demonstrar exemplos de startups de sucesso.
* Evidenciar um Modelo de Startup Enxuta.
* Apresentar Modelo de desenvolvimento ágil de projetos de software.
  + Apresentar o Manifesto Ágil.
  + Exemplificar modelos.
  + Realizar comparações.
* Descrever o Scrum.
  + Identificar os Perfis.
  + Apresentar as Regras.

## Justificativa

A estrutura scrum possibilita o máximo de aproveitamento da equipe refletindo assim na entrega do produto. Sendo assim é um grande diferencial ser utilizado para evolução dessas startups que estão à procura de um lugar no mercado.

Planejar o desenvolvimento de um software em uma startup não é nada fácil, pois a empresa não dispõe de recursos excedentes, o que faz com que o produto tenha que ser entregue o mais rápido possível. Essa presa em ter que entregar o produto faz com que o ambiente de trabalho fique caótico pois ao mesmo tempo é preciso lidar com a euforia de desenvolver algo inovador e também com a pressão de que nada pode dar errado.

Implantar a metodologia scrum juntamente com os conceitos de startup enxuta nesse meio de trabalho fara com que a equipe fique mais focada em seus objetivos e que os problemas encontrados no meio do desenvolvimento sejam resolvidos com maior rapidez o que consequentemente faz com que o produto que o público deseja seja entregue o mais rápido possível.

# FUNDAMENTAÇÃO TEÓRICA

O estudo dos tópicos a seguir procura-se dar uma ampla visão sobre o conceito de uma startup enxuta aplicando à mesma o modelo conceitual de gerenciamento de projetos baseado na metodologia scrum, procurando a melhora dos resultados da equipe e impactando diretamente na qualidade e prazos de entrega do produto.

## Projeto

Tudo que é construído necessita de um projeto, mesmo que simples como por exemplo apenas uma ideia. Projetos são segundo ANDREUZZA (2016), elaborados e executados pelas instituições para criação de novos produtos e serviços, implementação de mudanças e inovações em seus processos. Na verdade, projeto não é um conceito nada novo, a centenas de anos projetos são desenvolvidos, mesmo que ainda não fossem reconhecidos como projeto. Alguns exemplos a serem citados de projetos grandiosos são: a muralha da china, o canal do Panamá, as pirâmides do Egito e tantas outras

Projeto é uma construção própria do ser humano, que se concretiza a partir de uma intencionalidade representada por um conjunto de ações que ele antevê como necessária para executar, a fim de transformar uma situação problemática em uma situação desejada. A realização das atividades produz um movimento no sentido de buscar atingir, no futuro, uma nova situação que responda às suas indagações ou avance no sentido de melhor compreendê-las. Nesse processo de realização das atividades acontecem imprevistos, e mudanças fazem-se necessárias, evidenciando que o projeto traz em seu bojo as ideias de previsão de futuro, abertura para mudanças, autonomia na tomada de decisões e flexibilidade. (ALMEIDA, 2002)

## Gerenciamento de projetos

Gerenciamento de projetos é a aplicação de conhecimentos, habilidades e técnicas para a execução de projetos de forma efetiva e eficaz. Trata-se de uma competência estratégica para organizações, permitindo com que elas unam os resultados dos projetos com os objetivos do negócio – e, assim, melhor competir em seus mercados. (PMI, 2016)

Os princípios da gerencia de projetos surgiu na metade do século XIX, quando houve um aumento na complexidade dos novos negócios em escala mundial. Com a estrutura econômica do mundo ocidental profundamente alterada pela revolução industrial deu-se o desenvolvimento de uma de suas principais consequências, o capitalismo industrial. As formas de produção foram muito modificadas e se iniciou uma cadeia de transformações, que tornou cada vez mais difícil a tarefa de gerir as novas organizações. (ANDREUZZA, 2016)

Com isto surge a necessidade de alterar a forma de administração dessas instituições. Na maioria patrocinados pelo estado, os projetos foram decisivos para a criação de entidades que tentavam padronizar sua execução e para estimular a área de gerenciamento de projetos.

Frederick Taylor (1856-1915), no início do século XX, iniciou seus estudos de forma detalhada sobre trabalho. Ele aplicou raciocínio científico para mostrar que o trabalho pode ser analisado e melhorado focando em suas partes elementares. Ele aplicou sua teoria às atividades encontradas na indústria de aço (por exemplo, carregar areia, levantar areia).

Gantt construiu diagramas com barras de tarefas e marcos que esboçavam a sequência e a duração de todas as tarefas em um processo. Os diagramas de Gantt provaram ser uma ferramenta analítica tão poderosa para gerentes que se mantiveram inalterados por quase cem anos. (TORREÃO, 2007)

Mas foi por volta da segunda guerra mundial que começou a transformação do gerenciamento de projetos, para o que conhecemos hoje, ou seja, como uma disciplina. A construção da bomba atômica foi resultado do Projeto Manhattan, que envolveu mais de 100 mil pessoas – um projeto de pesquisa e desenvolvimento distribuído em treze locais diferentes e que em três anos apenas traria ao mundo um resultado que deixaria assustados até mesmo os envolvidos. (DE PAULA, 2009)

Essas técnicas espalharam-se para as indústrias e os negócios começaram a serem considerados como um organismo humano. Esta visão de organismo humano implica que para um negócio sobreviver toda sua estrutura precisar estar integrada e em constante funcionamento sempre visando as metas, as fases do projeto e o cumprimento dos cronogramas.

Surge em 1969 uma instituição para a associação de profissionais de gerenciamento de projetos, a PMI – Project Management Institute. Atualmente com mais de 240.000 membros em mais de 160 países, realiza pesquisas na área, promovendo acesso a uma grande quantidade de informações e recursos. Sediada em Atlanta, Geórgia, EUA. O grupo oficialmente incorporou a associação em Newtown Square, Pensilvânia, EUA. (TORREÃO, 2007)

## Padrões Tradicionais de Gerenciamento

O gerenciamento de projetos tradicional é, atualmente, o método mais utilizado no desenvolvimento de software. Baseia-se em processos bem definidos e documentados que passam por melhorias contínuas nas diversas organizações. (BOEHM, 2002)

### Características

O objetivo principal do gerenciamento tradicional está relacionado ao processo que suporte o desenvolvimento de software e permita o controle dos problemas durante o ciclo de vida do projeto (NERUR et al., 2005). A partir das informações históricas e da repetição obtém-se a melhoria da capacidade do processo através da padronização, medição e controle do projeto (BOEHM, 2002).

Segundo Massimus (2016), a metodologia tradicional é conhecida também como método iterativo, o foco está nos processos, no monitoramento do andamento do projeto através das entregas, ou seja, a ideia é de que só existe progresso quando é entregue ao todo, quer dizer, apenas com 100% do projeto concluído é que o cliente atestará a satisfação com o produto/serviço.

### Problemas

Segundo Massimus (2016), a metodologia tradicional “[...] é um método de estrutura mais rígida, pouco flexível a modificações e com grande exaltação da figura do gerente de projetos. Nessa categoria, se destacam as boas práticas oriundas do PMI e CMMi”.

O método tradicional é aplicado em qualquer ambiente de projeto, mas nas situações críticas envolvendo prazos restritos e com volatilidade de requisitos apresenta dificuldades em responder com rapidez as mudanças impostas pelos clientes causando, às vezes, conflitos no relacionamento e comprometimento do prazo. (RIBEIRO e ARAKAKI, 2006)

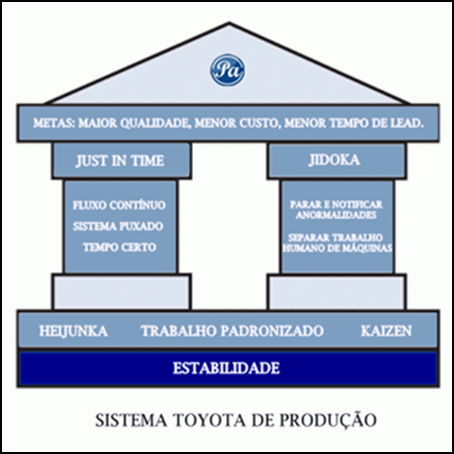
## Produção Enxuta

Segundo Ramos (2016) o termo significa “Filosofia produtiva essencialmente utilizada com o objetivo de eliminar ou minimizar atividades não agregadoras de valor ao produto final”, originado do Sistema Toyota de Produção também pode ser visto como Lean Manufacturing (Manufatura Enxuta) ou Lean Thinking (Pensamento Enxuto).

### Sistema Toyota de Produção (TPS)

Segundo Veras (2009) “[...] é uma combinação dos princípios e técnicas de qualidade total, da administração científica e das tradições culturais japonesas”.

O sistema Toyota foi concebido na década de 1950, quando Toyoda e Ohno, visitando os Estados Unidos, concluíram que o principal problema do modelo de Ford era o desperdício de recursos. Dessa observação nasceram os elementos básicos do Sistema Toyota de Produção, sendo também seus dois princípios mais importantes: a eliminação de desperdícios e a fabricação com qualidade. (PORTAL ADMINISTRAÇÃO, 2013)



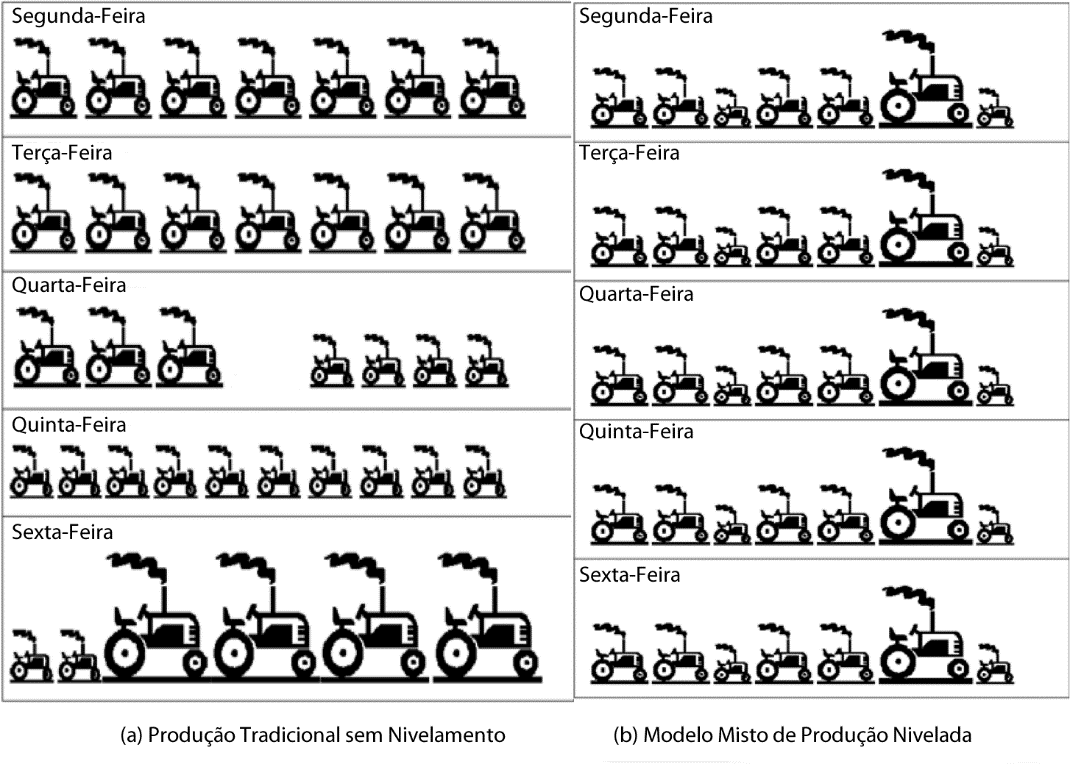
**Figura 1 -** Pilares do sistema Toyota de produção

Fonte: Portal Administração, 2013.

#### Heijunka

Como podemos ver na figura 1 está na base dos pilares do TPS. Esse conceito “[...] consiste na eliminação da variação da carga de trabalho (mura). Isto é feito através do nivelamento dos volumes de produção de forma a conseguir um fluxo suave, contínuo e eficiente”. (TOYOTA, 2016)

A figura a seguir exemplifica bem a diferença entre a produção tradicional da produção nivelada:



**Figura 2 -** sistema de produção Tradicional x Nivelada (heijunka)

Fonte: Silveira, 2016.

#### Kaizen

Filosofia japonesa central no sistema Toyota de produção, que significa melhoria contínua (TOYOTA, 2016). Segundo Silveira (2016) essa palavra japonesa é “[...] constituída de dois ideogramas: O primeiro (Kai) representa mudança e o Zen, bondade ou virtude”.

Para a Toyota (2016), “Na prática o Kaizen significa que todos os colaboradores em todas as partes da organização estão continuamente à procura de maneiras para melhorar as operações, e todas as pessoas em todos os níveis da empresa apoiam este processo de melhoria”.

A aplicação do Kaizen é ampla, podendo ser tanto na administração, como no grupo ou nas pessoas. No caso do Kaizen para o Grupo, um exemplo é o sistema PDCA (Plan-Do-Check-Action), que tem como meta tornar ágil a execução dos projetos e processos. Já o Kaizen para Administração tem como elemento principal o papel do Gerente, que deve se dedicar 50% do tempo no aperfeiçoamento de sua equipe. Enquanto isso, o Kaizen com foco nas pessoas deve envolver todos e para isso é preciso ser dinâmico. (HENRIQUE; FIORIO, 2013)

#### Normalização

Segundo a Toyota (2016) “O desenvolvimento com base em tarefas normalizadas assegura não só a consistência de elevados níveis de qualidade, mas também mantém o ritmo de produção e fornece termos de comparação para implementar uma estratégia de melhoria contínua”.

#### Jidoka

Como podemos ver na figura 1 é um pilar do TPS que se encontra sobre a base dos conceitos: Heijunka, Kazen e Normalização (trabalho padronizado). É constituído por elementos importantes tais como: Genchi Genbutsu, Painel Andon e Poka-Yoke. (TOYOTA, 2016)

Jidoka significa autonomação (automação com um toque humano) e é um dos pilares do Sistema Toyota de Produção. O conceito surgiu com uma máquina de tear auto-ativada inventada por Toyoda Sakichi pouco depois de 1890. Um dispositivo foi inserido na máquina de forma a distinguir condições normais e anormais, consequentemente produtos defeituosos não eram produzidos. Com isso, não era mais necessário possuir um operador vigiando a máquina enquanto ela estava funcionando, apenas sua intervenção caso ela parasse. Dessa forma, um único operador poderia cuidar de várias máquinas e a fábrica operaria com um número reduzido de pessoas, aumentando a eficiência da produção. (LEANTI, 2016)

Para esclarecer dúvidas entre autonomação e automação verifique o quadro exposto abaixo:

|  |  |
| --- | --- |
| **Autonomação** | **Automação** |
| Operador com controle restrito a comandos. Operador de máquina opera simultaneamente várias máquinas. | Controle realizado pela máquina com acompanhamento do operador. |
| Máquinas são providas de uma função de cérebro humano, ou seja, a capacidade de detectar anormalidades de forma autônoma. | Máquinas automatizadas não estão dotadas de cérebro humano. |
| Controle autônomo de qualidade, defeitos e quantidades em um processo. As causas dos defeitos são investigadas imediatamente, e uma ação corretiva é implementada. | O sistema calcula a ação corretiva mais apropriada |
| Produtos flexíveis, de baixo custo e qualidade superior. | Atividade meio para melhorar a integração e a flexibilização em um processo produtivo. |

**Quadro 1** - Autonomação x Automação

Fonte: Silveira, 2016.

Segundo o Lean Institute Brasil (2016) “Aplicar Jidoka evita que os defeitos de produção gerem retrabalhos, refugos, ou até mesmo que cheguem junto com o produto acabado às mãos do cliente”.

##### Genchi Genbutsu

Segundo o site Economist (2009) é uma expressão japonesa que significa “vai e veja você mesmo” não distante da ideia do conceito MBWA (Management By Walking Around).

Para Ferro (2010) “[...] é muito mais do que uma atividade adicional nas empresas. É mais que uma simples questão de caminhar e conversar”.

##### Painel Andon

De acordo com a Toyota (2016) painel andon “[...] é um simples quadro electrónico altamente visível que mostra o estado de cada linha de produção. Caso um operador detecte uma falha o quadro notifica imediatamente a gestão, indicando a sua localização precisa”.



**Figura 3** - Exemplo painel andon

Fonte: Silveira, 2016.

Segundo a própria Toyota (2016) “Os operadores são responsabilizados individualmente pela qualidade da produção e têm o poder de parar a linha de produção quando necessário. A linha de produção não recomeçará até que o problema tenha sido resolvido”.

##### Poka-Yoke

De acordo com o Endeavor Brasil (2015), é uma expressão japonesa que significa “à prova de erros” que surgiu nos anos 60 pelo então engenheiro líder da Toyota Shigeo Shingo.

O Poka-Yoke é uma técnica simples, mas extremamente poderosa, para eliminar falhas humanas no local de trabalho, usando dispositivos baratos que podem ser desenvolvidos no próprio chão de fábrica. Sistemas Poka-Yoke permitem que você atinja o Zero Defeito e elimine a inspeção. Ao mesmo tempo que o Poka-Yoke melhora a qualidade, também previne as falhas que podem levar a quebras ou a outros tipos de problemas com os equipamentos. (IMAM, 2016)

Segundo a Toyota (2016) “[...] este princípio é uma maneira simples contudo criativa e fiável de reduzir erros e manter a qualidade. Todos os itens usados com frequência estão claramente identificados para que possam ser usados e encontrados por qualquer pessoa”.

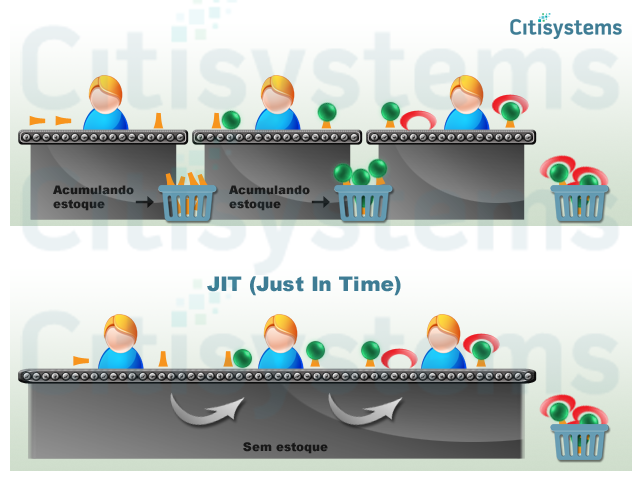
#### Just-in-time (JIT)

Como podemos ver na figura 1 é um pilar do TPS que se encontra sobre a base dos conceitos: Heijunka, Kazen e Normalização (trabalho padronizado). É constituído por princípios chaves tais como: Eliminação do desperdiço (muda), Takt time e Kanban. (TOYOTA, 2016)

O Just in Time surgiu no Japão, no princípio dos anos 50, sendo o seu desenvolvimento creditado à Toyota Motor Company, a qual procurava um sistema de gestão que pudesse coordenar a produção com a procura específica de diferentes modelos de veículos com o mínimo atraso. Quando a Toyota decidiu entrar em pleno fabrico de carros, depois da Segunda Guerra Mundial, com pouca variedade de modelos de veículos, era necessária bastante flexibilidade para fabricar pequenos lotes com níveis de qualidade comparáveis aos conseguidos pelos fabricantes norte-americanos. (DENES, 2008)

Segundo a Toyota (2016) o JIT “[...] assegura fluxos de trabalho suaves, contínuos e optimizados, com tempos de ciclo de trabalho planeados e medidos cuidadosamente e movimentos de produtos de acordo com a procura, reduzem custos com desperdícios de tempo, materiais e capacidade”.

Caso não soubéssemos mais a fundo sobre o conceito apenas traduzindo seu termo (na hora certa) chegaríamos à sua intenção. Observe a figura abaixo e compreenda como pode ser diminuído o estoque aplicando o JIT:



**Figura 4** - Just in Time Manufatura

Fonte: Silveira, 2016.

##### Eliminação do desperdiço

Segundo a Toyota (2016), “Desperdício (muda) é definido como qualquer coisa que não adiciona valor. Isto inclui áreas que normalmente não são consideradas como desperdício, tais como sobre produção, stocks, movimentos e processamentos em excesso e tempo de espera desnecessário”.

##### Takt time

De acordo com Saad (2012) “[...] vem da palavra alemã taktzeit, que pode ser traduzida como tempo de ciclo. ”. Para a Toyota (2016) o “Takt é a taxa de procura dos clientes”.

Ainda segundo Saad (2012) “Trata-se do período mais longo que pode ser gasto em cada unidade e em coesão com o nível de demanda. Essa medida pode ser particularmente útil para garantir que cada parte de uma linha de produção funcione de forma eficiente”.

A chave consiste em ter um ciclo de trabalho que esteja sincronizado com a procura, de forma a evitar tanto sub-processo como sobre produção. O Takt time determina a taxa do fluxo e permite o cálculo de quanto trabalho pode ser realizado. A optimização do tempo takt reduz o desperdício e a ineficiência eliminando o risco de atrasos temporais ou excesso de produção. (TOYOTA, 2016)

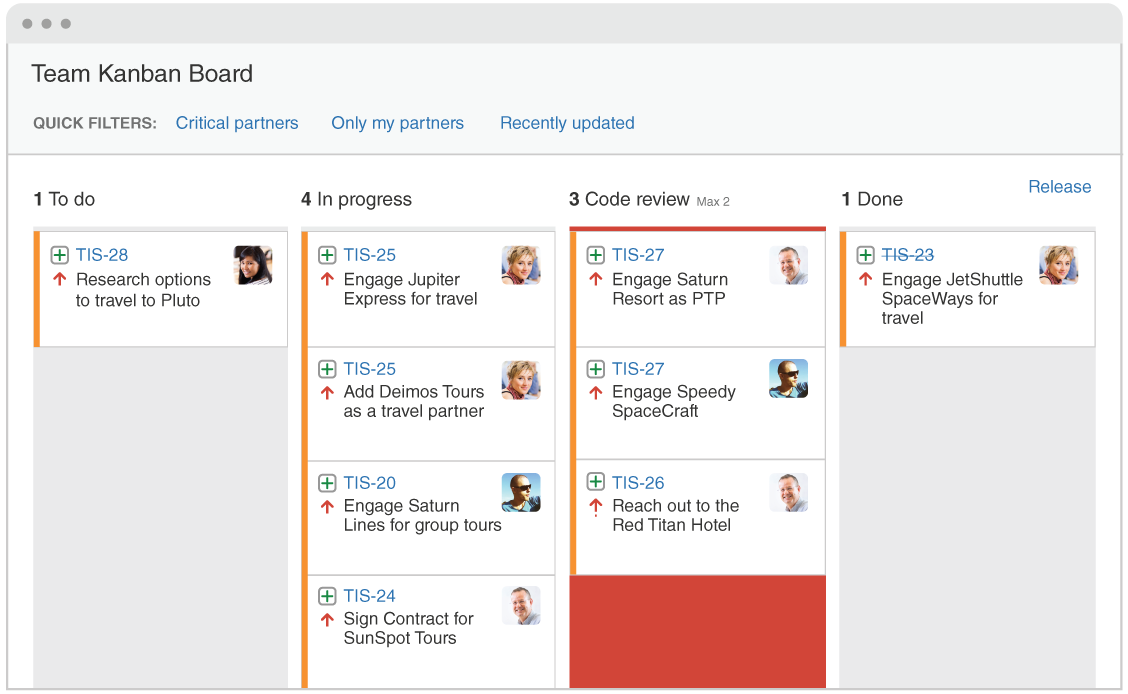
##### Kanban

Palavra de origem japonesa que pode ser traduzida como cartas (ou cartão) criada pela Toyota, fazendo parte do conceito de entrega Just-in-time e associado aos sistemas puxados. (SILVEIRA, 2016)

O cartão kanban é um meio simples e visível que o sistema de Produção Toyota usa para encomendar componentes à medida que eles são necessários, significando que um nível mínimo de componentes é mantido na área de montagem. Assim que os stocks necessitam reabastecimento, o operador através de um cartão kanban assegura uma entrega just-in-time. (TOYOTA, 2016)

Sendo assim, sua utilização possibilita a sintonia entre a gestão do estoque e a de produção, podendo ser subdividido em Kanban de produção e Kanban de movimentação. (SILVEIRA, 2016)

Esses quadros e cartões visuais integram o sistema Kanban, que ajudam os trabalhadores a planejarem a produção na indústria e a controlar o estoque. Assim, conforme a quantidade de cartões disponíveis nos quadros são tomadas decisões, priorizando o que é mais importante, realizando setup de máquinas e até mesmo as paradas para manutenção. (HENRIQUE; FIORIO, 2013)



**Figura 5** - Agile Kanban Board

Fonte: Radigan, 2016.

Kanban de produção

Segundo Silveira (2016) “O kanban de produção geralmente é utilizado no início da linha. Ele tem o objetivo de indicar a quantidade certa de peças e componentes necessários para determinada linha. É uma forma de solicitar peças para a linha de produção, facilitando e agilizando todo o processo”.

Kanban de movimentação

Segundo Silveira (2016) “Também chamado de kanban de transporte ou kanban de retirada, representa um outro momento da cadeia produtiva. O Kanban de movimentação indica a circulação de peças entre as áreas ou mesmo entre a empresa e seu fornecedor”.

## Startup

O termo Startup está cada vez mais comum no âmbito empresarial, porém vale ressaltar que:

[...] muita gente pensa que startup é toda empresa que está começando – já que “start up” pode ser traduzido como “começar”. Mas essa não é a definição mais aceita de startup. A definição mais aceita de startup é a do Steve Blank: uma empresa é considerada uma startup quando ainda está em busca de um modelo de negócios viável que seja repetível e escalável. Um modelo de negócios repetível é aquele capaz de vender o mesmo produto para todos os clientes. Já um modelo de negócios escalável é aquele capaz de crescer para atender grandes quantidades de clientes.

Além disso, as startups são empresas inovadoras, de maneira incremental ou disruptiva. Encontrar um modelo de negócios que seja viável, repetível, escalável e inovador não é nada fácil. Existem várias incertezas envolvidas, por exemplo: se existem clientes para o produto, se o modelo é realmente lucrativo, se é possível executar o modelo. E é por isso que toda startup é um empreendimento de alto risco – é muito mais seguro abrir um restaurante, por exemplo, do que começar uma startup. (KAYO, 2013)

De acordo com Ferrão (2013), startups são as empresas de pequeno porte, recém-criadas ou ainda em fase de constituição, com atividades ligadas a pesquisa e desenvolvimento de ideias inovadoras, cujo custos de manutenção são baixos, oferecendo a possibilidade de rápida e consistente geração de lucros.

“No comando desses empreendimentos, como não poderia deixar de ser, 55% são jovens entre 20 e 23 anos, que essencialmente têm formação em áreas como tecnologia, marketing ou administração”. (CANAL DO EMPREENDEDOR, 2014)



**Tabela 1** - Onde estão as startups

Fonte: Canal do empreendedor, 2014.

### Principais Características

Para ser considerada uma startup uma empresa deve possuir 5 características básicas, Inovação, Escalabilidade, ter potencial para atingir grandes mercados, sonho grande e serem enxutas e flexíveis. (SPINA, 2013)

Inovação: a startup precisa ter algo diferente do que já existe no mercado, seja no seu produto/serviço ou no modelo de negócio, de forma que isto seja uma vantagem competitiva.

Escalabilidade: o modelo de negócio de uma startup precisa ser escalável, isto é, poder crescer aceleradamente sem exigir na mesma proporção recursos humanos ou financeiros.

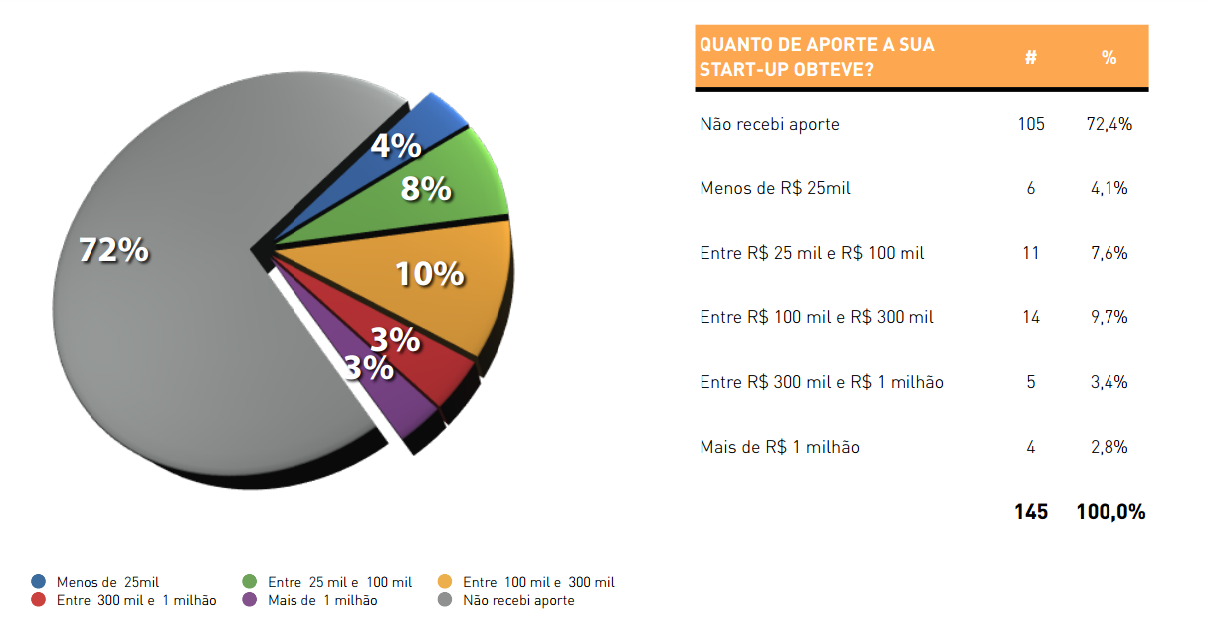
Tem potencial para atingir grandes mercados: as startups buscam criar produtos/serviços que possam ser utilizados potencialmente por milhares até milhões de clientes, para que possam assim crescer continuamente.

Sonho grande: as startups, mesmo começando como pequenas empresas, já nascem com a ideia de se tornarem grandes empresas no futuro, diferentemente das empresas chamadas de “estilo de vida”, que tem como objetivo apenas a realização pessoal do empreendedor.

São enxutas e flexíveis: mesmo pensando grande, as startups devem ser muito enxutas, pois pela necessidade de validar sua inovação, precisam estar prontas para mudar seu produto e/ou modelo de negócio de forma a adequá-los a demanda dos clientes. (SPINA, 2013)

Startups também convivem com alto grau de pressão na equipe, por ser uma empresa nova e de auto risco, uma startup não dispõe de muitos recursos para investimento, porém conta com projetos promissores, ligados ao desenvolvimento de ideias inovadoras, normalmente de base tecnológica buscando sempre um modelo de negócio inovador. (SEBRAE, 2015)

Segundo Broota (2014) “[...] o acesso a investimentos é um dos principais obstáculos das startups. A maioria não possui todo o dinheiro necessário para viabilização dos projetos, que envolvem conceitos de inovação e tecnologia e, portanto, não saem baratos”.



**Gráfico 1** - Quantidade de aporte recebido pelas startups

Fonte: Luz Consultoria, 2012.

O gráfico 1 é resultado de uma pesquisa realizada pela Luz Consultoria. Ele representa a quantidade de empresas startups que receberam aporte e o valor desses aportes, com isso é possível perceber que a carência dessas empresas quanto a investimentos é muito grande.

No Brasil existem algumas formas de conseguir impulsionar uma startup com alguns tipos de investimento como por exemplo: Bootstrapping, Investimento-Anjo, Capital semente, Incubadoras, etc. Dentre essas a forma de investimento a mais conhecida é o Investimento-Anjo, que segundo Anjos do Brasil (2015), “[...] é o investimento efetuado por pessoas físicas com seu capital próprio em empresas nascentes com alto potencial de crescimento”.

O Investidor-Anjo recebe, por seu investimento, uma participação societária minoritária no negócio, e não assume posição executiva na empresa, mas atua como um conselheiro orientando os empreendedores e participando das decisões estratégicas da empresa, aumentando muito suas chances de sucesso, bem como acelerando seu desenvolvimento. (SPINA, 2012)

### Exemplos

A revista Pequenas Empresas e Grandes Negócios (2014) diz que, “[...] criar uma empresa e levá-la ao sucesso não é nada fácil. Para atingir seus objetivos, o empreendedor precisa estar atento a uma série de fatores, da qualidade do produto à motivação de sua equipe”.

Uma pesquisa divulgada pelo Startup Genome Report, aponta que mais de 90% das startups quebram. (KFOURI, 2016)

Serão apresentadas abaixo exemplos de startups que deram certo e hoje são empresas reconhecidas nacionalmente, ou até mesmo internacionalmente.

Buscapé

De acordo com Bizstart (2015), este comparador online de preços veio ao mundo em 1999. O investimento inicial foi modestíssimo 400 reais do seu fundador, Romero Rodrigues, e três colegas. Após vencer a desconfiança de varejistas, que não queriam abrir seus preços para comparação, o site caiu rapidamente nas graças do consumidor, tornando-se uma das principais plataformas do comércio eletrônico brasileiro. Hoje a empresa é o maior site de comparação de preços da América Latina.

Easy Taxi

De acordo com Brancato (2015), após ter sido uma das vencedoras da Startup Weekend Rio de Janeiro 2011, a Easy Taxi foi oficialmente lançada em abril de 2012 por quatro empreendedores. Pioneira no serviço móvel de chamada de táxi na América Latina, atua hoje em 30 países, 420 cidades, sendo 350 delas no Brasil.

Os números atuais da empresa impressionam, são mais de 17 milhões de downloads, cerca de 400 mil taxistas cadastrados pelo mundo, aproximadamente 1300 funcionários diretos e indiretos e mais de 20 milhões de usuários.

Ao todo, a Easy Taxi já recebeu R$ 170 milhões de investimentos feitos pelo grupo alemão Rocket Internet; pelo Fundo Latin America Internet Holding (LIH); pela holding iMena e pelo Phenomen Ventures e Tengelmann.

Sambatech

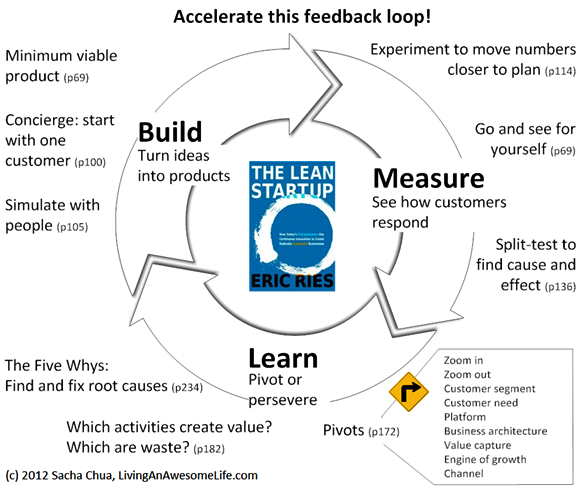
De acordo com Bizstart (2015), empresa é pioneira no mercado de distribuição de vídeos ao vivo, a startup integra o MIT Startup Exchange, programa exclusivo do MIT (Massachusetts Institute of Technology) com uma tecnologia própria.

Esta é apenas uma das grandes conquistas desta empresa mineira que oferece soluções para mais de 500 clientes em oito países e possui escritórios em Belo Horizonte, São Paulo, Miami e Buenos Aires. Auto referida como “o YouTube para empresas”, recebeu investimento da DFJ FIR Capital.

### Startup Enxuta

Para Gimenes (2013), “Na startup enxuta ao invés de fazer planos complexos baseados em várias hipóteses não-testadas, você deve fazer experimentos lançando "produto mínimo viável" (MVPs), que confirmem o que os clientes pagariam para ter, da forma mais rápida e barata possível”.

Gimenes (2013) também afirma que “O objetivo é chegar ao conceito certo do produto, identificar quem são os clientes dispostos a pagar por ele e se é possível montar um negócio sustentável - evitando desperdício de recursos (tempo, dinheiro, motivação, etc)”.



**Figura 6** - Resumo da Metodologia Lean Startup

Fonte: Gimenes, 2016.

Uma startup enxuta deve possuir as seguintes características: Empreendedores estão por toda parte; Empreender é administrar; Aprendizado validado; Construir-medir-aprender; Contabilidade para inovação. (RIES, 2011, p. 14)

## Desenvolvimento Ágil

Segundo BERNARDO (2015), nos últimos anos grandes empresas vêm gerando ganhos significativos em produtividade com reduções equivalentes no custo. A possibilidade de introduzir no mercado produtos com mais rapidez e níveis elevados de satisfação, tem feito com que os métodos ágeis fiquem mais popularizados ao redor do mundo. O retorno para empresas que utilizam os métodos ágeis pode ser grande, não à toa, empresas como Google, Yahoo, Microsoft e IBM os utilizam.

Agilidade em TI é: “a habilidade de criar e responder a mudanças, buscando a obtenção de lucro em um ambiente de negócio turbulento” (HIGHSMITH, 2004); ou ainda, a capacidade de balancear a flexibilidade e a estabilidade. HIGHSMITH (2004) enfatiza que a ausência de estrutura ou estabilidade pode levar ao caos, mas que a estrutura em demasia gera rigidez.

Ágil é uma nova forma de gestão e desenvolvimento de Software que usa uma abordagem de planejamento e execução iterativa e incremental voltado para processos empíricos que divide o problema em produtos menores e que visa entregar software funcionando regularmente [...] resumindo, seu principal objetivo é entregar o produto que o cliente realmente deseja e que será útil e com qualidade. (STEFFEN, 2012).

Métodos Ágeis para desenvolvimento de softwares visam uma melhoria constante do trabalho, ajudando a lidar melhor com a imprevisão e imprecisão inerentes ao processo e à tecnologia como um todo. (MASSIMUS, 2015)

### Manifesto Ágil

O manifesto é um documento elaborado por 17 pessoas de renome na área de desenvolvimento de software. Este documento foi elaborado no período de 11 a 13 de fevereiro de 2001, na região de “The Lodge”, nas montanhas “Wasatch” no Estado de Utah, EUA. Este encontro foi marcado com o propósito de tentar achar um denominador comum para a metodologia ágil, elaborando os princípios básicos para o desenvolvimento ágil de softwares, criando assim o encontro da “Formação da Aliança Ágil”. (RUGGIERI, 2016)

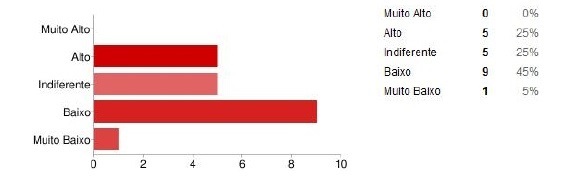
De acordo com o site Manifesto Ágil (2016) estavam presentes neste encontro os seguintes signatários: Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland e Dave Thomas. Como resultado desse encontro foi elaborado um documento, onde continha 12 princípios básico para o desenvolvimento ágil. Estes princípios são:

* Nossa maior prioridade é satisfazer o cliente, através da entrega adiantada e contínua de software de valor.
* Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se adequam a mudanças, para que o cliente possa tirar vantagens competitivas**.**
* Entregar software funcionando com frequência, na escala de semanas até meses, com preferência aos períodos mais curtos.
* Pessoas relacionadas à negócios e desenvolvedores devem trabalhar em conjunto e diariamente, durante todo o curso do projeto.
* Construir projetos ao redor de indivíduos motivados. Dando a eles o ambiente e suporte necessário, e confiar que farão seu trabalho.
* O Método mais eficiente e eficaz de transmitir informações para, e por dentro de um time de desenvolvimento, é através de uma conversa cara a cara.
* Software funcional é a medida primária de progresso.
* Processos ágeis promovem um ambiente sustentável. Os patrocinadores, desenvolvedores e usuários, devem ser capazes de manter indefinidamente, passos constantes.
* Contínua atenção à excelência técnica e bom design, aumenta a agilidade.
* Simplicidade: a arte de maximizar a quantidade de trabalho que não precisou ser feito.
* As melhores arquiteturas, requisitos e designs emergem de times auto organizáveis.
* Em intervalos regulares, o time reflete em como ficar mais efetivo, então, se ajustam e otimizam seu comportamento de acordo.

## Scrum

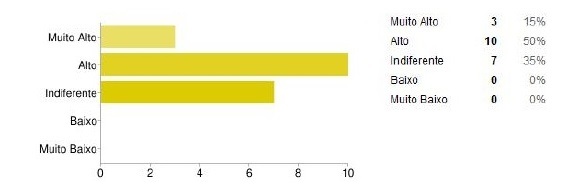
Para Filadoro (2014), “Quando o assunto é “fábrica de software”, vale a pena conhecer o SCRUM mais detalhadamente. Sua agilidade é responsável, em grande parte, pela inovação apresentada. Tudo é feito a partir das necessidades dos clientes e não a partir de soluções pré-programadas”.

“O cliente terá aquilo que precisa e valoriza no curto prazo e com facilidade de implantação e manutenção”. (FILADORO, 2014)



**Gráfico 2** - Grau de satisfação do cliente antes do SCRUM

Fonte: Miranda; Corrêa, 2012.



**Gráfico 3** - Grau de satisfação do cliente depois do SCRUM

Fonte: Miranda; Corrêa, 2012.

Para Schwaber e Sutherland (2013, p. 3) Scrum é “Um framework dentro do qual pessoas podem tratar e resolver problemas complexos e adaptativos, enquanto produtiva e criativamente entregam produtos com o mais alto valor possível”.

Quando Jeff Sutherland criou o processo scrum em 1993, ele emprestou o termo “scrum” de uma analogia apresentada em um estudo de 1986 por Takeuchi e Nonaka, publicado na Harvard Business Review. Nesse estudo, Takeuchi e Nonaka comparam equipes multifuncionais, alto desempenho para a formação de scrum usado pelas equipes de Rugby. (SCRUM ALLIANCE, 2016, tradução nossa)

Como um framework para desenvolvimento ágil o Scrum é consistente com os valores do manifesto ágil e para a equipe do desenvolvimento do produto que é a maior responsável pelos resultados tais valores como: foco, coragem, franqueza, comprometimento e respeito é essencial para a saúde e o sucesso de todo processo (SCRUM ALLIANCE, 2016). Baseia-se no sistema Toyota de produção desenvolvido por Taiichi Ohno e no ciclo OODA (Observe, Orient, Decide, Act) da aviação de combate (SUTHERLAND, 2014).

Scrum é fundamentado nas teorias empíricas de controle de processo, ou empirismo. O empirismo. O empirismo afirma que o conhecimento vem da experiência e de tomada de decisões baseadas no que é conhecido. O Scrum emprega uma abordagem iterativa e incremental para aperfeiçoar a previsibilidade e o controle de riscos. (SHWABER; SUTHERLAND, 2013, p. 4).

Segundo Schwaber e Sutherland (2013, p. 4) “Três pilares apoiam a implementação de controle de processo empírico: transparência, inspeção e adaptação”.

Transparência

Para Schwaber e Sutherland (2013, p. 4) “Aspectos significativos do processo devem estar visíveis aos responsáveis pelos resultados. Esta transparência requer aspectos definidos por um padrão comum para que os observadores compartilharem um mesmo entendimento do que está sendo visto”.

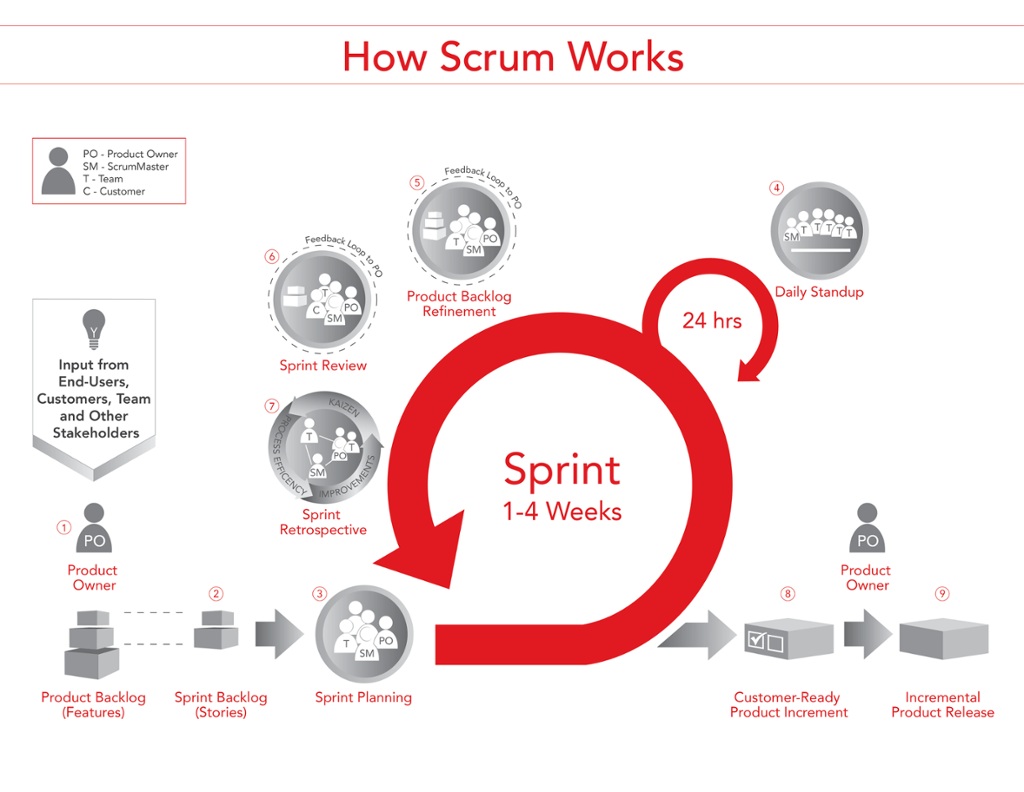
Inspeção

Devem ser inspecionados frequentemente os vários aspectos do processo para identificar variações inaceitáveis. A frequência da inspeção deve levar em consideração que os processos são alterados pelo próprio ato de inspeção, por isso deve ficar atento para que não exceda o limite e chega a atrapalhar a própria execução do processo. Outro fator na inspeção é o inspetor, que deve possuir as competências para avaliar o que ele ou ela está inspecionando. (SCHWABER, 2004)

Adaptação

Uma vez passado pelo processo de inspeção o consequentemente o resultado levará a uma adaptação visando melhorar o processo.

Se um inspetor determina que um ou mais aspectos de um processo desviou para fora dos limites aceitáveis, e que o produto resultado será inaceitável, o processo ou o material sendo produzido deve ser ajustado. O ajuste deve ser realizado o mais breve possível para minimizar mais desvios. (SHWABER; SUTHERLAND, 2013, p. 4)



**Figura 7** - Scrum Framework

Fonte: Scruminc, 2016.

### Time Scrum

Segundo Schwaber e Sutherland (2013, p. 5) “[...] é composto pelo Product Owner, o Time de Desenvolvimento e o Scrum Master. Times Scrum são auto-organizáveis e multifuncionais. [...] é projetado para aperfeiçoar a flexibilidade, criatividade e produtividade”.

Para Schwaber e Sutherland (2013, p. 5) o “Times Scrum entregam produtos de forma iterativa e incremental, maximizando as oportunidades de realimentação. Entregas incrementais de produto “Pronto” garantem que uma versão potencialmente funcional do produto do trabalho esteja sempre disponível”.

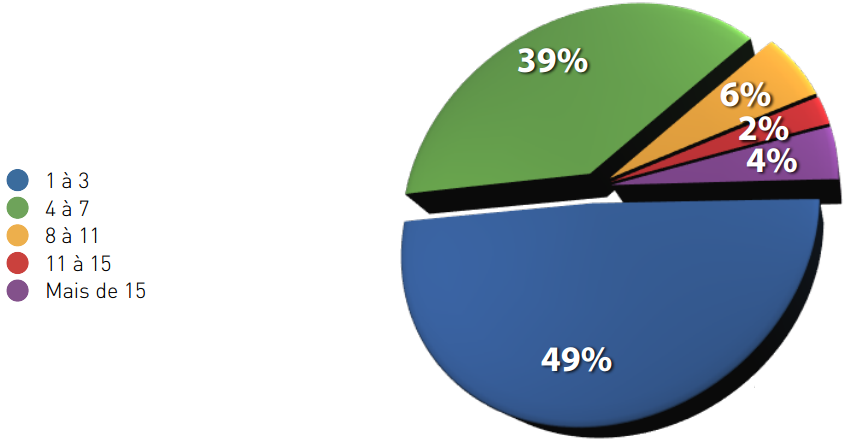
#### Product Owner

De acordo com Schwaber e Sutherland (2013, p. 5) “[...] é o responsável por maximizar o valor do produto e do trabalho do Time de Desenvolvimento” e “[...] a única pessoa responsável por gerenciar o Backlog do Produto”. Segundo Vieira (2014) “[...] é o ponto central com poderes de liderança sobre o produto. Ele é o único responsável por decidir quais recursos e funcionalidades serão construídos e qual a ordem que devem ser feitos”.

Segundo Vieira (2014) “Para garantir que a equipe construa rapidamente o que o Product Owner precisa, ele deve colaborar ativamente com o ScrumMaster e equipe de desenvolvimento e deve estar disponível para responder às perguntas tão logo estas são feitas”.

#### Time de Desenvolvimento

Para Schwaber e Sutherland (2013, p. 5) “[...] consiste de profissionais que realizam o trabalho de entregar uma versão usável que potencialmente incrementa o produto “Pronto” ao final de cada Sprint. Somente integrantes do Time de Desenvolvimento criam incrementos”. O time de desenvolvimento deve ser multifuncional e composto entre 3 a 9 integrantes. (SHWABER; SUTHERLAND, 2013)



**Gráfico 4** - Qual é o tamanho da sua equipe?

Fonte: Luz Consultoria, 2012.

O gráfico 4 é resultado de uma pesquisa realizada pela Luz Consultoria. Ele representa a quantidade de pessoas que compõem uma equipe nas startups, com isso é possível perceber que realmente o número ideal para as equipes é de 3 a 9 pessoas.

A principal abordagem para trabalhar com equipes grandes no Scrum é usando o conceito de “Scrum of Scrums”. Cada Scrum Team trabalha normalmente, mas cada equipe também contribui com uma pessoa que deverá freqüentar o Scrum of Scrums Meeting para coordenar o trabalho de múltiplas equipes Scrum. (DESENVOLVIMENTO ÁGIL, 2016)

De acordo com Vieira (2014) “A idéia principal é que a equipe de desenvolvimento se auto-organiza para determinar a melhor maneira de realizar o trabalho para atingir a meta estabelecida pelo Product Owner”.

#### Scrum Master

Segundo Auday (2014) “[...] trabalha focado em facilitar e potencializar o trabalho do Time de Scrum, utilizando seu conhecimento de Scrum, suas habilidades em lidar com pessoas, em facilitação, em comunicação, em resolução de conflitos, entre outras”.

O ScrumMaster faz tudo o possível para ajudar a equipe a se apresentar no seu nível mais elevado. Isso envolve a remoção de quaisquer impedimentos para o progresso, facilitando reuniões, e fazendo coisas como trabalhar para o Product Owner para certificar-se que o Product Backlog está em boa forma e pronto para o próximo Sprint. O papel do ScrumMaster é comumente preenchido por um gerente de projeto antigo ou um líder de equipe técnica, mas pode ser qualquer um. (MOUNTAIN GOAT SOFTWARE, 2016, tradução nossa)

Para Schwaber e Sutherland (2013, p. 7) “O Scrum Master ajuda aqueles que estão fora do Time Scrum a entender quais as suas interações com o Time Scrum são úteis e quais não são”.

### Eventos Scrum

De acordo com Schwaber e Sutherland (2013, p. 8) “[...] são usados no Scrum para criar uma rotina e minimizar a necessidade de reuniões não definidas no Scrum. Todos os eventos são eventos time-boxed, de tal modo que todo evento tem uma duração máxima”. Eventos como: Sprint, Reunião de Planejamento da Sprint, Reunião Diária, Revisão da Sprint e Retrospectiva da Sprint. (SHWABER; SUTHERLAND, 2013)

[...] cada evento no Scrum é uma oportunidade de inspecionar e adaptar alguma coisa. Estes eventos são especificamente projetados para permitir uma transparência e inspeção criteriosa. A não inclusão de qualquer um dos eventos resultará na redução da transparência e da perda de oportunidade para inspecionar e adaptar. (SHWABER; SUTHERLAND, 2013, p. 8)

#### Sprint

É o coração do Scrum, ou seja, contém todos os elementos do Scrum. Duração de não mais de um mês, e hoje comumente usada pelos times entre uma ou duas semanas. O objetivo é ter uma interação curta o suficiente para manter o time focado, mas tempo suficiente para entregar um incremento significativo do trabalho. (SCRUMINC, 2016)

Segundo Schwaber e Sutherland (2013, p. 9) “Uma Sprint pode ser cancelada antes do time-boxed da Sprint terminar. Somente o Product Owner tem a autoridade para cancelar a Sprint, embora ele (ou ela) possa fazer isso sob influência das partes interessadas, do Time de Desenvolvimento ou do Scrum Master”.

#### Reunião de Planejamento da Sprint

O site Desenvolvimento Ágil (2016) informa que “O Sprint Planning Meeting é uma reunião na qual estão presentes o Product Owner, o Scrum Master e todo o Scrum Team, bem como qualquer pessoa interessada que esteja representando a gerência ou o cliente”.

De acordo com Schwaber e Sutherland (2013, p. 9) a “Reunião de planejamento da Sprint possui um time-box com no máximo oito horas para uma Sprint de um mês de duração. Para Sprints menores, este evento é usualmente menor. O Scrum Master garante que o evento ocorra e que os participantes entendam seu propósito”.

O Sprint Planning Meeting é normalmente dividido em duas partes. Na primeira parte o Product Owner apresenta as prioridades do Product Backlog, descreve o que ele deseja ver construído e esclarece dúvidas. Ao final o time seleciona os itens do Product Backlog e define o Sprint Goal (Objetivo) juntamente com o Product Owner. (MITCH LACEY & ASSOCIATES, 2016)

Para a segunda parte:

Tendo definido o objetivo da Sprint e selecionado os itens de Backlog do Produto da Sprint, o Time de Desenvolvimento decide como irá construir essas funcionalidades durante a Sprint e transformá-las em um incremento de produto “Pronto”. Os itens de Backlog do Produto selecionados para a Sprint, junto com o plano de entrega destes itens é chamado de Backlog da Sprint. (SHWABER; SUTHERLAND, 2013, p. 10)

#### Reunião Diária

Chamada também de “Daily Scrum”, são reuniões diárias composta pelo mestre scrum e o time de desenvolvimento. São realizadas sempre no mesmo lugar e na mesma hora definida, com duração máxima de 15 minutos. Cada integrante do time deverá responder as três perguntas: O que eu fiz ontem que ajudou o Time de Desenvolvimento a atender a meta da Sprint? O que eu farei hoje para ajudar o Time de Desenvolvimento atender a meta da Sprint? Eu vejo algum obstáculo que impeça a mim ou o Time de Desenvolvimento no atendimento da meta da Sprint? (SHWABER; SUTHERLAND, 2013, p. 11)

O Daily Scrum não deve ser usado como uma reunião para resolução de problemas. Questões levantadas devem ser levadas para fora da reunião e normalmente tratadas por um grupo menor de pessoas que tenham a ver diretamente com o problema ou possam contribuir para solucioná-lo. (DESENVOLVIMENTO ÁGIL, 2016)

Reuniões Diárias melhoram as comunicações, eliminam outras reuniões, identificam e removem impedimentos para o desenvolvimento, destacam e promovem rápidas tomadas de decisão, e melhoram o nível de conhecimento do Time de Desenvolvimento. Esta é uma reunião chave para inspeção e adaptação. (SHWABER; SUTHERLAND, 2013, p. 11)

O site Desenvolvimento Ágil (2016) afirma que “Os impedimentos identificados no Daily Scrum devem ser tratados pelo Scrum Master o mais rapidamente possível”.

#### Revisão da Sprint

Realizada no final da Sprint com duração máxima de 4 horas para um Sprint de um mês, tem como objetivo inspecionar o incremento e adaptar o Backlog do Produto se necessário. É uma reunião informal composta pelo Time de Desenvolvimento, ScrumMaster, Product Owner e Stakeholders convidados. (SHWABER; SUTHERLAND, 2013, p. 12)

Durante essa reunião o Time Scrum mostra quais itens do Product Backlog que eles concluíram durante a Sprint. Isso pode ocorrer sob a forma uma demo das novas funcionalidades. É importante notar que os itens do Backlog que não estão concluídos não devem ser demonstrados. (SCRUM INSTITUTE, 2016)

De acordo com Shwaber e Sutherland (2013, p. 12) “O resultado da Reunião de Revisão da Sprint é um Backlog do Produto revisado que define o provável Backlog do Produto para a próxima Sprint. O Backlog do Produto pode também ser ajustado completamente para atender novas oportunidades”.

#### Retrospectiva da Sprint

De acordo com Shwaber e Sutherland (2013, p. 12) “A Retrospectiva da Sprint é uma oportunidade para o Time Scrum inspecionar a si próprio e criar um plano para melhorias a serem aplicadas na próxima Sprint”.

Após a reunião de revisão da Sprint tem lugar a reunião de retrospectiva da Sprint formada pelo Time Scrum e o Scrum Master. Nesta reunião, todos os membros da equipe refletem sobre a Sprint passada em verificar esses três pontos: o que ocorreu bem durante a Sprint, o que não e que melhorias poderiam ser feitas na próxima Sprint. A reunião deve ser time-boxed (por exemplo, 3 horas). (SCRUM INSTITUTE, 2016)

Ao final da Retrospectiva da Sprint, o Time Scrum deverá ter identificado melhorias que serão implementadas na próxima Sprint. A implementação destas melhorias na próxima Sprint é a forma de adaptação à inspeção que o Time Scrum faz a si próprio. A Retrospectiva da Sprint fornece um evento dedicado e focado na inspeção e adaptação, no entanto, as melhorias podem ser adotadas a qualquer momento. (SHWABER; SUTHERLAND, 2013, p. 13)

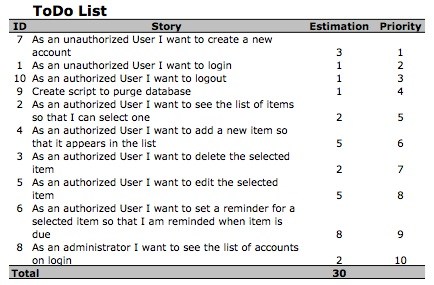
### Artefatos do Scrum

Segundo Shwaber e Sutherland (2013, p.13) “Os artefatos do Scrum representam o trabalho ou o valor para o fornecimento de transparência e oportunidades para inspeção e adaptação”.

#### Product Backlog

O Backlog do produto é uma lista priorizada de características, contendo descrições curtas das funcionalidades desejadas no produto (MOUNTAIN GOAT SOFTWARE, 2016). Segundo Schwaber e Sutherland (2013, p. 13) “O Product Owner é responsável pelo Backlog do Produto, incluindo seu conteúdo, disponibilidade e ordenação”. O Scrum Master, o Time de Desenvolvimento e outros Stackholders também contribuem no Product Backlog. (SCRUM INSTITUTE, 2016)

De acordo com Schwaber e Sutherland (2013, p. 14) “O Backlog do Produto lista todas as características, funções, requisitos, melhorias e correções que formam as mudanças que devem ser feitas no produto nas futuras versões. Os itens do Backlog do Produto possuem os atributos de descrição, ordem, estimativa e valor”. A melhor forma de expressar um item do backlog é na forma de user stories (histórias de usuário), exemplo: “Como um cliente da loja online eu gostaria de procurar por itens para adicionar ao meu pedido”. (MOUNTAIN GOAT SOFTWARE, 2016)



**Tabela 2** - Exemplo Scrum Product Backlog

Fonte: Scrum Institute, 2016.

Prioridade do cliente, Urgência em receber feedback, dificuldade de implementação e relações simbióticas entre os itens (B é mais fácil se fizer o A primeiro) podem influenciar na priorização do Product Owner. (RADIGAN, 2016)

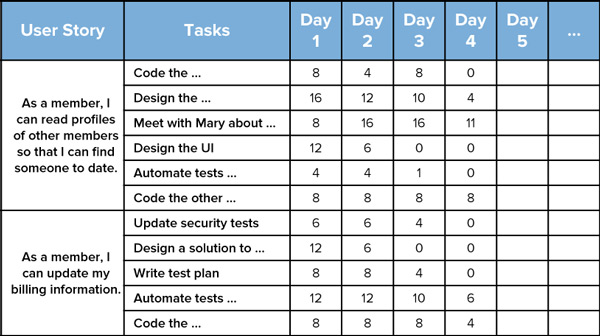
Para Schwaber e Sutherland (2013, p. 14) “O Time de Desenvolvimento é responsável por todas as estimativas. O Product Owner deve influenciar o Time, ajudando no entendimento e nas decisões conflituosas de troca, mas as pessoas que irão realizar o trabalham fazem a estimativa final”.

Várias práticas como burndown, burnup e outras práticas de estimativa tem sido usadas para prever o progresso. Estas tem se provado úteis. Contudo, não substituem a importância do empirismo. Em ambientes complexos, o que acontecerá é desconhecido. Somente o que tem acontecido pode ser usado para uma tomada de decisão a respeito do que virá. (SHWABER; SUTHERLAND, 2013, p. 15)

#### Sprint Backlog

O Sprint Backlog é uma lista de tarefas identificadas pelo time Scrum para ser concluída durante a Sprint. Durante a reunião de planejamento da Sprint, o time seleciona alguns itens (user story) do Product Backlog, e identifica as tarefas necessárias para completar cada user story. (MOUNTAIN GOAT, 2016)

Para Schwaber e Sutherland (2013, p. 15) “O Backlog da Sprint é a previsão do Time de Desenvolvimento sobre qual funcionalidade estará no próximo incremento e sobre o trabalho necessário para entregar essa funcionalidade em um incremento [...]”. É fundamental que o time seleciona os itens e tamanho o do Backlog da Sprint, porque eles são as pessoas comprometidas a concluir as tarefas. (MOUNTAIN GOAT, 2016)



**Tabela 3** - Sprint Backlog

Fonte: Mountain Goat, 2016.

Em qualquer ponto do tempo na Sprint, o total do trabalho remanescente dos itens do Backlog da Sprint pode ser somado. O Time de Desenvolvimento monitora o total do trabalho restante pelo menos a cada Reunião Diária. O Time de Desenvolvimento acompanha estes resumos diários e projeta a probabilidade de alcançar o objetivo da Sprint. Com o rastreamento do trabalho restante em toda a Sprint, o Time de Desenvolvimento pode gerenciar o seu progresso. (SHWABER; SUTHERLAND, 2013, p. 15)

#### Incremento

Para Bonfim (2015) “Ao final de uma sprint, todos os itens que foram finalizados formam um incremento do produto. Esse incremento deve ser entregável e utilizável, de maneira que o cliente perceba valor no produto a cada final de Sprint”.

Schwaber (2004) também afirma que “O incremento é a soma de todos os itens completados do Backlog do Produto. No final de um Sprint, o novo incremento deve estar “pronto”, o que significa que ele está em uma condição utilizável e atende à definição da Equipe do Scrum”.

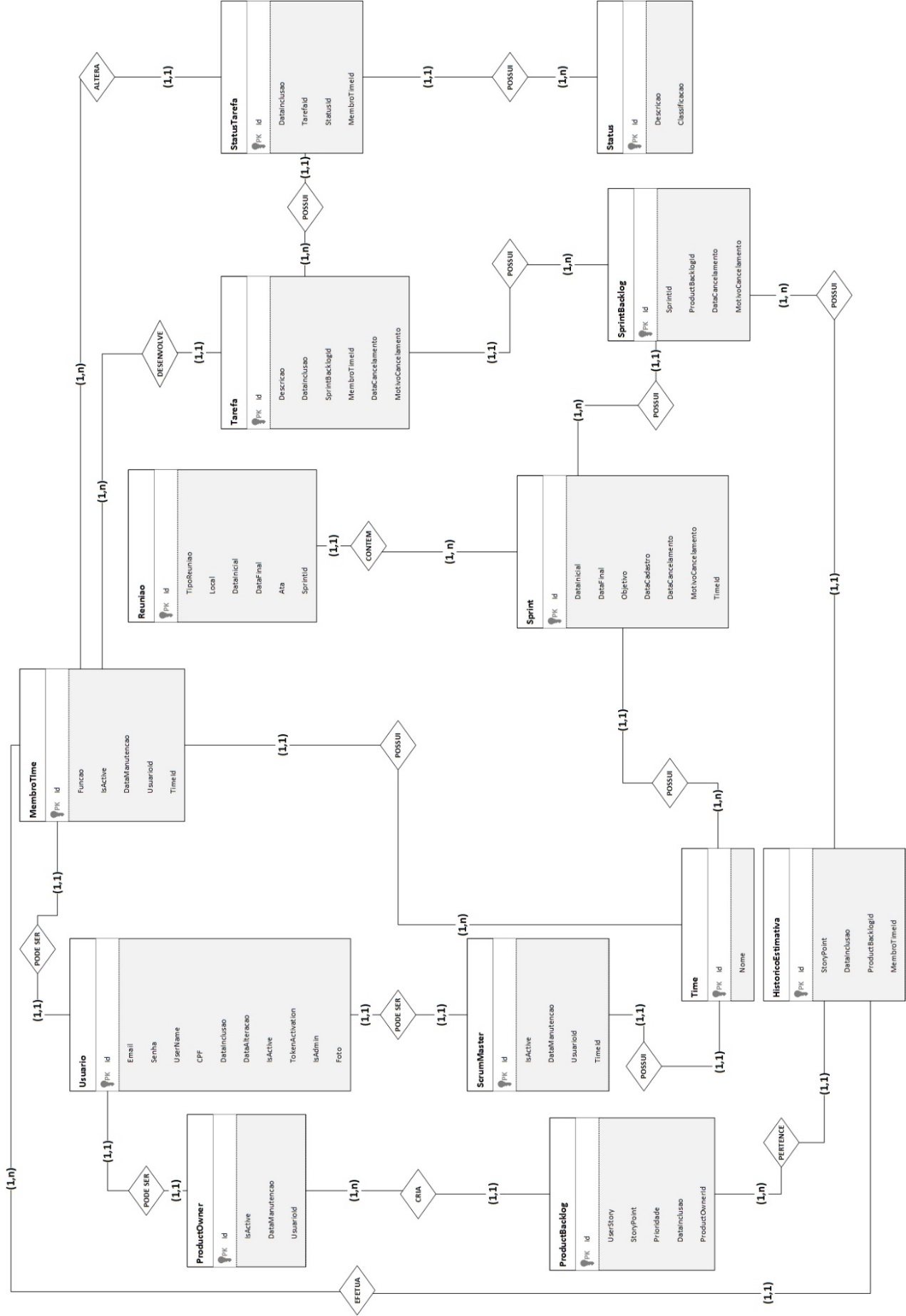
“É através da definição de pronto que se assegura a transparência de quando determinada parte do trabalho estará realmente finalizada”. (BONFIM, 2015)

Bonfim (2015) também afirma que “somente itens com definição de pronto clara são considerados aptos para terem sua estimativa realizada e seu desenvolvimento iniciado, uma vez que sem essa informação é impossível para o Time de Desenvolvimento estimar e realizar o trabalho”.

# DESENVOLVENDO O PROJETO PROPOSTO

## Diagramas UML

### MER – Modelo entidade e relacionamento



**Figura 8** - MER

### Dicionário de dados

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entidade: Usuário** | | | | | |
| Descrição: Armazenar os dados de cadastro do usuário | | | | | |
| Nro do Atributo | Atributo | Tipo | Chave Primária | Chave Estrangeira | Descrição |
| 1 | Id | Integer | X |  | Identificador do usuário |
| 2 | UserName | nvarchar(20) |  |  | Apelido de acesso do usuário |
| 3 | Email | nvarchar(200) |  |  | E-mail do usuário |
| 4 | Senha | nvarchar(max) |  |  | Senha de acesso do usuário |
| 5 | CPF | nvarchar(11) |  |  | CPF do usuário |
| 6 | DataInclusao | datetime |  |  | Data de inclusão do usuário |
| 7 | DataAlteracao | datetime |  |  | Data de alteração dos dados do usuário |
| 8 | IsActive | bit |  |  | Verificação se o usuário está ativo |
| 9 | TokenActivation | uniqueidentifier |  |  | Token de troca de senha |
| 10 | IsAdmin | bit |  |  | Verificação se o usuário é administrador do sistema |
| 11 | Foto | varbinary(max) |  |  | Foto do usuário |
| RELACIONAMENTOS: | | | | | |
| ID | Entidade | | Atributo Origem | Atributo Destino |  |
| Não há |  | |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entidade: MembroTime** | | | | | |
| Descrição: Armazena os dados de um membro do time | | | | | |
| Nro do Atributo | Atributo | Tipo | Chave Primária | Chave Estrangeira | Descrição |
| 1 | Id | Integer | X |  | Identificador do membro do time |
| 2 | TimeId | Integer |  | X | Identificador do time no qual o membro está vinculado |
| 3 | UsuarioId | Integer |  | X | Identificador do usuário no qual o membro está vinculado |
| 4 | Funcao | nvarchar(40) |  |  | Descrição da função do membro dentro do time |
| 5 | IsActive | bit |  |  | Verificação se o membro do time está ativo |
| 6 | DataManutencao | datetime |  |  | Data de manutenção do membro do time |
| RELACIONAMENTOS: | | | | | |
| ID | Entidade | | Atributo Origem | Atributo Destino |  |
| 1 | Time | | Id | TimeId |  |
| 2 | Usuario | | Id | UsuarioId |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entidade: ProductOwner** | | | | | |
| Descrição: Armazena os dados pertinentes ao dono do produto | | | | | |
| Nro do Atributo | Atributo | Tipo | Chave Primária | Chave Estrangeira | Descrição |
| 1 | Id | Integer | X |  | Identificador do dono do produto |
| 2 | UsuarioId | Integer |  | X | Identificador do usuário no qual o dono do produto está vinculado |
| 3 | IsActive | bit |  |  | Verificação se o product owner está ativo |
| 4 | DataManutencao |  |  |  | Data de manutenção do product owner |
| RELACIONAMENTOS: | | | | | |
| ID | Entidade | | Atributo Origem | Atributo Destino |  |
| 1 | Usuario | | Id | UsuarioId |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entidade: ScrumMaster** | | | | | |
| Descrição: Armazena os dados pertinentes ao scrum master | | | | | |
| Nro do Atributo | Atributo | Tipo | Chave Primária | Chave Estrangeira | Descrição |
| 1 | Id | Integer | X |  | Identificador do scrum master |
| 2 | UsuarioId | Integer |  | X | Identificador do usuário no qual o scrum master está vinculado |
| 3 | IsActive | bit |  |  | Verificação se o scrum master está ativo |
| 4 | TimeId | Integer |  | X | Identificador do time no qual o scrum master está vinculado |
| 5 | DataManutencao | datetime |  |  | Data de manutenção do scrum master |
| RELACIONAMENTOS: | | | | | |
| ID | Entidade | | Atributo Origem | Atributo Destino |  |
| 1 | Usuario | | Id | UsuarioId |  |
| 2 | Time | | Id | TimeId |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entidade: Time** | | | | | |
| Descrição: Armazena os dados pertinentes ao time | | | | | |
| Nro do Atributo | Atributo | Tipo | Chave Primária | Chave Estrangeira | Descrição |
| 1 | Id | Integer | X |  | Identificador do Time |
| 2 | Nome | nvarchar(100) |  |  | Nome do Time |
| RELACIONAMENTOS: | | | | | |
| ID | Entidade | | Atributo Origem | Atributo Destino |  |
| Não há |  | |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entidade: Sprint** | | | | | |
| Descrição: Armazena os dados da sprint | | | | | |
| Nro do Atributo | Atributo | Tipo | Chave Primária | Chave Estrangeira | Descrição |
| 1 | Id | Integer | X |  | Identificador da sprint |
| 2 | TimeId | Integer |  | X | Identificador do time vinculado a sprint |
| 3 | DataInicial | datetime |  |  | Data de início da sprint |
| 4 | DataFinal | datetime |  |  | Data do fim da sprint |
| 5 | Objetivo | nvarchar(100) |  |  | Objetivo da sprint |
| 6 | DataCadastro | datetime |  |  | Data de cadastro da sprint |
| 7 | DataCancelamento | datetime |  |  | Data de cancelamento da sprint |
| 8 | MotivoCancelamento | nvarchar(max) |  |  | Descrição do motivo de cancelamento |
| RELACIONAMENTOS: | | | | | |
| ID | Entidade | | Atributo Origem | Atributo Destino |  |
| 1 | Time | | Id | TimeId |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entidade: Reuniao** | | | | | |
| Descrição: Armazena os dados das reuniões | | | | | |
| Nro do Atributo | Atributo | Tipo | Chave Primária | Chave Estrangeira | Descrição |
| 1 | Id | Integer | X |  | Identificador da reunião |
| 2 | TipoReuniao | Integer |  |  | Classificação do tipo de reunião |
| 3 | Local | nvarchar(50) |  |  | Local da reunião |
| 4 | DataInicial | datetime |  |  | Data/Hora de início da reunião |
| 5 | DataFinal | datetime |  |  | Data/Hora do fim da reunião |
| 6 | Ata | nvarchar(max) |  |  | Observações sobre a reunião (ata) |
| 7 | SprintId | Integer |  | X | Identificador da sprint vinculada a reunião |
| RELACIONAMENTOS: | | | | | |
| ID | Entidade | | Atributo Origem | Atributo Destino |  |
| 1 | Sprint | | Id | SprintId |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entidade: SprintBacklog** | | | | | |
| Descrição: Armazena os dados pertinentes ao sprint backlog | | | | | |
| Nro do Atributo | Atributo | Tipo | Chave Primária | Chave Estrangeira | Descrição |
| 1 | Id | Integer | X |  | Identificador do sprint backlog |
| 2 | SprintId | Integer |  | X | Identificador da sprint vinculada ao sprint backlog |
| 3 | ProductBacklogId | nvarchar(max) |  | X | Identificador do sprint backlog item vinculado ao sprint backlog |
| 4 | DataCancelamento | datetime |  |  | Data do cancelamento da sprint backlog |
| 5 | MotivoCancelamento | nvarchar(max) |  |  | Motivo do cancelamento da sprint backlog |
| RELACIONAMENTOS: | | | | | |
| ID | Entidade | | Atributo Origem | Atributo Destino |  |
| 1 | Sprint | | Id | SprintId |  |
| 2 | ProductBacklog | | Id | ProductBacklogId |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entidade: ProductBacklog** | | | | | |
| Descrição: Armazena os dados pertinentes ao product backlog item | | | | | |
| Nro do Atributo | Atributo | Tipo | Chave Primária | Chave Estrangeira | Descrição |
| 1 | Id | Integer | X |  | Identificador do product backlog item |
| 2 | UserStory | nvarchar(150) |  |  | Descrição da implementação a ser feita |
| 3 | StoryPoint | integer |  |  | Nível de dificuldade do product backlog item |
| 4 | Prioridade | smallint |  |  | Prioridade do product backlog item |
| 5 | DataInclusao | datetime |  |  | Data de inclusão do product backlog item |
| 6 | ProductOwnerId | Integer |  | X | Identificador do dono do produto |
| RELACIONAMENTOS: | | | | | |
| ID | Entidade | | Atributo Origem | Atributo Destino |  |
| 1 | ProductOwner | | Id | ProductOwnerId |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entidade: HistoricoEstimativa** | | | | | |
| Descrição: Armazena os dados pertinentes aos históricos das estimativas | | | | | |
| Nro do Atributo | Atributo | Tipo | Chave Primária | Chave Estrangeira | Descrição |
| 1 | Id | Integer | X |  | Identificador da interação do product backlog item |
| 2 | StoryPoint | Integer |  |  | Classificação da interação entre product backlog item e os membros da equipe |
| 3 | DataInclusao | nvarchar(max) |  |  | Mensagem da interação |
| 4 | ProductBacklogId | datetime |  | X | Identificador do product backlog vinculado a estimativa |
| 5 | MembroTimeId | Integer |  | X | Identificador do membro do time responsável pela estimativa |
| RELACIONAMENTOS: | | | | | |
| ID | Entidade | | Atributo Origem | Atributo Destino |  |
| 1 | ProductBacklog | | Id | ProductBacklogId |  |
| 2 | MembroTime | | Id | MembroTimeId |  |

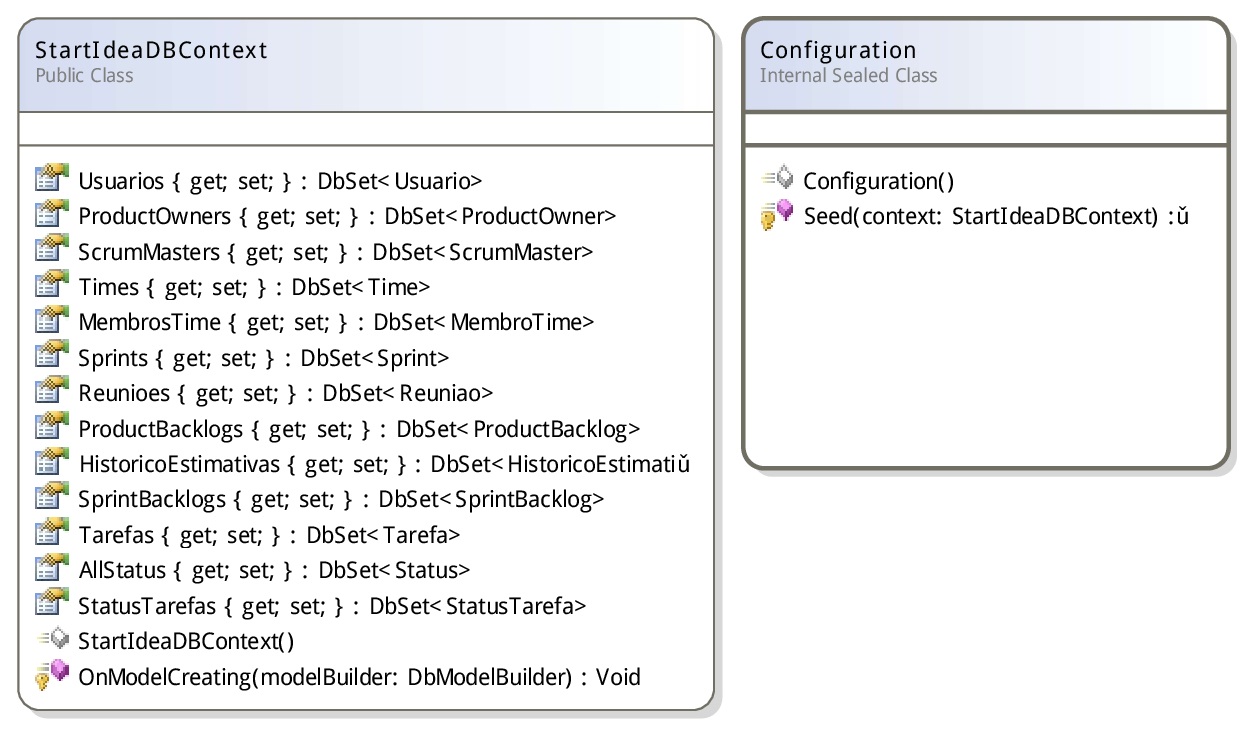
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entidade: Status** | | | | | |
| Descrição: Armazena os dados pertinentes aos status | | | | | |
| Nro do Atributo | Atributo | Tipo | Chave Primária | Chave Estrangeira | Descrição |
| 1 | Id | Integer | X |  | Identificador do status |
| 2 | Descricao | datetime |  |  | Descrição do status |
| 3 | Classificacao | Integer |  |  | Classificação do status |
| RELACIONAMENTOS: | | | | | |
| ID | Entidade | | Atributo Origem | Atributo Destino |  |
| Não há |  | |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entidade: Tarefa** | | | | | |
| Descrição: Armazena os dados pertinentes a tarefa | | | | | |
| Nro do Atributo | Atributo | Tipo | Chave Primária | Chave Estrangeira | Descrição |
| 1 | Id | Integer | X |  | Identificador da tarefa |
| 2 | Descricao | nvarchar(500) |  |  | Descrição da tarefa |
| 3 | DataInclusao | datetime |  |  | Data da inclusão da tarefa |
| 4 | SprintBacklogId | Integer |  | X | Identificador do sprint backlog a qual a tarefa está vinculada |
| 5 | MembroTimeId | Integer |  | X | Identificador do membro do time, na qual a tarefa está vinculada |
| 6 | DataCancelamento | datetime |  |  | Data de cancelamento da tarefa |
| 7 | MotivoCancelamento | nvarchar(max) |  |  | Motivo do cancelamento da tarefa |
| RELACIONAMENTOS: | | | | | |
| ID | Entidade | | Atributo Origem | Atributo Destino |  |
| 1 | SprintBacklog | | Id | SprintBacklogId |  |
| 2 | MembroTime | | Id | MembroTimeId |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entidade: StatusTarefa** | | | | | |
| Descrição: Armazena os dados pertinentes aos status da tarefa | | | | | |
| Nro do Atributo | Atributo | Tipo | Chave Primária | Chave Estrangeira | Descrição |
| 1 | Id | Integer | X |  | Identificador do status da tarefa |
| 2 | DataInclusao | datetime |  |  | Data de inclusão do status da tarefa |
| 3 | TarefaId | Integer |  | X | Identificador da tarefa à qual o status está vinculado |
| 4 | StatusId | Integer |  | X | Identificador do status |
| 5 | MembroTimeId | Integer |  | X | Identificador do membro do time relacionado ao status da tarefa |
| RELACIONAMENTOS: | | | | | |
| ID | Entidade | | Atributo Origem | Atributo Destino |  |
| 1 | Tarefa | | Id | TarefaId |  |
| 2 | Status | | Id | StatusId |  |
| 3 | MembroTime | | Id | MembroTimeId |  |

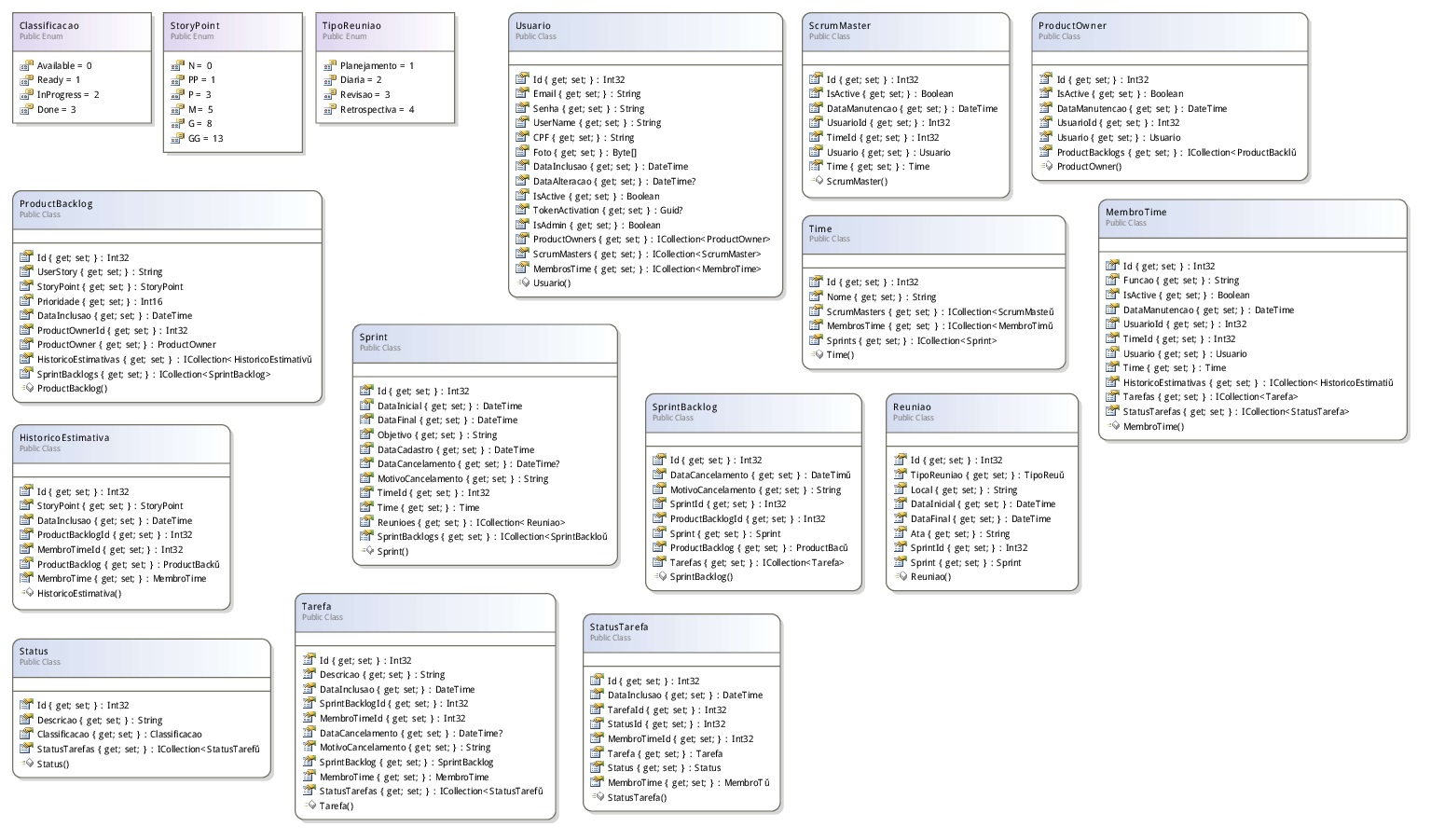
### Diagramas de classes

#### Configuração



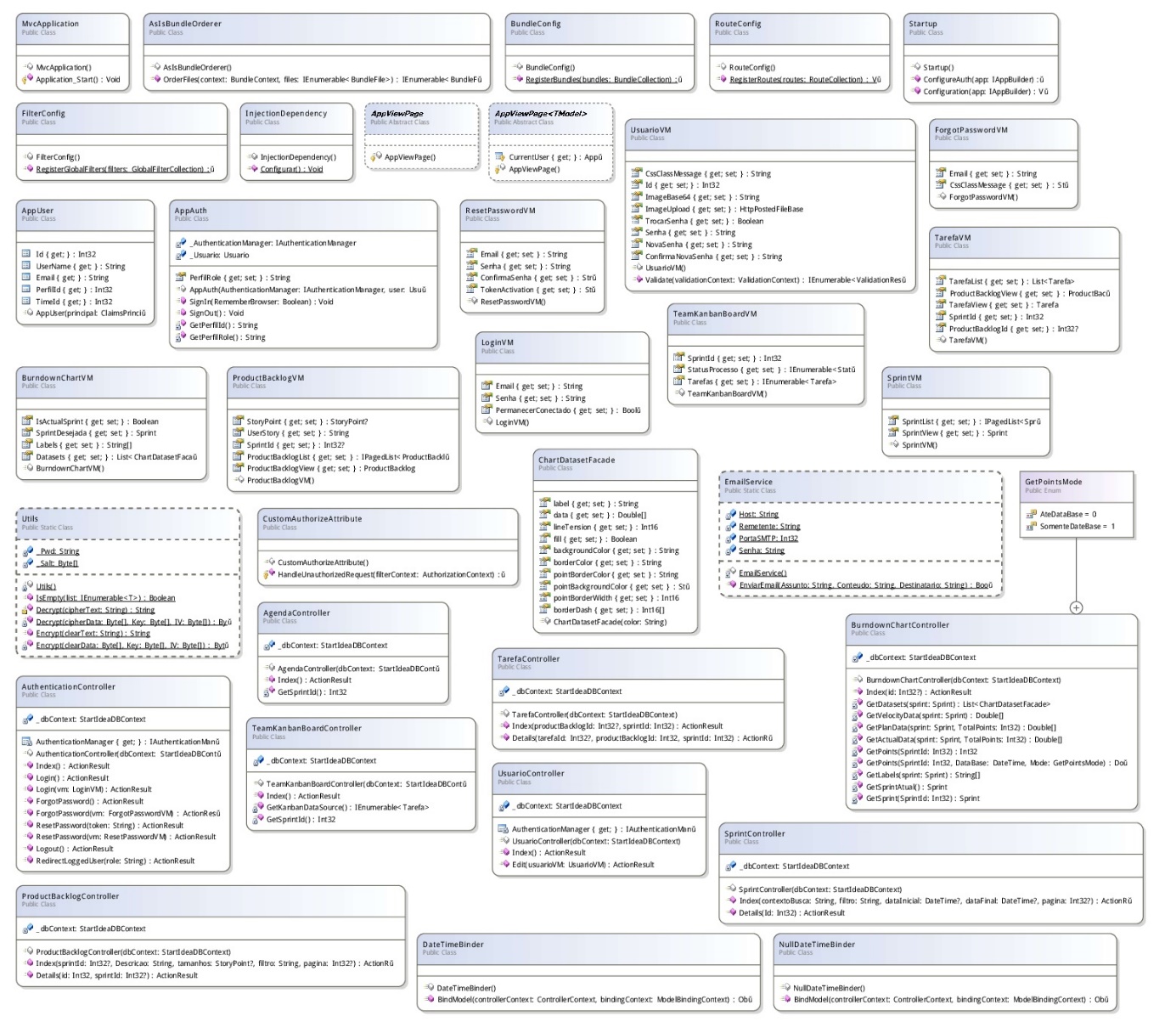
**Figura 9** - Diagrama de Classes - Configuração

#### Modelos



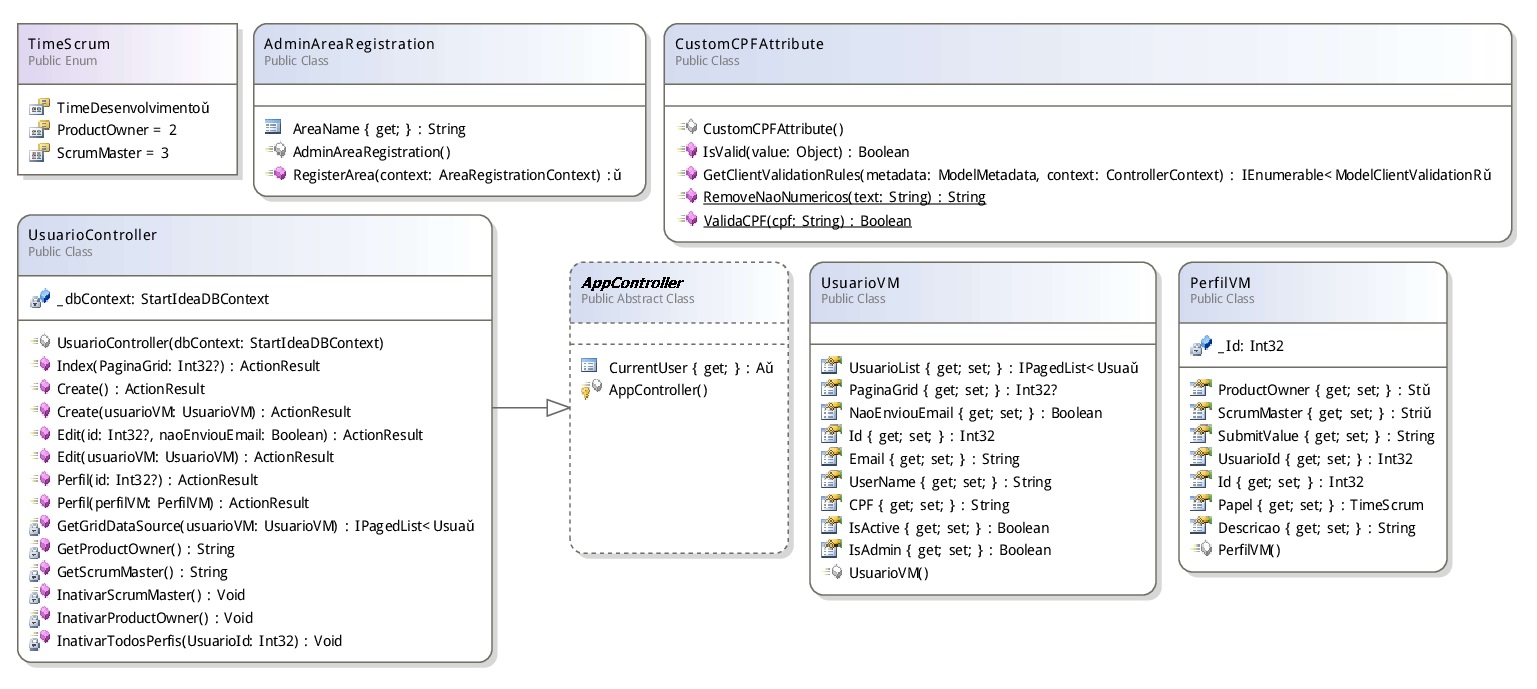
**Figura 10** - Diagrama de Classes - Modelos

#### Estrutura



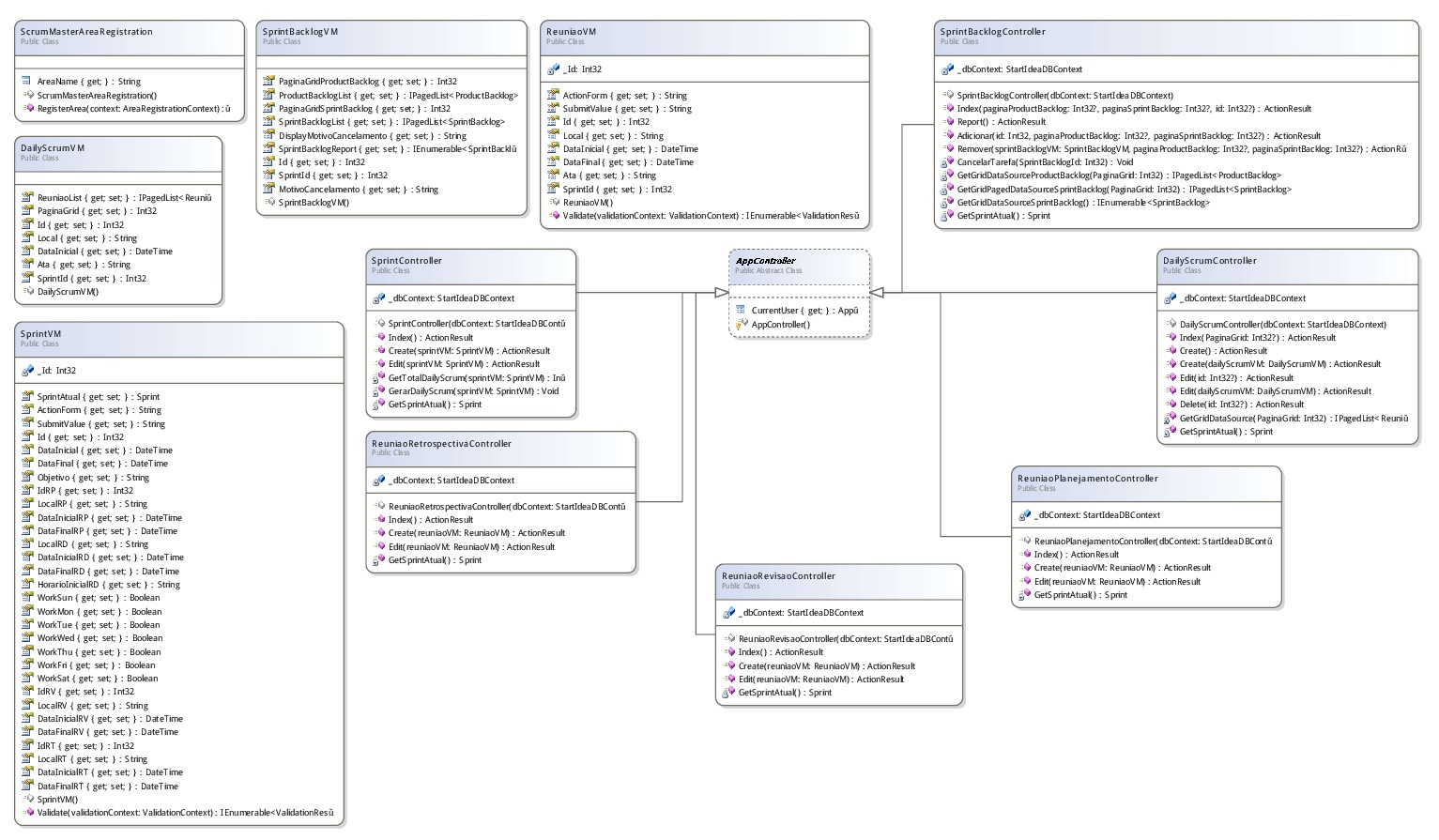
**Figura 11** - Diagrama de Classes - Estrutura

#### Administrador



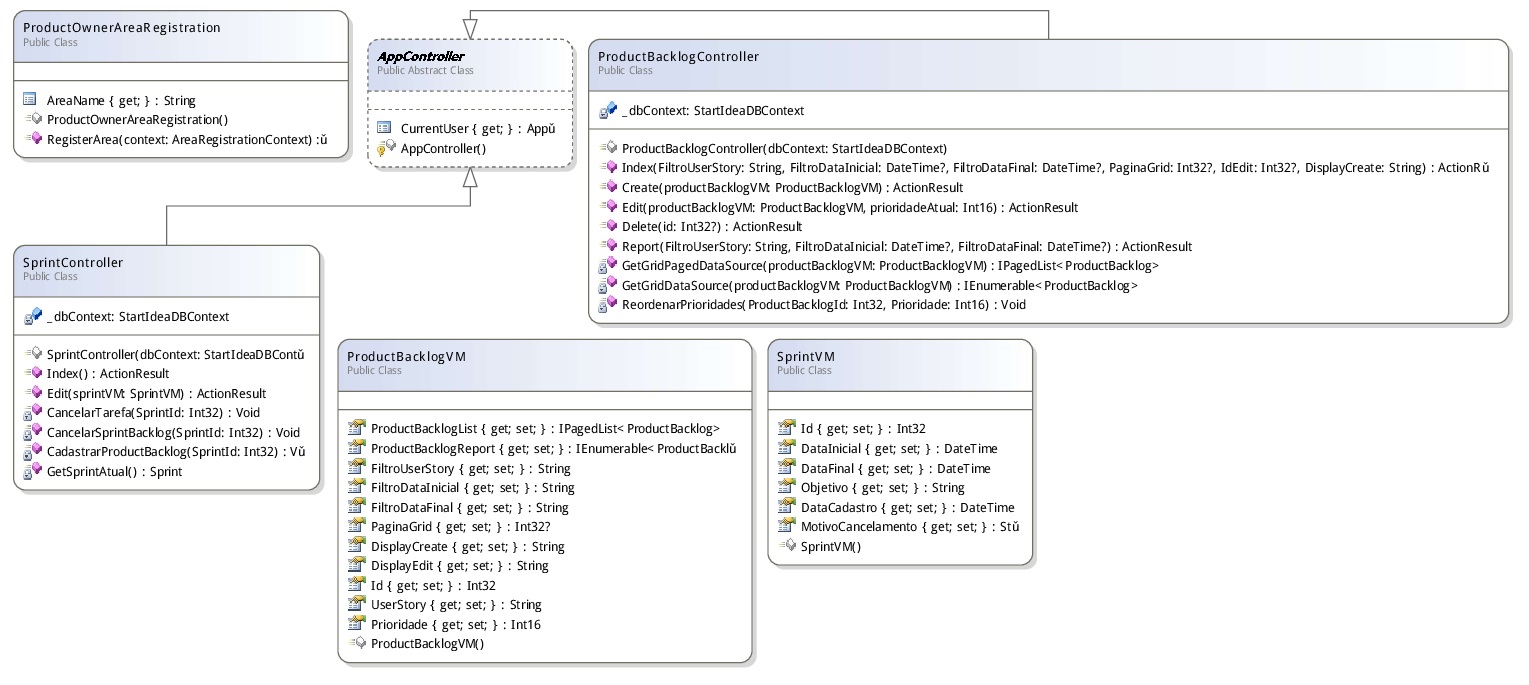
**Figura 12** - Diagrama de Classes - Administrador

#### Scrum Master



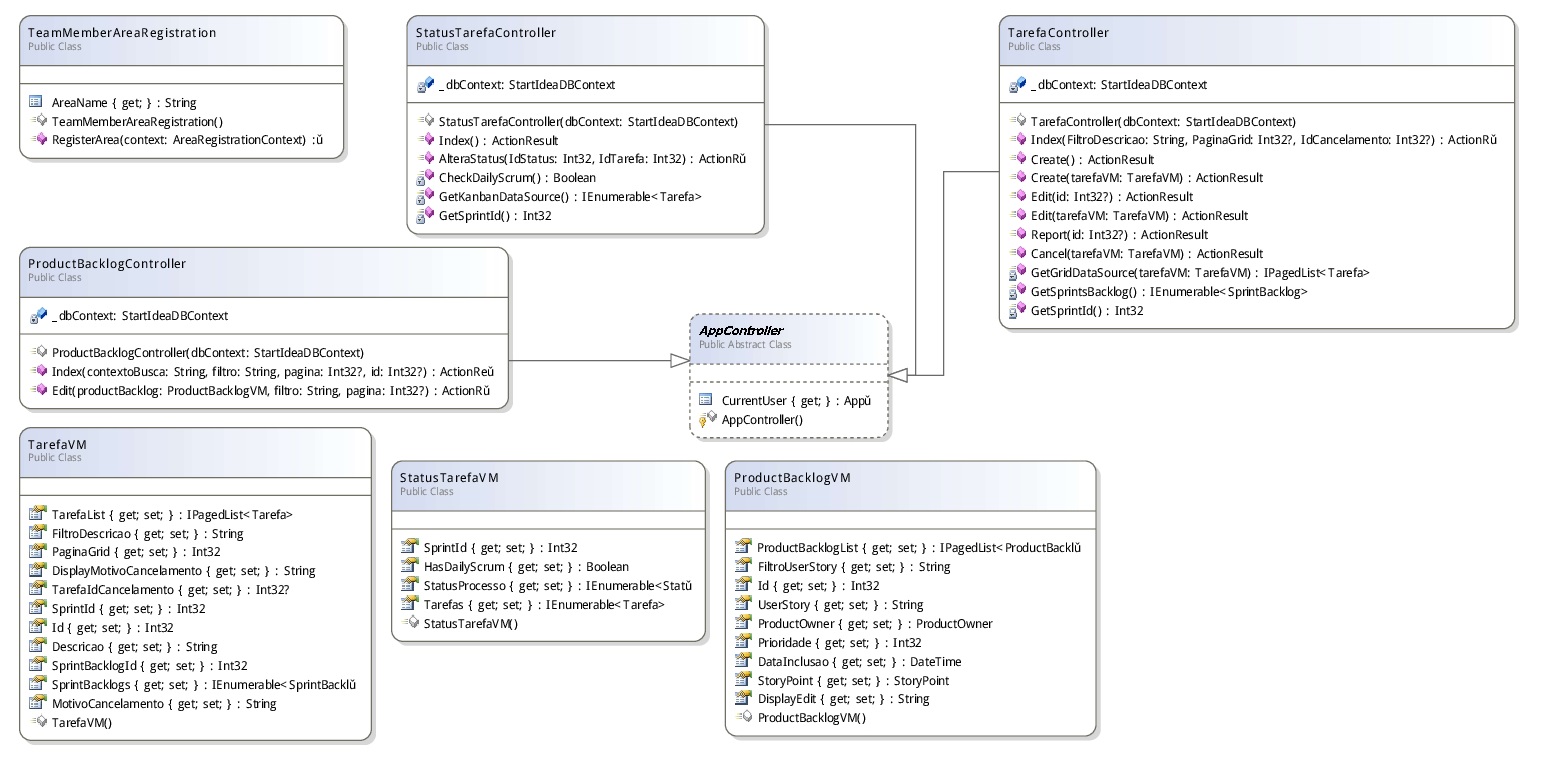
**Figura 13** - Diagrama de Classes - Scrum Master

#### Product Owner



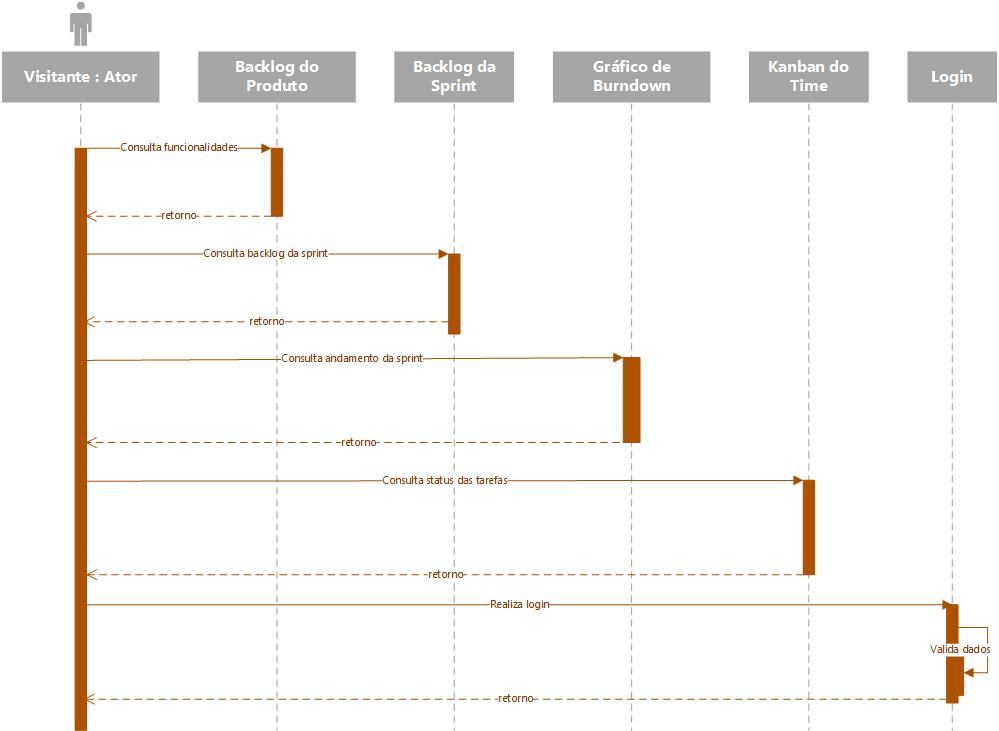
**Figura 14** - Diagrama de Classes - Product Owner

#### Membro do time

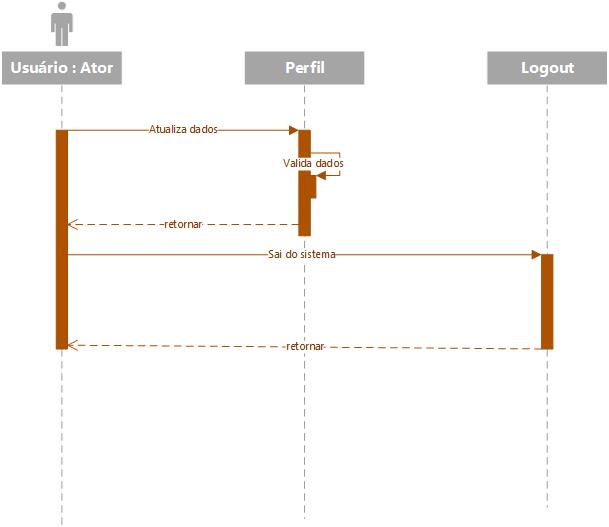


**Figura 15** - Diagrama de Classes - Membro Time

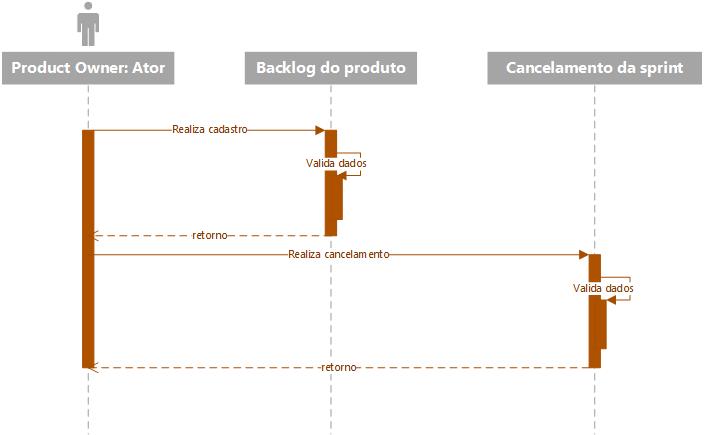
### Diagramas de sequência



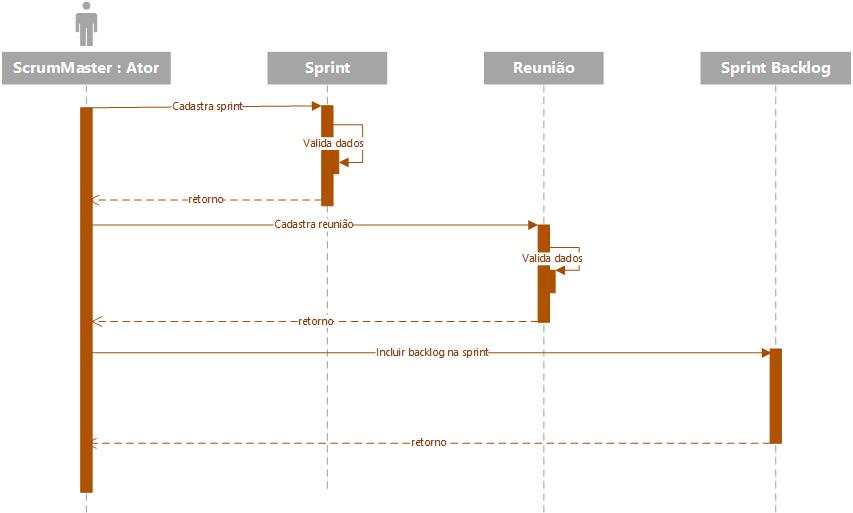
**Figura 16** - Diagrama de sequência – Visitante



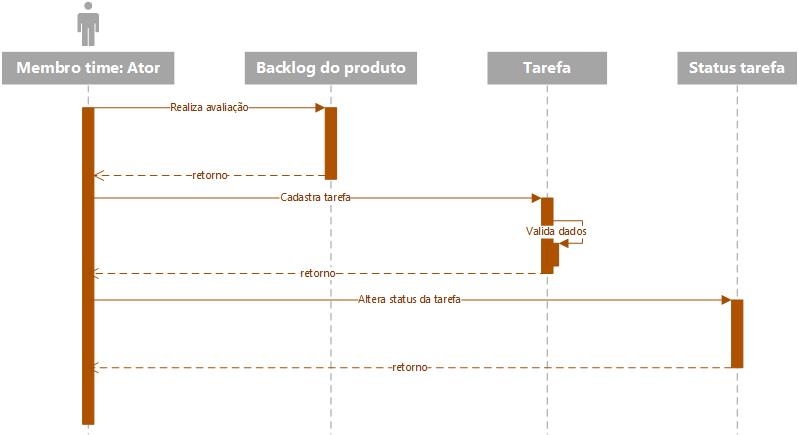
**Figura 17** - Diagrama de sequência – Usuário



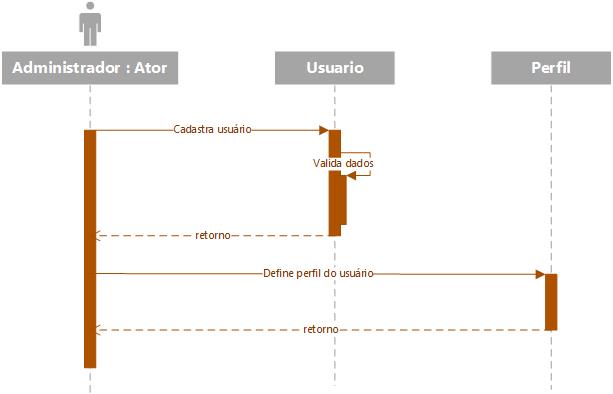
**Figura 18** - Diagrama de sequência - Product Owner



**Figura 19** - Diagrama de sequência - Scrum master



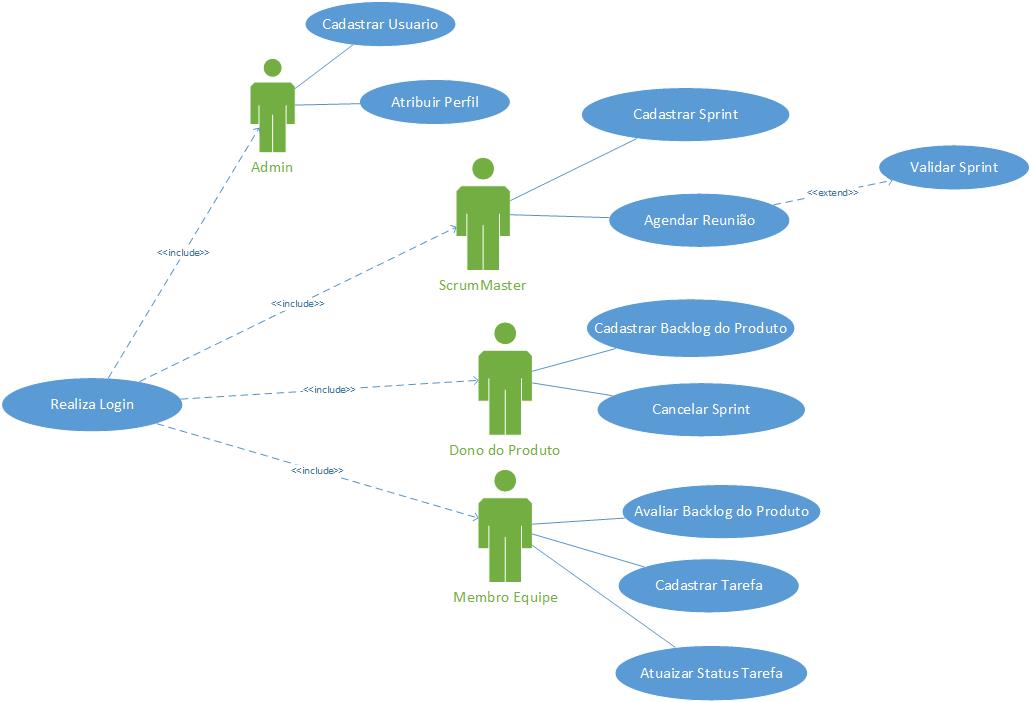
**Figura 20** - Diagrama de sequência - Membro time



**Figura 21** - Diagrama de sequência - Administrador

### Descrição de casos de uso

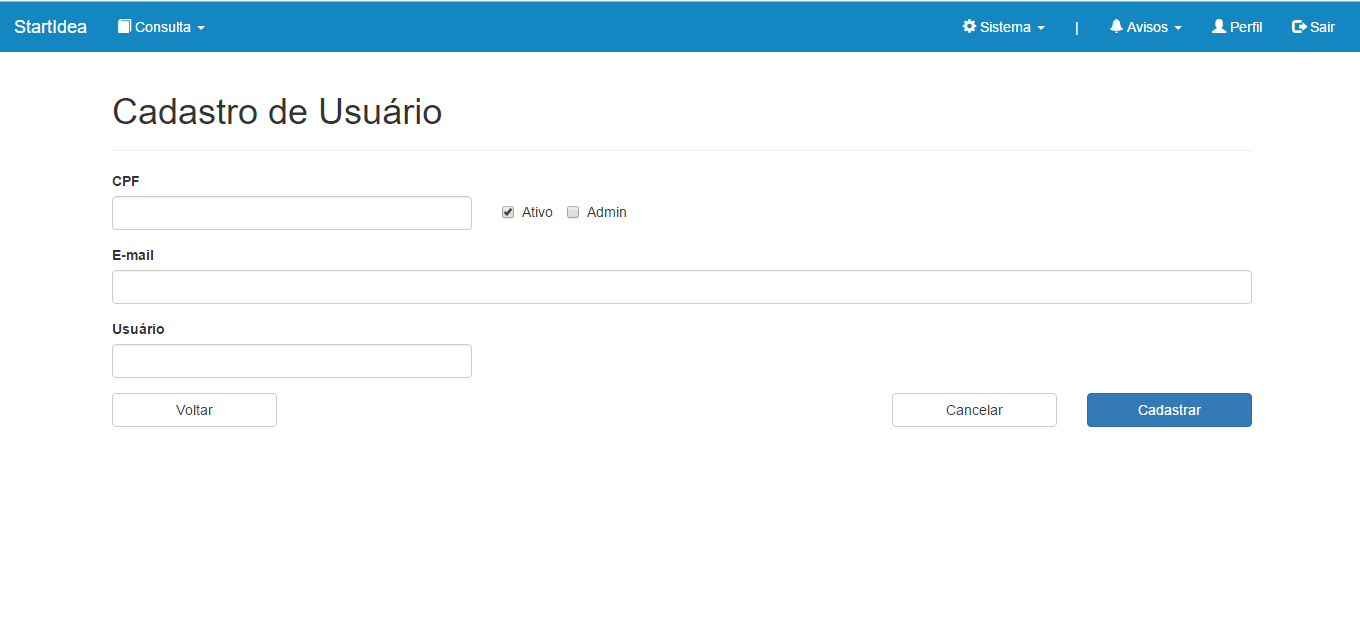
#### Diagrama de casos de uso



#### DESCRIÇÃO DE CASO DE USO: CADASTRO DE USUÁRIO

|  |  |
| --- | --- |
| **UC001 – CADASTRO DE USUÁRIO** | |
| **Descrição** | Permite ao administrador cadastrar um usuário do sistema |
| **Fluxo Básico de Eventos** | Administrador informa os dados básicos de cadastro, o sistema faz as validações de obrigatoriedade e inclui o novo usuário. |
| **Fluxos Alternativos** |  |
| **Cenários Chave** | Administrador tenta cadastrar um usuário sem informar os dados obrigatórios. |
| **Condições Prévias** | O usuário deverá estar logado como administrador do sistema e ter em mãos o cpf do usuário a ser cadastrado |
| **Condições Posteriores** | Se o use case for realizado com sucesso, um e-mail será disparado ao usuário para que ele possa realizar a troca da senha padrão |

#### CADASTRO DE USUÁRIO

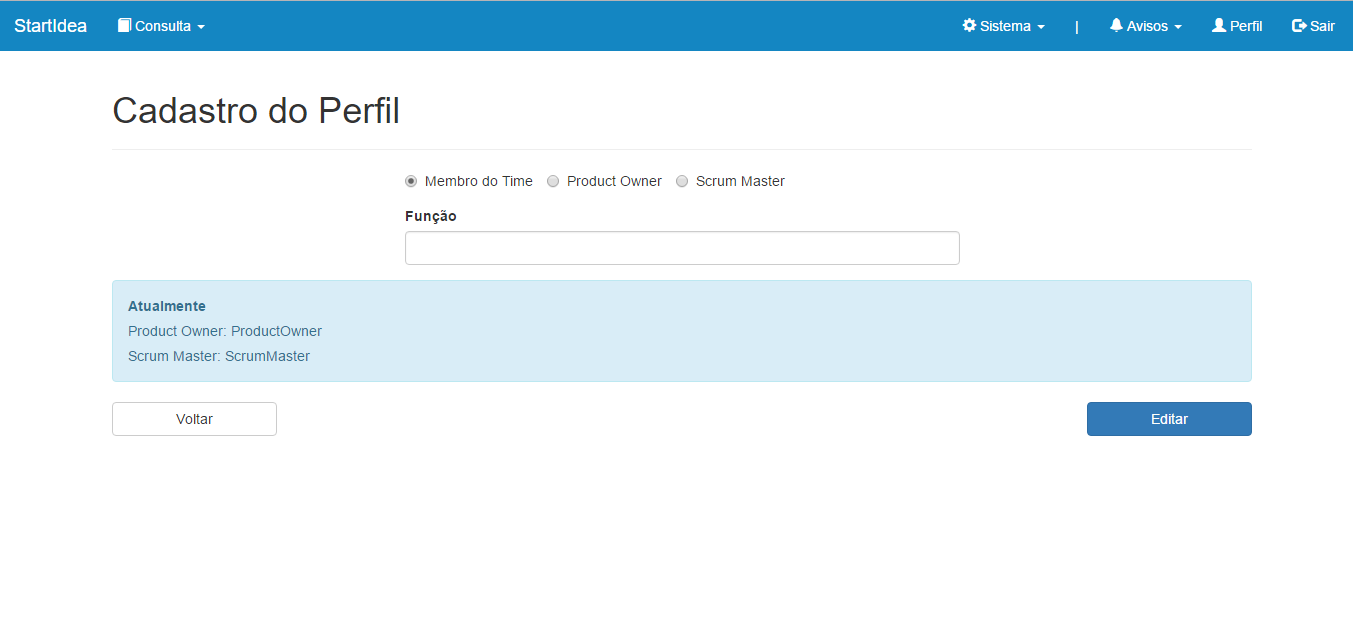


**Figura 22** - Tela de cadastro de usuário

#### DESCRIÇÃO DE CASO DE USO: ALTERAÇÃO DE FUNÇÃO DO USUÁRIO

|  |  |
| --- | --- |
| **UC002 – ALTERAÇÃO DE FUNÇÃO DO USUÁRIO** | |
| **Descrição** | Permite ao administrador definir a função do usuário. |
| **Fluxo Básico de Eventos** | Administrador seleciona o usuário, acessa a tela de perfil e seleciona a função do usuário. |
| **Fluxos Alternativos** | Caso já exista definido no sistema um product owner ou um scrum máster, ao definir outro usuário com essa função será substituído automaticamente. |
| **Cenários Chave** | Usuário tenta alterar o perfil para membro do time sem especificar sua função |
| **Condições Prévias** | O usuário deve estar logado com a conta de administrador. |
| **Condições Posteriores** | Se o use case for realizado com sucesso, será incluído no perfil do usuário sua atribuição no projeto. |

#### ALTERAÇÃO DE FUNÇÃO DE USUÁRIO

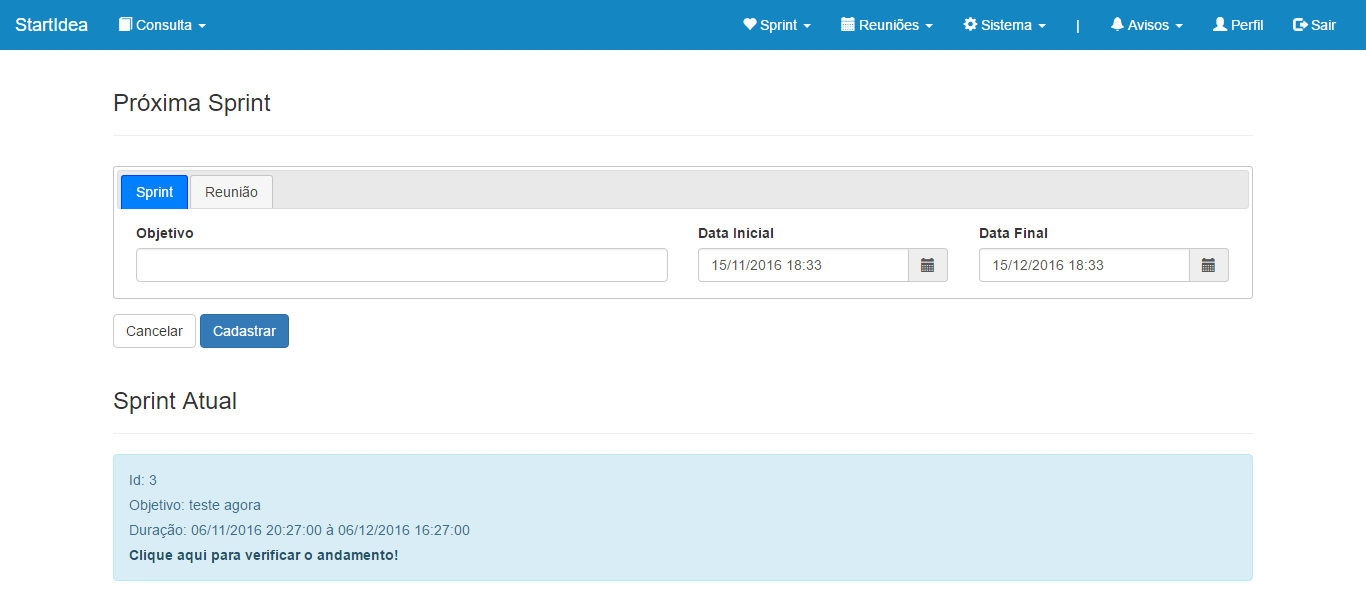


**Figura 23** - Tela de alteração de função de usuário

#### DESCRIÇÃO DE CASO DE USO: CADASTRO DA SPRINT

|  |  |
| --- | --- |
| **UC003 – CADASTRO DA SPRINT** | |
| **Descrição** | O usuário pode cadastrar as informações da Sprint atual e da próxima Sprint. |
| **Fluxo Básico de Eventos** | Usuário acessa a tela de sprint, informa os dados como, objetivo da Sprint, data inicial e data final e salva as informações |
| **Fluxos Alternativos** | Caso as datas das sprints estejam em conflito será apresentada uma mensagem de validação impedindo o cadastro da Sprint. |
| **Cenários Chave** | Usuário tenta realizar o cadastro de Sprint sem informar seu objetivo e data inicial e final. |
| **Condições Prévias** | O usuário deve estar logado com o perfil de scrum máster. |
| **Condições Posteriores** | Se o use case for realizado com sucesso será incluído no sistema uma nova Sprint. |

#### CADASTRO DA SPRINT

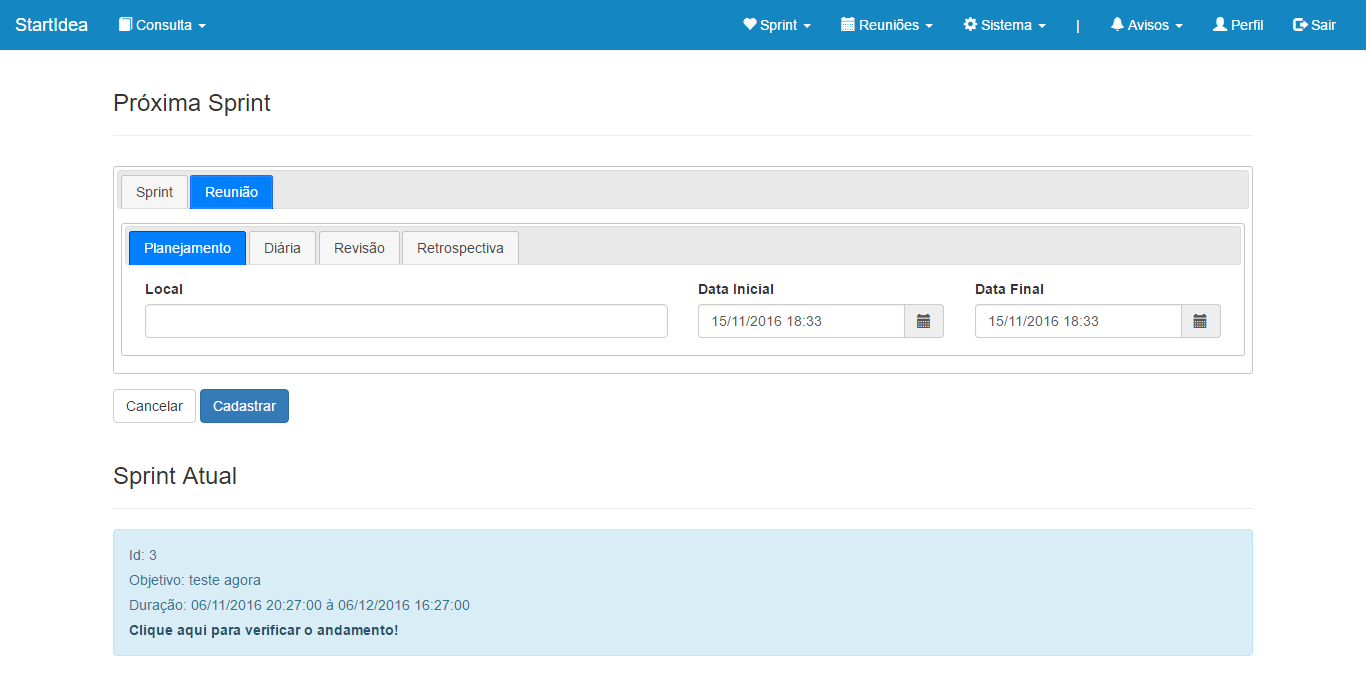


**Figura 24** - Tela de cadastro da sprint

#### DESCRIÇÃO DE CASO DE USO: CADASTRO DAS REUNIÕES

|  |  |
| --- | --- |
| **UC004 – CADASTRO DAS REUNIÕES** | |
| **Descrição** | O usuário pode cadastrar as informações das reuniões junto com o cadastro das sprints. |
| **Fluxo Básico de Eventos** | Usuário acessa a tela de sprint, informa os dados como, Local da reunião, data inicial e data final e salva as informações. |
| **Fluxos Alternativos** | Caso as datas das reuniões de planejamento, diária, revisão e retrospectiva estejam em conflito será apresentada uma mensagem de validação impedindo o cadastro da reunião/sprint. |
| **Cenários Chave** | Usuário tenta cadastrar as reuniões sem informar o local, data inicial e final. |
| **Condições Prévias** | O usuário deve estar logado com o perfil de scrum máster. |
| **Condições Posteriores** | Se o use case for realizado com sucesso será incluído no sistema uma nova Sprint e suas respectivas reuniões. |

#### CADASTRO DE REUNIÕES

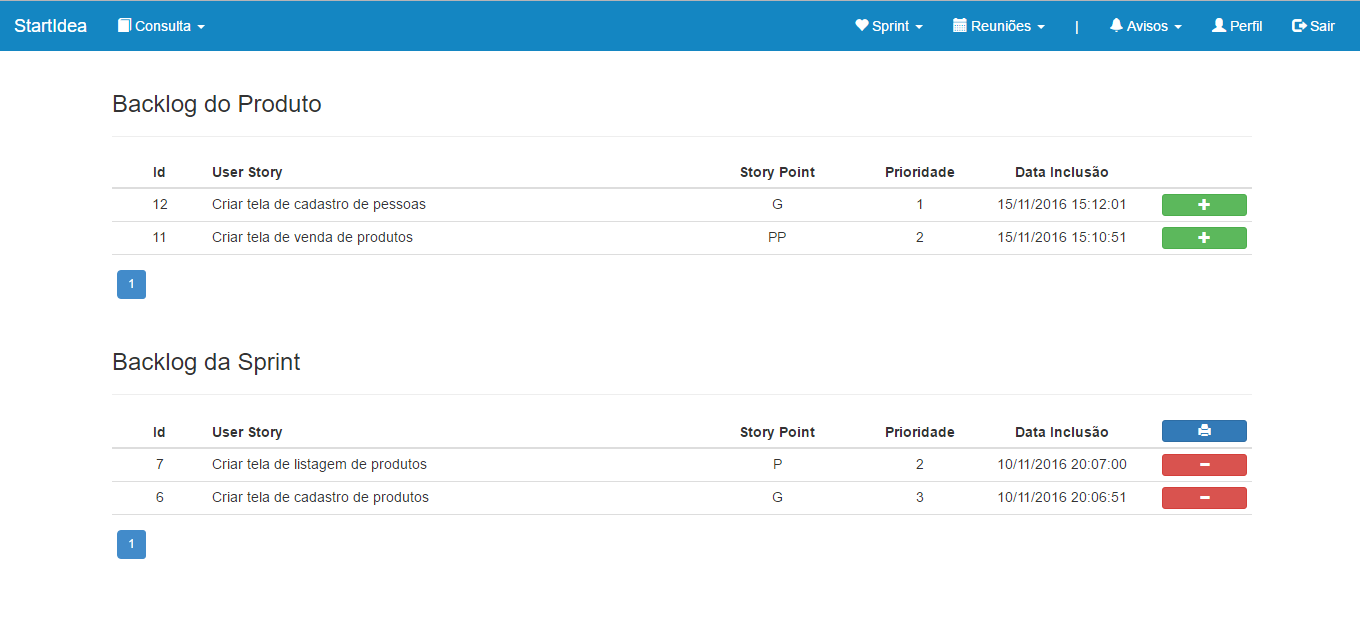


**Figura 25** - Tela de cadastro de reuniões

#### DESCRIÇÃO DE CASO DE USO: INCLUSÃO DO BACKLOG DO PRODUTO NA SPRINT

|  |  |
| --- | --- |
| **UC005 – INCLUSÃO DO BACKLOG DO PRODUTO NA SPRINT** | |
| **Descrição** | O scrum máster pode incluir na Sprint o backlog do produto para que seja desenvolvido |
| **Fluxo Básico de Eventos** | Usuário acessa a tela de backlogs da sprint, será apresenta dois grids com os backlogs incluídos e os backlogs que podem ser incluídos, ao clicar no botão “+” o backlog será incluído na Sprint. |
| **Fluxos Alternativos** | Ao clicar no botão “-” será aberta uma popup para informar o motivo do cancelamento do backlog na Sprint, assim que confirmado, o backlog será retirado da sprint |
| **Cenários Chave** |  |
| **Condições Prévias** | O usuário deve estar logado com o perfil de scrum máster, os backlogs do produto deverão estar avaliados pelo membro do time para poder ser incluso na sprint |
| **Condições Posteriores** | Se o use case for realizado com sucesso o backlog do produto será incluso na Sprint para ser criado tarefas para o mesmo para pôr fim ser desenvolvido. |

#### INCLUSÃO DO BACKLOG DO PRODUTO NA SPRINT

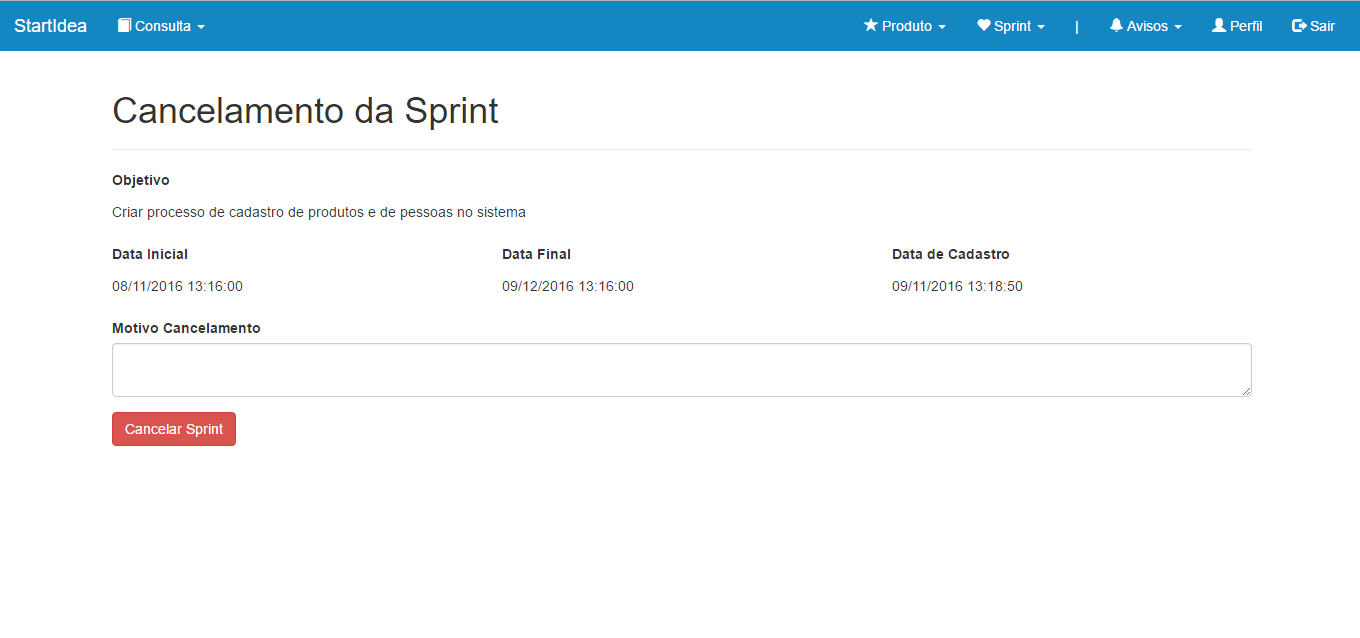


**Figura 26** - Tela de inclusão do backlog do produto na sprint

#### DESCRIÇÃO DE CASO DE USO: CANCELAMENTO DA SPRINT

|  |  |
| --- | --- |
| **UC006 – CANCELAMENTO DA SPRINT** | |
| **Descrição** | O product owner verificando a não necessidade da Sprint poderá cancelar a mesma. |
| **Fluxo Básico de Eventos** | Product owner acessa a tela da Sprint onde irá informar o motivo de cancelamento, ao confirmar a Sprint será cancelada. |
| **Fluxos Alternativos** | Caso não haja Sprint corrente será apresentada uma mensagem dizendo que não há Sprint em andamento. |
| **Cenários Chave** | Usuário tenta cancelar a Sprint sem informar o motivo. |
| **Condições Prévias** | O Usuário deverá estar logado com o perfil de product owner e deverá haver uma Sprint em andamento. |
| **Condições Posteriores** | Ao cancelar a Sprint será cancelado automaticamente os backlogs da Sprint e suas respectivas tarefas. |

#### CANCELAMENTO DA SPRINT



**Figura 27** - Tela de cancelamento da Sprint

#### DESCRIÇÃO DE CASO DE USO: CADASTRO DE BACKLOGS DO PRODUTO

|  |  |
| --- | --- |
| **UC007 – CADASTRO DE BACKLOGS DO PRODUTO** | |
| **Descrição** | O Product owner irá cadastrar um backlog para ser implementado no sistema. |
| **Fluxo Básico de Eventos** | Product owner acessa a tela da backlogs do produto onde irá informar a descrição da função a ser implementada no projeto e sua prioridade. |
| **Fluxos Alternativos** | Caso não seja informada uma prioridade já existente, a ordem das prioridades será refeita. |
| **Cenários Chave** | Usuário tenta cadastrar um backlog sem informar a user story e a prioridade. |
| **Condições Prévias** | O Usuário deverá estar logado com o perfil de product owner. |
| **Condições Posteriores** | Ao cadastrar um backlog do produto será automaticamente enviado aos membros do time para avaliação. |

#### CADASTRO DE BACKLOGS DO PRODUTO

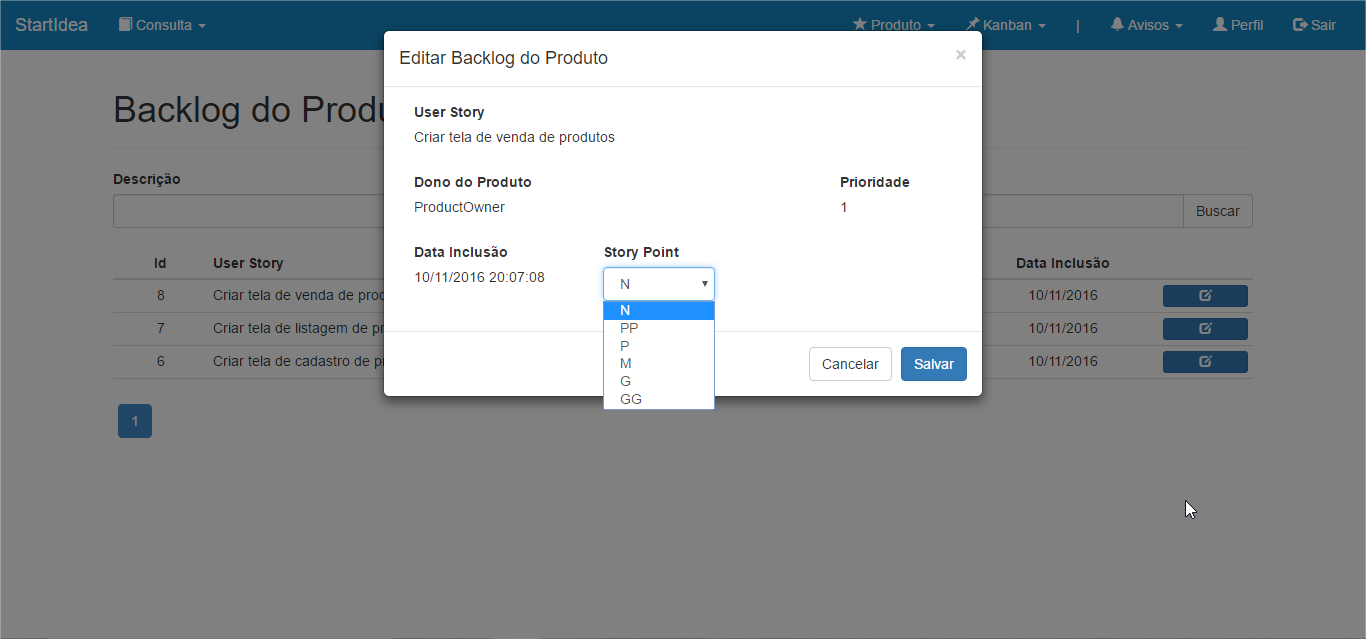


**Figura 28** - Tela de cadastro de backlogs do produto

#### DESCRIÇÃO DE CASO DE USO: AVALIAÇÃO DO BACKLOG DO PRODUTO

|  |  |
| --- | --- |
| **UC008 – AVALIAÇÃO DO BACKLOG DO PRODUTO** | |
| **Descrição** | O membro do time irá avaliar o nível de dificuldade do backlog cadastrado. |
| **Fluxo Básico de Eventos** | O membro do time acessa a tela de avaliação onde irá ler o backlog e avaliar sua dificuldade. |
| **Fluxos Alternativos** |  |
| **Cenários Chave** |  |
| **Condições Prévias** | O Usuário deverá estar logado com o perfil de membro do time e o backlog do produto não pode estar incluso em nenhuma sprint. |
| **Condições Posteriores** | Ao avaliar o backlog do produto o mesmo ficará disponível para ser incluso na Sprint. |

#### AVALIAÇÃO DO BACKLOG DO PRODUTO

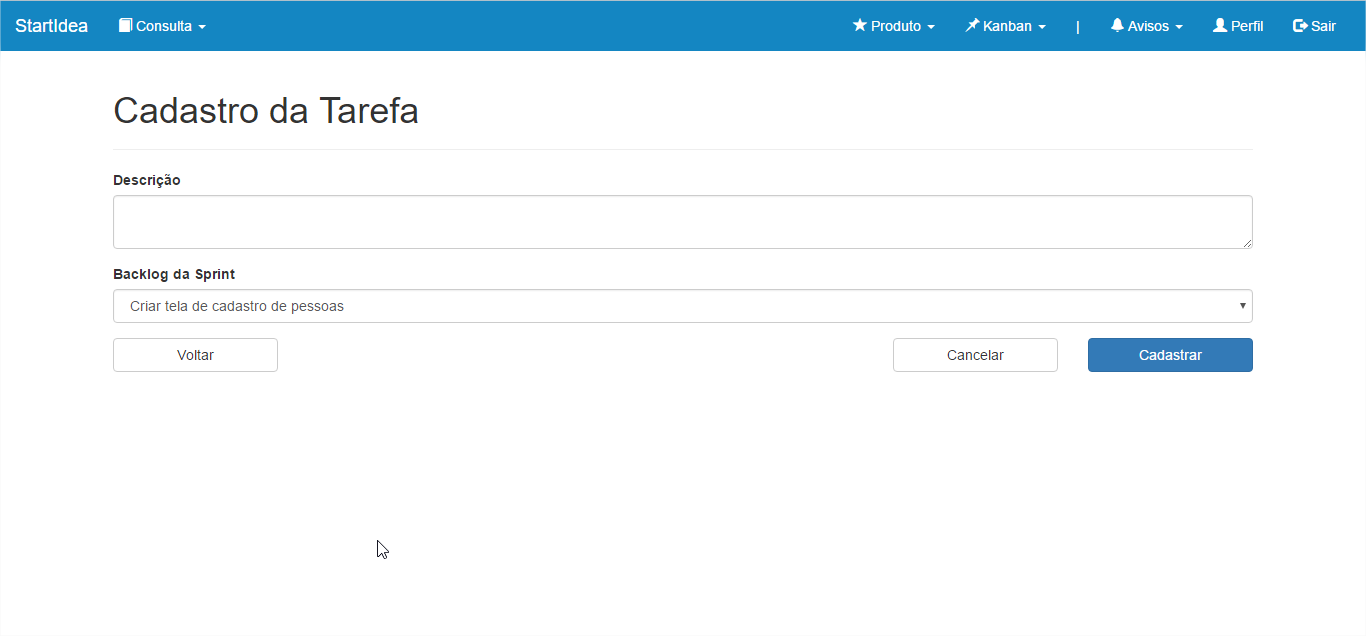


**Figura 29** - Tela de avaliação do backlog do produto

#### DESCRIÇÃO DE CASO DE USO: CADASTRO DE TAREFA

|  |  |
| --- | --- |
| **UC009 – CADASTRO DE TAREFA** | |
| **Descrição** | O membro do time irá cadastrar as tarefas referentes a um backlog do produto incluído na Sprint. |
| **Fluxo Básico de Eventos** | O membro do time acessa a tela de tarefa, onde irá incluir a descrição (analise) da tarefa e selecionar a qual backlog do produto aquela tarefa se refere. |
| **Fluxos Alternativos** | O usuário poderá também editar e excluir tarefas que ainda não estão prontas nem em desenvolvimento. |
| **Cenários Chave** | Usuário tenta cadastrar a tarefa sem informar a análise da mesma. |
| **Condições Prévias** | Para cadastrar uma tarefa é necessário que haja uma Sprint corrente e nesta Sprint haja backlogs do produto incluídos. |
| **Condições Posteriores** | Ao cadastrar uma tarefa, esta ficará disponível para os membros do time iniciarem o desenvolvimento. |

#### CADASTRO DE TAREFAS

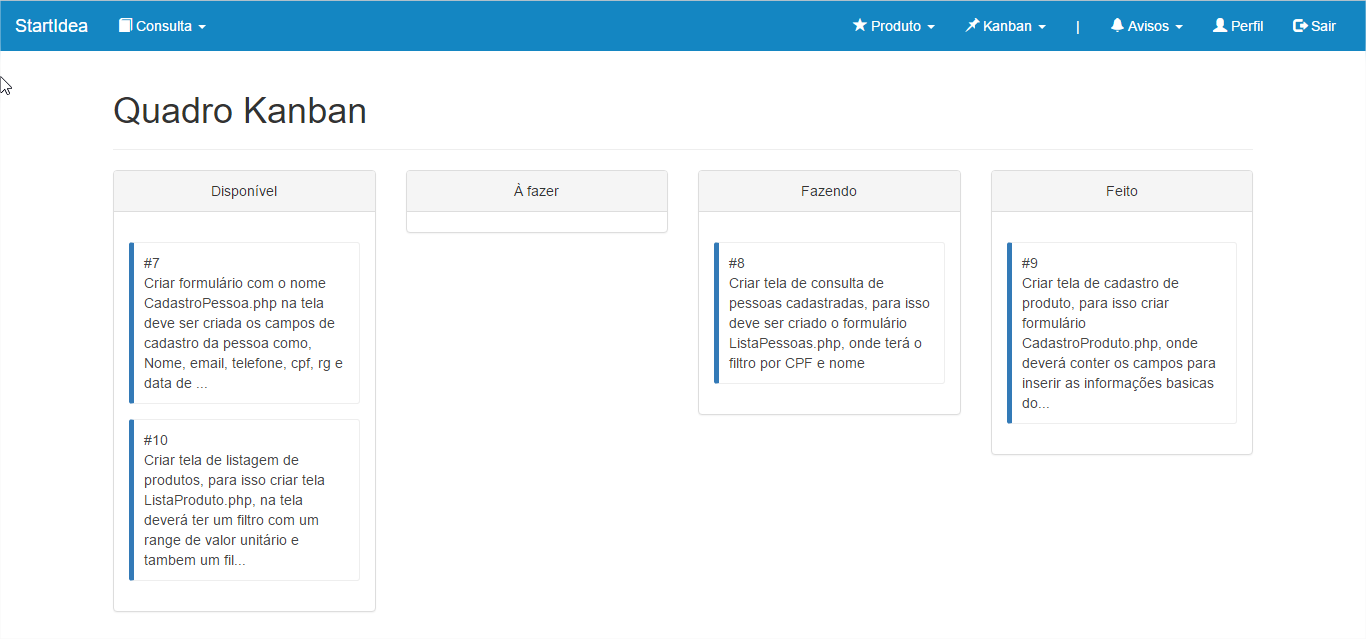


**Figura 30** - Tela de cadastro de tarefa

#### DESCRIÇÃO DE CASO DE USO: ALTERAÇÃO STATUS TAREFA

|  |  |
| --- | --- |
| **UC010 – ALTERAÇÃO STATUS TAREFA** | |
| **Descrição** | O membro do time poderá alterar o status do desenvolvimento da tarefa para os seguintes status, À fazer, Fazendo e Pronto. |
| **Fluxo Básico de Eventos** | O membro do time acessa a tela do quadro kanban, onde por meio de arrasta e solta poderá “pegar” a tarefa para desenvolvimento. |
| **Fluxos Alternativos** | O status da tarefa poderá ser alterado para qualquer status em qualquer momento, as tarefas que não estiverem com o status disponível, só poderão ser manipuladas pelo desenvolvedor que “pegou” a tarefa para desenvolvimento. |
| **Cenários Chave** |  |
| **Condições Prévias** | A tarefa não pode estar cancelada e o usuário logado deverá ter o perfil de membro do time. |
| **Condições Posteriores** | Ao manipular o status de uma tarefa, a mesma será automaticamente alocada ao desenvolvedor que à manipulou. |

#### ALTERAÇÃO STATUS TAREFA



**Figura 31** - Tela de alteração de status da tarefa

## Requisitos do sistema operacional

A ferramenta desenvolvida foi destinada para utilização na web, desta forma os requisitos de sistema operacional devem ser compatíveis com plataforma web.

Quanto ao servidor, devem-se levar em consideração as seguintes condições:

* Sistema Operacional Windows 7 ou superior
* Servidor IIS 10 ou superior
* Banco de dados SQL Server 2012
* Servidor web compatível com os elementos acima, sendo que o servidor usado durante o desenvolvimento foi o IIS 10.

Quanto ao cliente será necessário:

* Computador com acesso à rede;
* Navegador com o recurso de javascript ativo.
* Navegadores recomendados: Google Chrome, Mozilla Firefox

Os requisitos de hardware deverão ser aplicados de acordo com as especificações mínimas dos softwares acima citados. Esta configuração poderá ser aplicada de forma local ou através de uma hospedagem terceirizada.

### Tipo de hospedagem

A hospedagem poderá ser realizada em um servidor local ou terceirizado, tendo em vista as condições descritas no tópico anterior. Cabe ao administrador a responsabilidade da escolha, e devem ser levadas em considerações as questões de segurança, estabilidade e disponibilidade. Caso a escolha seja uma hospedagem terceirizada, orienta-se a contratação de um serviço gerenciável pela própria empresa contratada, tornando-a responsável por gerir as necessidades computacionais e de armazenamento. Caso a escolha seja um servidor local, é aconselhável a adoção de um padrão de gerenciamento que atenda as questões de necessidade computacional e armazenamento, levando sempre em consideração de demanda.

### Manutenção

A manutenção do sistema operacional, assim como dos demais softwares e hardwares, dependerá do tipo da hospedagem escolhida. Desta forma a hospedagem terceirizada é mais aconselhada, pois a maioria dos prestadores de serviço disponibilizam esta função inclusa no pacote, onde a mesma fica responsável pelas devidas atualizações em períodos predefinidos, a fim de manter o melhor funcionamento da ferramenta. Nos casos em que a hospedagem é em um servidor local, a responsabilidade fica a cargo do administrador do servidor, que deve manter e dar devida manutenção ao servidor, como licenciamento e devidas atualizações do software e hardware. É aconselhado que as atualizações de software sejam feitas semanalmente, não excedendo o limite de 4 horas.

### Suporte

Em casos de hospedagem terceirizada, é aconselhável que o suporte seja especificado no momento da contratação do serviço, sabendo como e onde será realizado, os meios de comunicação e de solicitação (telefone, e-mail, help-desk ou chat), bem como o período de atendimento e prazo para resposta as solicitações. É importante que toda manutenção seja previamente informada em tempo hábil sendo que este deve constar em contrato assim como tempo máximo para execução da manutenção. Para isso, é importante que se tenha claro todos os itens do Acordo de Nível de Serviço (Service Level Agreement - SLA) que define todas as questões de suporte, segurança, disponibilidade e de serviço. Em casos de servidor local, todas as questões relacionadas acima, devem ser discutidas e estabelecidas junto ao administrador do servidor.

### Segurança

Nas hospedagens terceirizadas, as políticas de segurança devem estar definidas em contrato, onde se estabelece as devidas obrigatoriedades de atualização de seus softwares de firewall, antivírus e outros, de modo a garantir o devido funcionamento de seu servidor e aplicação nele hospedada. O acesso deve ser restrito e identificado através de login e senha e criados logs de acesso e monitoramento que deverão estar disponíveis para possíveis consultas. Quanto à questão de segurança ambiental, a empresa deve prover o acesso restrito e monitorado em ambiente climatizado com controle de fornecimento de energia a fim de evitar paradas do fornecimento do serviço. Em servidores locais é aconselhado que se estabeleça o mesmo padrão de segurança orientado para uma empresa terceirizada.

### Infraestrutura

A infraestrutura de uma hospedagem terceirizada deve estar claramente descrita no contrato, com uma configuração mínima que é composta de dois servidores com as configurações mínimas, backup, um firewall e a conexão com internet compatível com a demanda.

## Métodos para controle de segurança do sistema

Segundo a norma ABNT NBR ISO/IEC 27002:2005:

[...] A segurança da informação é obtida a partir da implantação de um conjunto de controles adequados, incluindo políticas, processos, procedimentos, estruturas organizacionais e funções de software e hardware. Estes controles precisam ser estabelecidos, implementados, monitorados, analisados criticamente e melhorados, onde necessário, para garantir que os objetivos do negócio e de segurança da organização sejam atendidos. Convém que isso seja feito em conjunto com outros processos de gestão do negócio. [...]

### Controle de segurança lógica

De acordo com o site Dimep (2013), “[...] o controle de acesso lógico, por sua vez, permite que os sistemas de tecnologia da informação verifiquem a autenticidade e a identidade dos usuários que tentam adentrar em seus sistemas ou utilizar seus serviços”.

Um exemplo comum no controle lógico é a criação de senha para fazer o processo de identificação e autorização.

Segundo Morais (2011):

[...] em computação, uma senha faz parte do processo de autenticidade de um usuário para acessar determinada informação. Se analisarmos as propriedades de segurança da informação (Confidencialidade / Integridade / Disponibilidade), uma senha se enquadra no quesito confidencialidade. O processo de confidencialidade consiste em que a informação só pode ser acessada por usuários autorizados. [...]

No entanto hoje em dia existem muitas técnicas de roubo de senhas como engenharia social, ataque de força bruta, keyloggers, entre outros. Além dessas técnicas os usuários no ato da criação de suas senhas facilitam a descoberta das mesmas, pois criam com aspectos da vida pessoal. Exemplo: Nome da mãe, data de nascimento, número do telefone, nome da rua onde mora entre outras.

Para a elaboração de uma boa senha levamos em consideração as dicas do site BWNET (2000):

[...] Quanto mais “bagunçada” for a senha melhor, pois mais difícil será descobri-la. Assim, tente misturar letras maiúsculas, minúsculas, números e sinais de pontuação. Uma regra realmente prática e que gera boas senhas difíceis de serem descobertas é utilizar uma frase qualquer e pegar a primeira, segunda ou a última letra de cada palavra. Por exemplo, usando a frase “batatinha quando nasce se esparrama pelo chão” podemos gerar a senha “!BqnsepC” (o sinal de exclamação foi colocado no início para acrescentar um símbolo à senha). Senhas geradas desta maneira são fáceis de lembrar e são normalmente difíceis de serem descobertas. [...]

### Controle de segurança lógica

A política de senha adotada neste projeto, têm como regras as seguintes observações:

* Deve conter no mínimo uma letra maiúscula, um número e um caractere especial.
* Ao trocar a senha deve ser inserido uma senha diferente da anterior.

Com isso temos uma maior confiabilidade com nossas senhas.

|  |  |
| --- | --- |
| Níveis de acesso dos usuários utilizados no software desta monografia. | |
| Administrador | Terá acesso a tela de cadastro de usuários podendo cadastrar, editar, ativar e desativar os perfis cadastrados, podendo também acessar as telas compartilhadas entre todos os usuários. |
| Scrum Master | Terá acesso as telas de cadastro de sprints, sprint backlog, Reunião de planejamento, Reunião retrospectiva, Reunião diária e reunião de revisão, podendo também acessar as telas compartilhadas entre todos os usuários. |
| Product Owner | Terá acesso as telas de cadastro de Backlog do produto e a tela de cancelamento da Sprint, podendo também acessar as telas compartilhadas entre todos os usuários. |
| Team Member | Terá acesso a tela de avaliação do backlog do produto, assim como a tela de cadastro de tarefas e alocação de tarefas. |

**Quadro 2** - Controle de segurança

### Plano de contingência

NRFACIL (2009) salienta:

[...] Plano de contingência nada mais é, conforme seu próprio nome sugere, um documento que descreve, passo a passo, quais ações a empresa (ou, mais especificamente, a área de TI) deve tomar a fim de retomar normalmente seus processos de trabalho, após a ocorrência de um incidente de segurança (ou uma contingência). Ou seja, o plano de contingência detalha aonde, quando, como e quais ações corretivas devem ser efetuadas a fim de que as informações perdidas ou indisponibilizadas em um incidente sejam rapidamente recuperadas. [...]

Em TI, geralmente um plano de contingência é composto por um plano de backup e pelos planos de emergência e recuperação. Ele deverá ser acionado sempre que o plano de emergência não for capaz de evitar o desastre e ficará vigente até que a capacidade de processamento da empresa seja recuperada totalmente.

De acordo com Innarelli (2003):

[...] O Backup nada mais é que uma cópia de segurança dos dados armazenados no sistema de informação da instituição ou dos dados de uso pessoal. Este backup é fundamental para qualquer sistema de informação ou pessoa, pois ele é a garantia da restauração dos dados caso haja uma pane nos equipamentos da instituição ou pessoa. [...]

Para garantir a proteção dos dados do sistema criado neste projeto é fundamental a definição de uma política de backup a qual visa garantir de forma integra e confiável a restauração de todos os dados registrados no sistema. Algumas recomendações para a política de backup são:

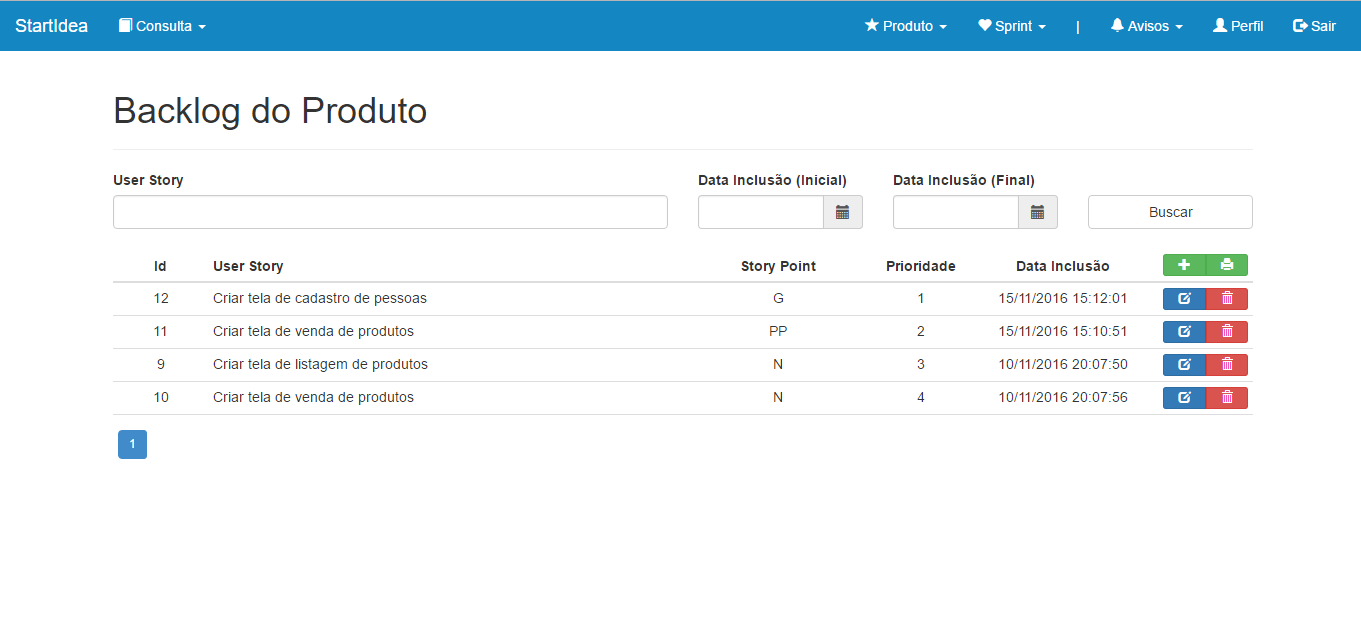
* Não permitir acesso de pessoas não autorizada ao sistema
* Fazer backup diariamente do sistema
* Guardar os backups em mídias virtuais e físicas de fácil acesso, porem apenas para pessoal autorizado

Sem esses passos a instituição fica falha a problemas com recuperação dos dados, podendo não os recuperar em caso de emergência ou desastres naturais por exemplo. O backup garante a integridade dos dados do sistema, sendo assim uma segurança para a mesma caso tenha problemas.

## Layout dos relatórios

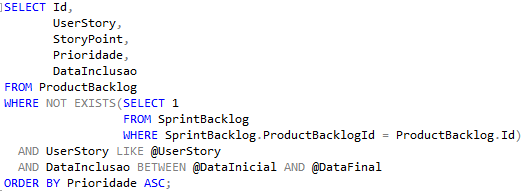
### Backlogs do produto

O relatório tem a função de listar todos os backlogs do produto cadastrados pelo dono do produto, podendo incluir, editar e excluir.



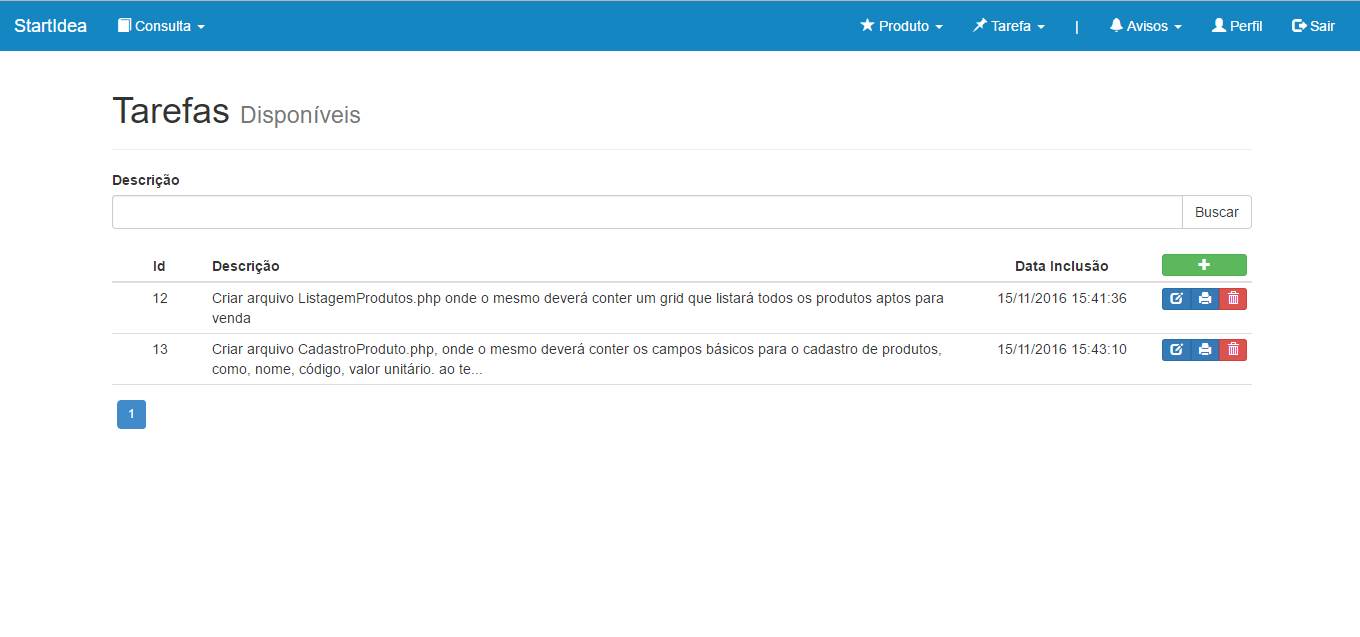
**Figura 32** - Tela de Backlog do Produto

**Instrução SQL**



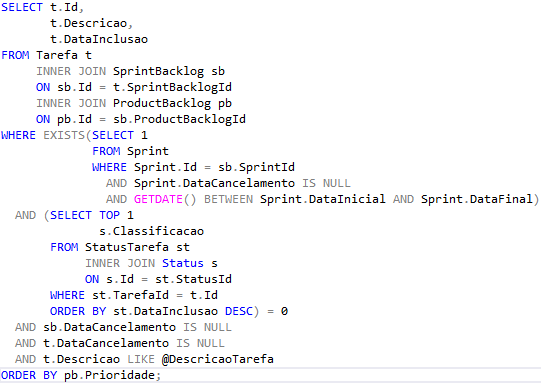
### Tarefas

O relatório tem a função de listar todas as tarefas cadastradas, podendo editar e excluir a mesma.



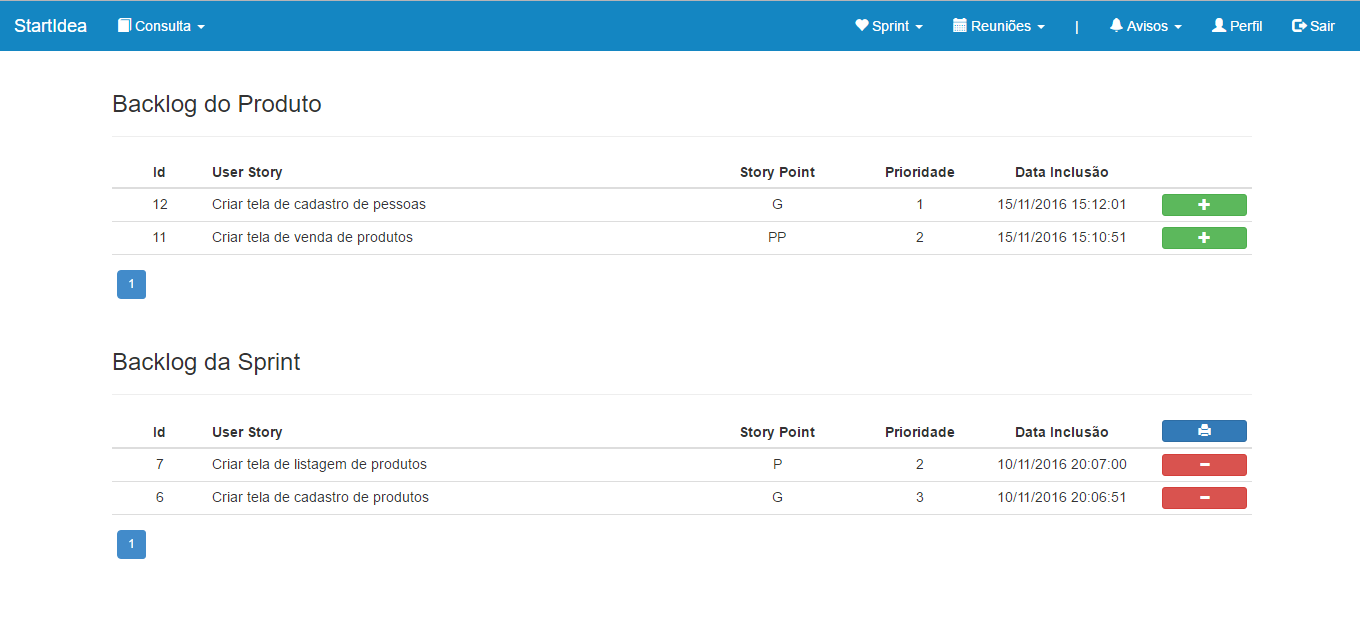
**Figura 33** - Tela de Tarefas

**Instrução SQL**



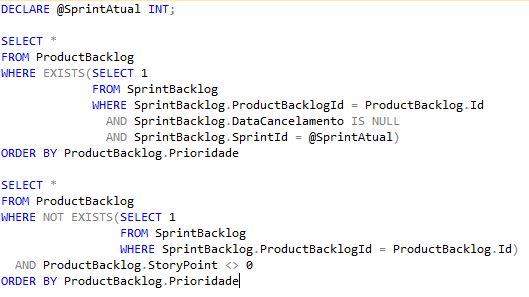
### Backlogs da Sprint

O relatório tem a função de listar todos os backlogs do produto que podem ser adicionados na Sprint atual e todos os que já estão adicionados na Sprint.



**Figura 34** - Tela de sprint backlog

**Instrução SQL**



# IMPLEMENTAÇÕES FUTURAS

O presente trabalho apresenta um resgate e estudo teórico sobre o gerenciamento de projeto em uma startup e a implementação de um sistema para auxilio no gerenciamento de projeto utilizando a metodologia scrum. O tema abordado possibilita futuros trabalhos complementares, algumas ideias propostas são o cadastro de projetos, o que possibilitaria o gerenciamento em empresas maiores que uma startup, outra implementação seria o cadastro de times, o que complementaria o cadastro de projetos.

Para maior flexibilidade do sistema a criação do o cadastro de status de tarefa possibilitaria o sistema ser implementado em diferentes empresas que possuem processos de desenvolvimento diferentes, fazendo com que o sistema abranja um maior número processos.

# CONCLUSÃO

# REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA, M. E. B. Escola em mudança: experiências em construção e redes colaborativas de aprendizagem. In ALONSO, M.; ALMEIDA, M. E. B.; MASETTO, M. T.; MORAN, J. M.; VIEIRA, A. **Formação de gestores escolares para utilização de tecnologias de informação e comunicação**. Brasília: Secretaria de Educação a Distância, 2002. pp. 41-62.

ANDREUZZA, Mario. **A história do gerenciamento de projetos**. Disponível em: <<http://www.sagres.org.br/artigos/historiagerenciamento.pdf>>. Acesso em 31 mai. 2016.

ANJOS DO BRASIL. **O que é um investidor anjo**. Disponível em: <http://www.anjosdobrasil.net/o-que-eacute-um-investidor-anjo.html>. Acesso em: 31 mai. 2016.

AUDAY, Rafael. **O QUE FAZ UM SCRUM MASTER?** Disponível em: <https://www.profissionaisti.com.br/2014/05/o-que-faz-um-scrum-master/>. Acesso em: 15 jul. 2016.

BERNARDO, Kleber. **Manifesto ágil, como tudo começou.** Disponível em: <http://www.culturaagil.com.br/manifesto-agil-como-tudo-comecou/>. Acesso em: 26 jul. 2016.

BIZSTART. **Startup: 12 cases de sucesso no Brasil**. Disponível em: <http://bizstart.com.br/startup-12-cases-de-sucesso-no-brasil/>. Acesso em: 01 jun. 2016.

BOEHM, B. W; TURNER R. **Balancing Agility and Discipline**. Boston; Addison Wesley,

2002.

BONFIM, Márcio. **Introdução ao scrum.** Disponível em: <http://www.devmedia.com.br/introducao-ao-scrum/27887>. Acesso em: 08 ago. 2016.

BRANCATO, Flavia. **10 startups de sucesso no Brasil**. Disponível em: <http://www.businessreviewbrasil.com.br/top10/1737/10-startups-de-sucesso-no-Brasil>. Acesso em: 01 jun. 2016.

BROOTA. **5 modalidades de investimento para a sua startup**. Disponível em: <http://blog.broota.com.br/5-modalidades-de-investimento-para-a-sua-startup/>. Acesso em: 31 mai. 2016.

BWNET. **CARTILHA DE SEGURANÇA PARA INTERNET.** Disponível em: <http://www.bwnet.com.br/cartilhadeseguranca.php>. Acesso em: 06 Nov. 2016.

CANAL DO EMPREENDEDOR. **Startups tem no comando 55% de jovens entre 20 e 23 anos.** Disponível em: < http://canaldoempreendedor.com.br/startup/startups-tem-no-comando-55-de-jovens-entre-20-e-23-anos/>. Acesso em: 13 ago. 2016.

CARVALHO, Rafael. **Escalabilidade: o que é, para que serve, por que ter**. Disponível em: <http://bizstart.com.br/escalabilidade-o-que-e-para-que-serve-por-que-ter/>. Acesso em: 01 jun. 2016.

DENES, Madson. **O que é Just in Time?** Disponível em: <http://www.administradores.com.br/artigos/carreira/o-que-e-just-in-time/21936/>. Acesso em: 24 jun. 2016.

DE PAULA, André Luis Lima. **Uma breve história do gerenciamento de projetos**. Disponível em: <<https://webinsider.com.br/2009/04/21/uma-breve-historia-do-gerenciamento-de-projetos/#sthash.tU6BpNFL.dpuf>>. Acesso em: 31 mai. 2016.

DESENVOLVIMENTO ÁGIL. **DAILY SCRUM**. Disponível em: <http://www.desenvolvimentoagil.com.br/scrum/daily\_scrum>. Acesso em: 25 jul. 2016.

DESENVOLVIMENTO ÁGIL. **SCRUM TEAM**. Disponível em: <http://www.desenvolvimentoagil.com.br/scrum/scrum\_team>. Acesso em: 08 jul. 2016.

DESENVOLVIMENTO ÁGIL. **SPRINT PLANNING MEETING**. Disponível em: <http://www.desenvolvimentoagil.com.br/scrum/sprint\_planning\_meeting>. Acesso em: 16 jul. 2016.

DIMEP. **Controle de acesso físico ou lógico: qual é a indicação de cada um.** Disponível em: <http://www.dimep.com.br/blog/sistemas-de-acessos/controle-acesso-fisico-logico-indicacao/>. Acesso em: 06 Nov. 2016.

ECONOMIST. **Genchi genbutsu**. Disponível em: <http://www.economist.com/node/14299017>. Acesso em: 24 jun. 2016.

ENDEAVOR BRASIL. **Poka Yoke: como ter uma empresa à prova de erros**. Disponível em: <https://endeavor.org.br/poka-yoke/>. Acesso em: 24 jun. 2016.

RIES, Eric. **A startup enxuta**. Disponível em: <http://s-inova.ucdb.br/wp-content/uploads/biblioteca/a-startup-enxuta-eric-ries-livro-completo.pdf>. Acesso em: 06 jul. 2016.

FERRÃO, Sara. **EMPREENDEDORISMO E EMPRESAS STARTUP: UMA NOVA VISÃO ESTRATÉGICA COMO MOTOR DE EMPREGABILIDADE JOVEM**. Boletim de Sociologia Militar: Centro de psicologia aplicada do exército, Lisboa, v. 4, p.96-26, 2013. Anual

FERRO, José Roberto. **Modelo de Gestão: Faça o 'Genchi Genbutsu'**. Disponível em: <http://www.administradores.com.br/noticias/negocios/modelo-de-gestao-faca-o-genchi-genbutsu/35326/>. Acesso em: 24 jun. 2016.

FILADORO, Adriano. **Por que o SCRUM vem sendo cada vez mais adotado nas empresas?** Disponível em: <http://corporate.canaltech.com.br/noticia/gestao/Por-que-o-SCRUM-vem-sendo-cada-vez-mais-adotado-nas-empresas/>. Acesso em: 15 ago. 2016.

FONSECA, Daniel. **Conceitos básicos sobre Metodologias Ágeis para Desenvolvimento de Software (Metodologias Clássicas x Extreme Programming)**. Disponível em: <<http://www.devmedia.com.br/conceitos-basicos-sobre-metodologias-ageis-para-desenvolvimento-de-software-metodologias-classicas-x-extreme-programming/10596>>. Acesso em: 31 mai. 2016.

GIMENES, Nicholas Fernandes. **Resumo Lean Startup - Eric Ries (Startup Enxuta).** Disponível em: <<http://www.nicholasgimenes.com.br/2012/09/resumo-lean-startup-eric-ries.html>>. Acesso em: 08 ago. 2016.

HENRIQUE, Fábio; FIORIO, Vivian. **O que é KAIZEN?** Disponível em: <http://www.industriahoje.com.br/kaizen>. Acesso em: 23 jun. 2016.

HENRIQUE, Fábio; FIORIO, Vivian. **O que é Kanban?** Disponível em: <http://www.industriahoje.com.br/o-que-e-kanban>. Acesso em: 27 jun. 2016.

HIGHSMITH, J., **Agile Project Management**, Creating innovative products, AddisonWesley, 2004.

IMAM. **POKA-YOKE (MÉTODOS A PROVA DE FALHAS)**. Disponível em: <http://www.imam.com.br/cursos/treinamento/aberto/?c=poka-yoke-metodos-a-prova-de-falhas&i=614>. Acesso em: 24 jun. 2016.

INNARELLI, Humberto Celeste. **Apostila Preservação de Documentos Digitais.** Disponível em: <http://cavalcanti.pro.br/textos/dspti/Politicas\_de\_Backup\_texto.pdf>. Acesso em: 06 Nov. 2016.

KAYO, Ramon. **O que é uma empresa startup?** Disponível em: <http://ramonkayo.com/conceitos-e-metodos/o-que-e-uma-empresa-startup>. Acesso em: 31 mai. 2016.

KFOURI, Eduardo. **Como não quebrar sua startup**. Disponível em: <http://convergecom.com.br/tiinside/webinside/03/03/2016/como-nao-quebrar-sua-startup/>. Acesso em: 01 jun. 2016.

LEAN INSTITUTE BRASIL. **Jidoka**. Disponível em: <http://www.lean.org.br/workshop/31/jidoka.aspx>. Acesso em: 23 jun. 2016.

LUZ CONSULTORIA. **Panorama das startups brasileiras**. Disponível em: <http://www.ufal.edu.br/empreendedorismo/downloads/manuais-guias-cartilhas-e-documentos-sobre-empreendedorismo-e-inovacao/panorama-de-startups-brasileiras>. Acesso em: 01 jun. 2016.

MANIFESTO ÁGIL. **Manifesto para o desenvolvimento ágil de software**. Disponível em: < http://www.manifestoagil.com.br/>. Acesso em: 03 ago. 2016.

MANIFESTO ÁGIL. **Princípios por trás do manifesto ágil**. Disponível em: <http://www.manifestoagil.com.br/>. Acesso em: 03 ago. 2016.

MASSIMUS. **O Manifesto ágil e seus 12 princípios**. Disponível em: <http://www.manifestoagil.com.br/principios.html>. Acesso em: 03 ago. 2016.

MASSIMUS. **Quais as diferenças entre Métodos Ágeis e Tradicionais na gestão de projetos?** Disponível em: <http://massimus.com/2016/04/metodos-ageis-e-tradicionais-na-gestao-de-projetos-qual-diferenca/>. Acesso em: 01 jun. 2016.

MICROSOFT. **Um histórico rápido do gerenciamento de projetos**. Disponível em: <<https://support.office.com/pt-br/article/Um-hist%C3%B3rico-r%C3%A1pido-do-gerenciamento-de-projetos-a2e0b717-094b-4d1e-878a-fcd0978891cd>>. Acesso em 31 mai. 2016.

MIRANDA, Fábio S.; CORRÊA, Chessman K.F. **Análise da Influência do uso de software para automatização de atividades gerenciais do SCRUM.** Disponível em: <http://pt.slideshare.net/fab1o/anlise-da-influncia-de-uso-de-software-para-automatizao-do-scrum>. Acesso em: 13 ago. 2016.

MITCH LACEY & ASSOCIATES. **Sprint Planning Meeting**. Disponível em: <https://www.mitchlacey.com/intro-to-agile/scrum/sprint-planning-meeting>. Acesso em 22 jul. 2016.

MORAIS, Danilo Augusto de Souza. **A Importância de uma senha segura.** Disponível em: <http://www.uenp.edu.br/2012-05-15-14-01-11/246-administrativo-e-tecnico/nucleo-tecnologia-da-informacao2/seguranca-info/426-a-importancia-de-uma-senha-segura>. Acesso em: 06 Nov. 2016.

MOUNTAIN GOAT SOFTWARE. **Sprint Backlog**. Disponível em: <https://www.mountaingoatsoftware.com/agile/scrum/sprint-backlog>. Acesso em 04 ago. 2016.

MOUNTAIN GOAT SOFTWARE. **ScrumMaster**. Disponível em: <https://www.mountaingoatsoftware.com/agile/scrum/scrummaster>. Acesso em 15 jul. 2016.

MOUNTAIN GOAT SOFTWARE. **Scrum Product Backlog**. Disponível em: <https://www.mountaingoatsoftware.com/agile/scrum/product-backlog>. Acesso em 01 ago. 2016.

NERUR S; MAHAPATRA, R; MANGALARA, G. **Challenges of Migrating to Agile**

**Methodologies**. Communications of the ACM, v.48, n.5, Maio/2005.

NOVY, Valquiria. **O que é inovação?** Disponível em: <https://clariceveras.wordpress.com/2009/03/11/o-que-e-inovacao/>. Acesso em: 01 jun. 2016.

NRFACIL. **PLANOS DE CONTINGENCIA: UM NOVO DESAFIO PARA O SESMT.** Disponível em: < http://nrfacil.com.br/blog/?p=1236#>. Acesso em: 06 Nov. 2016.

PMI. **O que é gerenciamento de projetos?** Disponível em: <<https://brasil.pmi.org/brazil/AboutUs/WhatIsProjectManagement.aspx>>. Acesso em: 31 mai. 2016.

PORTAL ADMINISTRAÇÃO. **Sistema Toyota de Produção**. Disponível em: <http://www.portal-administracao.com/2013/12/sistema-toyota-de-producao.html>. Acesso em: 22 jun. 2016.

RADIGAN, Dan. **A brief introduction to kanban**. Disponível em: <https://www.atlassian.com/agile/kanban>. Acesso em: 27 jun. 2016.

RADIGAN, Dan. **The product backlog: your ultimate to-do list**. Disponível em: <https://www.atlassian.com/agile/backlogs>. Acesso em: 08 ago. 2016.

RAMOS, Rogério. **Definições de Produção Enxuta**. Disponível em: <http://www.infoescola.com/administracao\_/definicoes-de-producao-enxuta/>. Acesso em: 01 jun. 2016.

REVISTA PEQUENAS EMPRESAS E GRANDES NEGÓCIOS. **4 dicas para montar uma startup de sucesso**. Disponível em: <http://revistapegn.globo.com/Como-comecar/noticia/2014/01/4-dicas-para-montar-uma-startup-de-sucesso.html>. Acesso em: 01 jun. 2016.

RIBEIRO, André Luiz Dias, ARAKAKI, Reginaldo. **GERENCIAMENTO DE PROJETOS TRADICIONAL X GERENCIAMENTO DE PROJETOS ÁGIL: UMA ANÁLISE COMPARATIVA**. São Paulo: 3º Congresso Internacional de Gestão da Tecnologia e Sistemas de Informação, 2006.

RUGGIERI, Ruggero. **Compreendendo os princípios do Manifesto Ágil.** Disponível em: <http://www.tiespecialistas.com.br/2014/08/compreendendo-os-principios-manifesto-agil/>. Acesso em: 26 jul. 2016.

SAAD, Flávia. **O que é takt time**. Disponível em: <http://www.manutencaoesuprimentos.com.br/conteudo/5777-o-que-e-takt-time/>. Acesso em: 27 jun. 2016.

SCHWABER, Ken; Sutherland, Jeff. **Guia do Scrum**. Disponível em: <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf>. Acesso em: 06 Jul. 2016.

SCHWABER, Ken. **Agile Project Management with Scrum**. Redmond: Microsoft Press, 2004.

SCRUM ALLIANCE. **Learn About Scrum**. Disponível em: <https://www.scrumalliance.org/why-scrum>. Acesso em: 06 jul. 2016.

SCRUM ALLIANCE. **Scrum Values**. Disponível em: <https://www.scrumalliance.org/why-scrum/core-scrum-values-roles>. Acesso em: 06 jul. 2016.

SCRUMINC. **Scrum Framework**. Disponível em: <https://www.scruminc.com/scrum-framework/>. Acesso em: 13 jul. 2016.

SCRUMINC. **The Sprint**. Disponível em: <https://www.scruminc.com/sprint/>. Acesso em: 16 jul. 2016.

SCRUM INSTITUTE. **Sprint Retrospective Meeting**. Disponível em: <http://www.scrum-institute.org/Sprint\_Retrospective\_Meeting.php>. Acesso em: 26 jul. 2016.

SCRUM INSTITUTE. **Sprint Review Meeting**. Disponível em: <http://www.scrum-institute.org/Sprint\_Review\_Meeting.php>. Acesso em: 25 jul. 2016.

SCRUM INSTITUTE. **The Scrum Product Backlog**. Disponível em: <http://www.scrum-institute.org/The\_Scrum\_Product\_Backlog.php>. Acesso em: 01 ago. 2016.

SEBRAE. **Mercado, necessidades e disposição para pagar pela solução são alguns dos questionamentos que devem ser feitos antes de obter um financiamento**. Disponível em:

<http://www.sebrae.com.br/sites/PortalSebrae/artigos/como-obter-financiamento-para-startup,201a5415e6433410VgnVCM1000003b74010aRCRD>. Acesso em: 31 mai. 2016.

SILVEIRA, Cristiano Bertulucci. **Andon**. Disponível em: <http://www.citisystems.com.br/andon/>. Acesso em: 24 jun. 2016.

SILVEIRA, Cristiano Bertulucci. **Heijunka – Flexibilizar e nivelar a produção**. Disponível em: <http://www.citisystems.com.br/heijunka/>. Acesso em: 23 jun. 2016.

SILVEIRA, Cristiano Bertulucci. **Jidoka: Automatização com um toque humano**. Disponível em: <http://www.citisystems.com.br/jidoka/>. Acesso em: 23 jun. 2016.

SILVEIRA, Cristiano Bertulucci. **Just in time: conceito, definição e objetivos**. Disponível em: <http://www.citisystems.com.br/just-in-time-conceito-significado/>. Acesso em: 24 jun. 2016.

SILVEIRA, Cristiano Bertulucci. **Kaizen**. Disponível em: <http://www.citisystems.com.br/kaizen-metodo-melhoria-continua/>. Acesso em: 23 jun. 2016.

SILVEIRA, Cristiano Bertulucci. **Sistema Kanban: conceito e softwares**. Disponível em: <http://www.citisystems.com.br/kanban-conceito-sistema-o-que-e-on-line/>. Acesso em: 27 jun. 2016.

SPINA, Cassio. **5 características indispensáveis de uma startup**. Disponível em: <<http://exame.abril.com.br/pme/noticias/5-caracteristicas-indispensaveis-de-uma-startup>>. Acesso em 31 mai. 2016.

SPINA, Cassio. **Afinal, o que é Investimento-Anjo?** Disponível em:

<https://endeavor.org.br/afinal-o-que-e-investimento-anjo/>. Acesso em: 01 jun. 2016.

SPINA, Cassio. **O processo de criação de startups**. Disponível em: <<http://info.abril.com.br/noticias/rede/invest/sem-categoria/o-processo-de-criacao-de-startups/>>. Acesso em 31 mai. 2016.

STEFFEN, Juliana Berossa. **O que são essas tais de metodologias Ágeis?** Disponível em:

<https://www.ibm.com/developerworks/community/blogs/rationalbrasil/entry/mas\_o\_que\_s\_c3\_a3o\_essas\_tais\_de\_metodologias\_\_c3\_a1geis?lang=en>. Acesso em 28 jul. 2016.

SUTHERLAND, Jeff. **Scrum - a arte de fazer o dobro de trabalho na metade do tempo**. São Paulo: LeYa, 2014.

TORREÃO, Paula. **História do Gerenciamento de Projetos**. Disponível em: <<https://pontogp.wordpress.com/2007/04/23/historia-do-gerenciamento-de-projetos/>>. Acesso em: 31 mai. 2016.

TOYOTA, **Jidoka**, Disponível em: <http://www.toyota-forklifts.com.pt/Pt/company/Toyota-Production-System/Jidoka/Pages/default.aspx>. Acesso em 23 jun. 2016.

TOYOTA, **Just-in-time**, Disponível em: <http://www.toyota-forklifts.com.pt/Pt/company/Toyota-Production-System/Just-in-time/Pages/default.aspx>. Acesso em 23 jun. 2016.

TOYOTA, **Kaizen**, Disponível em: <http://www.toyota-forklifts.com.pt/Pt/company/Toyota-Production-System/Kaizen/Pages/default.aspx>. Acesso em 23 jun. 2016.

VERA, Carlos. **SISTEMA TOYOTA DE PRODUÇÃO (TOYOTA WAY)**. Disponível em: <http://www2.ifma.edu.br/proen/arquivos/artigos.php/sistema\_toyota\_de\_producao.pdf>. Acesso em: 22 jun. 2016.

VIEIRA, Denisson. **Scrum: A Metodologia Ágil Explicada de forma Definitiva**. Disponível em: <http://www.mindmaster.com.br/scrum/>. Acesso em: 08 jul. 2016.