



MJ / DEPARTAMENTO DE POLÍCIA FEDERAL  
CGTI – COORDENAÇÃO-GERAL DE TECNOLOGIA DA INFORMAÇÃO  
DIVISÃO DE INFORMÁTICA

---

# **MDS METODOLOGIA DE DESENVOLVIMENTO DE SISTEMAS**

**Versão 2.0**



**MDS**  
**METODOLOGIA DE DESENVOLVIMENTO**  
**DE SISTEMAS**

**Página:** 2 / 21

**Data Emissão:**  
02/06/2015

**Versão :** 2.0

**HISTÓRICO DAS REVISÕES**

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>
17/08/2009	1.0	Elaboração da Minuta	Altamir Araujo Guimarães Junior
21/09/2009	1.0	Revisão da Minuta	Antonio Carlos Garcia Mendes
12/11/2009	1.0	Atualização para o novo layout	Maria Rosa S. Matozo
18/11/2009	1.0	Aprovação para publicação	José Luiz Povill de Souza
15/12/2009	1.1	Subdivisão da Fase de Elaboração	Altamir Araujo Guimarães Junior
03/04/2010	1.2	Alterar o logotipo da PF e atualizar o item 4. Artefatos, com a inclusão dos artefatos: Mensagens do Sistema, Regras de Negócio e Termo de Homologação.	Maria Rosa S. Matozo
17/05/2011	1.3	Inclusão no item 4 – Artefatos os documentos: “TP – Sigla Projeto – IC – Nome do Caso de Uso” e “TP – Sigla Projeto – Integração_Implementação”	Maria Rosa S. Matozo
25/10/2011	1.3	Alteração no número da versão que constou errada na capa e cabeçalho e o ano da alteração do histórico de: 17/05/2010 para 17/05/2011. Para este tipo de alteração não há necessidade de mudança de versão.	Maria Rosa S. Matozo
06/01/2012	1.4	Alteração do nome da CTI para “CGTI – Coordenação-Geral de Tecnologia da Informação	Maria Rosa S. Matozo
08/06/2012	1.5	Alterações realizadas: <ul style="list-style-type: none"><li>• Item 2.3 - Inclusão de uma Nota sobre artefatos já homologados recebidos na CGTI.</li><li>• Item 4 - Retirada dos artefatos relativos ao Gerenciamento de Projetos;</li><li>• Item 6 - retirada do conteúdo relativo ao Gerenciamento de Projetos;</li></ul>	Maria Rosa S. Matozo
30/11/2012	1.6	Alteração do tipo e tamanho da fonte, margens, rodapé, para obedecer ao novo padrão de documentação da CGTI. Inclusão de 3 novos templates na “Lista de Artefatos”, item 4, referentes as Especificações Técnicas.	Maria Rosa S. Matozo



**MDS**  
**METODOLOGIA DE DESENVOLVIMENTO**  
**DE SISTEMAS**

**Página:** 3 / 21

**Data Emissão:**  
02/06/2015

**Versão :** 2.0

30/09/2013	1.7	Inclusão do template “GO - Arquitetura de Referência Componentes Corporativos” no item 4 – Artefatos – no bloco “Guias Operacionais”.	Maria Rosa S. Matozo
27/03/2014	1.8	Inclusão do artefato GO - Versionamento de Código, na fase de Construção no item 4 – Artefatos – no bloco “Guias Operacionais”	Maria Rosa S. Matozo
05/10/2014	2.0	Reestruturação completa da metodologia para adequá-la ao paradigma ágil baseado em SCRUM	Frederico Borelli de Souza



## MDS METODOLOGIA DE DESENVOLVIMENTO DE SISTEMAS

Página: 4 / 21

Data Emissão:  
02/06/2015

Versão : 2.0

### 1. INTRODUÇÃO

Este documento tem o propósito de descrever e normatizar o processo de desenvolvimento, manutenção e testes de sistemas de informação da Polícia Federal. A Metodologia de Desenvolvimento de Sistemas (MDS) aqui descrita envolve diferentes processos, dentre os quais o processo para desenvolvimento de novos sistemas, baseado no *SCRUM*, um modelo de desenvolvimento ágil amplamente utilizado e consagrado no mercado mundial, além de adaptações e outras técnicas complementares, incluindo os processos de sustentação de sistemas e de realização de manutenções evolutivas de pequeno porte.

Nas seções a seguir, serão detalhados, dentre outros aspectos, o processo principal de desenvolvimento baseado em *SCRUM*, denominado Processo de Desenvolvimento Ágil (PDA) e os processos auxiliares, denominados Processo de Sustentação de Sistemas (PSS) e Processo de Evolução de Pequeno Porte (PEP).

### 2. DEFEITOS DE SOFTWARE

Com efeito na aceitação de entregas, estão definidos nesta seção os níveis de criticidade dos defeitos de software, cujo conceito se estende a quaisquer dos processos em questão.

#### 2.1. Defeito Impeditivo.

De caráter crítico, que envolve situações tais como:

- a) A impossibilidade de conclusão de um fluxo principal;
- b) Inoperância de integrações externas essenciais ao sistema;
- c) Divergência de regra de negócio em relação à especificação, que impossibilite o uso do sistema;
- d) Performance ou tempo de resposta em nível que impossibilite o uso do sistema;
- e) Corrupção de múltiplos registros de dados em produção.

#### 2.2. Defeito não impeditivo.

De caráter regular, envolve situações tais como:

- a) Dificuldades para concluir fluxos não essenciais do sistema e validações inconsistentes;
- b) Integrações não críticas inoperantes;
- c) Ausência de parâmetros ou tabelas de domínio que dificultem o uso do sistema;
- d) Não-conformidades de layout e aspectos ineficientes de usabilidade;
- e) Inconsistências de massa de dados para homologação que dificultem testes do sistema;



## MDS METODOLOGIA DE DESENVOLVIMENTO DE SISTEMAS

Página: 5 / 21

Data Emissão:  
02/06/2015

Versão : 2.0

### 3. PROCESSO DE DESENVOLVIMENTO ÁGIL (PDA)

Nesta seção será descrito o processo de desenvolvimento ágil baseado em SCRUM, que é utilizado para o desenvolvimento de novos sistemas e também para as evoluções elencadas como projeto, que tipicamente são evoluções de maior porte e/ou de alta criticidade, e cuja categorização é a cargo da CGTI/PF.

O PDA aqui descrito, seguindo as diretrizes que são empregadas no processo SCRUM, tem como principais valores:

- Interação e confiança entre os participantes;
- Janela fixa de tempo para cada ciclo de desenvolvimento;
- Adaptação rápida às mudanças;
- Documentação concisa e objetiva;
- Entrega rápida de produtos e satisfação das áreas de negócios;
- Revisão e melhoria contínuas no processo.

Considerando o exposto, é possível observar que neste processo, os produtos são continuamente incrementados, agregando valor à área de negócio desde os primeiros ciclos de desenvolvimento. O foco na documentação é reduzido, mas se mantém um conjunto de artefatos plenamente satisfatório para exprimir o sistema em termos documentais. Por fim, todo o trabalho é continuamente avaliado e monitorado, de forma que melhorias são aplicadas constantemente nas experiências entre os participantes.

#### 3.1. Termos Utilizados no PDA

- **Backlog** – coleção de funcionalidades definidas pelo cliente e que geram valor para o negócio;
- **Sprint** – iteração no processo de desenvolvimento, na qual é produzida uma parte do sistema, previamente definida pelo cliente;
- **Kanban** – técnica utilizada em processos industriais que consiste no simples mapeamento das atividades, e as unidades de trabalho responsáveis por elas, sendo aqui aplicada ao desenvolvimento de software através do Quadro Kanban de atividades;
- **História de usuário** – é a menor unidade de funcionalidade que possui valor para o cliente, e que normalmente representa um cenário de uso do sistema.

#### 3.2. Papéis do PDA

##### 3.2.1. Product Owner (PO)

É o representante da PF responsável por:

- Conhecer as necessidades relacionadas ao sistema;
- Definir a visão do produto;
- Descrever, priorizar e refinar as necessidades continuamente;
- Estar disponível para dúvidas e questionamentos do time de desenvolvimento;



## MDS METODOLOGIA DE DESENVOLVIMENTO DE SISTEMAS

Página: 6 / 21

Data Emissão:  
02/06/2015

Versão : 2.0

- Participar das reuniões de demonstração de Sprints e decidir pela aceitação de entregas;

### 3.2.2. Scrum Master

É o representante da contratada responsável por:

- Priorizar e remover os impedimentos da equipe de desenvolvimento;
- Garantir o funcionamento do processo, ou seja, que a equipe utilize corretamente a MDS;
- Evitar que membros da equipe implementem hierarquias;
- Facilitar e garantir as reuniões;

### 3.2.3. Gerente de Sistema

É o representante da CGTI/PF responsável por:

- Reforçar os fundamentos do processo e garantir a correta execução das tarefas, atuando como **Scrum Master Interno**;
- Garantir apoio ao Product Owner na priorização e demais atividades relacionadas às necessidades do produto, inclusive com o aprofundamento no entendimento do negócio;
- Definição de datas relacionadas ao plano de releases;
- Acompanhamento e discussão das atividades com a equipe de desenvolvimento diariamente, inclusive com inspeção dos resultados diários;
- Participar das reuniões de demo e retrospectiva de Sprints;
- Decidir pela homologação técnica de entregas;
- Definir questões que envolvam caráter técnico;
- Apoiar na resolução de conflitos e dificuldades da equipe contratada.

## 3.3. Visão Geral do Processo

Neste processo foi definida uma fase de **Iniciação**, onde se busca representar a visão inicial do produto. Após sua conclusão, se inicia a execução cíclica das Sprints, onde cada Sprint é composta de 3 fases: **Discovery**, que contempla o refinamento das histórias de cada ciclo; **Delivery**, que contempla a construção e implantação do produto planejado; e a **Homologação**, onde o Product Owner verifica em detalhes o produto entregue. Esta visão macro do processo consta na Figura 1.

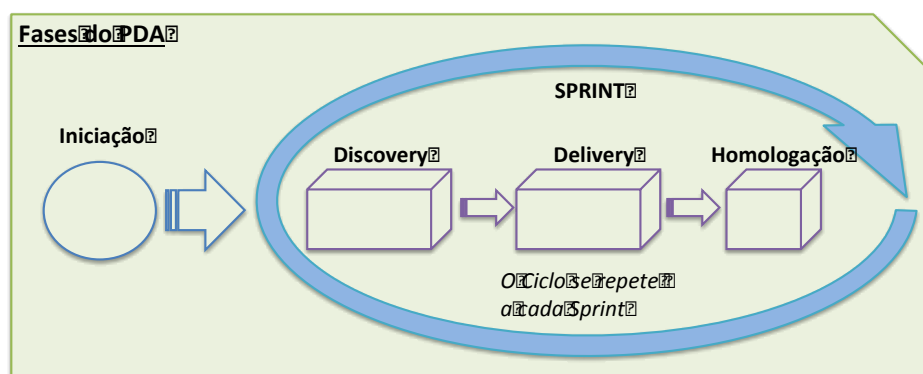


Figura 1 – Fases do PDA



## MDS METODOLOGIA DE DESENVOLVIMENTO DE SISTEMAS

Página: 7 / 21

Data Emissão:  
02/06/2015

Versão : 2.0

Aproximando o processo e decompondo as fases envolvidas, é possível observá-lo através de uma perspectiva mais detalhada (Figura 2). A iniciação, abrangendo a definição de visão do produto, compreende, em linhas gerais, os objetivos do sistema a ser desenvolvido, premissas arquiteturais, além do conjunto inicial previsto de histórias de usuário (backlog do produto), um plano de releases e demais necessidades de caráter não-funcional identificadas pela área demandante.

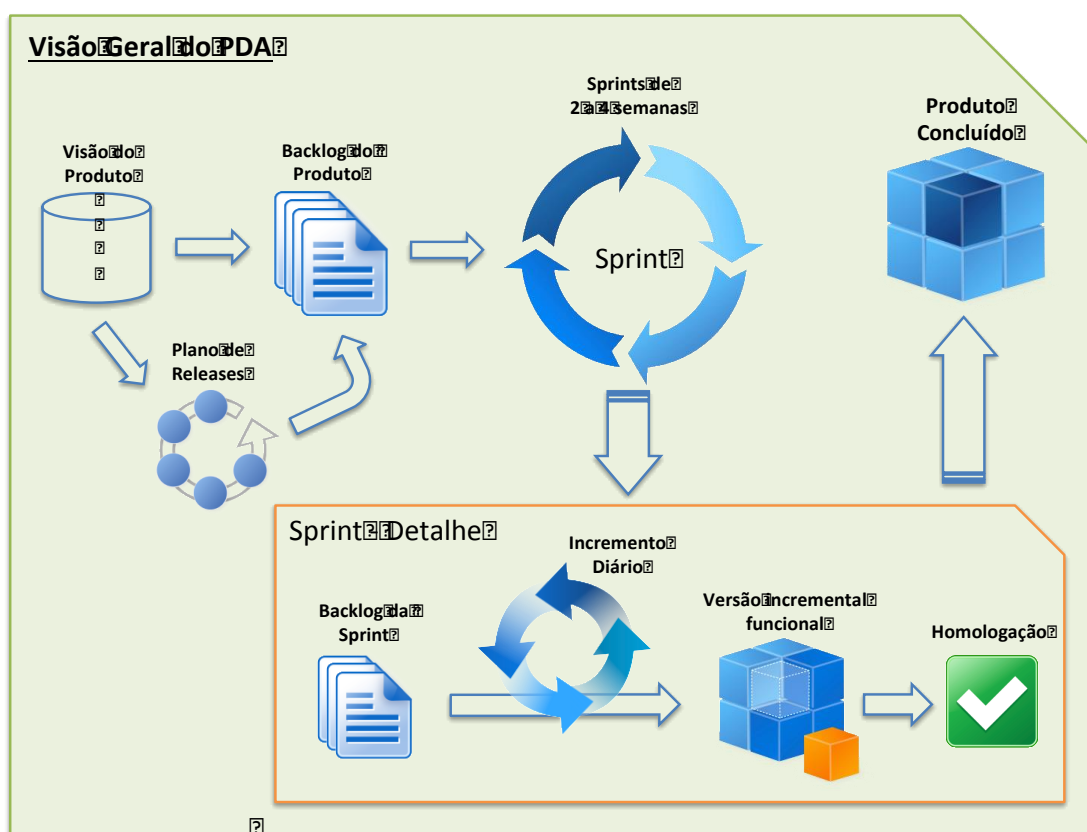


Figura 2 – Visão Geral Detalhada do PDA

Uma vez concluída a iniciação, o ciclo de Sprints começa a ser executado. Cada Sprint possui um conjunto de histórias selecionadas (Backlog da Sprint), que são refinadas (Discovery), e que através de incrementos diários, é concluída com um produto entregue contendo estas histórias planejadas (Delivery). Este fluxo permanece até que o produto seja completamente construído.

### 3.4. Fase de Iniciação

A fase de iniciação, **com duração fixa de uma semana**, é a primeira fase do projeto, onde se busca nivelar as necessidades e atingir um consenso entre todos os envolvidos sobre qual produto deverá ser desenvolvido. A Figura 3 ilustra as etapas e artefatos detalhados da fase de iniciação.

#### 3.4.1. Definição da Visão do Produto

A visão do produto é um documento contendo a missão do sistema a ser desenvolvido, os conceitos básicos relacionados à sua área de negócio, as necessidades que justificam seu



## MDS METODOLOGIA DE DESENVOLVIMENTO DE SISTEMAS

Página: 8 / 21

Data Emissão:  
02/06/2015

Versão : 2.0

desenvolvimento e macro objetivos a serem cumpridos. Além disso, devem estar explicitados requisitos arquiteturais específicos e demais informações de cunho geral relacionadas à visão do sistema.

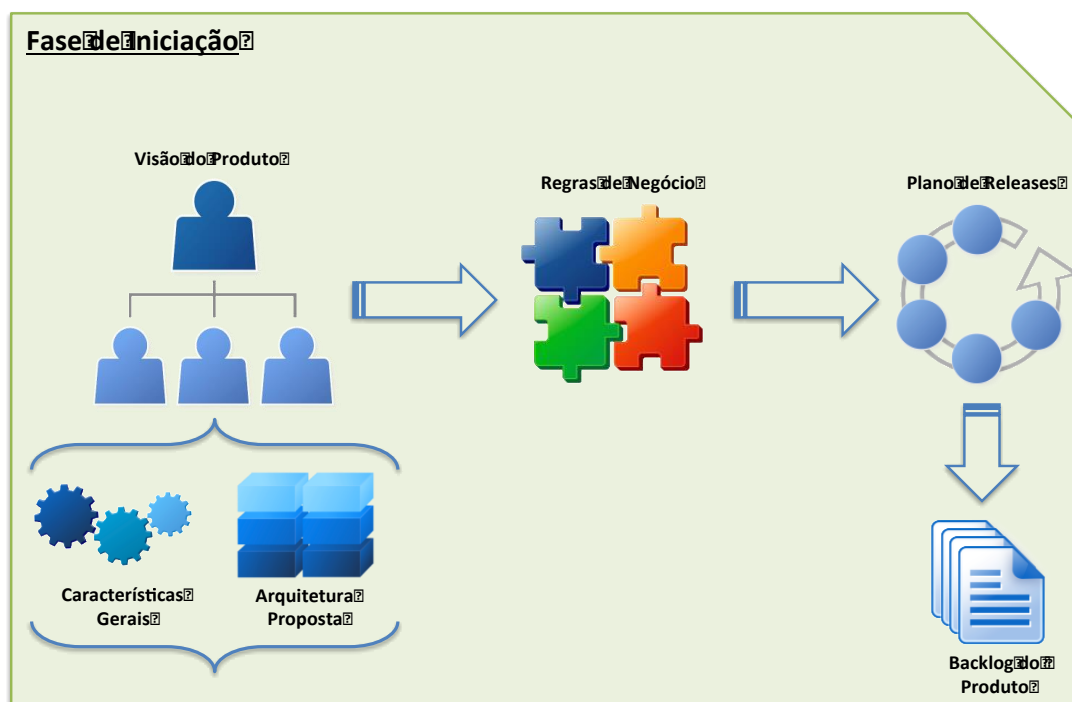


Figura 3 – Fase de Iniciação

### 3.4.2. Definição de Regras de Negócio

Após a delimitação da visão do produto, devem ser elencadas as principais regras de negócio relacionadas ao sistema. Tais regras tem grande importância para entender premissas concretas que orientem o levantamento do Backlog do Produto. Estas regras devem ser revisadas, ampliadas e aprimoradas a cada fase de Discovery.

### 3.4.3. Estabelecimento do Plano de Releases

O plano de releases envolve definir em alto nível as versões significativas do sistema que devem ser alcançadas. Eventualmente, dependendo do tipo de negócio ou tamanho do sistema, apenas uma versão é desejável. Mas em geral, vários marcos podem ser estabelecidos. Deste modo, é possível visualizar a associação destes objetivos de alto nível e funcionalidades com valor significativo para o Product Owner. Um exemplo de relação entre releases e Sprints está demonstrado na Figura 4.

### 3.4.4. Estabelecimento do Backlog do Produto

O backlog do produto é uma lista de todas as histórias que devem ser necessárias na construção do produto, de maneira ordenada por prioridade. É de responsabilidade do Product Owner elicitar estas histórias, e priorizá-las de maneira que a ordem do backlog reflita o grau de importância de cada história.

Este artefato está em constante evolução e é sempre passível de alterações. A versão concluída na iniciação reflete uma visão geral das funcionalidades que o produto deve conter, de forma a delimitar uma noção de escopo para o projeto. Inclusão e exclusão de necessidades é algo comum e rotineiro, cuja revisão é realizada a cada Sprint.





## MDS METODOLOGIA DE DESENVOLVIMENTO DE SISTEMAS

Página: 9 / 21

Data Emissão:  
02/06/2015

Versão : 2.0

Além do exposto, constam no backlog do produto eventuais correções necessárias e também alterações de funcionalidades existentes.

### 3.4.5. Artefatos Resultantes

Os artefatos resultantes da fase de Iniciação são:

- Documento de Visão;
- Regras de Negócio;
- Plano de Releases.
- Backlog do Produto;

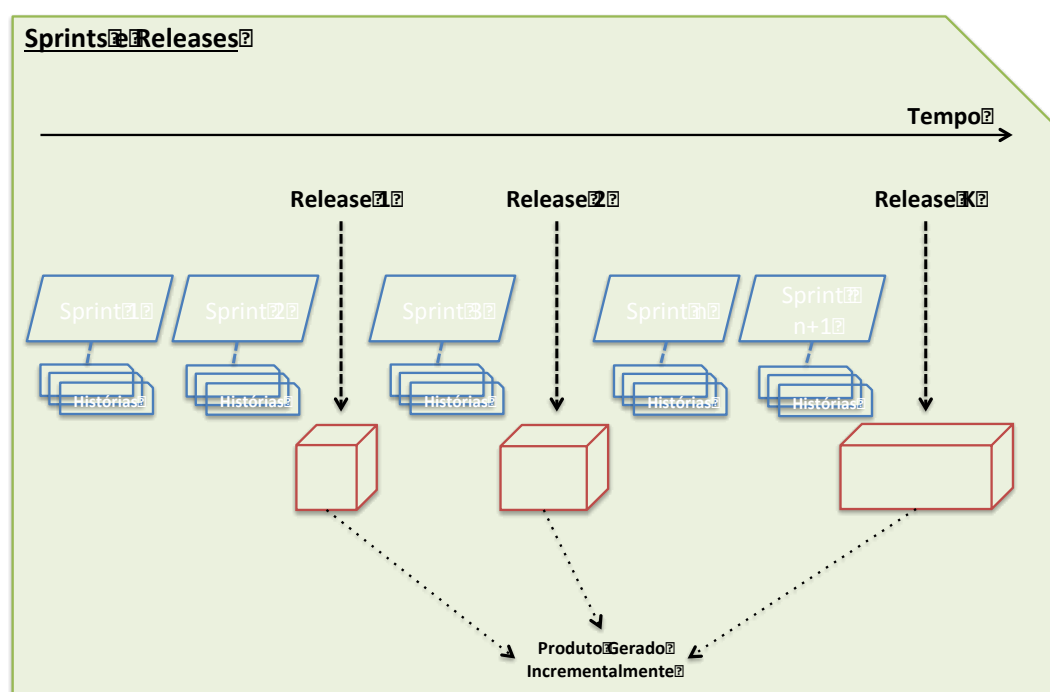


Figura 4 – Sprints e Releases

### 3.5. Fase de Discovery

A fase de Discovery, que é executada a cada Sprint e tem duração fixa de 2 semanas corridas, compreende a etapa relacionada ao planejamento do conteúdo de cada Sprint, e consequentemente, com o refinamento deste conteúdo. Para isso, são realizadas algumas etapas, descritas a seguir.

#### 3.5.1. Revisões da Visão, Backlog do Produto e Regras de Negócio

A cada Discovery, é realizada uma revisão da Visão do Produto, com o intuito de verificar se a missão, premissas e características gerais estão mantidos conforme foi planejado no início do projeto. Em adição, o backlog do produto também é revisado, tal que novas histórias possam ser incluídas, além de permitir-se a repriorização das histórias existentes, ou mesmo a exclusão de histórias que não se façam mais necessárias. Por último, devem ser revisadas também as regras de negócio do sistema, modificando-se regras existentes ou incluindo-se novas regras. Esta fase tipicamente envolverá até 4 horas.

### 3.5.2. Planejamento de Discovery

Faz-se necessário, após a ratificação do conteúdo de backlog e visão do produto, realizar o planejamento de discovery da Sprint a ser executada. Neste caso, a equipe se reúne com o PO e Scrum Master para definirem juntos as histórias a serem elencadas como **obrigatórias e opcionais** naquela Sprint. A quantidade de histórias deve ser ajustada de acordo com a produtividade da equipe. Naturalmente, a produtividade tende a melhorar à medida em que o time compreende mais profundamente o sistema.

Delimitado o escopo aproximado da Sprint, deve haver uma contagem estimada das funcionalidades previstas, para que verifique se há viabilidade na realização das histórias selecionadas.

Esta fase tipicamente envolverá até 4 horas.

### 3.5.3. Refinamento do Backlog da Sprint e Modelo de Dados

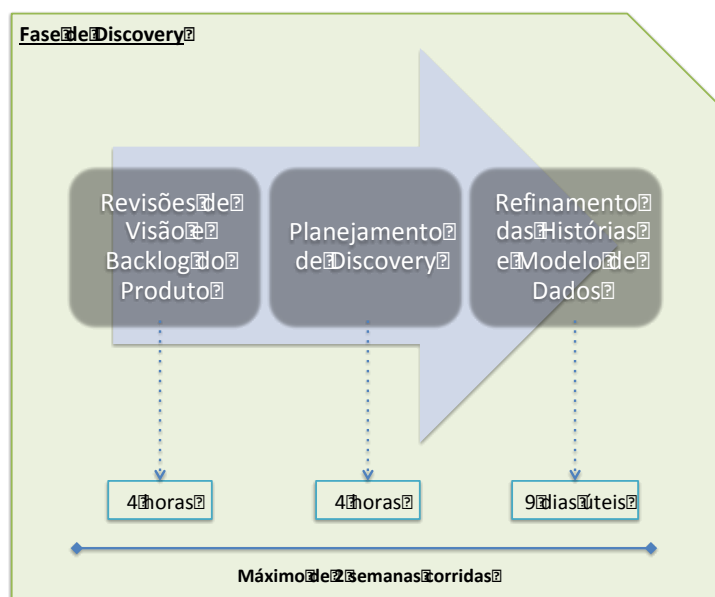
Uma vez definidas as histórias que devem compreender o backlog da Sprint, estas devem ser refinadas em conjunto com o Product Owner. Nesta fase, haverá reuniões envolvendo toda a equipe para o amplo registro e entendimento de cada história, além da geração do modelo de dados atualizado que reflita aquilo que foi elicitado pela equipe.

Várias técnicas podem ser utilizadas neste período de refinamento, incluindo Entrevistas, Brainstorming e Prototipação. Não há artefato documental a ser desenvolvido após a conclusão da fase, ou seja, todos são desenvolvidos em tempo real e de forma incremental, juntamente com o PO, de forma que a conclusão da fase já tem documentação auto-homologada.

Juntamente com as histórias, devem ser trazidos os critérios de aceitação providos pelo PO, que serão transformados em testes unitários na fase de Delivery. Além destes, também será considerada a necessidade, em determinadas histórias de usuário, dependendo da criticidade e complexidade das regras de negócio, que sejam feitos Testes Funcionais Automatizados de acordo com tais critérios de aceitação, utilizando ferramenta padronizada pela CGTI/PF.

Tipicamente, esta etapa deve durar até 9 dias úteis de trabalho.

Na Figura 5, é possível visualizar a sequência e duração de cada etapa do Discovery.



**Figura 5 – Fase de Discovery**



## MDS METODOLOGIA DE DESENVOLVIMENTO DE SISTEMAS

Página: 11 / 21

Data Emissão:  
02/06/2015

Versão : 2.0

---

### 3.5.4. Artefatos resultantes

Os artefatos resultantes da fase de Discovery são:

- Visão, Backlog e Regras de Negócio atualizados;
- Histórias de Usuário refinadas, com protótipos de tela e critérios de aceitação, além da indicação de necessidade de testes funcionais automatizados;
- Modelo de Dados atualizado;

### 3.6. Fase de Delivery

A fase de Delivery, que também é executada a cada Sprint logo após a fase de Discovery, também tem como prazo fixo duas semanas corridas, e compreende a construção das necessidades pactuadas no planejamento da Sprint, ou seja, envolve a codificação e a entrega de um incremento do produto que está sendo tratado no projeto ágil. Para isso, são realizadas algumas etapas, descritas a seguir.

#### 3.6.1. Planejamento de Delivery

No planejamento do delivery, as histórias da Sprint são sub-divididas em tarefas, que são distribuídas entre os integrantes do time. Há um planejamento de metas de curto prazo para cada tarefa, para que possa se alcançar a construção do produto.

#### 3.6.2. Criação de Testes Unitários

Esta metodologia preza pelo desenvolvimento orientado a testes (TDD – *Test Driven Development*), onde os testes unitários são construídos antes mesmo do código executável, para que, utilizando os critérios de aceitação definidos pelo Product Owner nas histórias de usuário, os testes orientem antecipadamente o próprio código a ser desenvolvido, em um processo incremental.

#### 3.6.3. Implementação das Histórias de Usuário

Como objetivo principal do Sprint, serão implementadas as tarefas planejadas de forma a se satisfazer as necessidades expressas nas histórias de usuário previstas para a Sprint.

#### 3.6.4. Criação de Testes Funcionais Automatizados

Esta etapa diz respeito à construção dos testes funcionais automatizados que podem ter sido planejados para a Sprint. Os testes funcionais serão desenvolvidos utilizando ferramenta que esteja padronizada pela CGTI/PF (Ex: Selenium).

#### 3.6.5. Execução de Testes

Antes de liberar o incremento do produto previsto na Sprint, a contratada deve executar os testes desenvolvidos para as histórias em questão, de forma que relatórios com o resultado da execução dos testes estejam disponíveis para verificação por parte da CGTI/PF.

#### 3.6.6. Pré-Homologação

Nesta etapa, a CGTI/PF recebe oficialmente o produto para uma pré-homologação, que tem como finalidade verificar:

- Se o produto entregue atende ao checklist para admissão, conforme subseção a seguir;
- Se todas as histórias planejadas na Sprint estão contempladas no produto entregue;



## MDS METODOLOGIA DE DESENVOLVIMENTO DE SISTEMAS

Página: 12 / 21

Data Emissão:  
02/06/2015

Versão : 2.0

- Se há defeitos de natureza impeditiva.

### 3.6.6.1. **Definição de PRONTO** (*Checklist de Admissão do Produto*)

Qualquer produto enviado para homologação por parte da CGTI/PF deve atender a uma série de critérios para sua admissão à implantação em ambiente de homologação, sem os quais o produto é rejeitado de imediato. Tais critérios estão listados a seguir:

- Código-fonte submetido ao controle de versões da CGTI/PF;
- Existência de testes unitários e do Relatório de Testes;
- Existência de scripts de banco de dados com **dicionário de dados embutido nos metadados** (ausência apenas quando não houver mudança no modelo de dados)
- Existência de arquivo para geração de Build (ex: Arquivo de projeto Maven).
- Existência de Manual de Implantação, conforme modelo disponível na MDS.
- Existência de Manual do Usuário, conforme modelo disponível na MDS.

### 3.6.6.2. *Aceitação da Demanda*

Após realizar a inspeção do produto quanto à sua admissibilidade (item anterior), o gerente de sistemas, juntamente com o Product Owner, poderão:

- Executar testes funcionais automatizados que tenham sido solicitados, e consequentemente verificar se estão corretamente implementados ou mesmo se existem, além de observar os resultados da execução;
- Executar testes unitários ou verificar relatórios de execução destes, que possam envolver porções críticas do produto;
- Realizar alguns testes funcionais, pelo menos nos principais fluxos do produto entregue.

Após a realização destes testes, pode se proceder a uma das ações a seguir:

- **Rejeição:** caso sejam percebidos defeitos de natureza impeditiva em alguma história implementada ou não tenha coberto o escopo planejado de tal forma que a entrega não seja passível de aceitação;
- **Aceitação parcial:** caso a demanda possua alguns defeitos significativos de natureza não-impeditiva ou não tenha coberto o escopo planejado de tal forma que ainda seja passível de aceitação;
- **Aceitação integral:** caso a demanda esteja em nível de qualidade tal que não sejam percebidos defeitos significativos, bem como envolva cumprimento do escopo planejado.

Todos os aspectos julgados relevantes devem ser registrados pelo Gerente de Sistemas e/ou Product Owner no Relatório de Não-Conformidade. Os defeitos percebidos na rejeição e na aceitação parcial devem **obrigatoriamente** fazer parte de um item de backlog da próxima Sprint, específico para correção dos defeitos, salvo determinação contrária do PO ou Gerente de Sistemas.

### 3.6.7. Retrospectiva

A etapa de retrospectiva diz respeito à melhoria contínua do processo. Nesta etapa, os integrantes se reúnem para discutir a Sprint que está sendo concluída, com foco nos desafios, oportunidades e problemas ocorridos. Não faz parte do escopo desta etapa a discussão sobre histórias



## MDS METODOLOGIA DE DESENVOLVIMENTO DE SISTEMAS

Página: 13 / 21

Data Emissão:  
02/06/2015

Versão : 2.0

de usuário e backlog do produto, ou seja, discute-se apenas o processo, e como melhorá-lo. Tipicamente, esta reunião leva até 4 horas.

### 3.6.8. Artefatos Resultantes

Os artefatos resultantes da fase de Delivery são:

- Histórias de Usuário implementadas e cujo código-fonte esteja submetido ao controle de versões;
- Testes unitários e funcionais implementados e executados, com o resultado de testes;
- Demais artefatos relacionados ao deployment do sistema: projeto para criação do build (Ex: Projeto Maven), scripts de banco e manual de implantação.

Na Figura 6, é possível visualizar a sequência e duração das etapas desta fase.

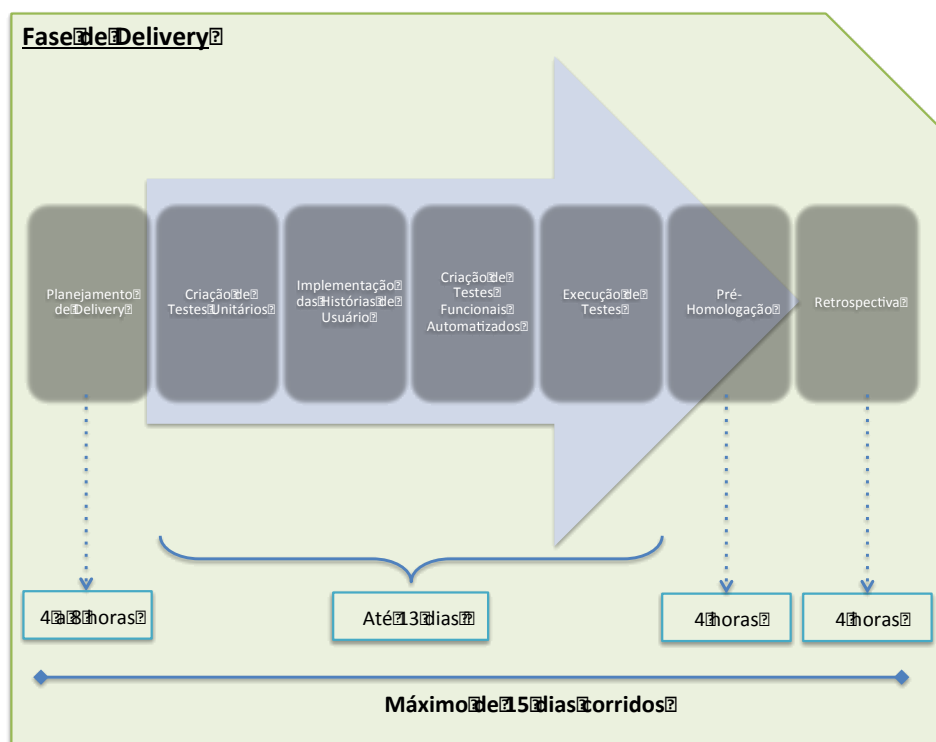


Figura 6 – Fase de Delivery

### 3.7. Fase de Homologação

Esta fase compreende apenas os testes e experimentação detalhados do produto entregue, por parte do PO e/ou Gerente de Sistemas, em até 5 dias úteis, para que possa haver alguma decisão de ordem negocial, como inclusão de novas regras, melhoria da implementação existente ou mesmo rejeição das regras implementadas. Qualquer problema ou observação deve ser acrescido ao Relatório de Não-conformidade criado na Pré-Homologação.



## MDS METODOLOGIA DE DESENVOLVIMENTO DE SISTEMAS

Página: 14 / 21

Data Emissão:  
02/06/2015

Versão : 2.0

### 3.8. Técnicas e Ferramentas Auxiliares ao Processo

Além das fases citadas anteriormente, o PDA envolve a utilização de técnicas e ferramentas auxiliares ao longo do processo, de forma a garantir a correta gestão e minimizar os riscos envolvidos. Tais práticas são detalhadas nas subseções a seguir.

#### 3.8.1. Quadro de Tarefas com Kanban

Todos os projetos desenvolvidos no PDA devem ter um quadro apoiado pela técnica KANBAN contendo as tarefas previstas no projeto, separadas em raias que significam etapas e estados relacionados à execução. Um exemplo deste quadro pode ser visto na Figura 7.

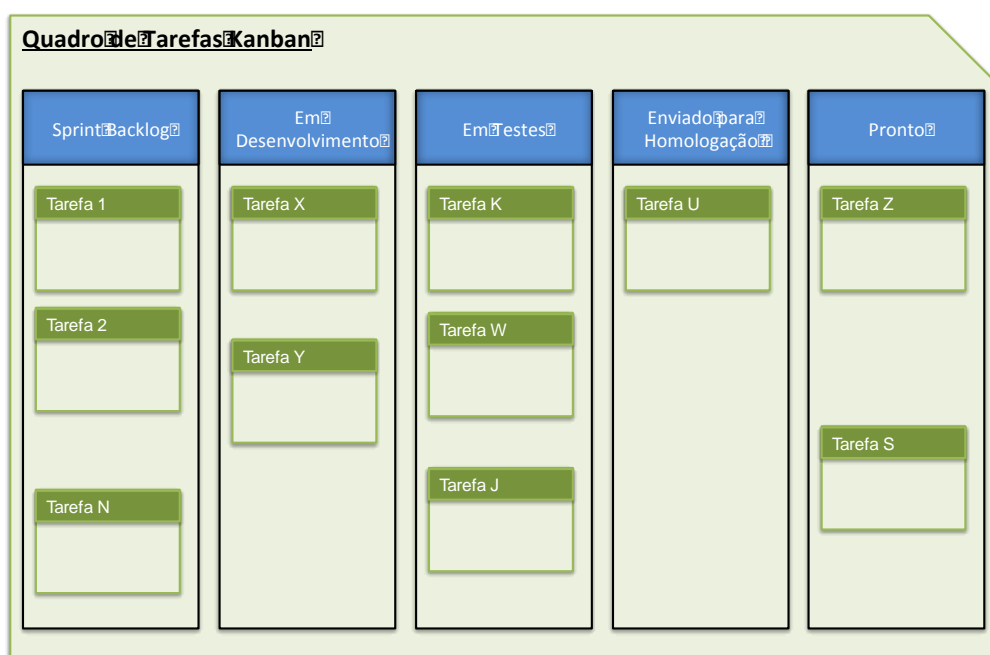


Figura 7 – Quadro de Tarefas Kanban

#### 3.8.2. Reuniões Diárias

Além das reuniões previstas no PDA, tais como Planejamento, Refinamento e Retrospectiva, as equipes devem obrigatoriamente fazer reuniões diárias de 15 minutos, com a participação do Gerente de Sistema, de forma que sejam apresentadas rapidamente e **por cada integrante** as metas do dia, problemas a serem enfrentados e pendências previstas.

Esta deve ser uma reunião transparente, onde todos os riscos são mapeados, e onde todos os integrantes apresentam sua visão diária do projeto.

#### 3.8.3. Gráfico de Burndown

Além do quadro Kanban, um gráfico de produtividade deve ser mantido visível para a equipe de desenvolvimento, exibindo, ao longo do tempo, a relação entre o trabalho planejado e efetivamente realizado. Na figura 8 é possível visualizar um exemplo de gráfico de Burndown. Este gráfico pode ser feito com diferentes índices que representem quantidade de trabalho, como por exemplo, o número de histórias restantes existentes. À medida que o time ganha maturidade, índices mais exatos podem ser utilizados, como horas ou Pontos de Função previstos.



## MDS METODOLOGIA DE DESENVOLVIMENTO DE SISTEMAS

Página: 15 / 21

Data Emissão:  
02/06/2015

Versão : 2.0

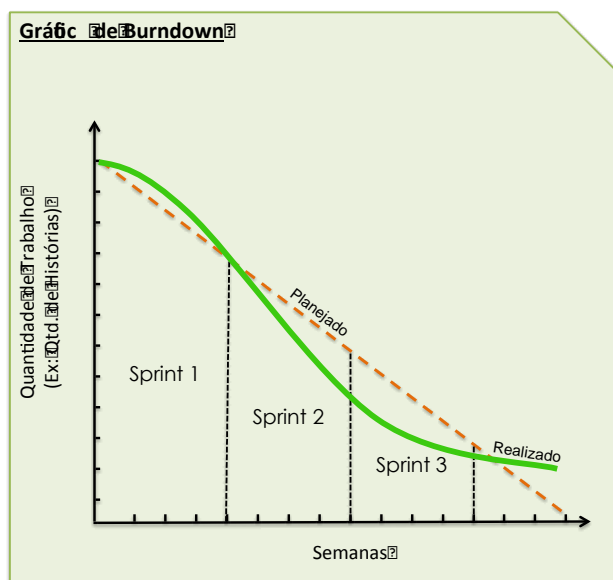


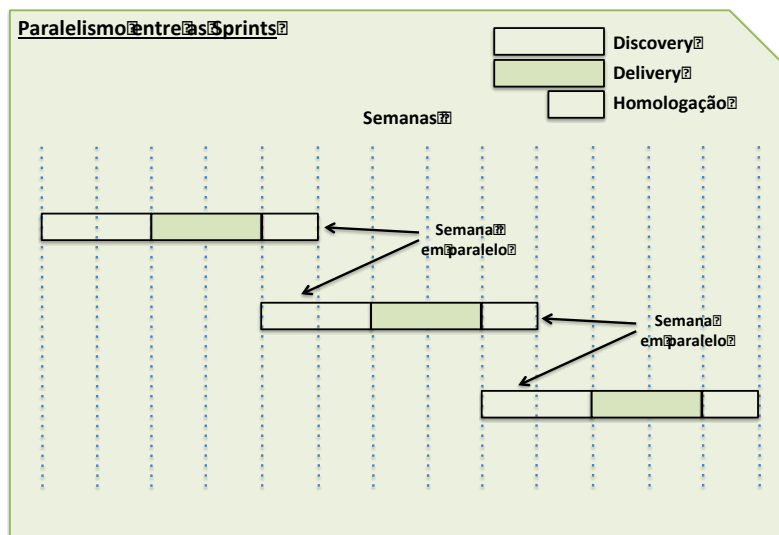
Figura 8 – Gráfico de Burndown

### 3.9. Paralelismo entre as Sprints

O modelo ágil permite que alguns passos entre as Sprints subsequentes sejam paralelizados. Este paralelismo depende da maturidade da organização, do projeto e da disponibilidade do PO, dentre outros fatores. Nesta metodologia, por padrão, será adotado o paralelismo conforme Figura 9.

Neste modelo, **a fase de homologação acontece em paralelo à primeira semana da fase de Discovery da próxima Sprint**. Na referida figura, as semanas estão representadas em raias verticais, e as fases coloridas representam etapas de Sprint, conforme legenda.

Tal paralelismo implica, naturalmente, que quaisquer alterações negociais decorrentes desta fase de homologação (percepção de mudança de negócio, novas regras relacionadas às funcionalidades implementadas, etc.) podem ser incorporadas diretamente na Sprint em andamento.





**Figura 9 – Paralelismo entre as Sprints**

### 3.10. Artefatos

Embora o Processo de Desenvolvimento Ágil preconize objetividade e código pronto, há uma série de artefatos a serem produzidos no processo, além do produto em si. Tais artefatos podem ser expressos em formato de documento, cujo template é fornecido, ou através de registro na ferramenta de projeto de software em uso na CGTI/PF.

Ainda existem, além dos artefatos gerais, artefatos específicos para desenvolvimento de componentes e webservices, e por fim os guias operacionais, que são padrões tecnológicos e metodológicos a serem adotados pela contratada.

#### 3.10.1. Artefatos Gerais de Desenvolvimento


**Legenda:** I – Iniciação Di – Discovery De – Delivery  
GO – Guia Operacional TP – Template

LISTA DE ARTEFATOS	DESCRIÇÃO	NOME DOS TEMPLATES	I	Di	De
Documento de Visão	Descreve uma visão geral do projeto, seus produtos e suas características	TP – Sigla Projeto – Visão	x		
Backlog do Produto	Lista das funcionalidades a serem desenvolvidas para o produto, em ordem decrescente de prioridade.	Registrado em ferramenta de projeto	x	x	x
História de Usuário	Especifica uma necessidade do Product Owner e seus critérios de aceitação, além de esboços de tela (quando necessário, pode ser protótipo navegável).	Registrado em ferramenta de projeto		x	
Regras de Negócio	Especifica as regras de negócio que farão parte das Histórias de Usuário	TP - Sigla Projeto - Regras de Negócios	x	x	
Modelo de dados	Projeto de modelo de dados para a aplicação	TP - Sigla Projeto – Modelagem de Dados		x	x
Manual de Implantação	Recursos de instalação, hardware, software	TP - Sigla Projeto - Manual de Implantação			x
Manual do Usuário	Descreve as funcionalidades do sistema para ajuda ao usuário.	TP WEB - Sigla Projeto - Manual do Usuário.doc			x
Relatório de Testes	Artefato gerado por ferramenta que execute os testes unitários	Não se aplica			x
Relatório de Não-conformidade	Relatar as não-conformidades detectadas.	Registrado em ferramenta de projeto			x

#### 3.10.2. Artefatos para Desenvolvimento de Componentes e WebServices

LISTA DE ARTEFATOS	DESCRIÇÃO	NOME DOS TEMPLATES
Especificação Técnica de	Especifica linguagens de programação,	ET - Sigla Projeto -



	<b>MDS</b> <b>METODOLOGIA DE DESENVOLVIMENTO</b> <b>DE SISTEMAS</b>	Página: 17 / 21
		Data Emissão: 02/06/2015
		Versão : 2.0

Componente	componentes, logs e tratamento de erro.	Especificação Técnica Componente
Especificação Técnica para Consumo WebService	Especifica a arquitetura para consumo WebService	ET - Sigla Projeto - Especificação Técnica para Consumo Webservice
Especificação Técnica Webservice	Especifica a representação da arquitetura WebService	ET - Sigla Projeto - Especificação Técnica Webservice

### 3.10.3. Padrões Tecnológicos – Guias Operacionais

GUIAS OPERACIONAIS	DESCRIÇÃO	NOME DOS TEMPLATES
Manual de Padrão de Telas	Define um padrão de telas para os sistemas da PF	GO - Camada de Apresentação de Telas
Arquitetura de referência JAVA	Documento que define a arquitetura base para os sistemas	GO - Sigla Projeto - Arquitetura de Referência Java
Arquitetura de Referência de Componentes Corporativos	Descrever a arquitetura padrão utilizada no desenvolvimento de componentes corporativos pelos fornecedores de sistemas do SDS/CGTI – DPF	GO - Arquitetura de Referência Componentes Corporativos
Padrões de Nomenclaturas para Banco de Dados	Padronizar a Nomenclatura para Banco de Dados	GO – Nomenclatura para Banco de Dados
Orientações sobre a confecção de modelos de dados	Orientar a criação de modelos de dados	GO - Elaboração de Modelos de Dados
Dicionário de dados	Dicionário de dados do sistema	GO - Dicionário de Dados Mainframe
Padrão de codificação JAVA	Padrão de codificação JAVA	GO - Codificação Java
Padrão de codificação Mainframe	Padrão de codificação Mainframe	GO – Codificação Ambiente Mainframe
Documentação de Sistemas Legados	Adoção de uma documentação mínima para os sistemas legados garantirá maior produtividade, qualidade e facilidade nos processos de manutenção	GO - Documentação de Sistemas legados
Versionamento de Código	Descrever as atividades do Processo de Controle de Versionamento de Programas a ser adotado no Processo de Desenvolvimento de Software utilizado no SDS/CGTI, de forma que a evolução dos sistemas ocorra de modo controlado.	GO – Versionamento de Código



## MDS METODOLOGIA DE DESENVOLVIMENTO DE SISTEMAS

Página: 18 / 21

Data Emissão:  
02/06/2015

Versão : 2.0

### 4. PROCESSO DE SUSTENTAÇÃO DE SISTEMAS (PDS)

Trata-se do processo que envolve as atividades de sustentação de sistemas, tratando-se de manutenção continuada e estendendo-se desde sua implantação até o momento em que for substituído ou descontinuado.

#### 4.1. Atividades de Sustentação

Dentro deste processo, estão englobadas uma série de atividades distintas, detalhadas a seguir.

##### 4.1.1. Manutenção corretiva

Consiste na eliminação de comportamentos do software que divirjam de suas especificações ou que provoquem a interrupção inesperada de seu funcionamento.

##### 4.1.2. Manutenção adaptativa tecnológica

Consiste na alteração do sistema para adaptá-lo às mudanças do ambiente computacional onde foi desenvolvido ou onde é executado, considerados aí os componentes tecnológicos passíveis de adaptação: Sistema Gerenciador de Bancos de Dados, Servidor de Aplicações, bibliotecas e/ou *frameworks* utilizados e as evoluções da própria linguagem computacional utilizada.

##### 4.1.3. Manutenção cosmética localizada

Consiste de alteração de interface de usuário que não implique alteração das regras de negócio do Caso de Uso e que seja realizada de forma localizada, isto é, pela intervenção em um único arquivo ou em um pequeno conjunto de arquivos. Tal manutenção pode ser exemplificada da forma que se segue:

- Fontes de letra, cores, logotipos, mudanças de botões, alteração na posição de campos e texto na tela;
- Mudanças de texto em mensagens do sistema, título de um relatório ou *labels* de uma tela de consulta;
- Mudanças de texto estático em e-mail enviado pelo sistema.

##### 4.1.4. Apurações especiais

Consiste na preparação de roteiros de execução em linguagem SQL, ou outra adequada ao caso, destinados às extrações de dados não cobertas pelos relatórios do sistema, à correção de inconsistências nos dados mantidos pelo sistema e não realizáveis por meio das interfaces de usuário disponíveis (ou cujo volume inviabilize a sua execução de forma manual), ou à inserção de dados não automatizada no sistema.

##### 4.1.5. Atendimento

Trata-se de prestação de esclarecimentos, à CGTI/PF, quanto à forma como foram implementados os requisitos do sistema, aos procedimentos requeridos ao seu correto funcionamento ou aos dados mantidos por ele. Em adição, também se enquadra nesta atividade o apoio à identificação e isolamento de falhas e problemas na execução do software.



## MDS METODOLOGIA DE DESENVOLVIMENTO DE SISTEMAS

Página: 19 / 21

Data Emissão:  
02/06/2015

Versão : 2.0

---

### 4.1.6. Rotinas operacionais

Consiste na execução de quaisquer procedimentos operacionais rotineiramente requeridos pelo sistema em função de suas regras de negócio ou forma de construção.

## 4.2. Tratamento de Demandas de Sustentação

Todas as atividades descritas anteriormente seguem fluxos a serem tratados pela contratada, que têm como ponto de partida o registro do incidente ou solicitação no Sistema de Gestão de Demandas em uso na CGTI/PF, que a partir de então deverá ser tratado pela contratada de acordo com as cláusulas de nível de serviço (que não são escopo desta metodologia) e tipo da demanda. Uma visão geral deste tratamento pode ser vista na Figura 10.

### 4.2.1. Manutenções Corretivas, Adaptativas e Cosméticas

Após a execução do serviço, a contratada deverá tomar as seguintes providências:

- Providenciar que o código com a mudança solicitada (correção, adaptação ou manutenção cosmética) seja enviado para geração de build em homologação, ou eventualmente diretamente em produção (dependendo da urgência ou tipo demanda, tratado a cada caso);
- Atualizar a versão do sistema conforme o guia operacional de versionamento;
- Registrar tudo o que for realizado no sistema de gestão de demandas, relacionando, se possível, com casos semelhantes já conhecidos.

### 4.2.2. Apurações Especiais

Após a execução do serviço, a contratada deverá tomar as seguintes providências:

- Providenciar que a intervenção em base de dados com a apuração solicitada (correção de registros, inclusão de dados em CODE TABLE, relatórios direto de base de dados, dentre outras) seja enviada para execução em base de dados de homologação, ou eventualmente, diretamente em base de produção (dependendo da urgência ou tipo demanda, tratado a cada caso);
- Registrar tudo o que for realizado no Sistema de Gestão de Demandas, relacionando, se possível, com casos semelhantes já conhecidos.

### 4.2.3. Atendimento e Rotinas Operacionais

Após a execução do serviço, a contratada deverá tomar as seguintes providências:

- Anexar evidências, caso possível, do atendimento realizado ou rotinas executadas;
- Registrar tudo o que for realizado no Sistema de Gestão de Demandas, relacionando, se possível, com casos semelhantes já conhecidos.



## MDS METODOLOGIA DE DESENVOLVIMENTO DE SISTEMAS

Página: 20 / 21

Data Emissão:  
02/06/2015

Versão : 2.0

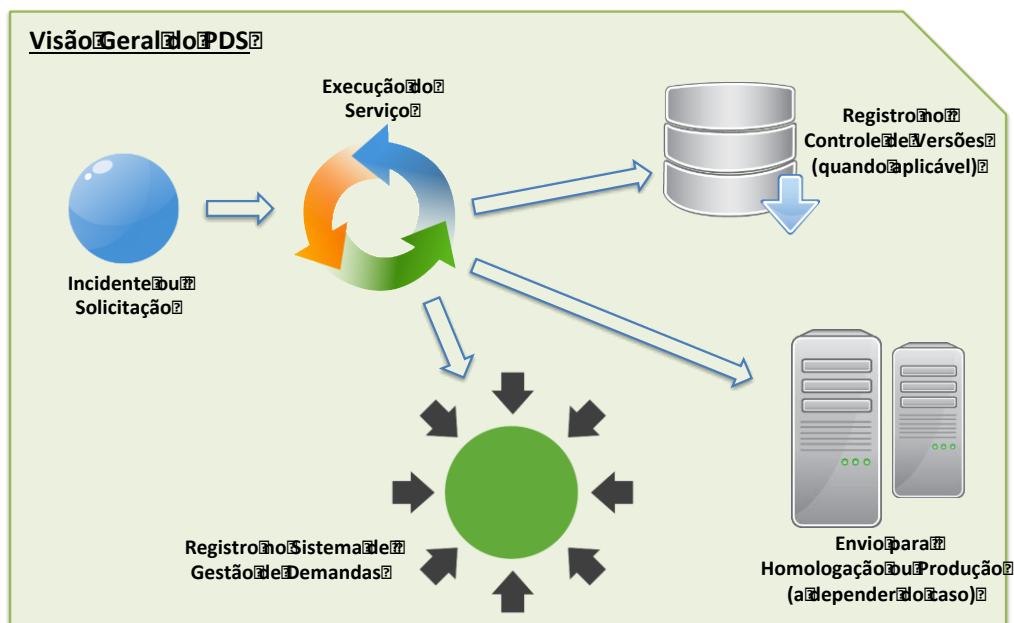


Figura 10 – Visão Geral do Processo de Sustentação de Sistemas

### 5. PROCESSO DE EVOLUÇÃO DE PEQUENO PORTE (PEP)

Trata-se de processo para lidar com situações em que as evoluções necessárias e/ou desejadas são de pequeno porte e não são elencadas como projeto. Cabe ressaltar que evoluções categorizadas como projeto são tratadas pelo PDA (Processo de Desenvolvimento Ágil).

Este processo, cuja visão geral pode ser observada na Figura 11, é composto de duas partes:

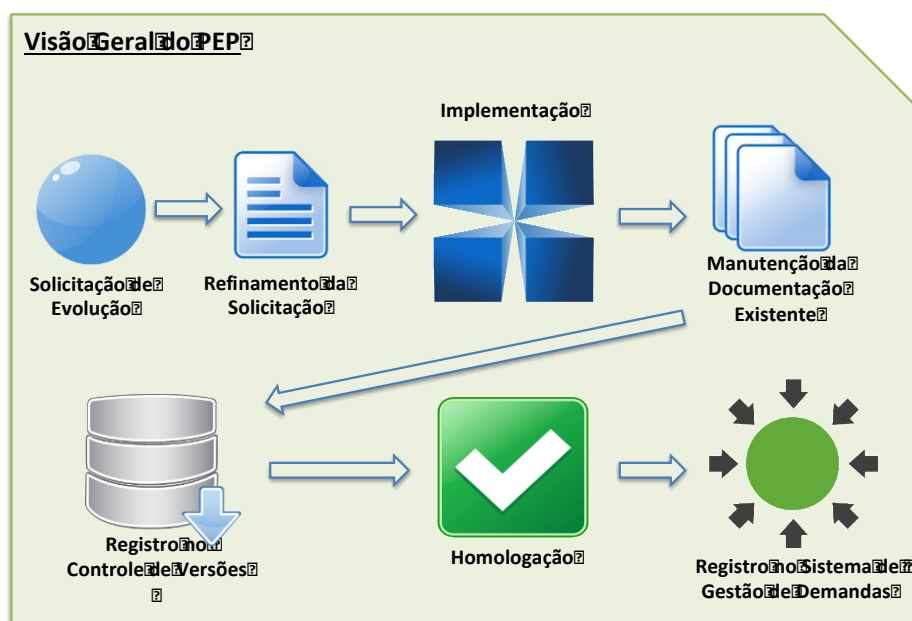


Figura 11 – Visão Geral do Processo de Evolução de Pequeno Porte



## MDS METODOLOGIA DE DESENVOLVIMENTO DE SISTEMAS

Página: 21 / 21

Data Emissão:  
02/06/2015

Versão : 2.0

### 5.1.1. Abertura da solicitação

A necessidade de evolução deve ser relatada por um usuário demandante no Sistema de Gestão de Demandas, que tipicamente pode ser um Gestor ou Gerente de Sistema. Devido ao pequeno porte envolvido, não há formato fixo para tal relato, que é livre.

### 5.1.2. Refinamento da solicitação

Após o recebimento da solicitação, deve haver uma reunião para refinar o entendimento sobre seu teor. Nessa reunião, quaisquer detalhes devem ser coletados e registrados no caso aberto no Sistema de Gestão de Demandas.

### 5.1.3. Implementação

Seguinte ao refinamento da solicitação, deve haver então a implementação conforme solicitado. Neste passo, é importante seguir boas práticas tais como desenvolver ou evoluir testes unitários correspondentes, e seguir as premissas arquiteturais existentes na CGTI/PF.

Ao final da implementação, devem ser executados os testes unitários correspondentes, cuja evidência deve ser anexada ao Sistema de Gestão de Demandas.

### 5.1.4. Manutenção da Documentação Existente

Em caso de haver documentação existente para o projeto em questão, esta deverá ser atualizada conforme as novas regras implementadas na evolução em questão.

### 5.1.5. Registro no Controle de Versões

O produto resultante da solicitação deve ser submetido ao Controle de Versões, e a ele atribuído versão conforme o Guia Operacional de Versionamento, anexo a esta MDS.

### 5.1.6. Homologação

Deve ser gerado um executável (build) da versão correspondente à evolução em questão, e submetido ao ambiente de homologação para que o usuário demandante possa verificar se o artefato construído está de acordo com a solicitação.

### 5.1.7. Registro no Sistema de Gestão de Demandas

Todo o histórico e demais informações da solicitação devem estar devidamente registradas no Sistema de Gestão de Demandas após a homologação por parte do demandante.

## 6. PROCESSO DE GERENCIAMENTO DE PROJETOS

Para os processos de gerenciamento de projetos e os artefatos a serem utilizados foi elaborada uma metodologia própria - Metodologia de Gerenciamento de Projetos da Polícia Federal – MGP/PF.

A MGP/PF e respectivos artefatos seguem um padrão de layout elaborado pelo Escritório de Gestão de Projetos Estratégicos – EGPE - EGPE/GAB/DG diferenciado do padrão da CGTI/PF e poderão ser acessados via Intranet endereço:

<http://intranet.dpf.gov.br/cti/servicos/projetos/metodologia-de-gerenciamento-de-projetos>