



TRIBUNAL REGIONAL ELEITORAL DO PIAUI

Ata de Reunião Nº 11 - TRE/PRESI/DG/STI

ATA DE REUNIÃO DO COMITÊ GESTOR DE TECNOLOGIA DA INFORMAÇÃO

IDENTIFICAÇÃO DA REUNIÃO

Data	Horário		Local	Coordenador da Reunião
24/03/2026	11:00h	12:32h	Videoconferência	Anderson Lima

PAUTA

1. Revisão do processo de Desenvolvimento e Sustentação de Sistemas.
2. Revisão da Metodologia de Gerenciamento de Projetos de TI.
3. Artefatos de Contratações de TI

PARTICIPANTES

Nome	Unidade
Ana Caroline Carvalho Portela	GABSTI
Anderson Cavalcanti de Lima	GABSTI
Danilo Nascimento Cruz	NSEGI
Etevaldo Cândido Custódio	SEAU
Márcio Igo Carvalho Ribeiro Gonçalves	SELOGI
Leonardo Moraes Júnior	COELEI
Marilza Ribeiro de Moraes	GABSTI
Nadja Marcela Melo Silva Santiago	COSUT
Paulo das Neves e Silva Junior	SEDESC
Rosemberg Maia Gomes	CODIN

APRESENTAÇÃO

Discussão	Decisão / Pendência	Responsável	Data Limite
Abertura	Recepcionados os presentes, o Secretário apresentou a pauta da reunião.	Anderson Lima	Não se aplica
	Contextualização acerca das informações relacionadas à aquisição/pauta da reunião.	Ana Caroline	Não se aplica
Metodologia de Gestão de Projetos de TI	Exposição da necessidade de revisão da atual Portaria nº 925/2022 para garantir maior aderência à realidade de execução da Secretaria de Tecnologia da Informação (STI). Entre as inovações apresentadas a classificação das iniciativas baseada em sua complexidade e impacto institucional: demandas mais simplificadas seguirão um fluxo direto sob a forma de "Planos de Ação", enquanto iniciativas de maior envergadura serão estruturadas formalmente como "Projetos".	Ana Caroline	Não se aplica
	O CGTI aprovou a Metodologia de Gestão de Projetos de TI sem ressalvas.	Membros do CGTI	Não se aplica
	O Gabinete da STI ficou responsável por promover os trâmites administrativos para a publicação da portaria regulamentadora.	Membros do CGTI	Não se aplica

Discussão	Decisão / Pendência	Responsável	Data Limite
Processo de Desenvolvimento e Sustentação de Sistemas.	<p>Apresentação do Manual do Processo de Desenvolvimento de Software (0002676508).</p> <p>A principal alteração metodológica apresentada foi a adoção de um modelo de trabalho híbrido, que consolida práticas do <i>Scrum</i> (como a estruturação em <i>Sprints</i>, reuniões de revisão e retrospectiva) com métricas de controle do <i>Kanban</i> (limites de <i>WIP</i>, <i>Lead Time</i> e <i>Cycle Time</i>). Também foi prevista a inclusão da sustentabilidade e acessibilidade em alinhamento às normas vigentes, especialmente oriundas do TCU e CNJ.</p>	Paulo das Neves	Não se aplica
	<p>O CGTI aprovou a revisão Manual do Processo de Desenvolvimento de Software na forma em que foi apresentada.</p>	Paulo das Neves	Não se aplica
	<p>A CODIN deverá encaminhar a minuta da nova portaria ao Gabinete STI em caráter de urgência, viabilizando a publicação a tempo de compor o portfólio de comprovações do iGovTI</p>	Rosemberg Maia	Não se aplica

Discussão	Decisão / Pendência	Responsável	Data Limite
Artefatos de Contratações de TI	<p>A Coordenadora de Suporte Técnico informou que os estudos relacionados à aquisição de no-breaks e estabilizadores já havia sido apreciada pelo CGTI, mas houve necessidade refazer os citados estudos, pois a pesquisa de preços utilizada para a composição do valor estimado do item Estabilizador foi baseada, em sua maioria, em equipamentos que não possuem Certificação de Conformidade com a norma ABNT NBR 14373. No novo estudo foi contemplado o acréscimo no quantitativo de estabilizadores para atender à demanda da COELEI.</p>	Nadja Marcela	Não se aplica
	<p>Apresentação dos artefatos de contratação relacionados à aquisição de estabilizadores e no-breaks constantes no Processo SEI nº 0017188-81.2025.6.18.8000, conforme ETP evento SEI nº 0002670563, Termo de Referência 49 (0002670566), Mapa de Riscos - IN05 0002670570.</p>	Etevaldo Custódio	Não se aplica
	<p>Os membros do CGTI aprovaram os Estudos Técnicos Preliminares apresentados pela COSUT, relacionados à aquisição em tela, com a ressalva de que seja retirado do Termo de Referência a dotação orçamentária vinculada ao pleito eleitoral, substituindo-a por fonte de custeio administrativo regular.</p>	Membros do Comitê	Não se aplica

Discussão	Decisão / Pendência	Responsável	Data Limite
Encerramento	Como parte integrante desta ata seguem anexos a minuta do manual de gestão de projetos de TI (evento 0002676485) e a minuta do manual do processo de desenvolvimento de software (0002676508).	Anderson Lima	Não se aplica
	Não havendo outros assuntos a serem tratados, o Secretário de Tecnologia da Informação agradeceu aos presentes e finalizou a reunião.	Anderson Lima	Não se aplica

ASSINATURA DOS MEMBROS DO COMITÊ GESTOR DE TECNOLOGIA DA INFORMAÇÃO

Nome	Unidade	Assinatura
Anderson Cavalcanti de Lima	GABSTI	Assinatura Eletrônica
Leonardo Moraes Júnior	COELEI	Assinatura Eletrônica
Rosemberg Maia Gomes	CODIN	Assinatura Eletrônica
Nadja Marcela Melo Silva Santiago	COSUT	Assinatura Eletrônica
Ana Caroline Carvalho Portela	GABSTI	Assinatura Eletrônica

Em 24 de março de 2026.



Documento assinado eletronicamente por **Rosemberg Maia Gomes, Coordenador de Desenvolvimento e Infraestrutura**, em 27/03/2026, às 08:25, conforme art. 1º, § 2º, III, "b", da Lei 11.419/2006.



Documento assinado eletronicamente por **Leonardo Moraes Junior, Coordenador de Eleições Informatizadas**, em 27/03/2026, às 09:47, conforme art. 1º, § 2º, III, "b", da Lei 11.419/2006.



Documento assinado eletronicamente por **Nadja Marcela Melo Silva Santiago, Coordenador(a) de Suporte Técnico**, em 27/03/2026, às 12:57, conforme art. 1º, § 2º, III, "b", da Lei 11.419/2006.



Documento assinado eletronicamente por **Anderson Cavalcanti de Lima, Secretário de Tecnologia da Informação**, em 30/03/2026, às 13:37, conforme art. 1º, § 2º, III, "b", da Lei 11.419/2006.



Documento assinado eletronicamente por **Ana Caroline Carvalho Portela**, Técnico Judiciário, em 30/03/2026, às 13:39, conforme art. 1º, § 2º, III, "b", da Lei 11.419/2006.



A autenticidade do documento pode ser conferida no site https://sei.tre-pi.jus.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 informando o código verificador **0002676334** e o código CRC **0E8DC737**.

0004553-34.2026.6.18.8000

0002676334v35



--

**METODOLOGIA DE GESTÃO DE PROJETOS DE
TECNOLOGIA DA INFORMAÇÃO**

ANEXO ÚNICO DA PORTARIA PRESIDÊNCIA Nº 165/2026

TERESINA – PI , março de 2026

TRIBUNAL REGIONAL ELEITORAL DO PIAUÍ

Praça Desembargador Edgar Nogueira s/n, Centro Cívico CEP 64000-920
Teresina - Piauí
Telefone: (86) 2107-9700

Elaboração

Antônio Manoel Silveira de Sousa – NSEGI
Marilza
Ana Caroline Carvalho Portela

Revisão

Anderson Cavalcanti de Lima - GABSTI

Validação

Comitê Gestor de Tecnologia da Informação

Aprovação

Desembargador ERIVAN LOPES
Presidente do TRE-PI

Capa

Breno Ponte de Brito – SECOM

Composição do Pleno do TRE-PI

Desembargador Sebastião Ribeiro Martins

Presidente

Desembargador Ricardo Gentil Eulálio Dantas

Vice-Presidente e Corregedor Regional Eleitoral

Dr. Gustavo André Oliveira dos Santos

Juiz Federal

Dra. Maria Luíza de Moura Mello e Freitas

Dr. Edson Alves da Silva

Juízes de Direito

Dr. Daniel Eufrásio de Sousa Alves

Jurista

Dr. Kelston Pinheiro Lages

Procurador Regional Eleitora

Composição do Órgão Consultivo do TRE-PI

Dra. Júnia Maria Feitosa Bezerra Fialho

Juiza Auxiliar da Presidência do TRE-PI

Dr. Melissa de Vasconcelos Lima Pessoa

Juiza Auxiliar da Corregedoria Regional Eleitoral

Composição da Secretaria do TRE-PI

Silvani Maia Resende Santana

Diretora-Geral

Anderson Cavalcanti de Lima

Secretario de Tecnologia da Informação

José Alves Siqueira Filho
Secretário de Administração, Orçamento e Finanças

Tadeu Almeida Martins
Secretário de Gestão de Pessoas

Walter Schel Alves da Costa Raposo
Secretária Judiciária

Composição do Comitê Diretivo de Tecnologia da Informação

Doutora Júnia Maria Feitosa Bezerra Fialho
Representante da Presidência – Presidente do Comitê

Elvira Maria Osório Pitombeira Meneses Carvalho
Magistrada de 1º Grau

Hediane Lima Xavier
Representante da Corregedoria

Silvani Maia Resende Santana
Diretora-Geral

Anderson Cavalcanti de Lima
Secretário de Tecnologia da Informação

Tadeu Almeida Martins
Secretária de Gestão de Pessoas

Walter Schel Alves da Costa Raposo
Secretário Judiciário

José Alves Siqueira Filho
Secretária de Administração, Orçamento e Finanças

Composição do Comitê Gestor de Tecnologia da Informação

Anderson Cavalcanti de Lima

Secretario de Tecnologia da Informação

Rosemberg Maia Gomes

Coordenador de Desenvolvimento e Infraestrutura

Leonardo Moraes Junior

Coordenador de Eleições Informatizadas

Nadja Marcela Melo Silva Santiago

Coordenadora de Suporte Técnico

Ana Caroline Carvalho Portela

Assistente de Governança de Tecnologia da Informação

1. INTRODUÇÃO.....	8
2. APLICABILIDADE.....	9
3. DEFINIÇÕES.....	9
4. PROCESSO GERENCIAMENTO DE PROJETOS DE TI.....	11
4.1 MACROPROCESSO GERENCIAMENTO DE PROJETOS DE TI.....	11
4.1.2 SUBPROCESSO INICIAÇÃO.....	12
4.1.3 SUBPROCESSO DE PLANEJAMENTO.....	16
4.1.4 SUBPROCESSO DE EXECUÇÃO.....	21
4.1.5. SUBPROCESSO DE MONITORAMENTO E CONTROLE.....	24
4.1.6 SUBPROCESSO ENCERRAMENTO.....	29
4.1.7. SUBPROCESSO GERENCIAR RISCOS.....	32
5. CONCLUSÃO.....	39

1. INTRODUÇÃO

A Metodologia de Gestão de Projetos de Tecnologia da Informação (MGP- TI), elaborada em consonância com a missão e visão de futuro do Tribunal Regional Eleitoral do Piauí (TRE-PI), com seus valores institucionais e com seus objetivos estratégicos.

A elaboração da MGP foi precedida do estudo das metodologias adotadas por diversos órgãos da Administração Pública Federal com o intuito de levantar informações acerca do gerenciamento de projetos nesses órgãos e de identificar sugestões de melhorias para as unidades do TRE-PI na gestão de seus projetos de Tecnologia da Informação (TI).

Algumas características desta versão:

- Diálogo com o PMBOK 7.0.
- Mais ágil e adaptável aos diferentes tipos de projetos da STI;
- Mais simples e menos burocrática;
- Modelos dos artefatos revisados para refletir maior eficiência e flexibilidade;
- Incorporação de sugestões de usuários e gestores (clientes);
- Aderência às melhores práticas de gerenciamento de projetos do mercado;
- Captura da satisfação dos clientes com a execução e gestão dos projetos;
- Tratamento mais adequado aos serviços de sustentação e pequenas evoluções de sistemas;
- Empoderamento dos times.

A MGP-TI foi desenvolvida para atender a gestores da STI, a gerentes e líderes de projeto, a equipes de projeto, bem como a qualquer colaborador que queira adquirir conhecimento ou melhorar suas competências na área de gerenciamento de projetos.

Ela também está disponível para outras Secretarias do Regional que queiram utilizá-la como referência na elaboração de suas próprias metodologias de gestão

de projetos.

A Metodologia de Gestão de Projetos de Tecnologia da Informação (MGP- TI), elaborada em consonância com a missão e visão de futuro do Tribunal Regional Eleitoral do Piauí (TRE-PI), com seus valores institucionais e com seus objetivos estratégicos.

2. APLICABILIDADE

Este documento tem aplicabilidade para toda a Secretaria de Tecnologia da Informação do TRE-PI.

3. DEFINIÇÕES

- **Iniciativa:** são esforços para a criação de soluções tecnológicas, o desenvolvimento de produtos e serviços, ou ainda a resolução de problemas. As iniciativas podem ser dos seguintes tipos:
- **Projeto:** um esforço temporário empreendido para criar um produto, serviço ou resultado único.
- **Programa:** um grupo de projetos, programas subsidiários e atividades relacionadas, gerenciado de modo coordenado visando à obtenção de benefícios que não estaria disponível se ele fosse gerenciado individualmente.
- **Plano de ação:** um esforço com o escopo reduzido e bem definido, para apoiar a resolução de problemas críticos ou a investigação de suas causas.
- **Gestão de Projetos:** Aplicação de conhecimentos, habilidades, ferramentas e técnicas necessárias à condução do projeto, a fim de atender aos seus objetivos e compatibilizar escopo, qualidade, cronograma, orçamento, recursos e riscos.
- **Portfólio de Projetos:** Conjunto de projetos, programas, portfólios subsidiários e operações gerenciado em grupo para alcançar objetivos

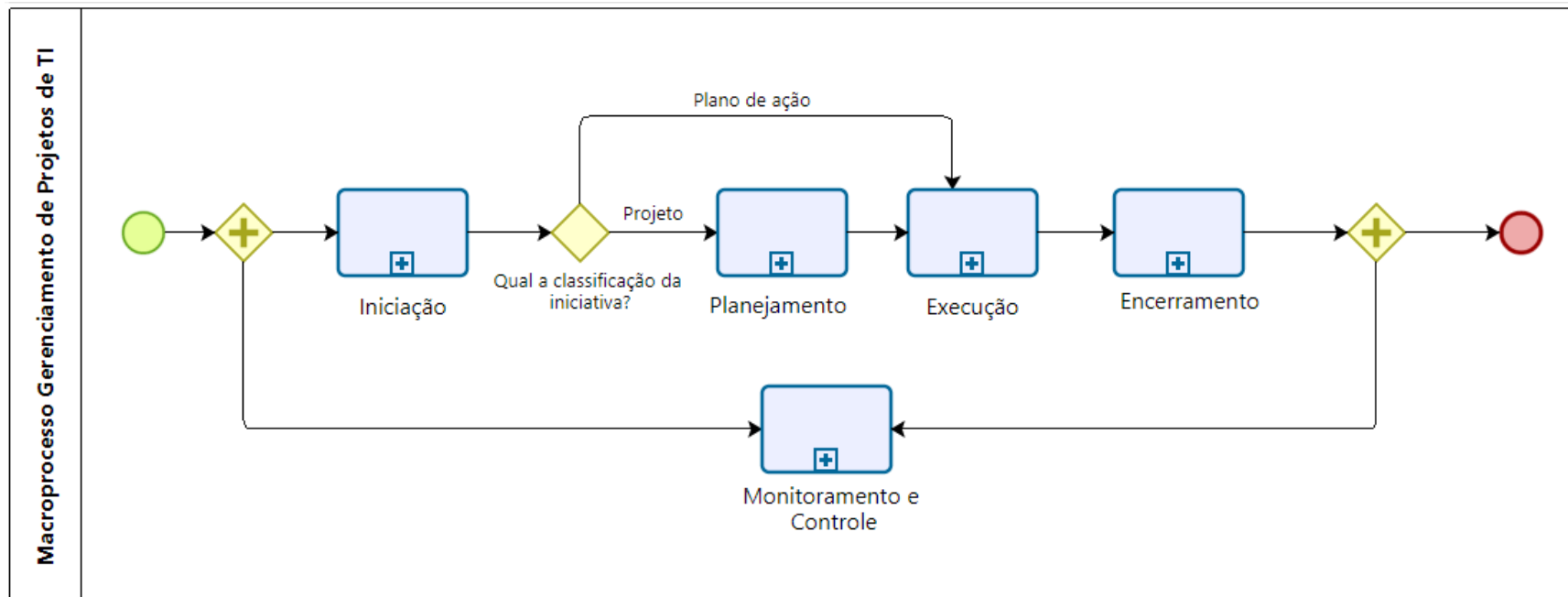
estratégicos. O portfólio tem um escopo organizacional que muda com os objetivos estratégicos da organização.

- **Gerente de Projeto:** Pessoa responsável pela gestão e condução de um projeto perante o Tribunal, com autoridade para tomar decisões com relação ao projeto.
- **Cliente:** gestor responsável por uma área de negócio, grupo de trabalho ou comissão responsável por definir as necessidades do projeto.
- **Unidade Gestora:** área de negócio responsável pela gestão da solução a ser criada pelo projeto.
- **Unidade Executora:** área responsável pela execução do projeto.
- **COGTI:** Coordenadoria de Governança da TI, unidade responsável por zelar para que as atividades dos projetos sejam realizadas de acordo com a metodologia vigente. Também apoia o incremento de qualidade e a melhoria da eficiência na condução dos projetos por meio da estruturação e da implantação de processos e métodos de gestão, realizando o apoio efetivo aos gerentes e às equipes na execução das funções de gestão.
- **CGTI:** Comitê de Gestão de TI responsável por assessorar a Secretaria de TI na aprovação de diretrizes, projetos de TI, planos de ação, investimentos, projetos básicos, termos de referência e outros documentos de relevância institucional.
- **CDTI:** Comitê Diretivo de Tecnologia da Informação responsável por assessorar a administração do Tribunal na definição das prioridades de investimento em TI, na aprovação e alocação de recursos orçamentários destinados à TI e na deliberação e priorização de planos e riscos decorrentes dos relatórios de gestão submetidos pelo CGTI

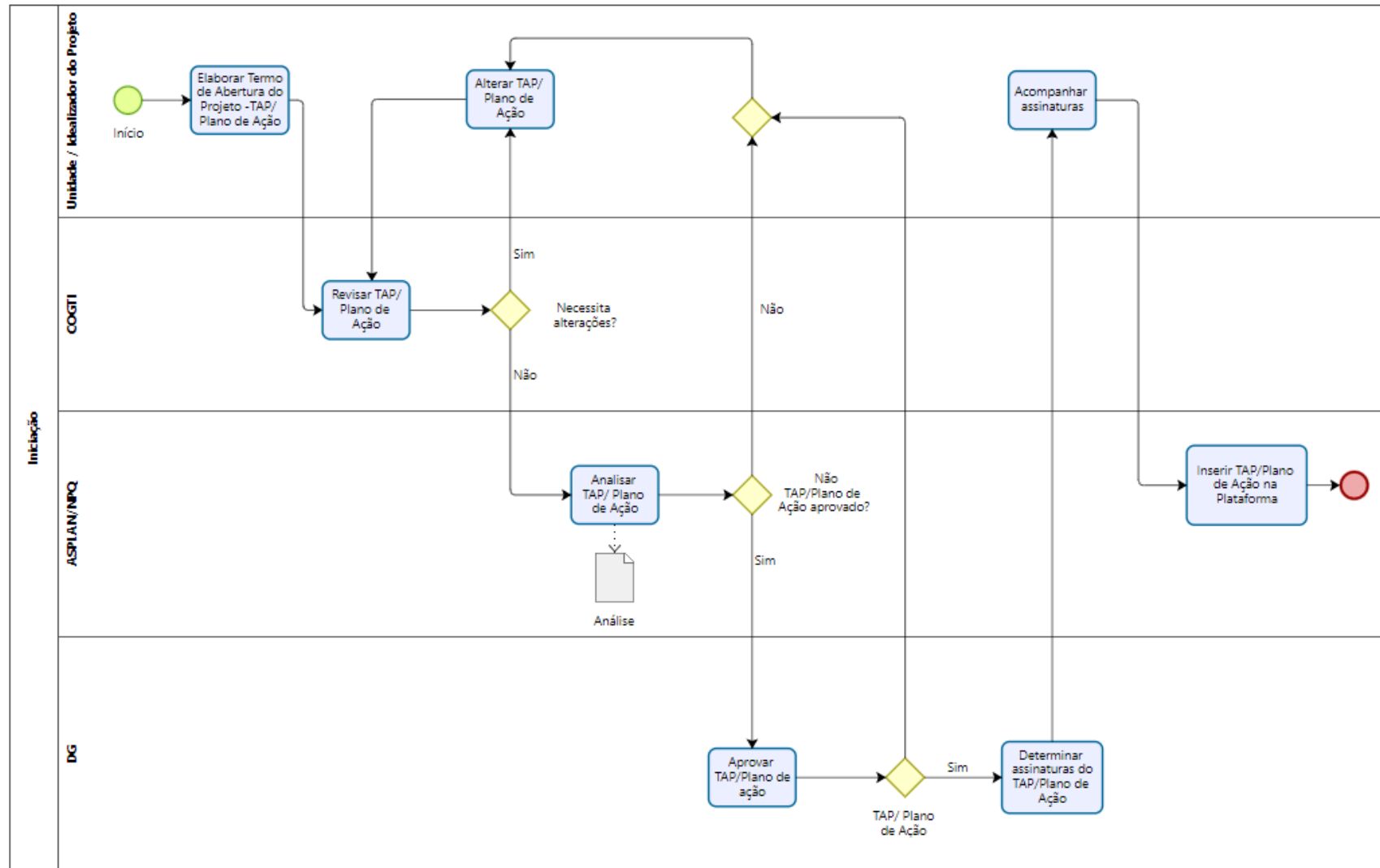
4. PROCESSO GERENCIAMENTO DE PROJETOS DE TI

4.1 Macroprocesso Gerenciamento de Projetos de TI

O macroprocesso Gerenciamento de Projetos de TI é composto por cinco fases denominadas iniciação, planejamento, execução, monitoramento e controle, e encerramento. Essas fases representam os grupos de processos necessários para o desenvolvimento de qualquer projeto, garantindo que ele seja executado de forma organizada e atinja seus objetivos.



4.15.2 Subprocesso Iniciação



Nesta fase, o projeto ou plano de ação é formalmente iniciado, com a definição de seu escopo, objetivos, viabilidade e autorização. É o momento de identificar as necessidades, os riscos, as restrições e as expectativas do projeto.

A classificação da iniciativa deverá observar seu grau de complexidade, abrangência e impacto institucional. Iniciativas mais simples, com escopo reduzido, menor nível de risco e execução direta, serão formalizadas como Plano de Ação. Por outro lado, iniciativas mais complexas, que envolvam maior esforço de planejamento, múltiplas entregas, maior risco ou impacto estratégico, deverão ser estruturadas como Projetos, mediante a elaboração do respectivo TAP.

4.1.3 Elaborar Termo de Abertura do Projeto -TAP/Plano de Ação

O idealizador é responsável pela elaboração do Termo de Abertura do Projeto (TAP) ou do Plano de Ação, documento que consolida as informações essenciais para análise e aprovação da iniciativa, tais como: identificação, justificativa, objetivos, resultados esperados, principais entregas, riscos preliminares com respectivas estratégias de resposta, definição da equipe técnica e coleta de assinaturas eletrônicas.

Responsável: idealizador(a) do projeto

4.1.4 Revisar TAP/Plano de Ação

A COGTI verifica o correto preenchimento dos campos, a coerência e clareza das informações, além da classificação e categorização do projeto. Assegura o alinhamento com outros projetos em desenvolvimento. Caso necessário, o TAP/Plano de Ação retorna ao idealizador para ajustes.

Responsável: COGTI

4.1.5 Analisar TAP/Plano de Ação

A ASPLAN analisa os objetivos macro, estratégicos e de desenvolvimento sustentáveis do TAP/Plano de Ação e gera um documento de análise. Se o parecer for favorável, o projeto segue para a Diretoria Geral para deliberação. Se não for

aprovado, o idealizador deverá realizar as alterações sugeridas antes de nova submissão.

Responsável: ASPLAN/NPQ

4.1.6 Alterar TAP/Plano de ação

Nesta etapa, o idealizador deve analisar as inconsistências apontadas e promover os ajustes necessários no Termo de Abertura de Projeto (TAP) ou no Plano de Ação.

Responsável: idealizador(a) do projeto

4.1.7 Determinar assinaturas do TAP/Plano de ação

Após aprovação, as partes envolvidas procedem com a assinatura do documento

Responsável: DG

4.1.8. Acompanhar assinaturas

O idealizador do projeto deve acompanhar as assinaturas do TAP/plano de ação.

Responsável: idealizador(a) do projeto

4.1.9 Inserir TAP/Plano de Ação na Plataforma

Finalização do fluxo com a inclusão do documento oficial na Plataforma.

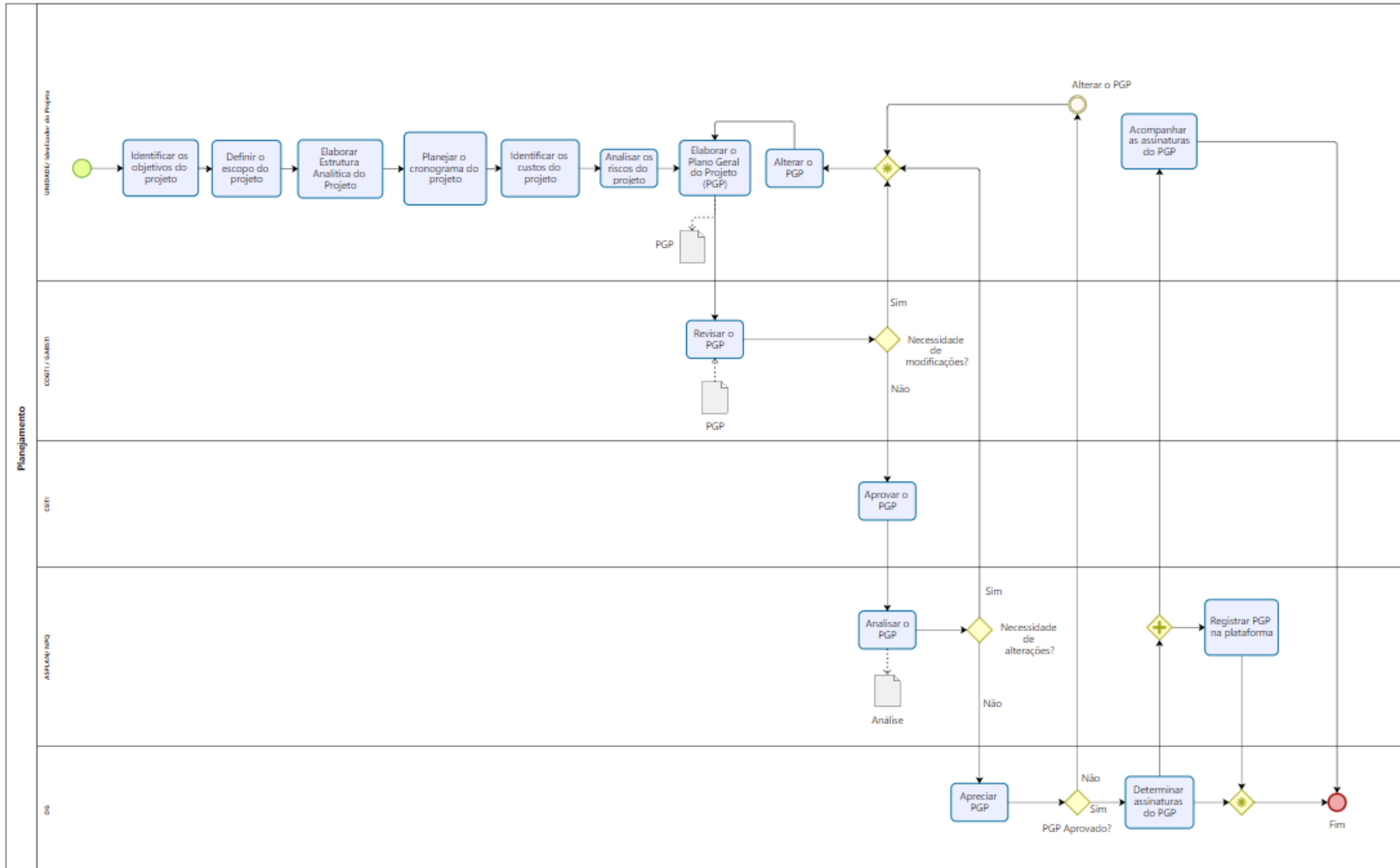
Responsável: ASPLAN

Uma vez formalizada a iniciativa, o fluxo adota uma Estrutura de Decisão obrigatória que direciona a demanda conforme sua classificação:

- Plano de Ação (PA): Iniciativas classificadas como Plano de Ação serão direcionadas para a fase de Planejamento, onde serão detalhados o cronograma tático, a distribuição de tarefas e os recursos operacionais necessários antes do início das atividades.

Projeto (TAP): Iniciativas formalizadas via Termo de Abertura de Projeto seguirão diretamente para a fase de Execução, considerando que as definições estratégicas e os requisitos técnicos consolidados no TAP já autorizam o início imediato das entregas e o consumo de recursos previstos.

4.2. Subprocesso de Planejamento



O planejamento detalha como o projeto será executado, monitorado controlado e concluído. Nesta fase, são definidos o escopo, o cronograma, os recursos, os padrões de comunicação e qualidade, os custos, os riscos e o plano de gerenciamento. Caso a iniciativa tenha sido classificada como plano de ação, o fluxo segue diretamente para o subprocesso “Execução”.

4.2.1 Identificar os objetivos do projeto

O processo inicia com a definição clara do que o projeto pretende alcançar. O idealizador estabelece as metas principais e os resultados esperados que justificam a existência da iniciativa.

Responsável: Unidade / Idealizador do Projeto

4.2.2 Definir o escopo do projeto

Nesta etapa, delimita-se o que será entregue e o que não faz parte do projeto. Define-se o trabalho necessário para entregar o produto ou serviço com as características especificadas.

Responsável: Unidade / Idealizador do Projeto

4.2.3 Elaborar Estrutura Analítica do Projeto (EAP)

O idealizador realiza a decomposição hierárquica do escopo total do trabalho em partes menores e mais manejáveis (pacotes de trabalho), facilitando a organização e o controle.

Responsável: Unidade / Idealizador do Projeto

4.2.4 Planejar o cronograma do projeto

O planejamento do cronograma visa definir a estratégia de execução no tempo, estabelecendo a ordem temporal e os prazos das atividades do projeto para monitorar se as datas acordadas estão sendo cumpridas.

Responsável: Unidade / Idealizador do Projeto

4.2.5 Identificar os custos do projeto

Esta tarefa envolve o levantamento e controle do orçamento disponível e das aquisições necessárias (como contratos, licenças e insumos), garantindo que os gastos estejam dentro da legalidade e do orçamento aprovado.

Responsável: Unidade / Idealizador do Projeto

4.2.6 Analisar os riscos do projeto

Consiste em identificar e monitorar ameaças potenciais antes que se tornem problemas, documentando o contexto, probabilidade e impacto para definir ações de mitigação ou contingência.

Responsável: Unidade / Idealizador do Projeto

4.2.7 Elaborar Plano Geral do Projeto (PGP)

O Líder do Projeto, com base nas diretrizes estabelecidas no TAP, elabora a primeira versão do Plano Geral do Projeto (PGP). Este documento detalha o planejamento operacional, incluindo o detalhamento dos objetivos, cronograma, orçamento, matriz de riscos e recursos necessários. Após a conclusão, o plano é encaminhado para a revisão técnica da COGTI/GABSTI.

Responsável: Unidade / Idealizador do Projeto

4.2.8 Revisar PGP

A COGTI e o GABSTI realizam a revisão técnica e validam o conteúdo do PGP. Nesta etapa, verifica-se a viabilidade técnica e a consistência do planejamento. Havendo necessidade de ajustes ou correções, o documento é devolvido à unidade idealizadora para atualização. Caso esteja em conformidade, o processo segue para a pauta do CGTI.

Responsável: A COGTI e o GABSTI

4.2.9 Alterar o PGP

O idealizador do projeto realiza as alterações necessárias no PGP.

Responsável: Unidade / Idealizador do Projeto

4.2.10 Aprovar PGP

O Comitê de Gestão de TI (CGTI) avalia o PGP sob a ótica de priorização e governança. Se o comitê estiver de acordo com planejamento apresentado, o plano é aprovado e encaminhado à ASPLAN/NPQ. Caso contrário, o PGP retorna para a unidade idealizadora para as alterações solicitadas pelo colegiado.

Responsável: CGTI

4.2.11 Analisar PGP

A Assessoria de Planejamento verifica a adequação dos dados preenchidos e a conformidade com o Termo de Abertura (TAP) original antes de submetê-lo à Diretoria-Geral.

Responsável: ASPLAN/NPQ.

4.2.12 Apreciar/Aprovar PGP

A Diretoria Geral (DG) realiza a apreciação final do PGP sob a ótica estratégica institucional. Se aprovado, o fluxo prossegue para a coleta de assinaturas formais e o encerramento da fase de planejamento, autorizando o início da execução. Caso a DG decida pela não aprovação, o PGP retorna à unidade idealizadora para alteração ou arquivamento definitivo.

Responsável: DG

4.2.13 Determinar assinaturas do PGP

Após a aprovação, a DG define quem são os atores responsáveis por assinar formalmente o documento para garantir o compromisso institucional.

Responsável: DG

4.2.14 Acompanhar as assinaturas do PGP

O idealizador garante que todos os responsáveis definidos pela DG realizem as assinaturas eletrônicas necessárias.

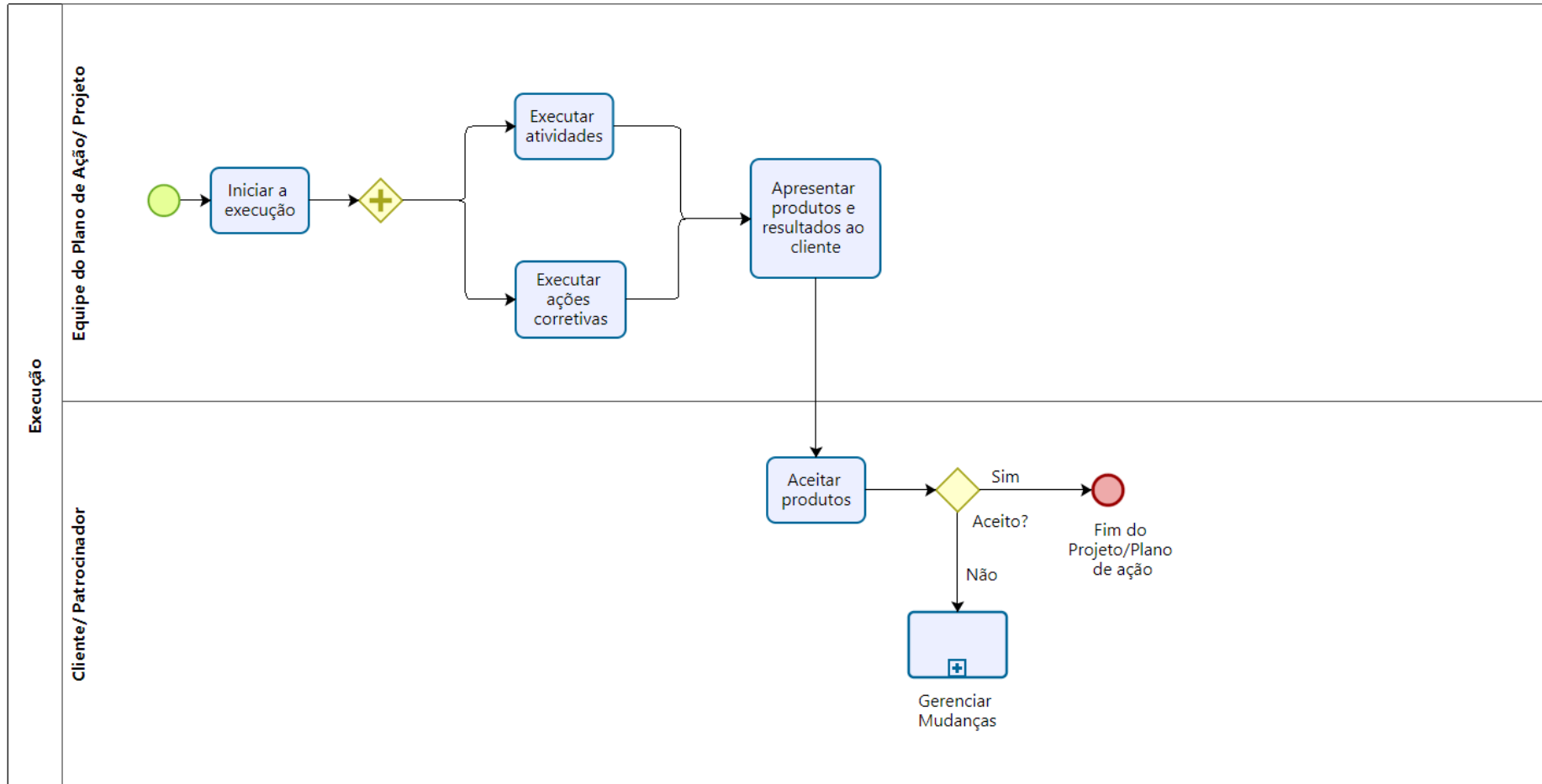
Responsável: Unidade / Idealizador do Projeto

4.2.15 Registrar PGP na plataforma

Em paralelo ao processo de assinaturas, a ASPLAN realiza o registro oficial dos dados do projeto aprovado na plataforma de gestão da instituição, finalizando o rito de planejamento.

Responsável: ASPLAN / NPQ

4.3 Subprocesso de Execução



É a fase onde o plano de gerenciamento do projeto é colocado em prática. Nesta etapa, os recursos são coordenados para realizar as atividades planejadas.

4.3.1 Iniciar a Execução

O processo é disparado a partir de uma "Chamada para o subprocesso", que ativa formalmente a tarefa de início da execução. Esta etapa marca a transição do planejamento para a operacionalização das frentes de trabalho.

Responsável: Equipe do Plano de Ação / Projeto

4.3.2 Executar atividades

A equipe realiza as tarefas técnicas e operacionais previstas no PGP/Plano de ação. Esta atividade ocorre de forma paralela e integrada à execução de correções, garantindo o fluxo contínuo de entregas.

Responsável: Equipe do Plano de Ação / Projeto

4.3.4 Executar ações corretivas

Simultaneamente à execução das atividades principais, a equipe implementa ajustes para corrigir desvios identificados. Essa prática assegura que as falhas sejam tratadas em tempo real, sem aguardar o encerramento da fase.

Responsável: Equipe do Plano de Ação / Projeto

4.3.5 Apresentar produtos e resultados ao cliente

Após a conclusão das frentes de trabalho, a equipe submete as entregas finais e os resultados alcançados ao Cliente/Patrocinador para análise detalhada, validação e aceite formal.

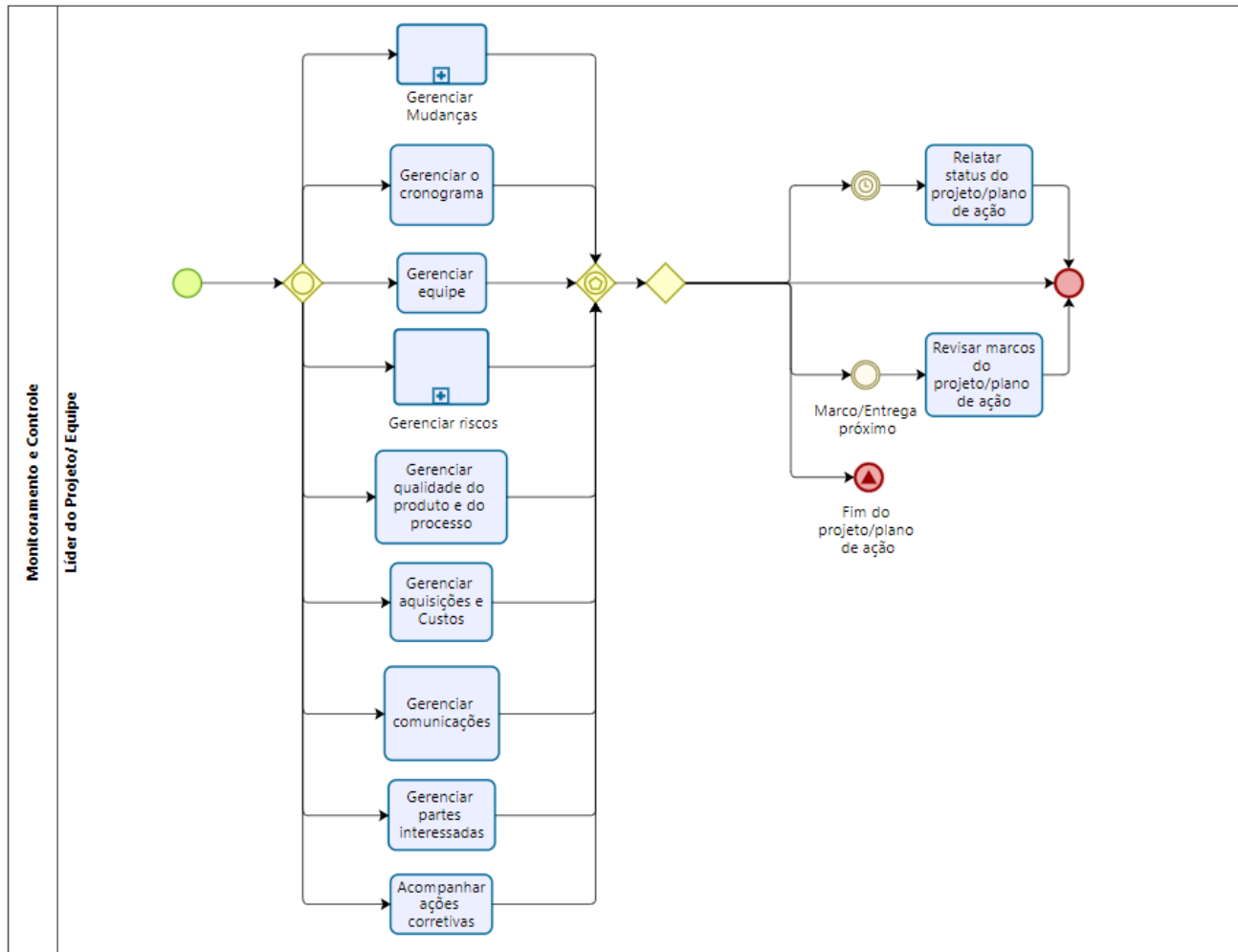
Responsável: Equipe do Plano de Ação / Projeto

4.3.6 Aceitar produtos

O Cliente/Patrocinador avalia se os resultados atendem aos requisitos estabelecidos. Se os produtos forem validados, o fluxo segue para o encerramento (Fim do Projeto). Caso contrário, o processo retorna via "Chamada para o subprocesso" para novas rodadas de execução e correção.

Responsável: Cliente / Patrocinador

4.4. Subprocesso de Monitoramento e Controle



Nesta fase, o progresso do projeto é avaliado em relação ao plano inicial. São verificadas as métricas de desempenho, identificados os desvios e implementadas as ações corretivas necessárias para garantir que o projeto permaneça no caminho certo.

Além disso, o processo atua como o eixo central de governança, exigindo que o líder e a equipe gerenciem múltiplas dimensões (cronograma, equipe, qualidade, custos e comunicações) simultaneamente. Após todas as atividades de gestão serem executadas e convergirem, o processo passa pelo gateway vazio, que atua como um filtro de decisão. O projeto então entra em uma rotina de revisão.

Dinâmica do Fluxo:

Execução Paralela: As atividades são iniciadas em conjunto através de um gateway paralelo, garantindo que todos os aspectos do projeto sejam acompanhados sem defasagem.

Barreira de Sincronização: O fluxo converge para um segundo gateway paralelo, que impede o avanço do processo até que todas as atividades anteriores sejam formalmente concluídas. Decisão (Gateway Exclusivo), Após a sincronização, o processo é direcionado conforme o gatilho.

Relatar status do projeto/plano de ação: Rotina periódica de comunicação baseada em tempo.

Revisar marcos do projeto: Focado na validação de entregas específicas (acionado por sinal/mensagem).

Fim do projeto/plano de ação: Encerramento definitivo e formal. A equipe monitora o progresso continuamente, comparando-o ao planejado. Quando desvios são detectados, ações corretivas são imediatamente implementadas para garantir a aderência do projeto ao plano original.

4.4.1 Gerenciar Mudanças

Lida com solicitações de alteração no escopo ou cronograma. É um subprocesso, sugerindo um nível maior de complexidade. Por ser um subprocesso, envolve um rito próprio de análise de impacto e aprovação antes da implementação.

Responsável: Líder do Projeto / Equipe

4.4.2 Gerenciar o Cronograma

Monitora o cumprimento das datas acordadas. Monitora se as datas acordadas estão sendo cumpridas. Identifica se há atrasos que podem comprometer o projeto e toma providências para realinhá-lo.

Responsável: Líder do Projeto / Equipe

4.4.3 Gerenciar Equipe

Gerencia recursos humanos, monitora produtividade, identifica necessidades de capacitação e realoca recursos conforme necessário.

Responsável: Líder do Projeto / Equipe

4.4.4 Gerenciar Riscos

Acompanha os riscos identificados, monitora os residuais, identifica novos riscos e avalia a eficácia do processo.

Responsável: Líder do Projeto / Equipe

4.4.5 Gerenciar Qualidade do Produto e do Processo

Verifica se o que está sendo construído segue os padrões técnicos exigidos e se o processo é eficiente.

Responsável: Líder do Projeto / Equipe

4.4.6 Gerenciar Aquisições e Custos

Controla o orçamento disponível e as compras necessárias (contratos, licenças, insumos). Garante que o gasto esteja dentro da legalidade e do orçamento aprovado.

Responsável: Líder do Projeto / Equipe

4.4.7 Gerenciar Comunicações

Mantém os stakeholders informados. Garante que a informação flua corretamente para que ninguém tome decisões com base em dados errados.

Responsável: Líder do Projeto / Equipe

4.4.8 Gerenciar Partes Interessadas (Stakeholders)

Identifica quem é impactado pelo projeto (Juízes, servidores, cidadãos, partidos). O objetivo é garantir que as expectativas deles sejam alinhadas e gerenciadas.

Responsável: Líder do Projeto / Equipe

4.4.9 Acompanhar Ações Corretivas

Ajusta a rota caso algo saia do planejado. Quando um erro é detectado na execução, esta atividade registra a correção para garantir que ele não se repita.

Responsável: Líder do Projeto / Equipe

4.9.10 Relatar Status do Projeto/Plano de ação

Acionado por um evento de tempo, representa o dever de prestar conta, indicando uma rotina periódica de comunicação (ex: relatórios semanais ou mensais).

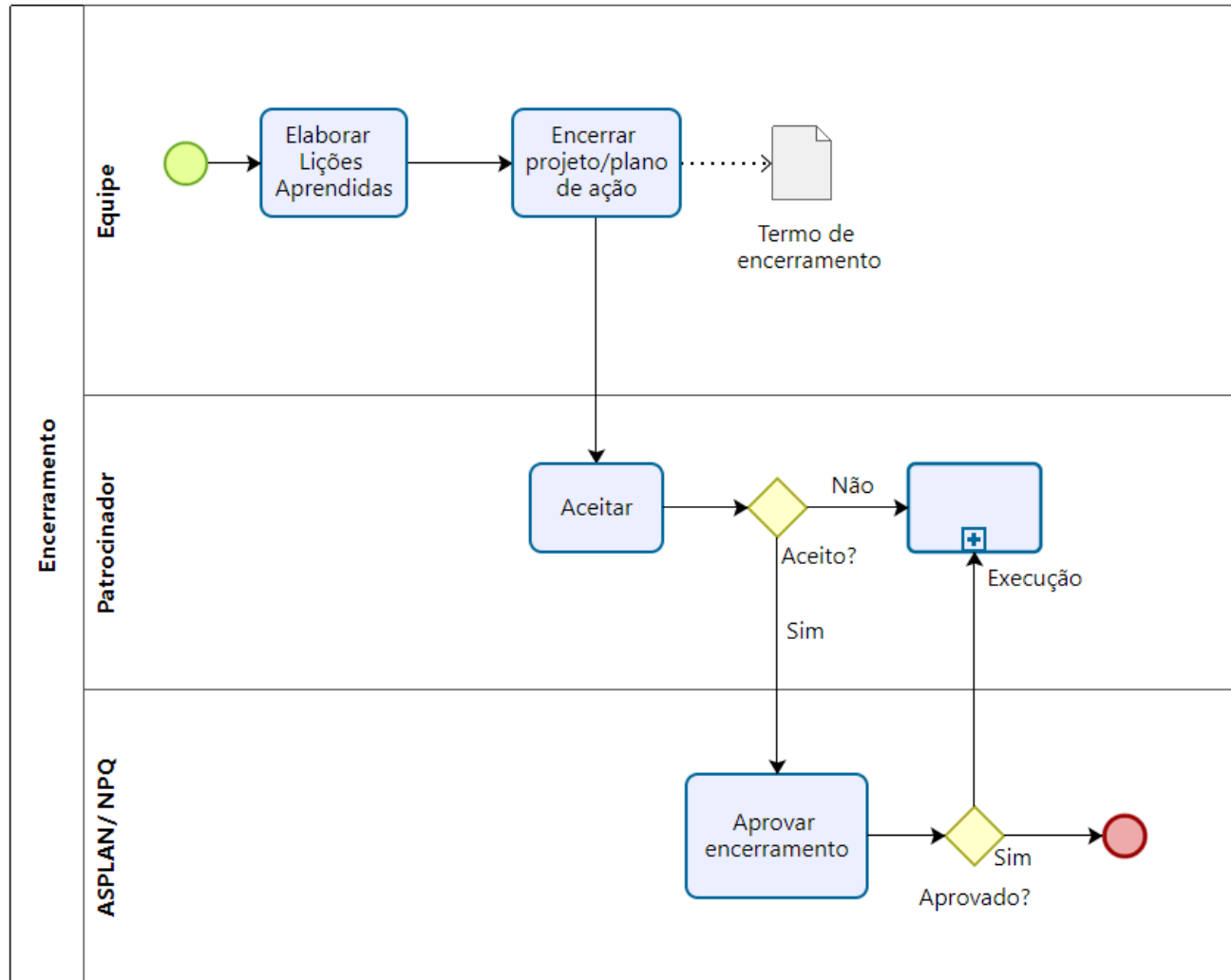
Responsável: Líder do Projeto / Equipe

4.4.11 Revisar Marcos do Projeto/Plano de ação

Esta atividade revisa se o marco foi alcançado com sucesso antes de seguir para a próxima fase ou para o fim do projeto/plano de ação.

Responsável: Líder do Projeto / Equipe

4.5 Subprocesso Encerramento



O encerramento formaliza a conclusão do projeto/plano de ação. Nesta fase, todos os trabalhos são finalizados, os objetivos alcançados e os entregáveis aceitos pelo cliente ou patrocinador.

4.5.1 Elaborar Lições Aprendidas

A equipe documenta as lições aprendidas e elabora o Termo de Encerramento do Projeto.

Responsável: Equipe

4.5.2 Encerrar projeto/plano de ação

Realiza transferência de conhecimento para equipes de sustentação. Arquiva toda a documentação projeto de forma organizada.

Responsável: Equipe

4.5.3 Validar Critérios de Aceite

O Patrocinador executa a tarefa 'Aceitar', confirmando a satisfação com o resultado. Caso aprovado, a ASPLAN/NPQ assume a governança do processo para realizar a aprovação final do encerramento. Se não aprovado, o fluxo é imediatamente enviado para o subprocesso de "Execução" e a equipe deve retomar o trabalho para corrigir as falhas apontadas, ajustar os entregáveis ou sanar as dúvidas que impediram o aceite.

Responsável: Patrocinador

4.5.4 Aprovar encerramento

Realiza homologação administrativa final, verifica conformidade com normas e regulamentações, processa finalização formal no sistema.

Responsável: ASPLAN/NPQ

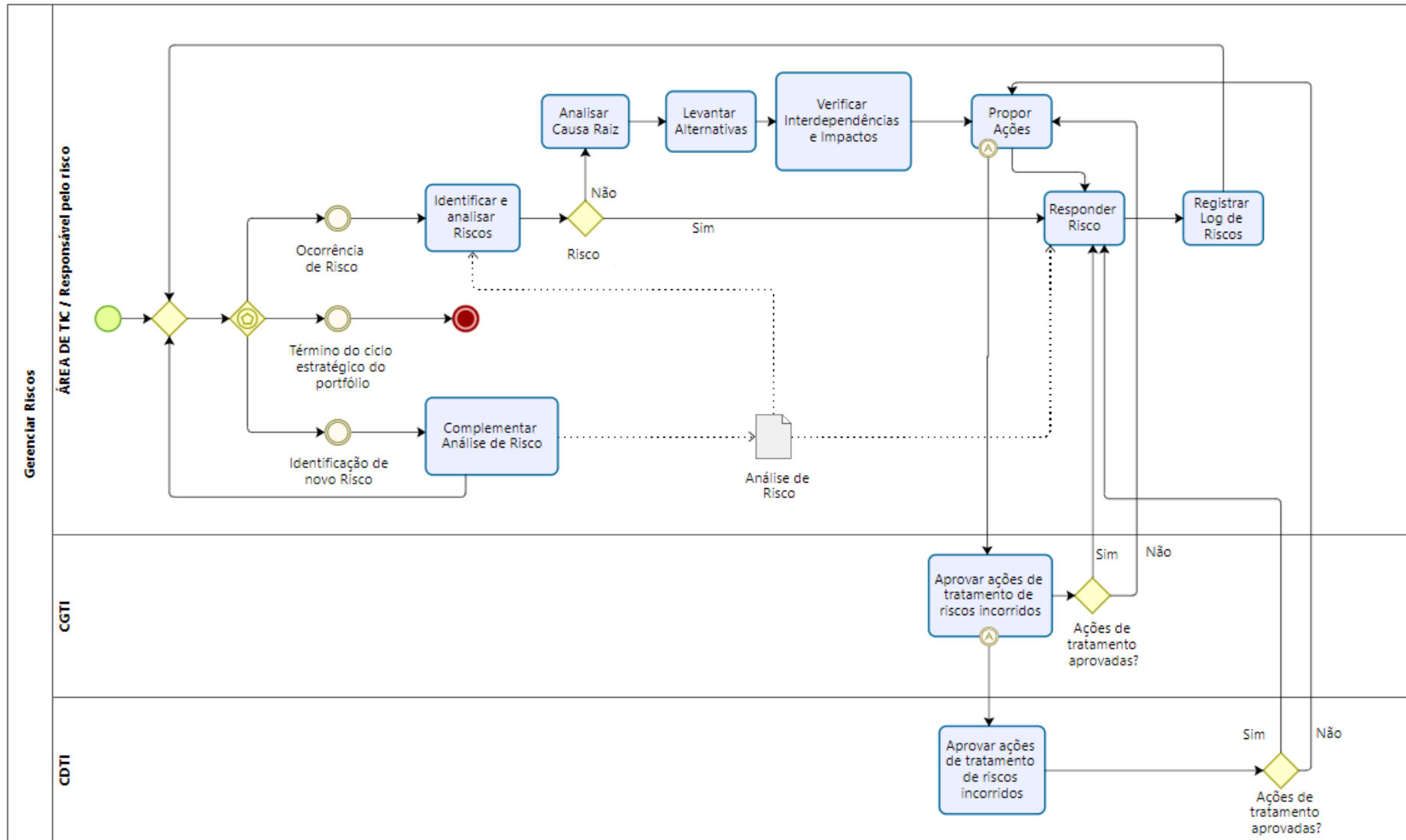
4.5.5 Aprovado? (Decisão Final)

Neste ponto de controle final:

- Sim: O projeto atinge o evento de término (Fim).
- Não: Caso haja pendências documentais ou normativas graves, o fluxo retorna para o subprocesso de Execução para regularização.

Responsável: ASPLAN/NPQ

4.6. Subprocesso Gerenciar Riscos



O subprocesso Gerenciar Riscos, consiste no acompanhamento contínuo e periódico dos riscos já identificados e das respostas implementadas. O objetivo é verificar se as respostas estão sendo eficazes (mitigando o risco conforme o esperado), identificar se novos riscos surgiram em decorrência das mudanças no ambiente tecnológico ou normativo do Tribunal, e observar se o perfil de probabilidade e impacto de riscos antigos se alterou.

Ele atua como um "termômetro", gerando alertas para a área de TI caso um risco ultrapasse a linha de apetite a risco definida pelo CGTI/CDTI, exigindo uma nova rodada de planejamento ou execução de contingências.

4.6.1 Identificar e analisar Riscos

Realizar o levantamento de possíveis eventos que possam impactar os objetivos do portfólio ou da área de TI. Nesta etapa, deve-se avaliar a probabilidade de ocorrência e o impacto potencial, classificando o nível de criticidade do risco.

Responsável: ÁREA DE TIC / Responsável pelo risco.

4.6.2 Complementar Análise de Risco

Atividade executada quando um novo risco é identificado ou quando uma revisão é necessária após o término do ciclo estratégico. Consiste em detalhar as características do risco e atualizar o artefato "Análise de Risco".

Responsável: ÁREA DE TIC / Responsável pelo risco.

4.6.3 Analisar Causa Raiz

Caso o risco identificado ainda não esteja mapeado, deve-se realizar um estudo aprofundado para entender as origens e os motivos que podem levar à concretização desse risco (ex: técnica de 5 porquês ou Diagrama de Ishikawa).

Responsável: ÁREA DE TIC / Responsável pelo risco.

4.6.4 Levantar Alternativas

Propor diferentes formas de lidar com o risco (evitar, transferir, mitigar ou aceitar), avaliando a viabilidade técnica e financeira de cada opção encontrada.

Responsável: ÁREA DE TIC / Responsável pelo risco.

4.6.5 Verificar Interdependências e Impactos

Avaliar como as alternativas propostas afetam outros processos, projetos ou áreas do TRE-PI. Garante que a solução para um risco não crie problemas em outras frentes de trabalho.

Responsável: ÁREA DE TIC / Responsável pelo risco

4.6.6 Propor Ações

Formalizar o plano de ação que será adotado para tratar o risco. Esta atividade gera uma proposta que aguardará a devida autorização instâncias superiores.

Responsável: ÁREA DE TIC / Responsável pelo risco.

4.6.7 Responder Risco

Executar as ações de resposta planejadas conforme a estratégia definida (seja para um risco já mapeado ou para uma nova proposição aprovada).

Responsável: ÁREA DE TIC / Responsável pelo risco.

4.6.8 Registrar Log de Riscos

Responsável: ÁREA DE TIC / Responsável pelo risco.

Documentar formalmente todas as ocorrências, análises e respostas dadas ao risco no registro histórico (Log de Riscos), garantindo a rastreabilidade e memória organizacional.

4.6.9 Aprovar ações de tratamento de riscos incorridos

Analisar as ações propostas pela área técnica sob a ótica da Governança de TIC, verificando o alinhamento com as diretrizes do Comitê de Gestão de Tecnologia da Informação.

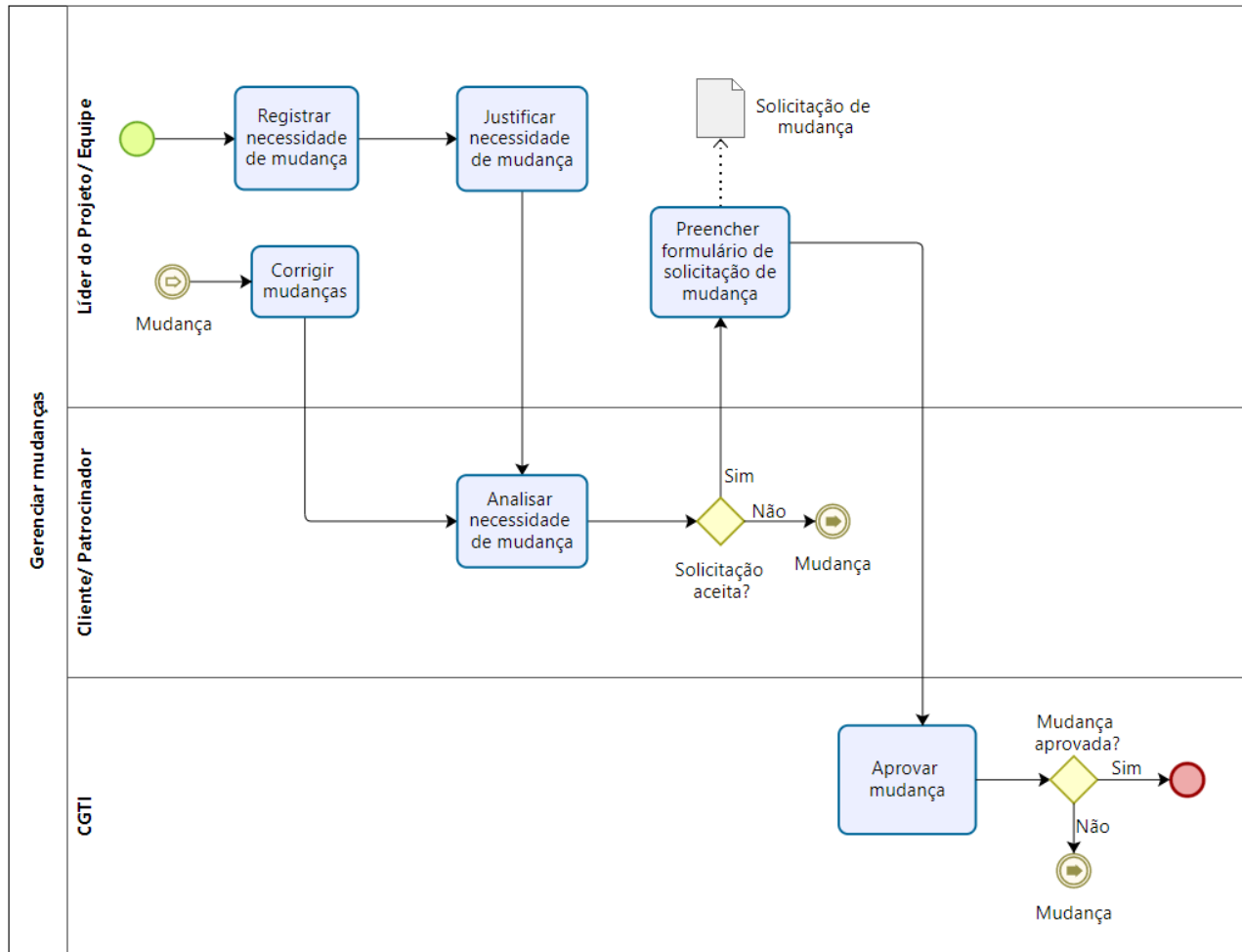
Responsável: CGTI (Comitê de Gestão de Tecnologia da Informação).

4.6.10 Aprovar ações de tratamento de riscos incorridos

Analisar as ações propostas pela área técnica sob a ótica da Governança de TIC, verificando o alinhamento com as diretrizes do Comitê Diretivo de Tecnologia da Informação.

Responsável: CDTI (Comitê Diretivo de Tecnologia da Informação).

4.7 Subprocesso de Gerenciamento de Mudança



Este subprocesso integra o fluxo de Monitoramento e Controle, sendo por ele acionado sempre que uma solicitação de mudanças é identificada. Uma vez a mudança aprovada pelas instâncias competentes, este subprocesso é formalmente encerrado

Após o encerramento, o fluxo de execução retorna automaticamente às demais atividades do processo principal de Monitoramento e Controle, garantindo a continuidade da gestão do projeto.

4.7.1 Corrigir Mudanças

Após a aprovação da mudança, o subprocesso é formalmente encerrado, devolvendo o fluxo às atividades subsequentes do processo de Monitoramento e Controle

Responsável: Líder do Projeto/ Equipe

4.7.2 Registrar necessidade de mudança

O processo é iniciado pelo Líder do Projeto/Equipe, que formaliza a intenção de alteração.

Responsável: Líder do Projeto/Equipe

4.7.3 Justificar necessidade Mudança

O líder do projeto/equipe realiza aceite inicial, avalia viabilidade, verifica alinhamento com objetivos estratégicos e descreve o motivo e a base para alteração.

Responsável: Líder do Projeto/ Equipe

4.7.3 Analisar necessidade de mudança

O cliente/patrocinador preenche o formulário de "Solicitação de Mudança", documenta impacto técnico e financeiro, analisa dependências.

Responsável: Cliente/Patrocinador

4.7.4 Preencher formulário de necessidade de mudança

Com o aval do Cliente/Patrocinador, a equipe formaliza a documentação técnica da mudança e submete ao CGTI.

Responsável: Líder do Projeto/Equipe

4.7.5 Aprovar mudança

O CGTI realiza a análise de conformidade. Se aprovada, segue para a implementação; caso contrário, o fluxo é retornado para ajustes necessários.

Responsável: CGTI

5. CONCLUSÃO

A Metodologia de Gestão de Projetos de Tecnologia da Informação (MGP-TI) consolida-se como um instrumento fundamental de governança no âmbito do Tribunal Regional Eleitoral do Piauí, ao estabelecer diretrizes, processos e responsabilidades claras para a condução dos projetos de TI. Sua estrutura contribui diretamente para a padronização das práticas institucionais, promovendo maior controle, transparência e eficiência na utilização dos recursos públicos.

Fundamentada em boas práticas reconhecidas de gerenciamento de projetos, a MGP-TI proporciona um modelo consistente e adaptável às diferentes naturezas e complexidades das iniciativas, assegurando que cada projeto seja devidamente planejado, executado, monitorado e encerrado com qualidade e conformidade institucional.

A metodologia fortalece, ainda, a cultura de gestão orientada a resultados, ao priorizar o alinhamento estratégico das iniciativas de TI com os objetivos institucionais do Tribunal. Nesse contexto, promove maior previsibilidade na execução dos projetos, aprimora a gestão de riscos e assegura que as entregas estejam direcionadas à geração de valor público e à melhoria contínua dos serviços prestados.

Adicionalmente, a MGP-TI amplia a integração entre as unidades envolvidas, estabelecendo um fluxo estruturado de governança que favorece a tomada de decisões, a responsabilização dos atores e o acompanhamento sistemático das iniciativas.

Assim, mais do que um instrumento técnico, a MGP-TI contribui para o fortalecimento da gestão institucional, a eficiência operacional e a prestação de serviços públicos com qualidade, integridade e responsabilidade.



STI
Secretaria da Tecnologia
da Informação

MANUAL DO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE



TRIBUNAL REGIONAL ELEITORAL DO PIAUÍ

Praça Desembargador Edgar Nogueira s/nº, Centro Cívico CEP 64000-830

Teresina – Piauí

Telefone: (86) 2107-9700 E-mail:

sti@tre-pi.jus.br

Elaboração:

Paulo das Neves e Silva Júnior

Revisão e Alteração para versão 5.0:

CODIN

Titular: Rosemberg Maia Gomes

SEDESC – Seção de Desenvolvimento de Soluções Corporativas Titular: Paulo das Neves e Silva Júnior



STI

Secretaria da Tecnologia da Informação

Validação:

Comitê Gestor de Tecnologia da Informação

Aprovação:

Comitê Gestor de Tecnologia da Informação

Editoração eletrônica e capa:

Breno Ponte de Brito – SECOM

Ficha catalográfica:

Jovita Maria Gomes Oliveira – SEJUB

Disponível também em: <<http://www.tre-pi.jus.br>>

Qualquer parte desta publicação pode

Dados Internacionais de Catalogação
na Publicação

Tribunal Regional Eleitoral do Piauí -
Biblioteca Des. Cristino Castelo Branco

Brasil. Tribunal Regional Eleitoral (PI).

Manual do processo de desenvolvimento de software [recurso eletrônico] / Tribunal Regional Eleitoral do Piauí. - Dados eletrônicos (62 páginas). - Teresina: Tribunal Regional Eleitoral do Piauí, 2026.

Elaboração: Seção de Desenvolvimento de Soluções Corporativas (SEDESC / CODIN / STI).

Revisão e alteração para versão 5.0: Coordenadoria de Desenvolvimento e Infraestrutura (CODIN / STI).

Versão eletrônica (PDF)

Modo de acesso: internet

<<https://www.tre-pi.jus.br/>>

1. Desenvolvimento de sistemas – Manual. I. Brasil. Tribunal Regional Eleitoral (PI). Secretaria de Tecnologia da Informação. II. Título.

SUMÁRIO

Apresentação

1. Introdução

2. Ciclo de Vida do Sistema

3. Processo de Concepção do Sistema

3.1 Introdução

3.2 Solicitar o Desenvolvimento do Sistema

3.3 Realizar Análise Preliminar

3.4 Priorizar o Desenvolvimento

3.5 Definir a Arquitetura

4. Processo de Desenvolvimento de Sistemas

5. Sustentação do Sistema

6. Declínio do Sistema

7. Escopo e Requisitos do Sistema

8. Diretrizes para o Desenvolvimento de Software

9. Conclusão



Apresentação

A Tecnologia da Informação (TI) é uma das grandes forças motrizes da administração do Tribunal Regional Eleitoral do Piauí (TRE-PI). Em constante busca pela excelência na prestação de seus serviços, as unidades administrativas do órgão demandam continuamente soluções de TI como suporte fundamental para a execução de suas atividades.

No contexto específico de desenvolvimento de software, o Tribunal de Contas da União (TCU) recomenda, no âmbito da Governança de TI, a institucionalização de uma metodologia que defina padrões claros para a criação de soluções corporativas. A adoção de um processo estruturado fortalece diretamente a missão da Secretaria de Tecnologia da Informação (STI) de “prover e manter soluções de Tecnologia da Informação para cumprimento da missão institucional”.

Nessa conjuntura, a gestão das demandas por soluções de TI já se encontra consolidada por meio da Resolução TRE-PI nº 320/2015, que regulamenta o Plano de Acompanhamento de Desenvolvimento de Sistemas (PADS). Todas as solicitações das unidades são submetidas ao Comitê Diretivo de Tecnologia da Informação (CDTI), responsável por priorizá-las de acordo com critérios objetivos previstos na resolução, com destaque para o alinhamento à estratégia institucional.

Acompanhando o amadurecimento contínuo dos processos da STI, o Manual de Desenvolvimento de Software evolui agora com a implementação de uma metodologia ágil híbrida, integrando as melhores práticas do Scrum e do Kanban (modelo conhecido como Scrumban). Essa atualização busca combinar a flexibilidade e o fluxo contínuo de entregas característicos do Kanban com a estrutura de planejamento e a cadência de cerimônias do Scrum, proporcionando maior previsibilidade, eficiência e alinhamento estratégico no desenvolvimento de software do Tribunal. Essa evolução metodológica também fortalece a padronização dos processos de desenvolvimento, aprimora a governança de TI e contribui para a entrega de soluções tecnológicas mais seguras, eficientes e alinhadas às necessidades institucionais.



1. Introdução

Tanto a elaboração como a revisão deste documento foi realizada pela **Secretaria de Tecnologia da Informação(STI)**, através da **Seção de Desenvolvimento de Soluções Corporativas (SEDESC)** e da **Coordenadoria de Desenvolvimento e Infraestrutura(CODIN)**, observando as melhores práticas de engenharia de software utilizadas no mercado e também instituídas no **Tribunal Superior Eleitoral e outros Tribunais Regionais Eleitorais**, no caso, a **Metodologia Ágil, padrões de segurança no desenvolvimento de sistemas**, integração de equipes, aliadas à experiência da equipe de desenvolvimento do TRE-PI.

Conforme apresentado na figura 02, neste processo serão detalhadas todas as etapas do ciclo de vida de uma solução corporativa. Desde o momento em que a mesma foi escalonada para atendimento, passando pelo processo de planejamento e desenvolvimento, gestão e, por fim, o momento em que não seja mais necessária sua utilização. São delineados e descritos os fluxos dos processos e subprocessos executados, os papéis e responsabilidades dos envolvidos, os recursos necessários e os resultados obtidos em cada uma das etapas.

Considera-se “Desenvolvimento” como sendo, não necessariamente, uma nova implementação, mas qualquer esforço que a equipe de desenvolvimento execute para atender a solicitação de uma unidade administrativa do TRE/PI, seja através de **estudos de**

viabilidade, análises de impactos, codificação, elaboração de documentos e realização de reuniões. Como “Solução Corporativa”, considera-se um **novo software**, a **adaptação ou melhoria de um software já existente e implantado no TRE/PI** ou a **implantação de software de terceiros**.

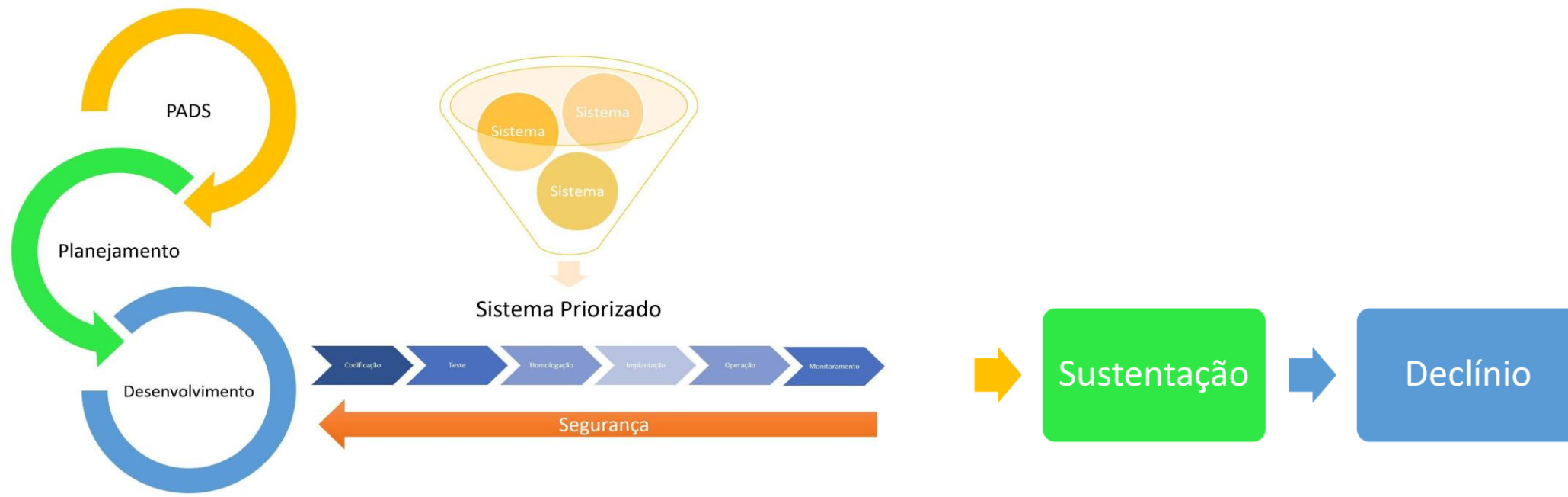


Figura 01 – Ciclo de vida da solução corporativa

2. Ciclo de vida do sistema

O ciclo de vida de um sistema corresponde às etapas pelas quais um software passa desde a sua idealização até entrar em desuso, que são: **Concepção**, **Desenvolvimento**, **Sustentação** e **Declínio**. Cada uma dessas etapas possui processos e atividades bem definidos. O fluxo apresentado na Figura 03 demonstra o macroprocesso das principais fases do ciclo de vida de um sistema.

A **etapa de concepção** corresponde ao nascimento do sistema e está descrita no capítulo 3 deste manual. Nessa fase tem-se a criação e detalhamento do escopo e requisitos da solução que deverá ser desenvolvida. Em outras palavras, é a fase em que **a ideia do cliente é traduzida em uma série de requisitos de software documentados e aptos a serem desenvolvidos**.

A **fase de desenvolvimento** refere-se à codificação, testes e implantação do sistema, processos detalhados no capítulo 4. Ainda no capítulo 4, é definida a Metodologia de Desenvolvimento de Software, aos moldes do Kanban.

A **fase de sustentação**, também conhecida como maturidade, é detalhada no capítulo 5. Refere-se à utilização plena do sistema após sua implantação em ambiente de produção e respectivas manutenções corretivas e evolutivas.

O **declínio**, definido no capítulo 6, corresponde ao momento em que o sistema passa a não mais atender o seu propósito, seja por obsolescência tecnológica, imposições legais ou decisões estratégicas de negócio. O declínio pode ter como resultado a simples desativação do sistema ou o início de um novo processo de desenvolvimento de sistemas para substituição por uma nova solução.



Figura 02 – Visão Geral do Ciclo de Vida de um sistema no TRE-PI

3. O Processo de Concepção do Sistema

3.1 Introdução

O Tribunal Regional Eleitoral do Piauí (TRE-PI) vivencia uma demanda crescente e contínua por soluções tecnológicas e desenvolvimento de sistemas que otimizem as rotinas e facilitem os processos de trabalho de suas diversas unidades administrativas e judiciárias. A Secretaria de Tecnologia da Informação (STI) atua como o setor estratégico responsável por receber e atender a essas necessidades. Contudo, diante da limitação natural de recursos e da pluralidade de requisições, torna-se imprescindível a adoção de um mecanismo formal e objetivo que permita elencar prioridades, garantindo que a execução dos projetos gere o maior valor possível para a instituição.

Para suprir essa necessidade de governança e organização, o TRE-PI instituiu o Plano de Acompanhamento de Desenvolvimento de Sistemas (PADS), devidamente regulamentado pela Resolução TRE nº 320/2015. Este normativo estabelece, no âmbito da Justiça Eleitoral do Piauí, diretrizes claras e critérios metodológicos não apenas para a apresentação formal de novas solicitações, mas, fundamentalmente, para a priorização do desenvolvimento

de soluções corporativas, alinhando as iniciativas de TI ao planejamento estratégico do Tribunal.

Do ponto de vista operacional, o ciclo de vida dessas soluções tem início no **processo de concepção de sistemas**. Esta etapa é estruturada em um fluxo contínuo que garante a análise adequada de cada pedido, ocorrendo da seguinte forma:

- **Solicitar Desenvolvimento:** Ponto de entrada onde as demandas das unidades são captadas e formalizadas.
- **Análise Preliminar:** Avaliação inicial que pondera a viabilidade técnica, a complexidade e o impacto institucional da solução proposta.
- **Priorizar Desenvolvimento:** Etapa em que os critérios do PADS são rigorosamente aplicados para definir a posição do projeto na fila de execução da STI.



Ainda na fase de concepção, ocorrem as definições estruturais críticas para o sucesso do projeto. É neste momento que a equipe técnica estabelece a **arquitetura do sistema** e executa as atividades iniciais de **gerenciamento de requisitos e escopo**, mitigando riscos e estabelecendo alicerces sólidos para as etapas de desenvolvimento subsequentes.

Para uma melhor compreensão visual desta esteira de trabalho institucional, a figura a seguir ilustra detalhadamente o fluxo de atividades e os subprocessos que compõem a fase de concepção de um sistema.

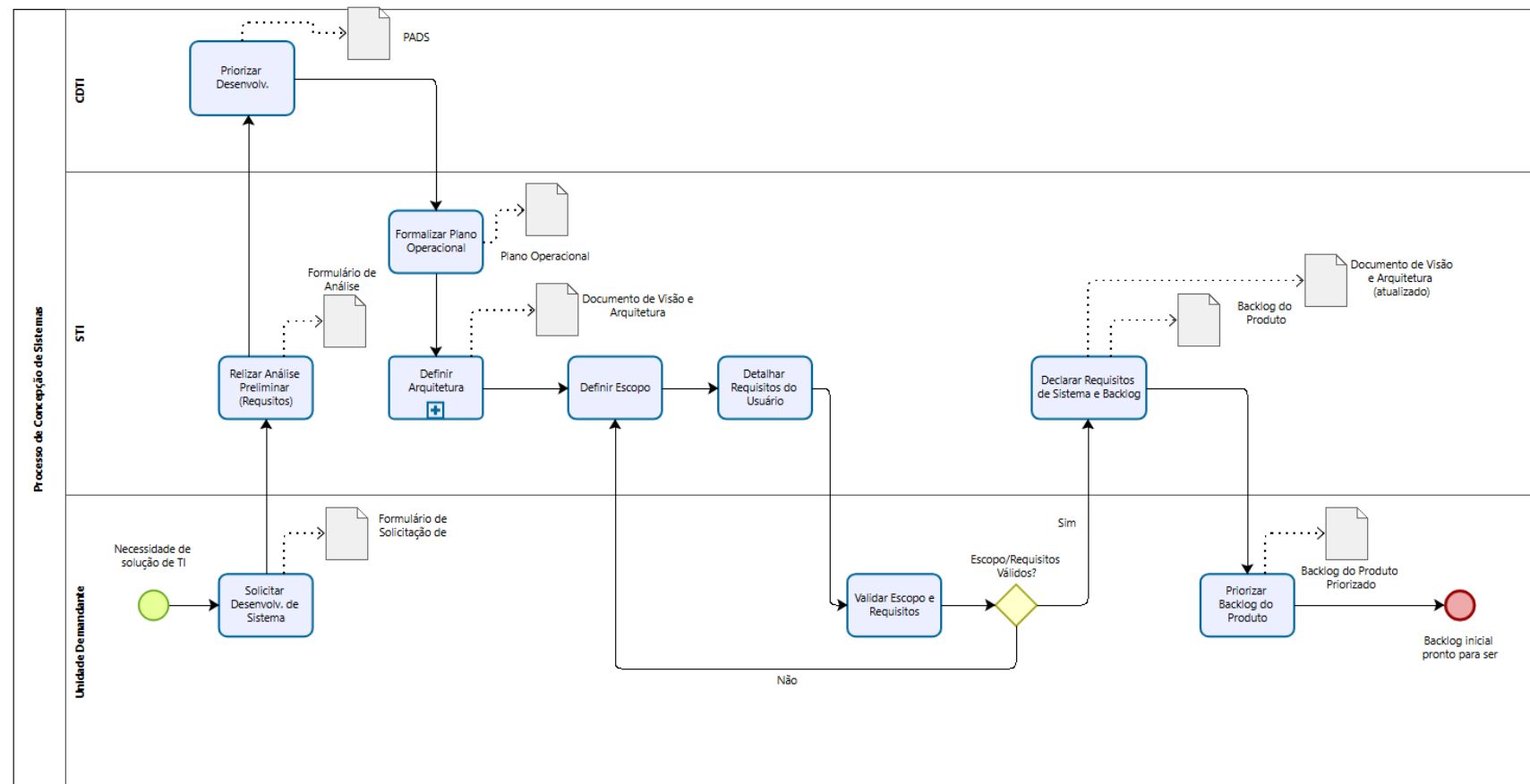


Figura 03 – Processo de Concepção de Sistemas

3.2 Solicitar o Desenvolvimento do Sistema

Representa a solicitação formal à Secretaria de Tecnologia da Informação – STI de uma demanda referente ao desenvolvimento, à evolução de um sistema ou a implantação de um sistema de terceiros, através do Formulário de Solicitação de Sistemas, cujo modelo é definido pela resolução do PADS. Nessa atividade, a unidade demandante descreve de forma sucinta as **informações iniciais** e essenciais a respeito do sistema que está sendo proposto, justificando a **sua necessidade**, indicando **os objetivos da estratégia institucional** que serão atendidos e **o gestor responsável** pelos procedimentos que serão informatizados.

O pedido é feito através do sistema eletrônico de tramitação dos processos administrativos utilizado no TRE/PI e obedece a um prazo estabelecido na Resolução do PADS. Essas informações são fundamentais para a realização da etapa seguinte, onde é executada a **Análise Preliminar**.

3.3 Realizar Análise Preliminar

Durante essa etapa, pretende-se obter de forma mais detalhada um entendimento da demanda. Ao receber a solicitação formalizada, o gestor responsável pelo desenvolvimento de sistemas escolhe um técnico ou equipe para coletar, através de entrevistas, mais informações junto ao gestor indicado no Formulário de Solicitação de Sistemas pela unidade demandante.

O técnico responsável por essa análise, após ter uma visão clara do sistema, expõe no Formulário de Análise Preliminar, cujo modelo é definido pela resolução do PADS, as **partes interessadas, escopo e requisitos iniciais**, além de uma **estimativa de prazos**.

As análises preliminares servirão de base para que o Comitê Diretivo de Tecnologia da Informação – CDTI determine uma **priorização de desenvolvimento de sistemas**, etapa descrita a seguir, de acordo com os critérios previstos na Resolução do PADS.

3.4 Priorizar o Desenvolvimento

Corresponde à fase de priorização das demandas de acordo com os critérios estabelecidos na resolução do PADS. Para isso, os membros do Comitê Diretivo de Tecnologia da Informação-CDTI se reúnem ordinariamente, em períodos definidos pela resolução do PADS, discutindo as solicitações feitas através dos formulários de análise preliminar.

O resultado dessa priorização constituirá uma ordem a qual a unidade responsável pelo desenvolvimento obedecerá para escalonar a sequência de atendimento. O **PADS será apreciado pela Presidência do TRE/PI para aprovação através de portaria**.

As demandas não previstas para a última reunião ordinária do CDTI serão solicitadas utilizando o mesmo Formulário de Solicitação de Sistemas citado no item 2.2.1. A análise preliminar será realizada e apreciada na próxima reunião ordinária do CDTI. Para analisar as solicitações urgentes, poderá ser convocada uma reunião extraordinária por determinação da Presidência do TRE/PI.

3.5 Formalizar Plano Operacional

Cada um dos Sistemas componentes do PADS será considerado um Projeto. Logo, a STI, na figura do Gestor de Desenvolvimento, irá formalizar um Plano Operacional, o qual seguirá os trâmites previstos em normas internas.

3.6 Definir a Arquitetura

A arquitetura do sistema é **o conjunto de soluções tecnológicas** que, estabelecido de maneira integrada e coordenada, **provê os recursos necessários para que o software desempenhe seu propósito**. A definição dessa arquitetura envolve a análise, caso a caso, do tipo de solução a ser desenvolvida, requisitos funcionais e não funcionais existentes, entre outras variáveis aplicáveis ao caso concreto.

Essa definição envolve o time de desenvolvimento, que fará a **sugestão inicial**, e o departamento de desenvolvimento, que terá decisão

final sobre a aprovação da arquitetura a ser utilizada. O resultado desse processo é o **documento de declaração da visão e arquitetura**, que pode vir a ser atualizado conforme o projeto avança. O processo de definição da arquitetura é ilustrado no diagrama a seguir:

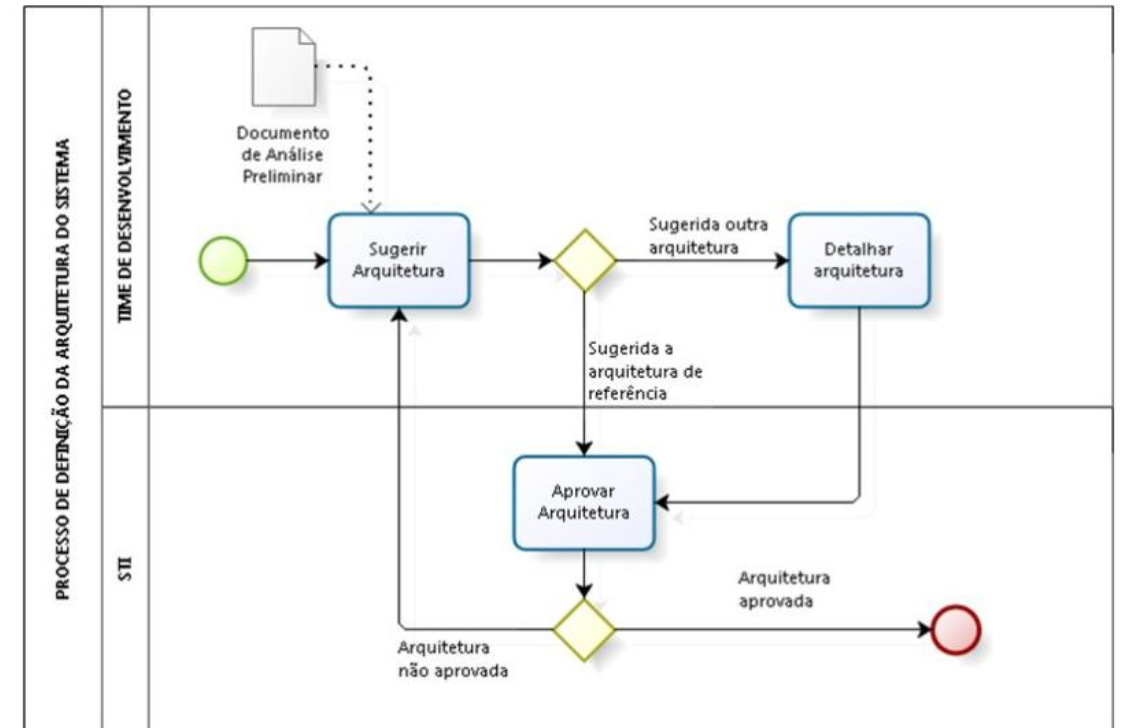


Figura 04 – Processo de Definição da Arquitetura

3.6.1 Sugerir a Arquitetura

O Time de Desenvolvimento, de posse do Documento de Análise Preliminar, fará um estudo da solução a ser implementada e fará a sugestão do modelo arquitetural a ser utilizado. Sempre que possível, o Time deverá optar por adotar o modelo arquitetural sugerido no item 3.5.4, de modo a favorecer a padronização e a maximização no uso dos conhecimentos já consolidados pelo quadro de colaboradores da STI.

A sugestão da arquitetura a ser utilizada, deve ser encaminhada à chefia da seção de desenvolvimento de software.

3.6.2 Detalhar a Arquitetura

Esta atividade é ativada quando o Time de Desenvolvimento opta por sugerir o uso de um modelo arquitetural diverso do proposto no item 3.5.4. Nesses casos, deve ser preenchido o Formulário de Arquitetura Detalhado, e o mesmo deve ser submetido à aprovação da chefia da seção de desenvolvimento de software.

3.6.3 Aprovar a Arquitetura

Nesta atividade a chefia da seção de desenvolvimento de

software, juntamente com Núcleo de Cibersegurança, analisam o projeto a ser desenvolvido e manifesta-se sobre a adoção ou não do modelo arquitetural proposto pela equipe de desenvolvimento.

3.6.4 Arquitetura de Referência

Na versão 1.0 do Manual de Processo de Software do TRE-PI, a arquitetura padrão era baseada em um modelo de arquitetura monolítico, onde todos os elementos e requisitos do sistema a ser desenvolvido eram implementados e consolidados em um único aplicativo executável.

Apesar do uso de boas práticas de padrões de projeto e aliadas a tecnologias corporativas, como Java EE, o uso desse padrão arquitetural, ao longo do tempo, começou a apresentar desafios para a equipe de desenvolvimento, como por exemplo:

- Aumento da complexidade e tamanho do sistema
- Alta dependência de componentes de código
- Dificuldade de escalabilidade
- Pouca flexibilidade para sistemas mais complexos

- Dificuldade para colocar as alterações em produção

Para a versão 2.0 do MPS foi estabelecido pela equipe de desenvolvimento do TRE-PI o uso da arquitetura de microsserviços, e nesta versão (3.0) se mantém.

A arquitetura de microsserviços consiste em uma coleção de pequenos serviços autônomos. Cada serviço é independente e deve implementar uma única funcionalidade do negócio dentro de um contexto limitado. Um contexto limitado é uma divisão natural de uma regra de negócio e fornece um limite explícito dentro do qual um modelo de domínio existe.

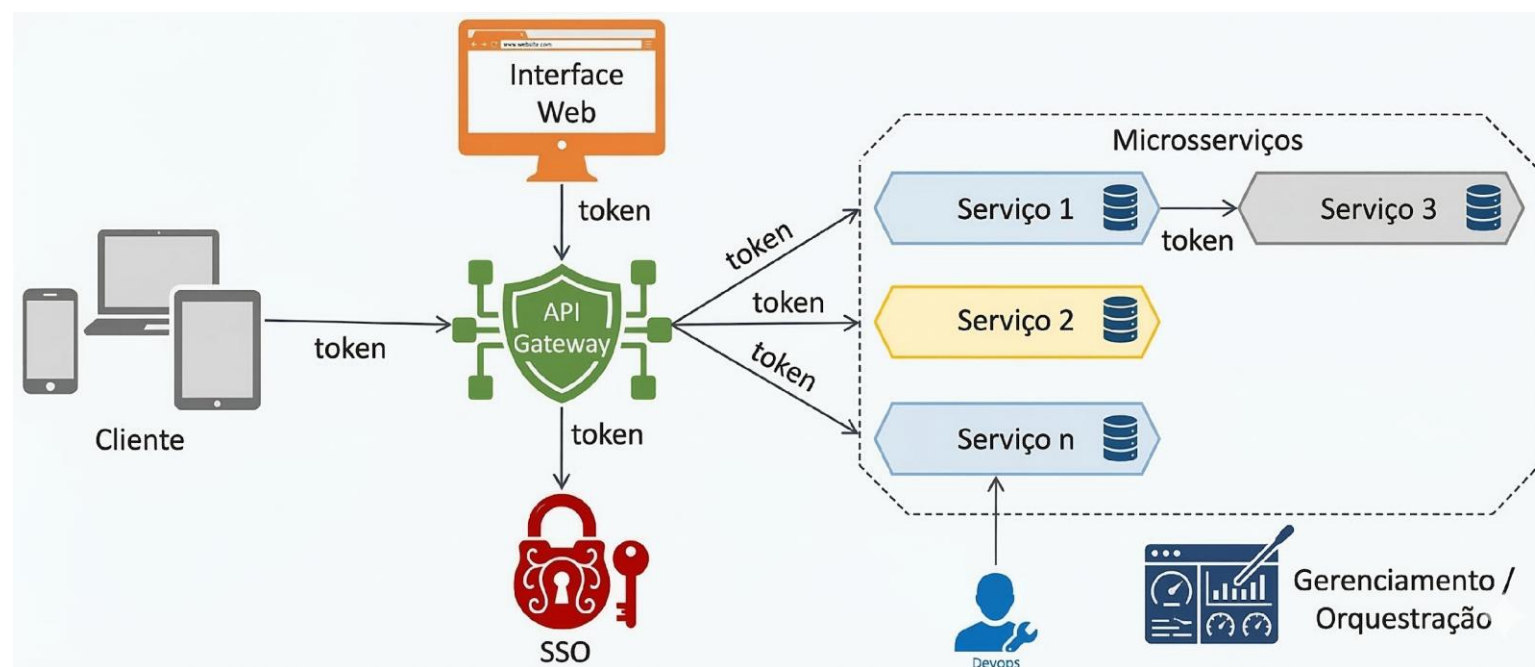


Figura 05 – Arquitetura de Referência

3.6.4.1 Microserviços

Os microserviços consistem em uma abordagem arquitetônica na qual um único aplicativo é composto de muitos componentes ou serviços menores que são implementáveis de forma independente e possuem as seguintes características:

- São pequenos, independentes e fracamente acoplados.
- Os serviços podem ser implantados de maneira independente.
- Cada serviço possui uma base de código em separado, sendo responsáveis por manter os seus próprios dados.
- Os serviços comunicam-se entre si por meio de APIs RESTful bem definidas.
- Podem ser implementados em tecnologias distintas, não precisando compartilhar a mesma pilha de tecnologias, bibliotecas ou estruturas.

3.6.4.2 Padrão API Gateway

O gateway de API é o ponto de entrada para os clientes. Em vez de chamar os serviços diretamente, os clientes chamam a API Gateway, que encaminha a chamada para os serviços adequados.

As vantagens de usar uma API Gateway incluem:

- Desacoplar os clientes dos serviços.
- Utilização de outros protocolos de mensagens e não somente o HTTP.
- Executar outras funções abrangentes, como autenticação, registro em log, terminação SSL e balanceamento de carga.

3.6.4.3 Gerenciamento / Orquestração

Esse componente é responsável por colocar serviços em nós—também conhecidos como “Containers”, identificar falhas, balancear a carga entre os serviços e assim por diante. Normalmente, esse componente é uma tecnologia pronta para uso, como Kubernetes, em vez de algo criado de maneira personalizada.

Com o gerenciamento e a orquestração dos microserviços, implantados em nós do tipo “Container”, pode-se automatizar e gerenciar tarefas como:

- Provisionamento e implantação
- Configuração da aplicação com base no container em que ela será executada
- Alocação de recursos

- Escalabilidade e disponibilidade dos containers
- Balanceamento de carga e roteamento de tráfego
- Monitoramento
- Proteção das interações entre os serviços

3.6.4.4 SSO—Single Sign-On

Single sign-on (SSO) é um esquema de autenticação que permite que um usuário efetue login com um único ID em qualquer um dos vários sistemas de software relacionados, mas independentes.

Em uma arquitetura de microsserviço, o cliente, seja um usuário ou um serviço, interage com uma coleção de outros serviços, cada um com uma necessidade potencial de saber quem é o usuário/cliente. Assim, um serviço centralizado de autenticação e autorização permite armazenar as credenciais usadas para a autenticação inicial e convertê-las nas credencias necessárias para os diversos serviços desenvolvidos ou em uso pelo TRE-PI.

Os benefícios do uso dessa abordagem incluem:

- Redução do risco de acesso por site de terceiros—autenticação federada
- Redução do excesso de entradas de senhas por meio da

combi- nação usuário/senha

- Administração simplificada de autenticação e autorização a re- cursos
- Melhor controle administrativo

No MPS versão 5.0, o SSO na arquitetura de referência irá adicionar uma camada extra de segurança a API Gateway, aumentando a visibilidade de autorização e roteamento e diminuindo a quantidade de API expostas, garantindo a diminuição das superfícies de ataques aos sistemas.

3.6.4.5 Interface Web

Essa é a camada de apresentação do sistema, ela apresenta a interface do usuário e de comunicação do aplicativo. Seu principal objetivo é exibir as informações e coletar os dados dos usuários. Essa camada será desenvolvida utilizando as tecnologias web, como html, css e Javascript.

3.6.4.6 DevSecOps

DevSecOps significa desenvolvimento, segurança e operações. É uma abordagem à cultura, automação e design da plataforma que integra segurança como uma responsabilidade compartilhada em todo o ciclo de vida de um produto.

O uso dessa abordagem implicará em uma evolução natural e necessária na forma como a equipe de desenvolvimento do TRE-PI abordará as questões relativas à segurança em todo o processo de desenvolvimento e manutenção de software.

3.6.4.6.1 Integração e Entrega Contínua (CI/CD)

Este procedimento, conhecido como **Pipeline de Integração Contínua, Entrega Contínua e/ou Implantação Contínua (CI/CD)**, é um dos pilares essenciais para a efetiva implementação da cultura e práticas **DevSecOps**. Consiste fundamentalmente em utilizar um conjunto robusto de ferramentas e automações para otimizar e proteger o ciclo de vida de desenvolvimento de software. O objetivo primordial é automatizar as etapas de **integração de código, construção da aplicação, execução de testes (funcionais, de segurança e de qualidade) e a implantação** das novas versões do software. Esta automação é projetada para ocorrer de forma frequente e com mínima intervenção humana, resultando em lançamentos mais rápidos, consistentes, seguros e confiáveis.

O processo automatizado, que integra a segurança desde as fases iniciais (Shift Left Security), geralmente se desenrola através dos seguintes passos interconectados:

Versionamento e Disparo do Pipeline

O desenvolvedor consolida suas alterações de código e as envia (através de um commit e push) para um **servidor de controle de versão centralizado**, como o GitLab.

Essa ação de enviar o código ao repositório principal ou a uma *branch* específica serve como o **gatilho (trigger)** que inicia automaticamente o pipeline de CI/CD.

Construção, Testes Automatizados e Análise de Segurança

Uma vez que o novo código é recebido pelo servidor de versionamento, uma **rotina automatizada (o pipeline de CI/CD)** entra em ação, executando uma sequência de tarefas críticas:

- **Build (Construção):** O código fonte é compilado e empacotado, gerando um artefato de software executável.
- **Testes Automatizados:** Diversos níveis de testes são executados, como testes unitários (para verificar componentes individuais), testes de integração (para verificar a interação entre componentes) e, possivelmente, testes de aceitação.
- **Análise Estática de Qualidade de Código:** Ferramentas analisam o código em busca de "code smells" (maus odores de

código), complexidade ciclomática elevada, duplicação de código e aderência a padrões de estilo.

- **Verificação Estática de Vulnerabilidades de Segurança (SAST):** Ferramentas de *Static Application Security Testing (SAST)*, como o Fortify, inspecionam o código-fonte, bytecode ou código binário em busca de padrões que indiquem vulnerabilidades de segurança conhecidas (ex: SQL Injection, Cross-Site Scripting, etc.) antes mesmo da aplicação ser executada. Esta é uma etapa crucial do "Sec" em DevSecOps.

Decisão e Implantação Automatizada (Entrega ou Implantação Contínua):

Após a execução de todas as verificações anteriores:

- **Validação:** O pipeline avalia os resultados. Se o código passar em todos os testes (funcionais e de qualidade) e, crucialmente, não apresentar vulnerabilidades de segurança críticas ou que violem as políticas estabelecidas, ele é considerado apto para a próxima fase.
- **Implantação:** Conforme os parâmetros configurados no pipeline (que

podem variar de acordo com a *branch*, *tags* ou gatilhos manuais), o artefato de software aprovado é automaticamente implantado. Isso pode ocorrer em diferentes ambientes:

- **Ambiente de Desenvolvimento:** Para testes exploratórios e integração

contínua com outras funcionalidades em desenvolvimento.

- **Ambiente de Homologação (Staging/QA):** Uma réplica do ambiente de produção, onde são realizados testes de aceitação do usuário (UAT), testes de carga, e validações finais antes da produção.
- **Ambiente de Produção:** Onde a aplicação fica disponível para os usuários finais. A implantação direta em produção após todos os checks caracteriza a **Implantação Contínua**. Se houver uma aprovação manual antes da produção, mesmo com o processo automatizado até este ponto, caracteriza-se como **Entrega Contínua**.



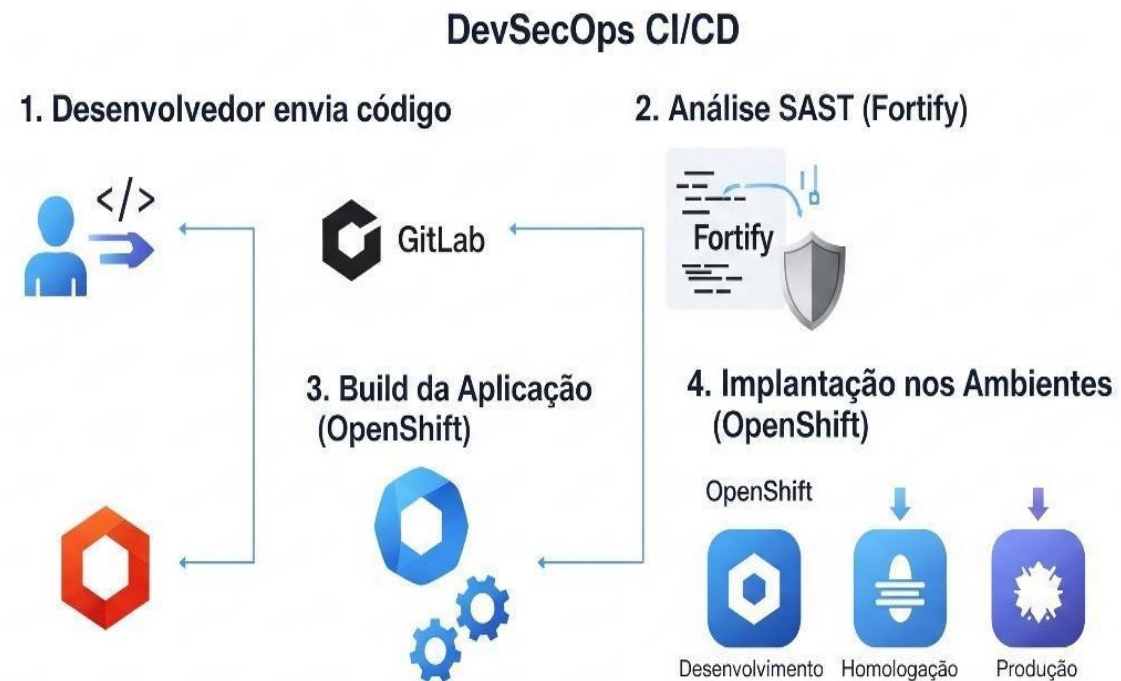


Figura 06—Ilustração do Processo CI/CD

3.6.4.6.2 Desenvolvimento Seguro de Software

O segurança no desenvolvimento de sistemas já é contemplado na cultura DEVSECOPS, sendo que o TRE-PI normatizou na Portaria Presidência nº 158/2023 uma série de procedimentos para atingir esse objetivo, considerando os seguintes tópicos:

3.6.4.6.3 Análise de Vulnerabilidades

Juntamente com a equipe de Segurança da Informação, a equipe de desenvolvimento deverá realizar o tratamento de vulnerabilidades do processo de software, tais como:

1. Recebimento de notificação de vulnerabilidade
2. Classificação do Risco da vulnerabilidade
3. Notificação da correção da vulnerabilidade
4. E análise da causa raiz da vulnerabilidade

3.6.4.6.4 Boas práticas recomendadas para proteção de dados pessoais

Algumas práticas devem ser adotadas, no processo de concepção e desenvolvimento das aplicações, obedecendo a padrões

estabelecidos na indústria de desenvolvimento de software, tais como:

I - **Privacy By Design**: assegura a proteção de dados pessoais deverá ser estabelecida desde a concepção do software ou componentes compreendendo todo o ciclo de vida, onde a equipe deverá realizar uma abordagem proativa na proteção de dados pessoais; e

II – **Privacy By Default**: o software deverá resguardar a exposição de dados pessoais salvaguardando a privacidade, sendo o mais restritivo possível tanto na exposição/visualização de dados pessoais quanto na coleta;

III—As vulnerabilidades que incidem sobre dados pessoais devem ser tratados com prioridade sobre qualquer outro tipo;

IV - Qualquer componente de terceiros que manuseie dados pessoais devem sofrer uma análise adicional, sendo inventariado e validado.

3.6.4.6.5 Acessibilidade e Usabilidade

O Conselho Nacional de Justiça (CNJ), alinhado às melhores práticas de inclusão, estabeleceu por meio da Resolução nº 401/2021 diretrizes

claras para a promoção da acessibilidade, determinando que os sistemas informatizados sejam desenvolvidos sob premissas rigorosas de acessibilidade e usabilidade.

Embora essa regulamentação seja direcionada precipuamente aos sistemas judiciais, é fundamental e altamente recomendável que essas premissas sejam adotadas como padrão na construção de todos os sistemas da instituição. A diversidade do público interno e externo exige interfaces inclusivas, preparadas para atender a usuários com diferentes necessidades e características físicas, sensoriais ou cognitivas.

Além da acessibilidade, a usabilidade é um pilar indispensável para o sucesso de qualquer aplicação. Desenvolver interfaces atrativas, intuitivas e fáceis de usar vai muito além da estética: trata-se de garantir a eficiência da ferramenta. Sistemas com alta usabilidade reduzem a curva de aprendizado, minimizam a ocorrência de erros operacionais e diminuem a sobrecarga cognitiva dos usuários. Quando o design é centrado no usuário, a tecnologia deixa de ser uma barreira e passa a atuar como um verdadeiro facilitador, promovendo maior produtividade, satisfação e engajamento no uso diário.

Assim, foi incluído, durante a etapa de desenvolvimento, a implementação das referidas recomendações, fazendo parte de análise posterior, ao final de cada sprint, de cada entregável.

3.7 Detalhar Requisito do Usuário

Esta atividade consiste no detalhamento dos requisitos inicialmente levantados junto ao usuário na atividade Realizar Análise Preliminar. Tem-se como objetivo elaborar uma lista de requisitos, funcionalidades que o sistema deve executar e/ou restrições sobre as quais deve operar.

3.8 Definir o Escopo

Esta atividade visa a definição dos limites do sistema. Sua execução ocorre de maneira paralela ao detalhamento dos requisitos devido à forte integração e interdependência dessas duas atividades. Alterações nos requisitos certamente terão impacto no escopo, e vice-versa. O propósito do sistema deve ficar claro na Declaração do escopo, que deverá ser explicitamente validada pela Unidade Demandante.

Essa tarefa complementa e elabora o escopo inicialmente levantado na tarefa Realizar Análise Preliminar, e pode ser ativada através de eventos das fases de Desenvolvimento e Sustentação.

3.9 Validar Escopo e Requisito

Esta atividade tem por objetivo obter, junto à unidade demandante, a confirmação de que o escopo e os requisitos apresentados de fato

refle- tem suas expectativas quanto à solução que deve ser implementada.

3.10 Declarar Requisitos do Sistema e Backlog

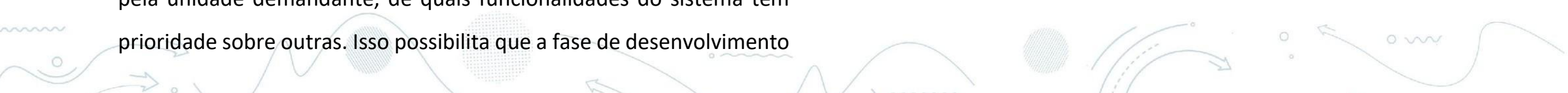
É nesta atividade que a unidade responsável pelo desenvolvimento da solução fará a tradução dos requisitos validados pelo usuário em requisitos de sistema. Ter-se-á ao fim dessa tradução o artefato chamado *Backlog* do Produto, que reúne todos os requisitos do sistema escritos em linguagem técnica, de modo a subsidiar o time de desenvolvimento com uma visão abrangente do trabalho a ser feito. Uma outra saída possível, é a atualização do Documento de Visão e Arquitetura, nos casos em que os detalhamentos e consecutivas validações no escopo e requisitos venham a justificar essa atualização.

Uma parte dos requisitos do backlog irá compor um documento descrevendo o **Produto Mínimo Viável**, o qual representará um entregável possível de ser usado pelo demandante em que será avaliado a viabilidade do projeto, com atendimento das necessidades prementes do Gestor do Sistema.

3.10 Priorizar Backlog do Produto

A priorização do *backlog* do produto consiste na definição, pela unidade demandante, de quais funcionalidades do sistema têm prioridade sobre outras. Isso possibilita que a fase de desenvolvimento

seja conduzida de modo a seguir essa prioridade, proporcionando a entrega artefatos de software tidos como mais importante num menor espaço de tempo, acelerando a geração de valor. Ao fim desta atividade, ter-se-á uma versão inicial do *backlog* do produto e a fase de desenvolvimento poderá ser iniciada tão logo o sistema em questão adquira prioridade no PADS.



3.11 Matriz de Responsabilidades

ATIVIDADES/PAPÉIS	GESTOR DE DESENVOLVIMENTO	GESTOR DO SISTEMA	ANALISTA DE REQUISITOS	EQUIPE
Solicitar desenvolvimento de Sistema	-	R/E	-	-
Realizar análise preliminar	R/E	I	I	I
Priorizar desenvolvimento	-	R/E	E	-
Formalizar Plano de Ação	R/E	I	-	-
Definir arquitetura	C/I	-	C/I	R/E
Detalhar requisitos de usuário	-	C/I	R/E	C/I
Definir escopo	-	R/E	E	-
Validar escopo e requisitos	-	R/E	E	-
Detalhar requisitos de sistema e backlog	R/E	C/I	C/I	C/I
Priorizar backlog do produto	-	R/E	E	-

Legenda: R – Responsável, E – Executor, C – Consultado, I – Informado.

Tabela 01 – Processo de Concepção de Sistemas

3.12 Recursos e Resultados

ATIVIDADES	RECURSOS	RESULTADOS
Solicitar desenvolvimento de Sistema	Modelo do Formulário de Solicitação de Sistema	Formulário de Solicitação de Sistema preenchido
Realizar análise preliminar	Formulário de Solicitação de Sistema preenchido	Formulário de análise preliminar
Priorizar desenvolvimento	Formulário de Análise Preliminar; Atas das reuniões com o gestor do sistema	PADS
Formalizar Plano de Operacional	Formulário de Análise Preliminar Atas de reuniões com o gestor do sistema	Plano Operacional
Definir arquitetura	Formulário de Análise Preliminar.	Documento de visão e arquitetura
Detalhar requisitos de usuário	Documento de Visão e Arquitetura do Produto.	Documento de Visão e Arquitetura do Produto (Atualizado).
Definir escopo	Documento de Visão e Arquitetura do Produto.	Documento de Visão e Arquitetura do Produto (Atualizado).
Validar escopo e requisitos	Documento de Visão e Arquitetura do Produto.	Validação do Documento de Visão e Arquitetura do Produto (Atualizado).
Declarar requisitos de sistema e backlog	Documento de Visão e Arquitetura do Produto.	Documento de Visão e Arquitetura do Produto (Atualizado);
Priorizar backlog do produto	Backlog do produto	Backlog do produto priorizado.

Tabela 02 – Recursos e Resultados do Processo de Concepção de Sistemas

4. Processo de Desenvolvimento de Sistemas

4.1 Introdução

O Tribunal Regional Eleitoral do Piauí apresenta uma grande demanda quanto ao desenvolvimento de sistemas que facilitem os processos de trabalho das suas unidades administrativas. A Secretaria de Tecnologia da Informação é a unidade responsável pelo atendimento dessas demandas, sendo necessária, no entanto, a definição de um mecanismo que permita objetivamente a priorização da ordem de realização dos projetos. Uma vez priorizados, cada um dos projetos deve ser conduzido através de um processo de desenvolvimento consolidado, que seja capaz de coordenar a evolução das etapas de trabalho e, ao mesmo tempo, ágil o suficiente para absorver as mudanças que se fazem necessárias ao longo da execução de todo projeto de software.

O TRE-PI instituiu o Plano de Acompanhamento de Desenvolvimento de Sistemas (PADS) por meio da Resolução TRE nº 320/2015, que disciplina no âmbito da Justiça Eleitoral do Piauí os critérios para apresentação e priorização de solicitação de desenvolvimento de soluções corporativas. Uma vez que um projeto de software recebe a prioridade necessária para que tenha seu desenvolvimento iniciado, esse trabalho deve ser realizado observando o

Processo de Desenvolvimento definido neste capítulo.

4.2 Metodologia de Desenvolvimento de Sistemas

O desenvolvimento de software de maneira indisciplinada eleva as chances de retrabalho, desperdício, insegurança e frustração para a unidade demandante de um sistema. Em razão disso, é imprescindível definir e seguir padrões preestabelecidos que norteiem o fluxo de construção de um programa, considerando as melhores práticas para modelagem, programação, métrica, arquitetura e testes.

A institucionalização de uma metodologia padronizada fortalece a missão da Secretaria de Tecnologia da Informação - STI de “Prover e manter soluções de Tecnologia da Informação para cumprimento da missão institucional”, além de atender as boas práticas de Governança de TI.

4.2.1 Metodologia Ágil

Entre as melhores práticas em termos de processo de desenvolvimento destaca-se a Metodologia de Desenvolvimento Ágil, cujos princípios foram relacionados em 2001, quando profissionais veteranos da área de

construção de software criaram o **Manifesto para o Desenvolvimento Ágil de Sistemas**, chamado apenas de **Manifesto Ágil**, o qual descrevia abordagens de desenvolvimento que passassem a valorizar:

Indivíduos e interação entre eles mais que processos e ferramentas;

Sistema em funcionamento mais que documentação abrangente;

Colaboração com o cliente mais que negociação de contratos;

Responder a mudanças mais que seguir um plano.

Observa-se que, mesmo havendo valor nos itens à direita, valoriza-se mais os itens à esquerda.

O Manifesto Ágil tem seus princípios baseados no *Extreme Programming (XP)*, uma metodologia de desenvolvimento de software, nascida nos Estados Unidos ao final da década de 90, a qual propõe criar sistemas de melhor qualidade, produzidos em menos tempo e de forma mais econômica que o habitual. Tais objetivos são alcançados através de um pequeno conjunto de **valores, princípios e práticas**, que diferem bastante da forma tradicional de desenvolver software.

O Manifesto Ágil recomenda ainda 12 princípios:

- 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 a 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.

4.2.2 Scrum + Kanban = Scrumban

O Scrumban é uma metodologia ágil que une a estrutura prescritiva do framework Scrum com o foco em melhoria contínua de processos do Kanban. Criado por Corey Ladas em 2008, o método busca extrair o melhor de ambas as ferramentas.

Aqui está um resumo dos principais pontos da metodologia apresentada na fonte:

Como funciona O Scrumban mantém os papéis e as cerimônias do Scrum, mas adota as seguintes características fundamentais do Kanban:

- Visualização do fluxo de trabalho por meio de cartões em um quadro.
- Limitação das tarefas em andamento para evitar sobrecarga.
- Adoção de um sistema "puxado" (just-in-time).

Principais Diferenças

- **Em relação ao Kanban:** Diferente do Kanban, que não prescreve papéis ou reuniões, o Scrumban exige a definição de responsabilidades para a equipe e a realização de reuniões diárias. Reuniões de revisão e retrospectiva também podem ser feitas para otimizar processos.
- **Em relação ao Scrum:** O Scrumban substitui as iterações fixas (Sprints) por um fluxo de trabalho contínuo. Novas demandas podem ser adicionadas conforme a necessidade, em vez de aguardarem a próxima Sprint. Além disso, o único artefato utilizado é o quadro visual, dispensando backlogs e burndowns complexos.

Vantagens

A adoção do Scrumban oferece métricas de eficiência e ajuda a reduzir a *lead time* (tempo médio de execução) e os desperdícios. Além de incentivar a melhoria contínua, o método proporciona maior autonomia para que os membros da equipe escolham e finalizem suas tarefas.

Quando utilizar O método é bastante indicado para:

- Projetos de manutenção ou de suporte (orientados a eventos).



- Desenvolvimento de novos produtos ou projetos com histórico de erros de programação.
- Equipes iniciantes em metodologias ágeis ou que enfrentam dificuldades para fechar Sprints devido à constante entrada de demandas de última hora.

O Quadro Scrumban O fluxo de trabalho é visualizado em um quadro geralmente dividido em colunas como "A fazer", "Em andamento", "Em teste" e "Concluído". É fundamental que a equipe estabeleça um limite de tarefas por coluna, garantindo que cada integrante realize apenas uma atividade de cada vez para otimizar a produtividade e a qualidade das entregas. Os status de coluna são exemplificativos, sendo que mais ou menos colunas podem ser criadas ao longo do tempo.

4.2.2.1 Termos Importantes

Alguns termos são necessários compreender para seguir com o entendimento deste manual:

Sprint: é um ciclo de tempo fixo (geralmente de 2 a 4 semanas) onde um incremento de produto "Pronto" (funcional) é criado;

Daily Scrum: Uma reunião de 15 minutos, diária, para os

desenvolvedores inspecionarem o progresso em direção à Meta da Sprint. É tempo dedicado a identificação de impedimentos e planejamento do trabalho para as próximas 24 horas.

Sprint Review (Revisão da Sprint): ocorre no final da Sprint para inspecionar o incremento gerado e adaptar o Product Backlog, se necessário. Tem por objetivo apresentar o trabalho concluído aos demandantes e receber feedback para decidir os próximos passos.

Sprint Restrospective (Retrospectiva da Sprint): é a última reunião da sprint, onde o time faz uma análise do processo de trabalho, objetivando identificar o que funcionou bem, o que pode ser melhorado e criar um plano de ação para aumentar a qualidade e eficácia da próxima Sprint.

Backlog: é uma visão nocional do local que mostra uma lista consolidada de itens de trabalho (Histórias de usuários/defeitos/problemas) que devem ser trabalhados. Pode-se pesquisar através do Backlog e adicionar cartões ao Software de Quadro Kanban.

Quadro Kanban: é uma das ferramentas para implementar o método Kanban. O quadro é dividido em, no mínimo, 3 colunas – Para

Fazer, Fazendo, Feito, representando as etapas de um processo.

Raia: são divisões horizontais de um quadro Kanban, que ajudam a otimizar o fluxo de trabalho. As colunas representam etapas e as raias categorizam o trabalho. Raias podem ser usadas para representar times, classes de serviço, prioridade, etc.

Cartão kanban: representam visualmente os itens de trabalho. Cada cartão é uma tarefa, que se move pelas colunas do quadro Kanban. Os cartões contêm informações sobre o item de trabalho. Eles não possuem diferença em tamanho, porque a ideia é dividir um projeto em pequenas tarefas que podem ser concluídas rapidamente.

WIP (Work in Progress): ou Trabalho em Progresso, é a quantidade de trabalho sendo iniciada, independentemente da sua subcoluna atual.

Limites de WIP: é uma estratégia para evitar a sobrecarga e a mudança de contexto enquanto focamos em coisas importantes. A aplicação de limites de WIP é a segunda prática central do Kanban e assegura um fluxo saudável.

Lead Time: é o tempo total que uma tarefa leva, desde a solicitação do demandante até a sua entrega final.

Cycle Time: é o tempo de execução ativa, medindo o tempo de execução

ativa, a partir da real execução da tarefa até a sua conclusão.

4.2.3 Objetivo do Manual do Processo de Software

O Manual do Processo de Software do TRE/PI (MPS-TRE/PI) visa, dentre outros fins, determinar padrões para o desenvolvimento de sistemas no âmbito deste Regional, seguindo e incorporando os parâmetros da metodologia ágil, definindo etapas, papéis e artefatos necessários para a correta realização dos processos de gerenciamento, desenvolvimento e manutenção de sistemas, bem como na implantação de soluções desenvolvidas por outros órgãos do Poder Judiciário. Visa-se portanto, prover à equipe técnica uma maior efetividade nas suas atividades e às unidades do TRE/PI maior satisfação quanto ao atendimento de soluções de seu interesse.

4.2.4 Papéis e Responsabilidades

Os papéis e respectivas responsabilidades dos envolvidos na fase de Desenvolvimento do MPS são descritos neste tópico.

4.2.4.1 Gestor de Desenvolvimento de Sistema

Responsável por gerenciar o processo de atendimento das demandas referentes ao desenvolvimento de sistemas no TRE/PI. São suas respon-

sabilidades:

- Garantir o uso das recomendações descritas nesta metodologia;
- Designar analistas de requisitos para realização de análises preliminares;
- Gerenciar a realização das Análises Preliminares;
- Acompanhar a execução dos cronogramas definidos;
- Reconhecer perfis profissionais para alocá-los nas atividades adequadas;
- Definir as equipes de desenvolvimento;
- Garantir a integração entre a equipe de desenvolvimento e o gestor do sistema.
- Controlar as atividades de mudanças de requisitos e a entrada de novos requisitos;
- Gerenciar os fatores de risco quanto ao atendimento de mudanças nos requisitos;
- Fazer a mediação entre o gestor do sistema e a equipe de desenvolvimento;
- Aceitar ou rejeitar o que for entregue nas *sprints*;
- Acompanhar a homologação de documentos junto ao gestor do sistema;

- Modificar o *backlog* do produto;

4.2.3.2 Gestor do Sistema

Pessoa indicada pela unidade demandante de um sistema para acompanhar e fornecer todas as informações necessárias para a unidade de desenvolvimento. É responsável por:

- Fornecer as informações necessárias em todas as etapas de desenvolvimento;
- Participar de reuniões acordadas;
- Acompanhar a execução das atividades junto à unidade de desenvolvimento;
- Validar os documentos de requisitos gerados;
- Validar os produtos elaborados na fase de desenvolvimento;
- Homologar os produtos finais.

4.2.3.3 Analista de Requisitos

Técnico designado para realizar a análise preliminar de um sistema, coletando junto ao gestor do sistema os dados indispensáveis para o reconhecimento inicial do problema que a unidade demandante precisa solucionar. A ele compete:

- Acordar reuniões com o gestor do sistema para captar

informações e dirimir dúvidas;

- Elaborar o documento “Formulário de Análise Preliminar”.

4.2.3.4 Scrum Master

É o líder facilitador e responsável pelo alinhamento das práticas e trabalhos da Equipe de Desenvolvimento siga os valores da Metodologia constante deste Manual, competindo:

- Identificar e eliminar obstáculos técnicos, processuais ou de comunicação que impedem a equipe de trabalhar.
- Organizar e conduzir as reuniões principais (cerimônias): Daily Scrum (diária), Sprint Planning (planejamento), Review (revisão) e Retrospective (retrospectiva)
- Orientar o time Scrum em autogestão e alta performance, promovendo um ambiente colaborativo
- Ajudar a equipe a definir metas claras, gerenciar o backlog do produto e melhorar a comunicação
- Educar a organização e stakeholders sobre a metodologia Scrum, atuando como um agente de mudança
- Garantir que a equipe não assuma compromissos além de sua capacidade durante a Sprint

Este papel é **obrigatório** e poderá ser desempenhado pelo Gestor de Desenvolvimento de Sistema

4.2.3.2 Product Owner (PO)

É o responsável máximo por maximizar o valor do produto e gerenciar o Product Backlog. Suas competências incluem definir a visão do produto, priorizar funcionalidades, alinhar interesses dos stakeholders (partes interessadas) com o time de desenvolvimento e garantir que o produto atenda às necessidades dos usuários finais. Tem as seguintes responsabilidades:

- Criar, ordenar (priorizar) e comunicar claramente os itens do backlog, garantindo que ele seja transparente e compreendido por todos.
- Estabelecer o objetivo do produto (Product Goal) e comunicá-lo claramente.
- Garantir que o time de desenvolvimento trabalhe nas funcionalidades de maior valor para o negócio primeiro.
- Agir como o elo entre os clientes, usuários e a equipe de desenvolvimento, coletando feedbacks e alinhando expectativas.

- Definir critérios de aceite e validar se o incremento do produto desenvolvido atende às expectativas e aos requisitos de negócio.
- Ter autoridade para decidir o "quê" será construído e quando, baseando-se em pesquisas de mercado e feedback do usuário.

É um papel não obrigatório, podendo a depender da maturidade do time, ser exercido por um integrante do time de desenvolvimento em colaboração com representante do demandante.

4.2.3.3 Equipe de Desenvolvimento

Grupo de pessoas responsáveis pelo desenvolvimento de um sistema. Também denominado Equipe de Desenvolvimento. De acordo com o Método Ágil, possuem habilidades multidisciplinares para que sejam capazes de se auto-organizarem durante o projeto. Não há uma divisão funcional através de papéis tradicionais, tais como programador, *designer* ou analista de testes. Os integrantes do projeto trabalham juntos para entregar as funcionalidades ou sistema. A equipe é responsável por:

- Seguir as diretrizes fundamentadas no MPS;
- Codificar em linguagem de programação a solução de sistema, de acordo com as regras do negócio, os requisitos e documentos de apoio;

- Tomar decisões técnicas que orientem o design e a implementação do projeto;
- Produzir e validar a documentação a ser entregue;
- Definir a arquitetura dos sistemas;
- Prezar pela qualidade do código-fonte, documentando a codificação para futuras manutenções;
- Analisar e modelar o banco de dados de desenvolvimento e testes;
- Criar as interfaces solicitadas pelo usuário;
- Inspecionar código de outros desenvolvedores;
- Elaborar o manual do usuário, quando necessário;
- Propor soluções ou alternativas para problemas;
- Identificar e construir casos e modelos de teste, segundo os testes de aceitação derivados das histórias dos usuários;
- Realizar testes automatizados das unidades implementadas e das interfaces;

- Realizar testes de integração, de sistema e de aceitação em ambiente de homologação similar ao ambiente de produção.

4.3 Processo de Desenvolvimento

O macroprocesso de Desenvolvimento de Software é ilustrado na figura 6 e posteriormente detalhado nas seções a seguir.

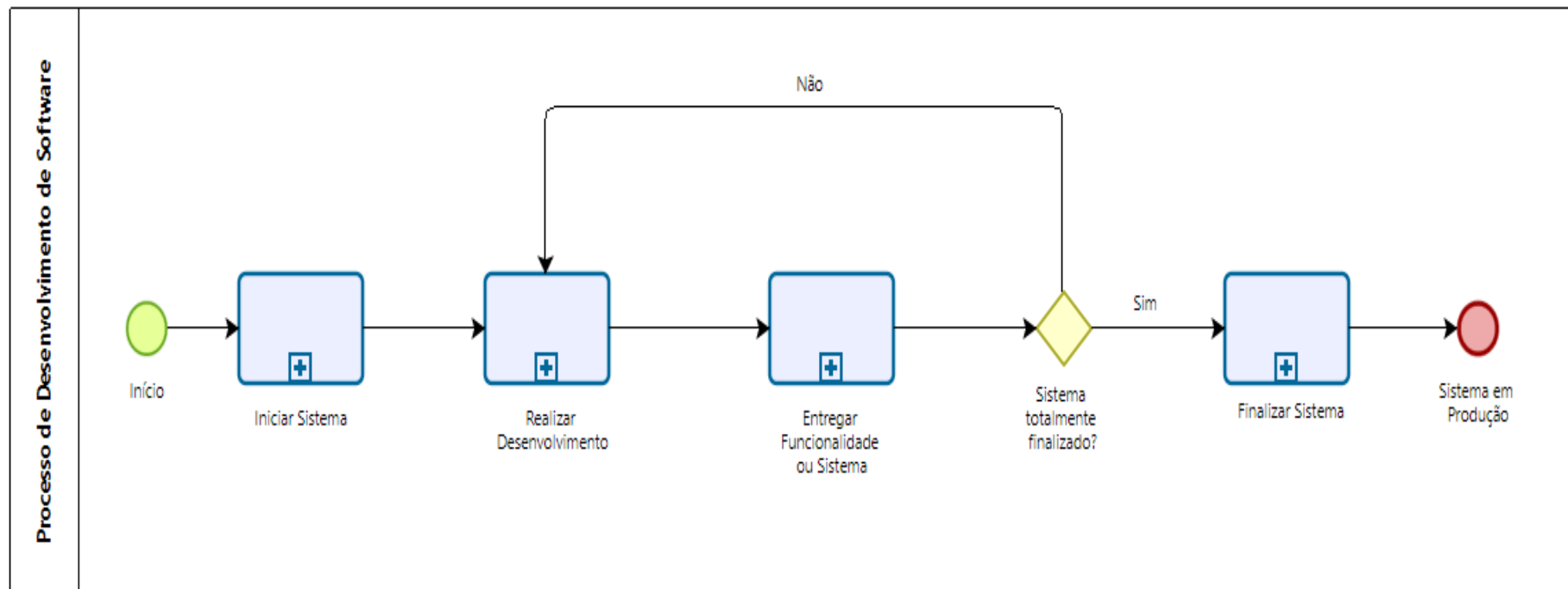


Figura 07 - Etapa do Desenvolvimento de Sistemas do TRE-PI

4.3.1 Iniciar Sistema

O início do projeto envolve as atividades, que são ilustradas no diagrama abaixo e detalhadas nos próximos tópicos.

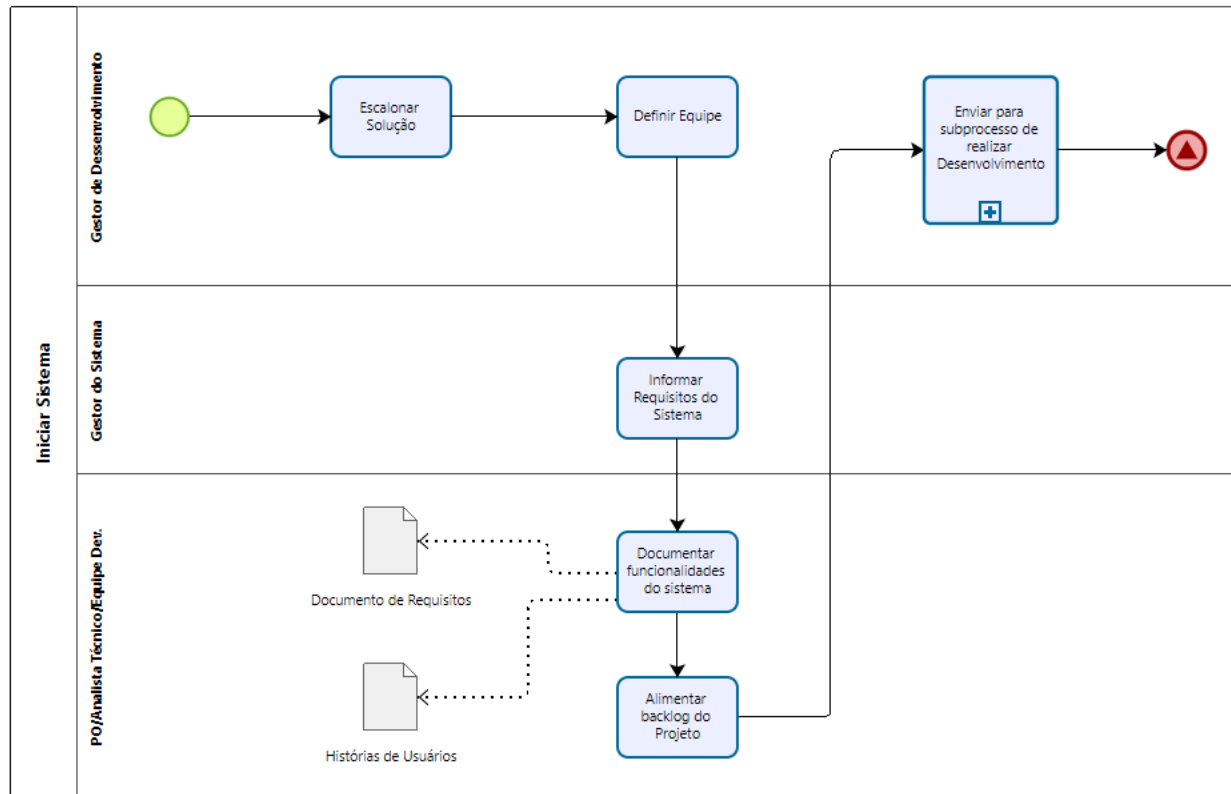


Figura 08 – Fluxo do Subprocesso Iniciar Projeto

4.3.1.1 Escalonar Solução

Nesta fase, o gestor de desenvolvimento de sistemas seleciona o próximo projeto a ser atendido, seguindo a sequência estabelecida na Por-

taria do PADS, resultante da última priorização do CDTI. Havendo disponibilidade de equipes, várias soluções podem ser escalonadas simultaneamente.

No caso de ocorrer algum impedimento para que um sistema seja escalonado em razão de premissas não atendidas, a demanda ficará pendente e o próximo sistema será escolhido no intuito de não haver ociosidade da equipe. Uma vez resolvida a pendência, será analisado se alguma das soluções em andamento poderá ser interrompida. Caso contrário, a demanda permanecerá aguardando uma equipe ser liberada.

4.3.1.2 Definir Equipe

Corresponde à montagem da equipe pelo **Gestor de Desenvolvimento de Sistemas**. O grupo terá perfil para o atendimento da demanda indicada e será composto por profissionais que exercerão os seguintes papéis: Analista de Requisitos e Equipe de desenvolvimento.

4.3.1.3 Informar Requisitos do Sistema

A equipe se reunirá com o Gestor do Sistema, e este irá informar todos os requisitos funcionais e não funcionais do sistema,

validando e complementando o que foi proposto no documento inicial de Solicitação de Desenvolvimento de Sistema.

4.3.1.3 Documentar Funcionalidades do Sistema

A equipe irá coletar as funcionalidades catalogadas durante a fase inicial de Concepção, analisando a documentação produzida e se necessário realizando novas entrevistas com o Demandante, a fim de confirmar as funcionalidades que serão desenvolvidas.

4.3.1.4 Alimentar Backlog do Projeto

Com base no backlog já produzido na fase de concepção e na definição das funcionalidades verificadas, a equipe irá produzir os cartões correspondentes às funcionalidades que serão desenvolvidas, inserindo-as na área do backlog do Projeto.

Ao final, ter-se-á o backlog alimentado com os cartões do sistema.

4.3.2 Realizar Desenvolvimento

É o subprocesso principal, correspondendo a execução de todas as atividades necessárias ao desenvolvimento do Sistema.

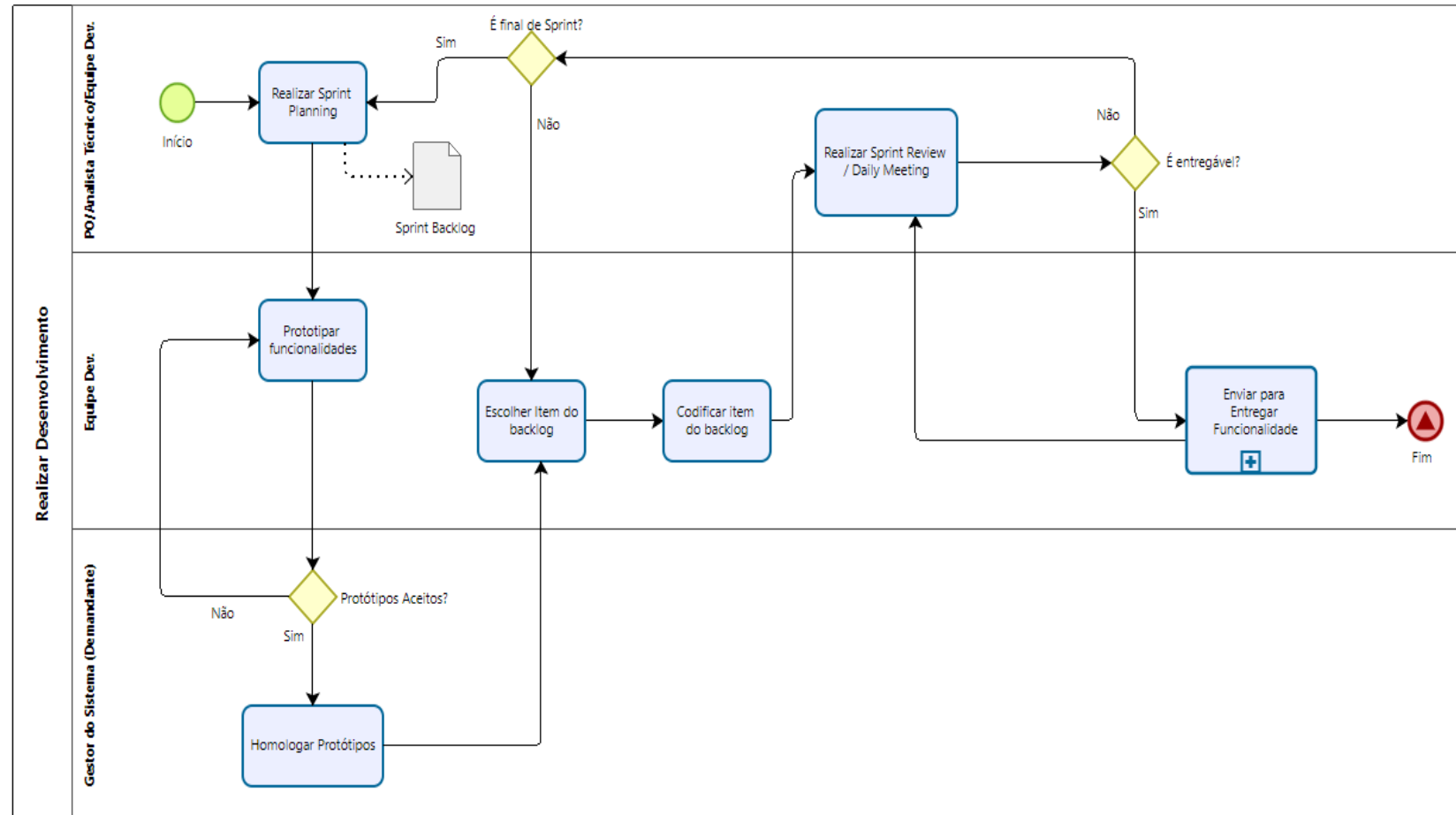


Figura 09 – Fluxo do Subprocesso Realizar Desenvolvimento

Este subprocesso corresponde ao desenvolvimento do sistema em si, por meio de codificação e construção de interfaces.

Neste subprocesso, a equipe de desenvolvimento irá fazer o planejamento da sprint, com todas as funcionalidades que serão

implementadas. O processo é cíclico para cada conjunto de funcionalidades, o qual será finalizado somente quando o sistema for totalmente concluído.

4.3.2.1 Realizar *Sprint Planning*

O Gestor de Desenvolvimento de Sistema irá reunir a equipe designada e irá realizar reunião. É o momento em que o Time Scrum se reúne para definir o que será entregue e como o trabalho será realizado. Para que ele seja eficiente, o Product Owner já deve chegar com um **Backlog do Produto** refinado e priorizado.

A primeira tarefa é estabelecer um objetivo claro. O Product Owner apresenta o valor que a Sprint deve entregar, e o time todo colabora para definir a **Meta da Sprint**. Ela serve como um farol: se as tarefas mudarem durante a semana, a meta permanece a mesma.

Com base na meta e na **capacidade (velocity)** do time, os desenvolvedores selecionam os itens do *Product Backlog* que acreditam ser capazes de concluir.

- O time analisa o que é prioritário.
- Verifica-se se os itens atendem à **Definição de Preparado (Definition of Ready)**.
- O grupo decide a quantidade de trabalho, sem pressão externa, respeitando seu histórico de entregas.

4.3.2.2 Prototipar funcionalidades

As funcionalidades do sprint backlog são prototipadas, pela Equipe de Desenvolvimento, por meio de wireframes ou ferramentas específicas, objetivando a validação de requisitos e regras de negócio do sistema.

4.3.2.3 Homologar Protótipos

O Gestor do Sistema valida todos os protótipos, preferencialmente, em reunião com a equipe de desenvolvimento. Caso um ou mais protótipos não sejam aceitos, estes retornam para que a equipe os refaça. Os protótipos aceitos serão implementados pela Equipe de Desenvolvimento.

4.3.2.4 Escolher item do backlog

A equipe de desenvolvimento irá escolher na sprint backlog, priorizado, as funcionalidades correspondentes aos protótipos validados.

4.3.2.5 Codificar item do backlog

A equipe de desenvolvimento irá implementar o item do

backlog e a funcionalidade correspondente.

4.3.2.6 Realizar Sprint Review / Daily Meeting

Diariamente, o Daily Meeting, será feita reunião com a Equipe de Desenvolvimento, conduzida pelo Scrum Master / Gestor de Desenvolvimento, no qual terá como objetivo alinhar as atividades em desenvolvimento, o que foi feito no dia anterior, o que será feito no dia e os impedimentos ocorridos.

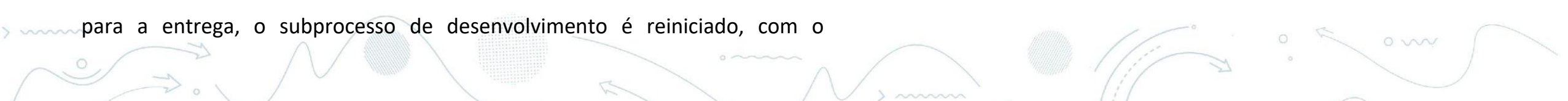
Já a reunião de Sprint Review, ou reunião de revisão, será apresentada as funcionalidades desenvolvidas, com a participação do Gestor do Sistema e do Product Owner. Pela flexibilidade trazida do Kanban, essa verificação poderá ser durante o transcorrer da sprint, desde que tenha um ou mais funcionalidades relevantes e que entreguem valor para o Demandante.

4.3.2.7 Enviar para Entregar Funcionalidade

Após a reunião de **Sprint Review**, ao se verificar que se tem um grupo de funcionalidades que constitui um entregável, a Equipe de Desenvolvimento/PO irá deliberar se inicia o subprocesso de Entregar Funcionalidade.

Caso se observe que o grupo de funcionalidades ainda não estão prontas para a entrega, o subprocesso de desenvolvimento é reiniciado, com o

planejamento da próxima sprint ou a continuidade da sprint atual.



4.3.2.8 Matriz de Responsabilidades

ATIVIDADES/PAPÉIS	PRODUCT OWNER/ANALISTA TÉCNICO	GESTOR DE DESENVOLVIMENTO/SCRUM MASTER	GESTOR DO SISTEMA	EQUIPE
Realizar Sprint Planning	R/E	I	C/I	E
Prototipar Funcionalidades	C/I	I	C/I	R/E
Homologar Protótipos	I	I	R/E	I
Escolher Item da sprint backlog	I	I	I	R/E
Codificar Item do Backlog	I	I	I	R/E
Realizar Sprint Review	R/E	R/E	I	E
Realizar Daily Meeting	R/E	R/E	I	I
Enviar para Subprocesso Entregar Funcionalidade	C/I	I	I	R/E
Legenda: R—Responsável; E—Executor; C—Consultado; I—Informado.				

Tabela 03 – Matriz de responsabilidades do subprocesso Realizar Desenvolvimento

4.3.2.9 Recursos e Resultados

ATIVIDADES	RECURSOS	RESULTADOS
Realizar Sprint Planning	Backlog do Projeto	Sprint Backlog
Prototipar Funcionalidades	Sprint Backlog	Protótipos prontos para homologação
Homologar Protótipos	Protótipos	Protótipos prontos para homologação
Escolher Item da sprint backlog	Sprint Backlog	Lista de itens da sprint backlog referentes aos protótipos homologados
Codificar item do Backlog	Itens da Sprint Backlog referentes aos protótipos homologados	Funcionalidades implementadas
Realizar Sprint Review	Funcionalidades implementadas	Funcionalidades prontas para homologação
Realizar Daily Meeting	Itens da sprint backlog em status de desenvolvimento	Informação sobre o status de cada item ou lista de impedimentos encontrados
Enviar para Subprocesso Entregar Funcionalidade	Funcionalidades prontas para homologação	Lista de funcionalidades homologadas ou retornadas para desenvolvimento

Tabela 04 – Recursos do subprocesso Realizar Desenvolvimento



4.3.3 Entregar Funcionalidade ou Sistema

É o subprocesso executado após o fornecimento de parte do sistema plenamente funcional ou o sistema completo.

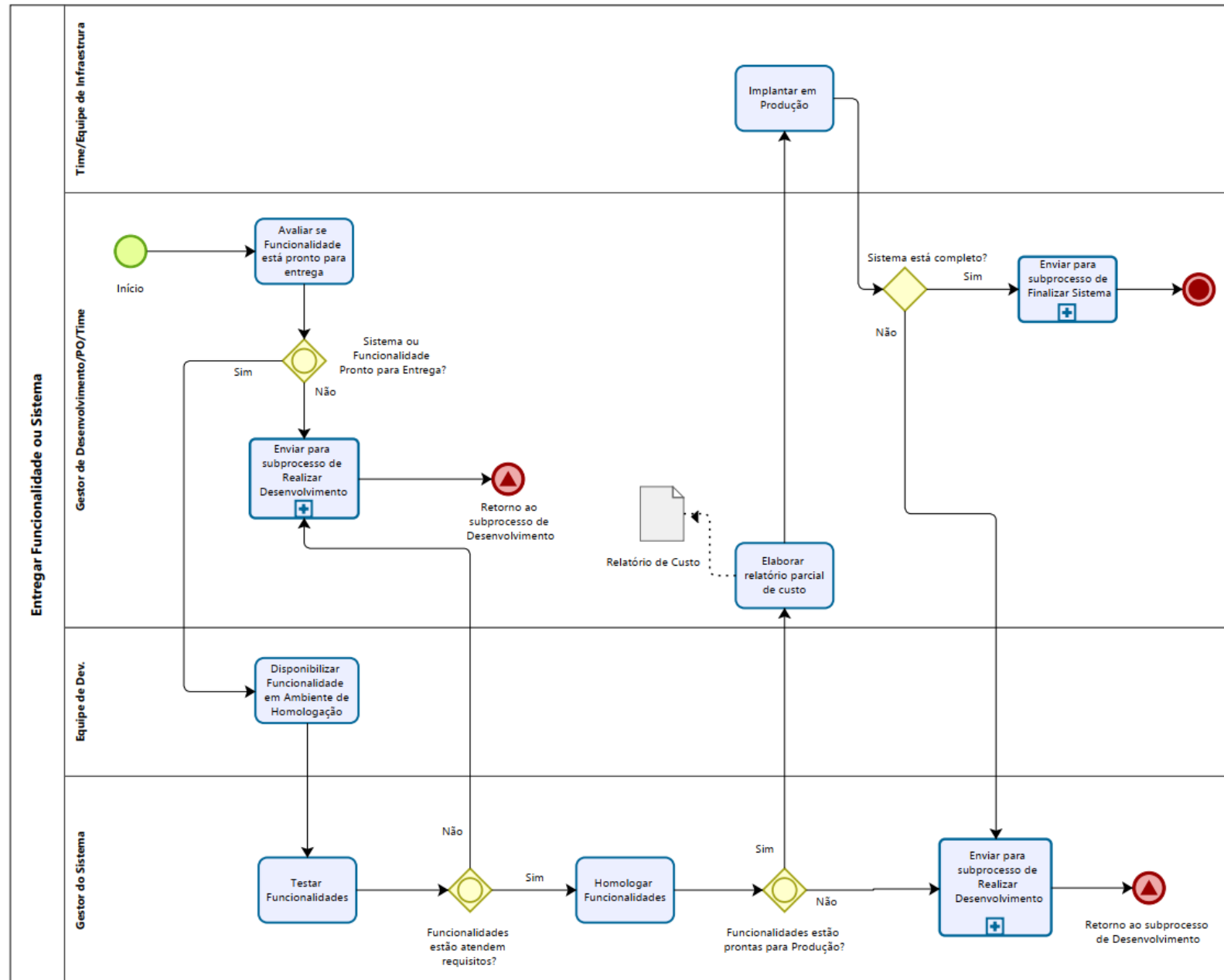


Figura 10 – Fluxo do Subprocesso de Entregar Funcionalidade ou Sistema

4.3.3.1 Avaliar se Funcionalidade ou Sistema está pronto para Entrega

O Gestor de Desenvolvimento, juntamente com a Equipe e *Product Owner* realizam a última avaliação se as funcionalidades ou todo o sistema estão prontas para homologação e posterior implantação em produção

4.3.3.2 Enviar para Subprocesso de Realizar Desenvolvimento

Julgado que as funcionalidades ou o sistema ainda não atende aos requisitos ou apresenta defeitos, o Gestor do Sistema mobilizará novamente a equipe de Desenvolvimento para corrigir eventuais problemas verificados no entregável. O subprocesso de Realizar Desenvolvimento será novamente iniciado.

Este subprocesso poderá ser iniciado antes do envio das funcionalidades para homologação quanto após a atividade de homologação.

4.3.3.3 Disponibilizar Funcionalidade(s) em Ambiente de Homologação

Depois da avaliação das funcionalidades e decidido pela entrega ao Gestor de Sistemas, a equipe desenvolvimento implanta, em ambiente de homologação, as funcionalidades ou todo o sistema.

4.3.3.4 Testar Funcionalidades

O Gestor do Sistema (Demandante) irá testar todas as funcionalidades disponibilizadas em ambiente de produção.

As funcionalidades testadas e com erros ou não atendendo aos requisitos propostas são enviadas para o Subprocesso de Realizar Desenvolvimento.

Já as funcionalidades sem erros e alinhadas aos requisitos propostos podem ser homologas pelo Gestor do Sistema.

4.3.3.5 Homologar Funcionalidades

As funcionalidades que atendem os requisitos propostos pelo Gestor do Sistema passam pela última validação, as funcionalidades que estão prontas para implantação em ambiente de produção são homologadas e as não estão prontas voltam para o subprocesso de Realizar Desenvolvimento de Sistemas.

4.3.3.6 Elaborar Relatório Parcial de Custo

Todas as funcionalidades preparadas para implantação tem o seu custo mensurado, objetivando compor, ao final, quando o sistema estiver totalmente concluído, o Relatório de Entrega

Cada funcionalidade tem o seu curso mensurado por meio da aplicação da seguinte fórmula:

Custo da Funcionalidade = quantidade horas * valor da Hora do Desenvolvedor

Custo Parcial Total = Somatório de todos os custos individuais por Funcionalidade

O Gestor de Desenvolvimento é o responsável por essa tarefa.

4.3.3.7 Implantar em Produção

A equipe de desenvolvimento e a equipe de infraestrutura poderá, de posse das funcionalidades homologadas, implantar em ambiente de produção.

Caso o conjunto de funcionalidades contemplem todos os requisitos do Sistema, o Gestor de Desenvolvimento, o Product Owner ou a Equipe inicia o Subprocesso de Finalizar Sistema.



4.3.3.8 Matriz de Responsabilidades

ATIVIDADES/PAPÉIS	GESTOR DE DESENVOLVIMENTO	GESTOR DO SISTEMA	PRODUCT OWNER	EQUIPE
Avaliar se Funcionalidades estão prontas para entrega	R/E	I	R/E	R/E
Enviar para subprocesso de Realizar Desenvolvimento	R/E	C/I	R	R/E
Disponibilizar Funcionalidade em Ambiente de Homologação	I	I	C/I	R/E
Testar Funcionalidades	I	R/E	I	I
Homologar Funcionalidades	I	R/E	I	I
Elaborar Relatório de Custos	R/E	-	-	-
Implantar em Produção	I	I	C/I	R/E
Enviar para o Subprocesso de Finalizar Sistema	R/E	I	E	I
Legenda: R—Responsável; E—Executor; C—Consultado; I—Informado.				

Tabela 05 – Matriz de responsabilidades do subprocesso Entregar Funcionalidade ou Sistema

4.3.3.9 Recursos e Resultados

ATIVIDADES	RECURSOS	RESULTADOS
Avaliar se Funcionalidades estão prontas para entrega	Funcionalidades ou Sistema	Funcionalidade ou Sistema
Enviar para subprocesso de Realizar Desenvolvimento	Erros ou defeitos encontrados Não conformidade com os requisitos	Relatório de erros ou defeitos encontrados Novos itens para Backlog
Disponibilizar Funcionalidade em Ambiente de Homologação	Funcionalidades ou Sistema	Funcionalidades ou Sistema em execução
Testar Funcionalidades	Funcionalidade ou Sistema	Relatório de Funcionalidades Testadas
Elaborar Relatório de Custos	Funcionalidade ou Sistema	Relatório Parcial de Custo
Implantar em Produção	Funcionalidade ou Sistema	Funcionalidade ou Sistema em Execução
Enviar para o Subprocesso de Finalizar Sistema	Sistema	Relatório de todas as funcionalidades do Sistema

Tabela 06 – Recursos e Resultados do subprocesso Entregar Funcionalidade do Sistema



4.3.4 Finalizar Sistema

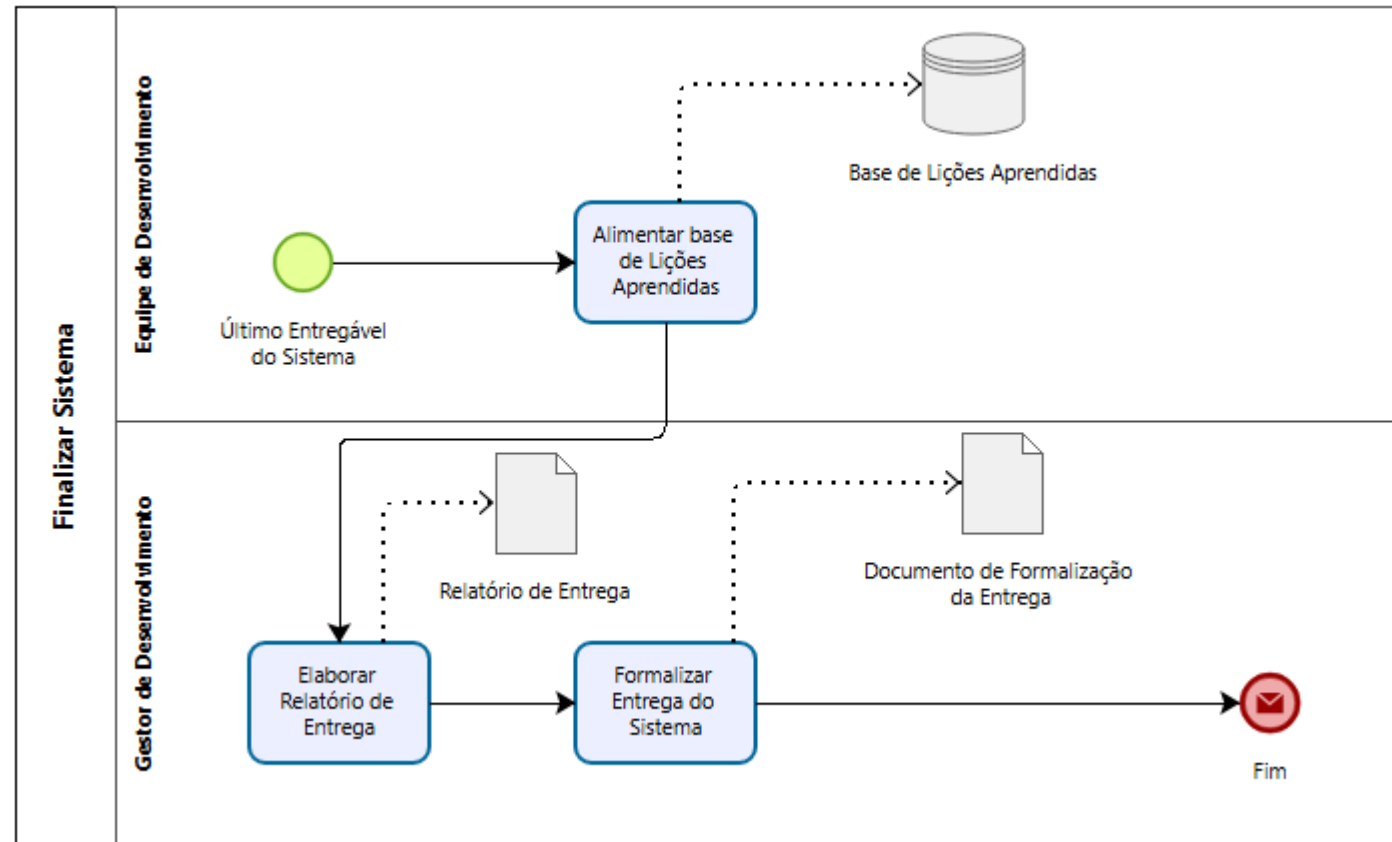


Figura 11 – Fluxo do Subprocesso Encerrar Projeto

4.3.4.1 Alimentar Base de Lições Aprendidas

Corresponde à realização de documentação e registro, pela Equipe de Desenvolvimento, diante de todos os eventos bloqueadores e problemas verificados, as soluções encontradas para resolução de tais obstáculos.

Estes registros poderão feitas em sistema próprio de base de conhecimentos.

4.3.4.3 Formalizar Entrega do Sistema

Representa a formalização da entrega do produto final junto ao gestor do sistema. O projeto é considerado encerrado quando o último entregável ou o Sistema completo for homologada pelo gestor do sistema e não haja mais pendências. Será preenchido um documento de homologação do sistema pelo gestor do sistema para formalizar o encerramento do projeto.

4.3.4.2 Elaborar Relatório de Entrega

O Gestor de Desenvolvimento, antes de formalizar a entrega do sistema, elaborará relatório contendo todos o recursos utilizados para desenvolver, tais como: tempo de desenvolvimento, quantidade de desenvolvedores (terceirizados e efetivos) que participaram do

desenvolvimento, o custo por hora de trabalho (se terceirizados, a base será o salário previsto em contrato, se servidor efetivo, a base será o valor de sua remuneração constante de lei). Se houve necessidade de aquisição de licenças de software ou ferramentas de desenvolvimento, acrescentar, o valor despendido.

4.3.4.4 Matriz de Responsabilidades

ATIVIDADES/PAPÉIS	GESTOR DE DESENVOLVIMENTO	EQUIPE
Alimentar Base de Lições Aprendidas	C/I	R/E
Elaborar Relatório de Entrega	R/E	-
Formalizar Entrega do Projeto	R/E	-
Legenda: R—Responsável; E—Executor; C—Consultado; I—Informado.		

Tabela 07 – Matriz de Responsabilidades do Subprocesso de Finalizar Projeto

4.3.4.5 Recursos e Resultados

ATIVIDADES	RECURSOS	RESULTADOS
Alimentar Base de Lições Aprendidas	Problemas e eventos bloqueadores encontrados	Base de Conhecimento alimentada
Formalizar Entrega do Projeto	Funcionalidade ou Sistema	Documento de Formalização de Entrega

Tabela 08 – Recursos e Resultados do Subprocesso de Finalizar Projeto



5 Sustentação do Sistema

5.1 Introdução

A fase de sustentação do sistema consiste na etapa em que o mesmo, uma vez desenvolvido e implantado em ambiente de produção, começa a operar e a cumprir seu propósito. Essa é uma das fases mais longas de seu ciclo de vida, momento em que far-se-ão necessárias manutenções corretivas e/ou evolutivas que visarão, respectivamente, corrigir problemas identificados com o uso ou inserir novas funcionalidades percebidas como necessárias.

O diagrama abaixo ilustra este processo:

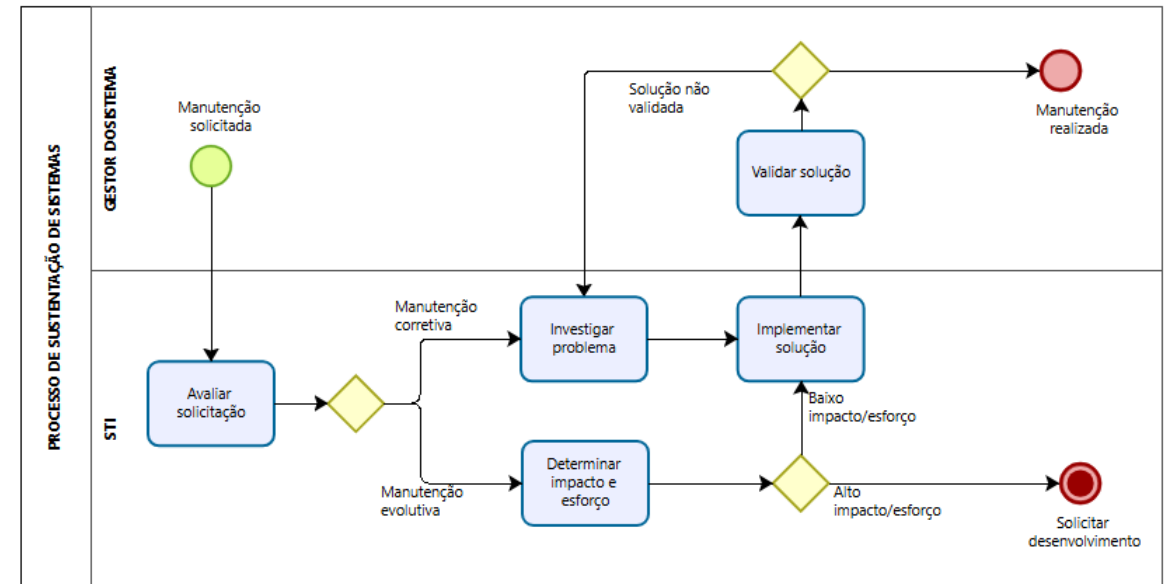


Figura 12 – Fluxo do Processo de Sustentação de Sistemas

5.2 Avaliar Solicitação

A avaliação da solicitação consiste na análise da demanda de manutenção recebida com o objetivo de classificá-la em manutenção 1 – corretiva; ou 2 – evolutiva. A Unidade de Desenvolvimento deverá identificar o tipo de manutenção solicitada e prosseguir com este fluxo de acordo com essa classificação. Esta atividade, uma vez concluída, desencadeará uma das duas atividades abaixo: Investigar Problema – Fluxo ativado caso a solicitação seja classificada como manutenção corretiva; Determinar Impacto/Esforço – Fluxo ativado caso a solicitação seja classificada como manutenção evolutiva.

5.3 Investigar Problema

Uma vez que a solicitação de manutenção foi classificada como corretiva, esta atividade consiste em diagnosticar o problema relatado pela unidade demandante a fim de identificar a solução apropriada.

Determinar Impacto/Esforço

5.5 Validar Solução

Esta atividade é responsável por obter da unidade demandante o aceite da solução implementada pela STI. A saída desta atividade pode direcionar o fluxo do processo de volta à atividade ‘Implementar Solução’, caso a resposta dada ao problema não atenda ao esperado, ou encerrar o processo, quando a

Uma vez que a solicitação de manutenção foi classificada como evolutiva, faz-se necessária a análise do impacto que essa evolução causará na arquitetura geral do sistema, bem como do esforço necessário para implementá-la. Toda manutenção evolutiva terá, ao final desta tarefa, uma das duas classificações a seguir: Baixo Impacto/Esforço – desencadeará a atividade Implementar Solução; Alto Impacto/Esforço – encerrará este processo e iniciará a atividade ‘Solicitar desenvolvimento de Sistema’, do processo de desenvolvimento de sistemas, tratado no capítulo 4.

5.4 Implementar Solução

Esta atividade é responsável por implementar a solução do problema, definida na atividade ‘Investigar Problema’, ou por desenvolver a nova funcionalidade classificada como de baixo impacto/esforço pela atividade ‘Determinar Impacto/Esforço’. Quando de seu término, esta atividade deve notificar a unidade demandante que uma solução para sua solicitação foi implementada e está sujeita a validação.

solução atende à expectativa da unidade demandante.

5.6 Matriz de Responsabilidades

ATIVIDADES/PAPÉIS	GESTOR DE DESENVOLVIMENTO	EQUIPE
Alimentar Base de Lições Aprendidas	C/I	R/E
Formalizar Entrega do Projeto	R/E	-
Legenda: R—Responsável; E—Executor; C—Consultado; I—Informado.		

Tabela 09 – Matriz de Responsabilidades do Processo Sustentação do Sistema

5.7 Recursos e Resultados

ATIVIDADES	RECURSOS	RESULTADOS
Alimentar Base de Lições Aprendidas	Problemas e eventos bloqueadores encontrados	Base de Conhecimento alimentada
Formalizar Entrega do Projeto	Funcionalidade ou Sistema	Documento de Formalização de Entrega

Tabela 10 – Recursos e Resultados do Processo Sustentação do Sistema

6 Declínio do Sistema

6.1 Introdução

Considera-se que um sistema entra em declínio quando é preciso atender novas necessidades advindas de evoluções tecnológicas, organizacionais ou legais. Há situações em que se torna muito oneroso realizar adaptações nos sistemas para atender tais requisitos. Pode ocorrer também do sistema não ser mais necessário para o cliente, devendo apenas ser retirado do ambiente de produção. A decisão por desativar um sistema pode partir do próprio gestor do sistema (ou gestor de negócio) ou então o Gestor Técnico do Sistema. A seguir, são descritas as etapas que serão executadas quando um sistema já implantado entra em declínio.

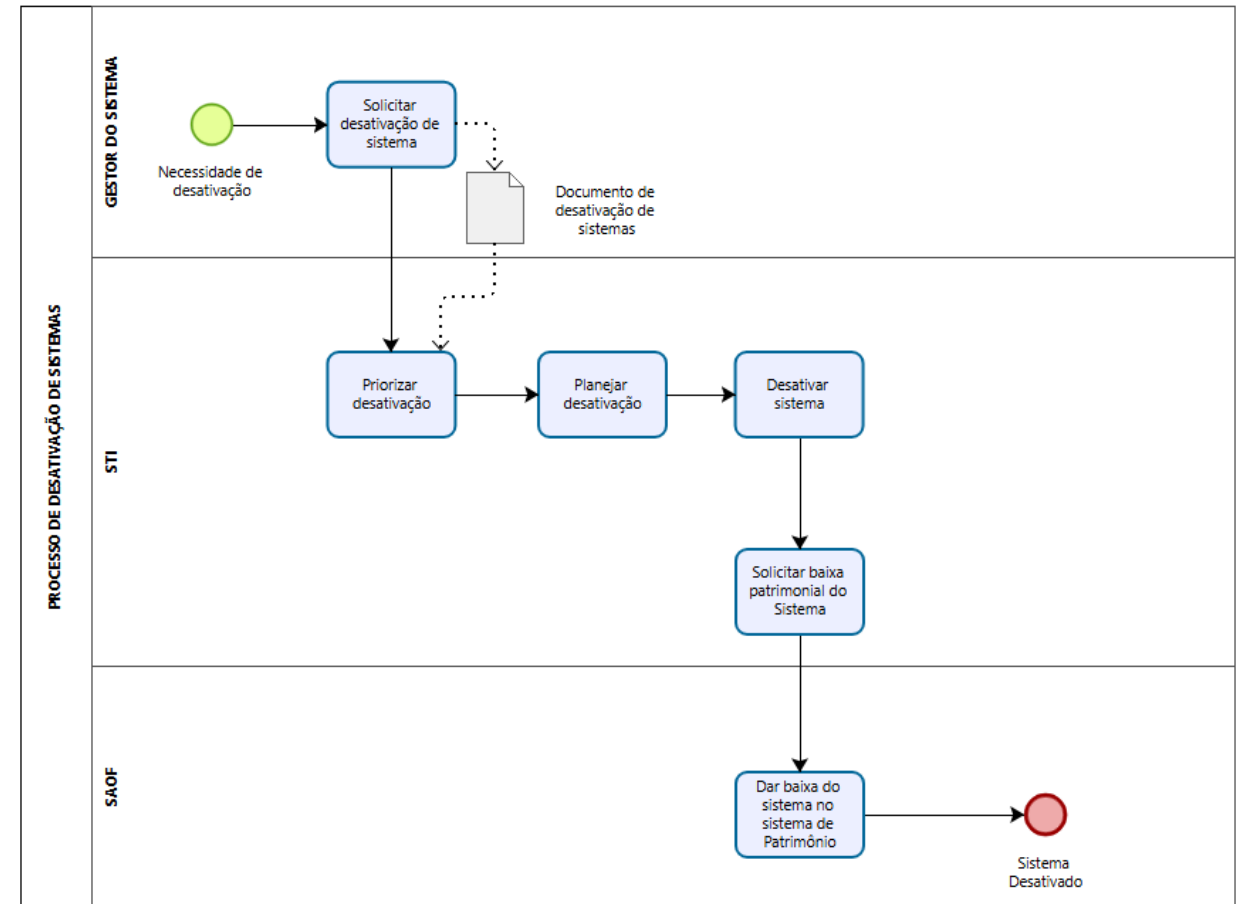


Figura 13 – Fluxo do Processo de Desativação do Sistema

6.2 Verificar Sistema Passível de Desativação

O Gestor Técnico irá, periodicamente, verificar quais sistemas podem ser desativadas, por falta de uso ou obsolescência, entrando em contato com o Gestor de Desenvolvimento sobre a possibilidade de desativação.

6.3 Consultar Desativação

O Gestor técnico comunicará ao Gestor de Desenvolvimento qual sistema poderá ser desativado, e este consultará ao Gestor do Sistema se realmente o sistema poderá ser desativado.

6.4 Autorizar Desativação

O Gestor técnico comunicará ao Gestor de Desenvolvimento qual sistema poderá ser desativado, e este consultará ao Gestor do Sistema se realmente o sistema poderá ser desativado.

Se o Gestor do Sistema autorizar a desativação, será comunicado ao Gestor de Desenvolvimento, passando para o passo seguinte.

Caso não seja autorizado, o processo será finalizado, sem a desativação do sistema.

6.5 Solicitar Desativação

Atividade em que a Unidade Demandante, na figura do Gestor do Sistema, formalmente se manifesta solicitando a desativação de um sistema sob sua responsabilidade. Essa formalização se dá através do preenchimento e entrega, via sistema de processo administrativo, do Documento de Desativação de sistemas.

6.6 Priorizar Desativação

Uma vez formalizado junto ao Gestor de Desenvolvimento, o pedido para desativação de um sistema, é necessário que o gestor da referida unidade verifique as demandas correntes para que a atividade seja priorizada e encaminhada ao gestor técnico do sistema para definir o planejamento.

6.7 Planeja a Desativação

Nesta etapa, o gestor técnico analisa o impacto da desativação do sistema e elabora um cronograma para que a transição ocorra de maneira segura, sem transtornos para o usuário e respectivas unidades impactadas. O planejamento para a desativação do sistema deve contemplar: As atividades a serem executadas e respectivos responsáveis; Prazos para execução; Se necessário, definição de uma etapa piloto a fim de analisar possíveis impactos não previstos inicialmente; Política de backup dos dados.

6.8 Desativar Sistema

Corresponde à execução do planejamento definido na etapa anterior no que diz respeito ao backup dos dados e liberação de recursos dos servidores de produção. Será realizada pela unidade de infraestrutura do TRE-PI.

6.9 Solicitar Baixa do Sistema

Todos os sistemas produzidos internamente foram patrimoniados. Assim, quando da sua desativação, a STI, por meio do seu Gestor de Desenvolvimento, deverá solicitar à SAOF a baixa do sistema do base patrimonial do TRE-PI.

6.10 Dar Baixa do Sistema

A SAOF, após receber a solicitação de baixa do sistema, deverá proceder a baixa do sistema na base de dados do Sistema de Patrimônio do TRE-PI.



6.11 Matriz de Responsabilidade

ATIVIDADES/PAPÉIS	GESTOR DO SISTEMA	GESTOR DE DESENVOLVIMENTO	EQUIPE DE INFRA-ESTRUTURA	SAOF
Verificar Sistema Passível de Desativação	-	I	-	
Consultar Desativação	-	C	-	
Autorizar Desativação	R/E			
Solicitar desativação	R/E	C/I	-	
Priorizar Desativação	C/I	R/E	-	
Planejar Desativação	C	C/I	I	
Desativar Sistema	I	I	R/E	
Solicitar Baixa Patrimonial do Sistema	-	R/E		I
Dar Baixa do Sistema		I	-	R/E

Tabela 11 – Matriz de responsabilidades do processo de Declínio de um sistema



6.12 Recursos e Resultados

ATIVIDADES	RECURSOS	RESULTADOS
Verificar Sistema Passível de Desativação	Não atualização de dados no sistema Relatório de acessos ao sistema	Relatório de sistemas passíveis de desativar
Consultar Desativação	Relatório de sistemas passíveis de desativar	Documento enviado ao Gestor de Sistema consultando sobre desativação
Autorizar Desativação	Documento enviado ao Gestor de Sistema consultando sobre desativação	Documento de Desativação
Solicitar desativação de sistema	Justificativas para a necessidade de desativação do sistema.	Documento de solicitação de desativação de sistema;
Priorizar desativação	Documento de solicitação de desativação de sistema; PADS atualizado; Relação de outras demandas do setor.	Desativação priorizada entre as outras demandas da unidade.
Planejar desativação	Documento de solicitação de desativação de sistema; Desativação priorizada entre as outras demandas da unidade.	Cronograma de desativação do sistema.
Desativar sistema	Cronograma de desativação do sistema.	Sistema desativado.

Tabela 12 – Recursos e Resultados do processo de Declínio de um sistema

7 Escopo e Requisitos do Sistema

7.1 Introdução

Fundamentais ao completo entendimento da necessidade do cliente, o escopo e os requisitos configuram dois dos principais artefatos que precisam ser bem definidos e gerenciados. Isso posto, e considerando ainda as boas práticas de desenvolvimento de software, faz-se necessária a efetiva gestão dos mesmos. Essa gestão busca estabelecer um entendimento comum da natureza do sistema, seus limites e funcionalidades, bem como manter esse entendimento atualizado conforme as mudanças ocorrem. Busca-se também, dessa forma, manter o controle das futuras evoluções que o mesmo venha a receber.

7.2 Gerenciamento de Escopo e Requisitos

O gerenciamento do escopo tem por objetivo estabelecer de maneira clara para as partes interessadas os limites do sistema. Isso significa que ambos, cliente e unidade de desenvolvimento, tenham uma visão consensual das capacidades esperadas para o sistema.

É importante que, quando da definição do escopo, fique claro o que não deve fazer parte do escopo do sistema. Isso reduz ambiguidades e aumenta a clareza, alinhando as expectativas. A definição do escopo acontece quando

da atividade de definição do documento de visão e arquitetura, no processo de desenvolvimento de sistemas. O escopo pode ainda ser ampliado ou reduzido com eventuais manutenções evolutivas. A depender do tipo e abrangência de determinada evolução, essa pode alterar significativamente as fronteiras do sistema, adicionando ou removendo funcionalidades relevantes. Essas mudanças devem, necessariamente, ser refletidas no escopo documentado.

Ao passo que o escopo define os limites do sistema, os requisitos estabelecem as funcionalidades que os usuários esperam que o mesmo execute (requisitos funcionais) e as restrições sob as quais o mesmo tenha que operar (requisitos não funcionais). O gerenciamento desses requisitos envolve identificar, controlar e acompanhar suas necessidades e mudanças.

Essa gestão, semelhante à do escopo, ocorre de maneira dispersa em todo o ciclo de vida do produto. Os requisitos são inicialmente levantados na fase de concepção do projeto e documentados no artefato denominado documento de análise preliminar. Esses requisitos são posteriormente especificados de maneira mais detalhada, considerando as restrições

impostas pela arquitetura do sistema e por outras definições que irão compor o documento de visão e arquitetura do sistema. Por fim, existe a constante alteração de requisitos, cenário comum em projetos de software. Sejam alterações de requisitos que surgem ao longo do desenvolvimento do software, sejam necessidades de mudanças após este estar em produção, ambas precisam ser gerenciadas de maneira coordenada. Todo o exposto deixa claro que a gestão de escopo e requisitos ocorre de maneira fragmentada ao longo do ciclo de vida do produto, tendo suas atividades detalhadas nos capítulos anteriores, mas pode ser ilustrada no diagrama abaixo:

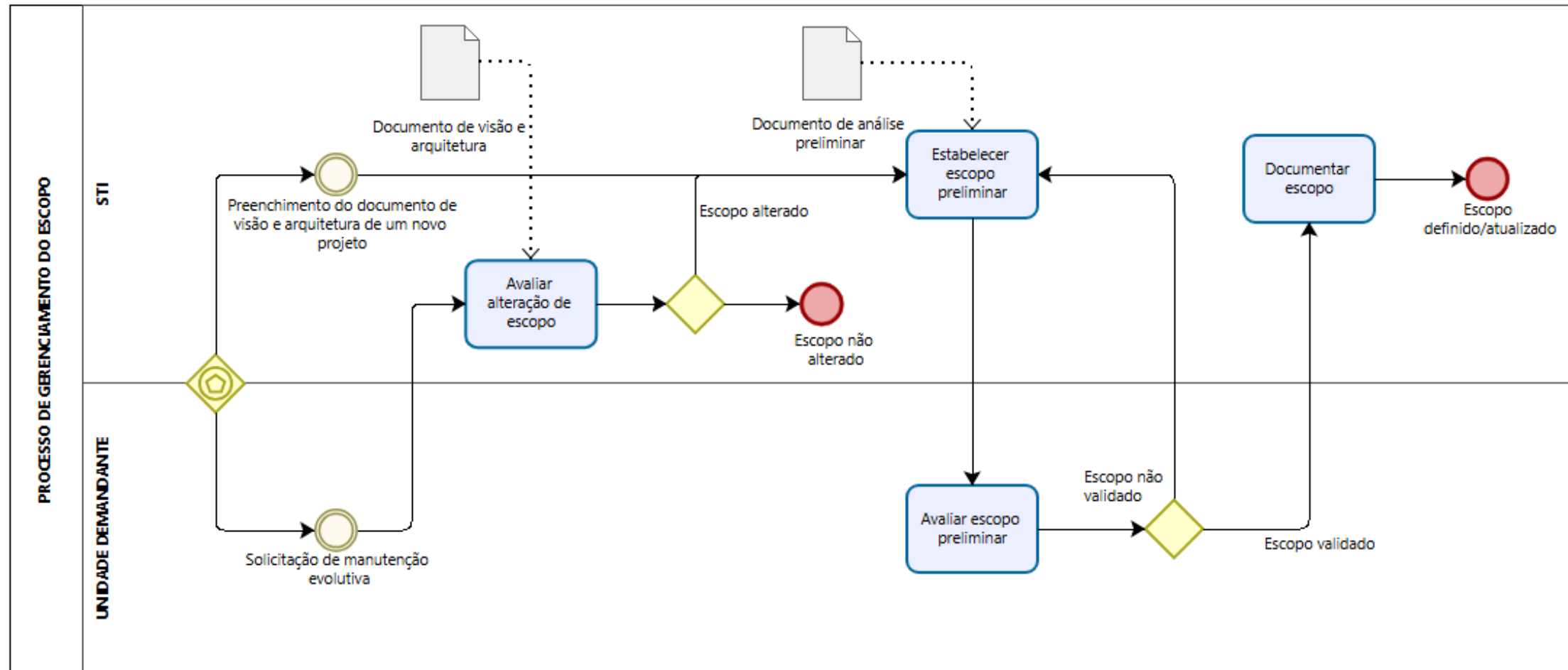


Figura 14 – Fluxo do Processo de Gerenciamento de Escopo

8 Diretrizes para o Desenvolvimento de Software

8.1 Introdução

A busca pela conformidade, no âmbito de um produto de software, tem por objetivo garantir que o mesmo atende aos requisitos estabelecidos em normas, procedimentos e na legislação aplicável. Em síntese, a conformidade é a avaliação da adequação do software aos requisitos de legislação e outros tipos de padronização aplicáveis. Nesse contexto, esta sessão estabelece as diretrizes que devem nortear as atividades de desenvolvimento de software, com o objetivo de garantir essa conformidade.

8.1 Diretrizes de Conformidade

Esta sessão define, de maneira não exaustiva, as diretrizes a serem observadas ao desenvolver soluções de software. Estimular a adoção de metodologia de desenvolvimento de sistemas, buscando assegurar padronização, integridade e segurança; Criar e manter, para toda aplicação em produção, documentação que oriente o usuário no uso do sistema; Adotar padrões abertos no desenvolvimento de tecnologia da informação; Observar as recomendações da Política de Segurança da Informação instituído pelo TRE-PI;

Garantir que as aplicações desenvolvidas sejam responsivas; Garantir que as aplicações desenvolvidas sejam adequadas à exibição em dispositivos móveis; Observar as recomendações do Modelo de Acessibilidade em Governo Eletrônico (e-MAG); Observar as recomendações do Modelo de Requisitos para Sistemas Informatizados de Gestão de Processos e Documentos da Justiça Federal – MoReq-Jus; Observar o Modelo Nacional de Interoperabilidade, de acordo com as metas do termo de cooperação técnica nº 58/2009; Sempre que aplicável, prover suporte à assinatura digital, conforme padrões estabelecidos pela ICP Brasil.



9 Conclusão

O presente Manual do Processo de Desenvolvimento de Software (MPS) consolida o compromisso do Tribunal Regional Eleitoral do Piauí (TRE-PI) com a excelência, a segurança e a governança de TI. Ao definir de maneira estruturada as quatro fases do ciclo de vida de uma solução corporativa — Concepção, Desenvolvimento, Sustentação e Declínio —, o manual garante que todas as demandas tecnológicas institucionais sejam alinhadas estrategicamente e priorizadas de forma objetiva através do Plano de Acompanhamento de Desenvolvimento de Sistemas (PADS).

Uma das maiores evoluções desta versão é a adoção da metodologia ágil híbrida Scrumban, que une a previsibilidade de planejamento e as cerimônias do Scrum à flexibilidade e ao fluxo contínuo de entregas do Kanban. Aliada a essa nova dinâmica de trabalho colaborativo, a consolidação da arquitetura de microsserviços e a implementação da cultura DevSecOps — sustentada por ferramentas de automação e pipelines de CI/CD — garantem lançamentos mais rápidos e confiáveis. A segurança da informação deixa de ser apenas uma etapa final e passa a ser uma responsabilidade contínua desde a escrita do código, com análises estáticas de vulnerabilidades preventivas.

Além da excelência técnica, o MPS reafirma a responsabilidade institucional com o usuário final. As diretrizes de acessibilidade e

usabilidade transformam a tecnologia em um verdadeiro facilitador inclusivo. Simultaneamente, a aplicação rigorosa dos conceitos de Privacy by Design e Privacy by Default assegura que todas as soluções desenvolvidas já nasçam em plena conformidade com a proteção de dados pessoais.

Por fim, compreende-se que este manual não é um artefato estático, mas sim um guia vivo. Ele reflete o amadurecimento constante da Secretaria de Tecnologia da Informação (STI). Através do registro constante de lições aprendidas e do aprimoramento contínuo, a metodologia aqui descrita continuará a evoluir, garantindo que o TRE-PI forneça as melhores soluções tecnológicas para o cumprimento de sua missão institucional.



STI
Secretaria da Tecnologia
da Informação

MANUAL DO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE



Secretaria de Tecnologia da Informação
Coordenadoria de Desenvolvimento e Infraestrutura
Seção de Desenvolvimento de Soluções Corporativas