



**PODER
JUDICIÁRIO**
DO ESTADO DO ACRE

METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE (MDS)

Guia de Referência

Março **2026**





SUMÁRIO

1. INTRODUÇÃO.....	2
2. APRESENTAÇÃO DA METODOLOGIA	3
3. CICLO DE VIDA	14
4. FERRAMENTAS DE APOIO.....	18
5. PROCESSOS DE COMUNICAÇÃO	23
6. ANEXOS.....	26
7. HISTÓRICO DE REVISÕES.....	27



1. INTRODUÇÃO

O presente documento tem como objetivo descrever a Metodologia de Desenvolvimento de Software adotada pela Subsecretaria de Sistemas de Informações (SUSIS), da Secretaria de Tecnologia da Informação e Comunicação (SETIC) do Tribunal de Justiça do Acre (TJAC). A metodologia aqui apresentada foi desenvolvida para assegurar que todos os projetos sejam conduzidos de maneira eficiente e eficaz, atendendo às necessidades do Tribunal e garantindo a entrega de resultados de alta qualidade.

A SUSIS coordena e gerencia projetos de sistemas de informação críticos para o TJAC. Devido à complexidade desses projetos, a adoção de uma Metodologia de Desenvolvimento de Software estruturada é vital. Esta metodologia da SUSIS estabelece um *framework* claro e consistente para o planejamento, execução, monitoramento e encerramento de projetos, alinhado com as melhores práticas e adaptado às necessidades do Tribunal.

Este documento é destinado a todos os envolvidos no ciclo de vida dos projetos da SUSIS, incluindo gestores, membros da equipe de projeto, stakeholders e outros colaboradores. Ele serve como um guia para compreender as etapas, processos, papéis e responsabilidades que compõem a metodologia, promovendo uma abordagem padronizada e colaborativa para o gerenciamento de projetos.

Este documento é um guia para todos os envolvidos nos projetos da SUSIS (gestores, equipe, stakeholders e colaboradores), padronizando a metodologia, processos, papéis e responsabilidades. A adoção visa aumentar a previsibilidade, eficiência, minimizar riscos e garantir que os objetivos sejam atingidos dentro do prazo, orçamento e qualidade esperada, fortalecendo a gestão de projetos.



2. APRESENTAÇÃO DA METODOLOGIA

A Metodologia de Desenvolvimento de Software adotada pela SUSIS no Tribunal de Justiça do Acre (TJAC) foi desenvolvida para oferecer uma abordagem sistemática e consistente para a condução de projetos de tecnologia da informação. A metodologia é composta por uma série de processos, ferramentas e técnicas que são aplicados em todas as fases do ciclo de vida de um projeto, a aplicação desta metodologia promove a transparência, facilita a comunicação entre os stakeholders e minimiza os riscos, contribuindo para o sucesso dos projetos e a melhoria contínua dos serviços prestados pelo TJAC.

Importante ressaltar, que, visando adequação às exigências do iGovTIC-JUD, em março de 2026, foi realizadas revisões e ajustes nas normas que regulamentam os Processos de Gerenciamento de Projetos de TIC, de Desenvolvimento e de Sustentação de Sistemas no âmbito da SETIC, resultando na revogação da então vigente Portaria nº 2.969, de 11/07/2024 e publicação da Portaria nº 1135, de 27 de março de 2026.

Esta mudança fez com que a presente Metodologia passe a atuar como parte do Processo de Gerenciamento de Projetos da SETIC.

2.1. Qual o objetivo da Metodologia de Desenvolvimento de Software na SETIC?

O objetivo da Metodologia de Desenvolvimento de Software na SETIC é proporcionar um conjunto de diretrizes e melhores práticas para a condução de projetos de tecnologia da informação no TJAC. Esta metodologia visa assegurar que todos os projetos sejam gerenciados de forma consistente e padronizada, promovendo a entrega de resultados de alta qualidade, dentro dos prazos estabelecidos e conforme os orçamentos alocados. Os objetivos específicos incluem:

- **Melhorar a eficiência e a eficácia:** Otimizar o uso de recursos e garantir a entrega de projetos de TI que atendam às necessidades do Tribunal.
- **Minimizar riscos:** Identificar, analisar e mitigar riscos ao longo do ciclo de vida do projeto.



- **Promover a transparência:** Fornecer uma visão clara do andamento dos projetos para todas as partes interessadas.
- **Assegurar a qualidade:** Implementar práticas de controle de qualidade para garantir que os produtos e serviços entregues estejam em conformidade com os padrões estabelecidos.
- **Facilitar a comunicação:** Estabelecer canais de comunicação eficientes entre todos os envolvidos nos projetos.

2.2. Divisão desta Metodologia

A Metodologia de Desenvolvimento de Software adotada pela SUSIS é dividida em quatro fases principais, que abrangem desde a concepção inicial do projeto até a sua entrega final. A seguir, apresentamos a estrutura desta metodologia:

1) Abertura do Projeto

- a) SETIC recebe do solicitante o pedido de abertura do projeto pelo SEI. Cada projeto deve ter um processo próprio no Sistema Eletrônico de Informações (SEI). Juntamente com o pedido de abertura, o solicitante deverá anexar o formulário FOR-SETIC-001 (Anexos). SETIC encaminha o pedido à SUSIS, que deve tomar conhecimento da necessidade do solicitante e verificar a viabilidade do projeto.
- b) SUSIS manifesta viabilidade de execução e disponibilidade de equipe. A SUSIS pode opinar sobre a viabilidade da demanda, contudo, a decisão final é do GAPRE.
- c) SETIC encaminha instruções para o CGTIC. A SETIC instruirá ao CGTIC a demanda, juntamente com sua avaliação técnica e disponibilidade, para ser apreciado pelo Comitê. CGTIC avalia o pedido de abertura do projeto considerando as demandas existentes. Nesta etapa, o CGTIC deve realizar uma avaliação de viabilidade, considerando as habilidades disponíveis na SUSIS e projetos já em andamento, bem como a prioridade de cada projeto.
- d) CGTIC encaminha ao GAPRE para deliberação final.
O CGTIC encaminha o processo ao GAPRE



(Gabinete de Presidência) para deliberação via SEI.

GAPRE informa a decisão final ao solicitante e SETIC pelo SEI.

A resposta deve ser incluída no mesmo processo em que o solicitante requisitou o software.

Atores envolvidos: SETIC, SUSIS, CGTIC e GAPRE e solicitante.

Artefatos produzidos: Processo no SEI.

2) Modelagem

a) SUSIS designa uma equipe de desenvolvimento para atender a demanda
Considerando as habilidades técnicas de cada equipe.

b) SUSIS define Product Owner (PO)
Considerando o conhecimento negocial e disponibilidade das partes interessadas.

c) SUSIS agenda reunião com solicitante
Via processo no SEI.

d) Product Owner se reúne com solicitante para realizar Levantamento de Requisitos, Diagrama de caso de uso e definir a frequência de reuniões
A partir deste Levantamento de Requisitos o PO deve criar o registro do backlog de modo que a equipe de desenvolvedores consiga acessar para definir as backlogs das Sprints. O PO também deve confeccionar a Ata da Reunião.

e) Equipe de desenvolvimento realiza: Protótipo, Diagrama de classes e Cronograma do projeto
Com base nas informações disponibilizadas pelo PO, que deve avaliar os artefatos produzidos posteriormente, para verificar se condiz como escopo que o solicitante informou.

Atores envolvidos: SUSIS, equipe de desenvolvimento, Product Owner e solicitante.

Artefatos produzidos: Levantamento de Requisitos, Diagrama de caso de uso, Ata da reunião, Protótipo, Diagrama de classes e Cronograma do Projeto.

3) Desenvolvimento

a) Product Owner monitora e altera o Backlog do produto quando necessário



Quando observado incoerência com o pedido do solicitante, ou quando o solicitante informar alterações no pedido, o PO altera o backlog e informa as alterações para a equipe de desenvolvimento.

- b) Equipe de desenvolvimento, a partir do Backlog, define o Backlog da Sprint

A cada início de Sprint o planejamento deve ser realizado a partir das informações contidas no Backlog.

- c) Equipe de desenvolvimento realiza as entregas da Sprint e apresenta ao Product Owner

A cada final de Sprint as alterações são informadas ao PO, que avaliará a coerência do feito.

- d) Product Owner realiza reunião com solicitante na frequência combinada para mostrar as entregas

Garantindo que o produto está sendo construído de acordo com as necessidades do solicitante.

A etapa de desenvolvimento se repete até a completa execução da construção do pedido do solicitante.

Atores envolvidos: Product Owner, equipe de desenvolvimento, solicitante.

Artefatos produzidos: Backlog das Sprints.

4) Entrega

- a) SUSIS agenda entrega com o solicitante

i) Via processo no SEI.

- b) Product Owner realiza entrega, treinamento e define período de teste

i) Na data e local combinados via processo no SEI, o PO realiza a entrega e faz o treinamento com o solicitante. O período de teste pode variar e deve ser combinado com o solicitante.

- c) Solicitante testa o software

i) O teste deve ser realizado em seu real ambiente de utilização pelo período acordado anteriormente com o PO. **A partir do teste, o solicitante pode encontrar falhas ou aceitar a solução.**



ii) Solicitante encontrou falhas

(1) Solicitante indica alterações necessárias

Via processo no SEI, preferencialmente com detalhes e prints da atividade que estava realizando quando o erro ocorreu (se for um caso de erro).

iii) Solicitante aceita solução

E informa via processo no SEI.

(1) Equipe de desenvolvimento e solicitante celebram Termo de Homologação, via processo no SEI.

Atores envolvidos: SUSIS, Product Owner e solicitante.

Artefatos produzidos: Relatório de Testes de Homologação e Termo de Aceite Final.

2.3. Atores envolvidos

1) SETIC (Secretaria de Tecnologia da Informação e Comunicação):

- a) Papel: Facilitador inicial e administrador do processo.
- b) Responsabilidades:
 - i) Receber o pedido de abertura do projeto pelo SEI.
 - ii) Encaminhar o pedido à SUSIS para avaliação de viabilidade.
 - iii) Instruir o processo com base nas decisões da SUSIS.
 - iv) Coordenar a comunicação entre os diferentes atores do projeto.
- c) Competências:
 - i) Conhecimento no uso do SEI para registro e acompanhamento dos processos.
 - ii) Habilidade de comunicação e coordenação entre equipes.
 - iii) Compreensão das necessidades de TI e das prioridades organizacionais.

2) SUSIS (Subsecretaria de Sistemas de Informações):

- a) Papel: Avaliador técnico e gestor do desenvolvimento.
- b) Responsabilidades:



- i) Avaliar a viabilidade de execução e disponibilidade de equipe para o projeto.
 - ii) Designar a equipe de desenvolvimento e definir o Product Owner (PO).
 - iii) Viabilizar reuniões com o solicitante.
- c) Competências:
- i) Profundo conhecimento técnico e de recursos disponíveis.
 - ii) Capacidade de avaliação de viabilidade e planejamento de recursos.
 - iii) Habilidade de gestão de projetos e liderança de equipes.

3) Product Owner (PO):

- a) Papel: Representante dos stakeholders e gerente de requisitos.
- b) Responsabilidades:
- i) Realizar o levantamento de requisitos junto ao solicitante.
 - ii) Definir e priorizar o backlog do produto e das Sprints.
 - iii) Monitorar o progresso do projeto e ajustar conforme necessário.
 - iv) Realizar reuniões periódicas com o solicitante para revisão das entregas.
- c) Competências:
- i) Habilidade de comunicação e negociação com stakeholders.
 - ii) Conhecimento em gerenciamento ágil de projetos e definição de requisitos.
 - iii) Capacidade de priorização e tomada de decisões.

4) Equipe de Desenvolvimento:

- a) Papel: Implementadores técnicos do projeto.
- b) Responsabilidades:
- i) Desenvolver o software conforme os requisitos definidos no backlog.
 - ii) Realizar protótipos, diagramas e outros artefatos técnicos necessários.
 - iii) Participar de reuniões de revisão de Sprint e ajustar conforme feedback.
- c) Competências:



- i) Conhecimento técnico em desenvolvimento de software e metodologias ágeis.
- ii) Habilidade de trabalhar em equipe e seguir cronogramas definidos.
- iii) Capacidade de adaptação e resolução de problemas técnicos.

5) CGTIC (Comitê de Governança de Tecnologia da Informação e Comunicação):

- a) Papel: Avaliador estratégico e decisor.
- b) Responsabilidades:
 - i) Avaliar o pedido de abertura do projeto considerando as demandas existentes.
 - ii) Decidir sobre a aprovação ou negação do projeto.
 - iii) Encaminhar o processo ao GAPRE para deliberação final.
- Competências:
 - iv) Capacidade de avaliação estratégica e alinhamento com prioridades organizacionais.
 - v) Conhecimento em governança de TI e gestão de portfólio de projetos.
 - vi) Habilidade de tomada de decisões informadas e justificadas.

6) GAPRE (Gabinete da Presidência):

- a) Papel: Deliberador final e comunicador.
- b) Responsabilidades:
 - i) Deliberar sobre os projetos encaminhados pelo CGTIC.
 - ii) Informar ao solicitante e à SETIC sobre as decisões tomadas.
 - iii) Redefinir prioridades com base nas novas aprovações de projetos.
- c) Competências:
 - i) Conhecimento em planejamento estratégico e recursos organizacionais.
 - ii) Habilidade de comunicação clara e eficaz com todos os atores.
 - iii) Capacidade de gestão de prioridades e alocação de recursos.

7) Solicitante:

- a) Papel: Iniciador do projeto e validador final.



- b) Responsabilidades:
 - i) Iniciar o processo solicitando a abertura do projeto pelo SEI.
 - ii) Participar das reuniões com SUSIS e PO para definição de requisitos e revisão de entregas.
 - iii) Realizar testes finais e homologação do software entregue.
- c) Competências:
 - i) Habilidade de definição clara e detalhada das necessidades e requisitos do projeto.
 - ii) Capacidade de comunicação e colaboração com a equipe de desenvolvimento.
 - iii) Conhecimento em testes de homologação e validação de software.

Estes atores interagem ao longo das fases de abertura, modelagem, desenvolvimento e entrega do projeto, garantindo que as necessidades do solicitante sejam atendidas de forma eficiente e alinhada às prioridades organizacionais.

2.4. Artefatos do Projeto

Os projetos gerenciados pela metodologia da SUSIS na SETIC envolvem a produção de diversos artefatos ao longo de suas fases. Cada artefato desempenha um papel crucial no planejamento, desenvolvimento e entrega do software, garantindo o alinhamento com as expectativas do solicitante e a qualidade do produto final. Abaixo estão os principais artefatos utilizados:

1) Processo no SEI:

Documento eletrônico que registra todas as etapas e decisões relacionadas ao projeto, desde sua abertura até a conclusão. É utilizado para formalizar o pedido de abertura, decisões de aprovação, comunicação com os atores envolvidos e documentação final.

2) Levantamento de Requisitos:



Documento que detalha as necessidades e expectativas do solicitante em relação ao software a ser desenvolvido. Inclui requisitos funcionais e não funcionais, restrições e outras especificações relevantes.

3) Diagrama de Caso de Uso:

Representação gráfica dos principais cenários de uso do software, identificando atores (usuários ou sistemas externos) e suas interações com o sistema. Ajuda a visualizar como o sistema será utilizado na prática.

4) Ata da Reunião:

Registro detalhado das discussões, decisões e ações definidas durante as reuniões entre a equipe de desenvolvimento, o Product Owner e o solicitante. Inclui pontos discutidos, responsáveis pelas ações e prazos estabelecidos.

5) Protótipo:

Versão inicial do software que simula o funcionamento e a aparência da interface do usuário. É utilizado para validar conceitos, coletar feedback inicial e ajustar requisitos antes da implementação completa.

6) Diagrama de Classes:

Representação estrutural do sistema, mostrando classes, atributos, métodos e suas relações. Auxilia no design da arquitetura do software e na definição das entidades principais do sistema.

7) Cronograma do Projeto:

Planejamento detalhado das atividades, marcos e prazos do projeto. Define as datas de início e fim de cada fase, entregas esperadas e recursos necessários para cumprir os objetivos do projeto.

8) Backlog das Sprints:



Lista de todas as funcionalidades a serem desenvolvidas, priorizadas pelo Product Owner e divididas em incrementos menores chamados de sprints. Inclui itens do backlog do produto que serão trabalhados em cada iteração.

9) Relatório de Testes de Homologação:

Documento que registra os resultados dos testes finais realizados pelo solicitante para validar se o software atende aos requisitos acordados. Inclui casos de teste, resultados obtidos e problemas identificados.

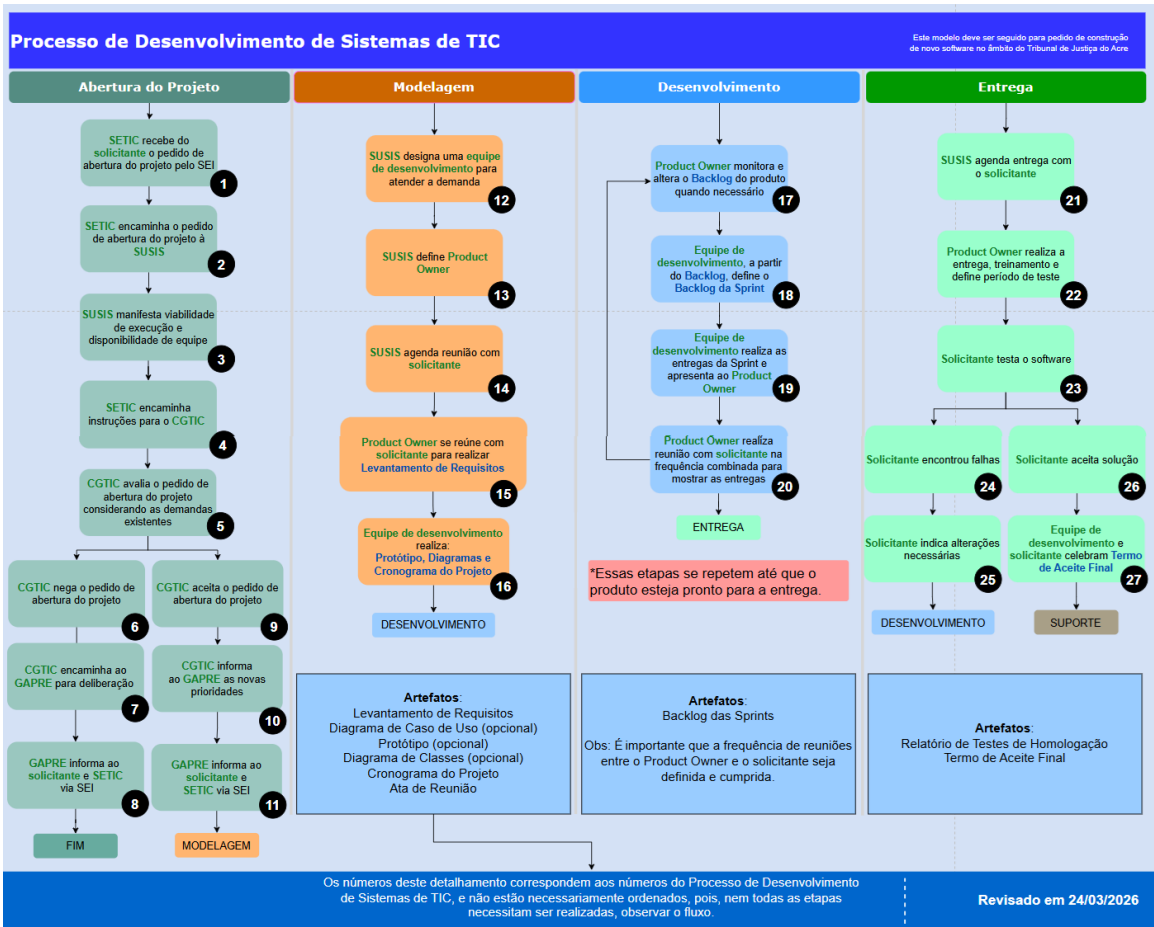
10) Termo de Aceite Final:

Documento formal assinado pelo solicitante, confirmando que o software entregue atende às suas expectativas e está pronto para ser colocado em produção. Marca o encerramento do projeto.

Esses artefatos são essenciais para garantir a transparência, qualidade e sucesso dos projetos desenvolvidos pela SUSIS na SETIC, facilitando a comunicação entre as partes envolvidas e assegurando que todas as etapas sejam documentadas de forma adequada.

2.5. Fluxo do Gerenciamento de Projetos TIC

Para garantir a clareza e a eficiência no gerenciamento de projetos, a metodologia adotada pela SUSIS na SETIC é representada por um fluxo detalhado que abrange todas as fases do ciclo de vida do projeto. Este fluxo ilustra a sequência de atividades, os pontos de decisão e a interação entre os diversos atores envolvidos. A seguir, apresentamos o desenho do fluxo do ciclo de vida do projeto.





3. CICLO DE VIDA

O ciclo de vida dos projetos gerenciados pela metodologia da SUSIS na SETIC segue uma abordagem estruturada para garantir que cada fase seja cuidadosamente planejada, executada e controlada. Este ciclo de vida é composto por várias etapas inter-relacionadas, cada uma com seus objetivos específicos e entregáveis. Abaixo está uma visão geral das principais fases:

1) **Abertura do Projeto:**

Nesta fase inicial, o processo é iniciado com a solicitação de abertura do projeto pelo solicitante através do Sistema Eletrônico de Informações (SEI).

A SETIC recebe e encaminha o pedido à SUSIS para avaliação de viabilidade, considerando recursos disponíveis e alinhamento com as prioridades estratégicas.

O Comitê de Governança de Tecnologia da Informação e Comunicação (CGTIC) avalia o pedido e decide sobre sua aprovação ou negação, encaminhando-o ao GAPRE para deliberação final.

2) **Modelagem:**

Uma vez aprovado, o projeto avança para a fase de modelagem, onde a SUSIS designa uma equipe de desenvolvimento e define o Product Owner (PO) responsável por representar os interesses do solicitante.

São realizadas reuniões para o levantamento detalhado de requisitos, elaboração do Diagrama de Caso de Uso e definição do cronograma do projeto.

A equipe de desenvolvimento cria protótipos e diagramas de classes que servirão como base para o desenvolvimento futuro, garantindo que o escopo do projeto esteja claramente definido e compreendido por todos os envolvidos.

3) **Desenvolvimento:**

Na fase de desenvolvimento, a equipe de desenvolvimento trabalha de acordo com o backlog das Sprints definido pelo PO.

São realizadas iterações curtas e incrementais, conhecidas como Sprints,



onde as funcionalidades são implementadas, testadas e entregues.

O PO monitora e ajusta o backlog do produto conforme necessário, garantindo que as prioridades estejam alinhadas com as necessidades atuais do solicitante.

4) Entrega:

Após o desenvolvimento das funcionalidades necessárias, o projeto avança para a fase de entrega.

O PO coordena a entrega oficial do software ao solicitante, realiza treinamentos se necessário, e define um período de teste para que o solicitante possa validar o funcionamento do sistema em seu ambiente real de operação.

O solicitante realiza testes de homologação para verificar se todas as suas expectativas foram atendidas. Caso identifique problemas, estes são registrados e endereçados pela equipe de desenvolvimento.

5) Encerramento:

Com a aprovação do solicitante, é assinado o Termo de Aceite Final, formalizando o encerramento do projeto.

São compilados todos os artefatos gerados durante o ciclo de vida do projeto, incluindo documentação, diagramas, relatórios de testes e qualquer outro registro relevante.

O projeto é arquivado e lições aprendidas são documentadas para futuras referências e melhoria contínua dos processos.

Este ciclo de vida estruturado permite à SUSIS na SETIC gerenciar eficazmente seus projetos, assegurando que cada etapa seja executada de forma transparente, com foco na qualidade e na entrega de valor ao solicitante.



3.1 Eventos no Ciclo de Vida do Projeto

Durante o ciclo de vida dos projetos gerenciados pela metodologia, diversos eventos são realizados para garantir o alinhamento, monitoramento e tomada de decisões estratégicas. Esses eventos são essenciais para manter o projeto dentro dos prazos, orçamentos e padrões de qualidade estabelecidos. A seguir, são destacados os principais eventos:

1) Solicitação de Abertura do Projeto:

Evento inicial onde o solicitante formaliza a necessidade de um novo projeto através do Sistema Eletrônico de Informações (SEI).

Este evento inicia o fluxo de aprovação e avaliação pelo Comitê de Governança de Tecnologia da Informação e Comunicação (CGTIC), que decide sobre a viabilidade e prioridade do projeto.

2) Avaliação de Viabilidade pela SUSIS:

A SUSIS realiza uma avaliação técnica e de recursos para determinar a viabilidade de execução do projeto.

Durante este evento, são analisadas as capacidades da equipe disponível, a complexidade técnica e o alinhamento estratégico com as metas organizacionais.

3) Deliberação pelo CGTIC e GAPRE:

Após a avaliação inicial, o CGTIC decide sobre a aprovação ou negação do projeto, com base na recomendação da SUSIS.

Projetos são encaminhados ao GAPRE para deliberação final, garantindo uma decisão alinhada com as prioridades organizacionais e disponibilidade de recursos.

4) Reuniões de Levantamento de Requisitos:

Eventos periódicos onde o Product Owner (PO) e a equipe de desenvolvimento se reúnem com o solicitante para detalhar os requisitos do projeto.



Nestas reuniões, são discutidas as funcionalidades desejadas, restrições, prioridades e qualquer outra informação relevante para o desenvolvimento do software.

5) Revisões de Sprint:

Realizadas ao final de cada Sprint, as revisões são eventos onde a equipe de desenvolvimento demonstra as funcionalidades implementadas ao PO e ao solicitante.

Durante estas revisões, são identificados ajustes necessários e novas prioridades, garantindo a entrega contínua de valor ao longo do projeto.

6) Entregas e Treinamentos:

Eventos onde o PO realiza a entrega formal das funcionalidades desenvolvidas ao solicitante, acompanhado de treinamentos se necessário.

Estes eventos são essenciais para garantir que o solicitante esteja preparado para usar o software de forma eficaz e realizar testes de homologação adequados.

7) Homologação e Aceitação Final:

Evento final onde o solicitante realiza testes de homologação para validar se todas as suas expectativas foram atendidas.

Após a aprovação pelo solicitante, é assinado o Termo de Aceite Final, marcando o encerramento formal do projeto.

Estes eventos são planejados e executados de forma a garantir a transparência, comunicação eficaz e tomada de decisões informadas ao longo do ciclo de vida do projeto, assegurando que todos os envolvidos estejam alinhados com os objetivos e resultados esperados.



4. FERRAMENTAS DE APOIO

Para assegurar a eficiência e a eficácia no gerenciamento dos projetos, a metodologia adotada utiliza diversas ferramentas de apoio. Essas ferramentas auxiliam nas diferentes fases do ciclo de vida do projeto, desde a concepção até a entrega, proporcionando suporte técnico, organizacional e colaborativo. A seguir, são apresentadas as principais ferramentas de apoio utilizadas.

4.1. GitLab

O GitLab é uma plataforma de DevOps completa que fornece uma solução integrada para controle de versão, integração contínua (CI), entrega contínua (CD), e monitoramento de desempenho. Ele permite que equipes de desenvolvimento colaborem de forma eficaz em projetos de software, oferecendo ferramentas para planejamento, gerenciamento de código-fonte, automação de pipelines de CI/CD, e muito mais.

A fase de desenvolvimento é onde o GitLab se torna uma ferramenta central para a equipe de desenvolvimento.

- **Controle de Versão:** Desenvolvedores utilizam Git para versionar o código-fonte, permitindo rastrear alterações, colaborar em branches, e revisar código por meio de merge requests.
- **Pipelines de CI/CD:** Pipelines de integração contínua e entrega contínua são configurados no GitLab para automatizar a construção, teste e implantação do código. Isso assegura que o código integrado ao repositório é sempre testado e pronto para produção.
- **Issues e Boards:** O GitLab é usado para gerenciar tarefas e bugs através de issues e boards. O Product Owner pode criar e priorizar issues, que são atribuídas aos desenvolvedores para execução.
- **Deploy Automatizado:** Utilizando pipelines de CD, o software é automaticamente implantado nos ambientes de teste e produção, minimizando erros humanos e acelerando o processo de entrega.
- **Documentação:** a documentação do projeto (cronograma, protótipos e diagramas), é armazenada no repositório GitLab, assegurando que todas as informações relevantes estão centralizadas e acessíveis.



4.2. Metodologia Ágil

A Metodologia Ágil é um conjunto de práticas de gerenciamento de projetos e desenvolvimento de software que enfatiza a flexibilidade, a colaboração e a entrega contínua de valor. Baseada no Manifesto Ágil, que valoriza indivíduos e interações sobre processos e ferramentas, software funcionando sobre documentação extensiva, colaboração com o cliente sobre negociação de contratos e resposta a mudanças sobre seguir um plano, a metodologia ágil visa adaptar-se rapidamente às mudanças e proporcionar entregas frequentes e incrementais de valor ao cliente.

A Metodologia Ágil é aplicada pela SUSIS para garantir que os projetos de desenvolvimento de software sejam conduzidos de maneira eficiente, colaborativa e adaptável. A seguir, o detalhamento de como a Metodologia Ágil é implementada em cada fase do ciclo de vida do projeto:

1) Abertura do Projeto

Na fase de abertura do projeto, os princípios ágeis são adotados para garantir que todos os requisitos e expectativas sejam claros desde o início. Isso é feito através de:

- Reuniões Iniciais: Encontros com stakeholders para discutir as necessidades e objetivos do projeto.
- Manifestação de Viabilidade: Avaliação rápida e colaborativa da viabilidade do projeto.

2) Modelagem

Na fase de modelagem, a equipe utiliza práticas ágeis para planejar e preparar o desenvolvimento:

- Designação de Equipe e PO: A equipe de desenvolvimento é designada, e um Product Owner (PO) é nomeado para representar os interesses do cliente.
- Reuniões de Levantamento de Requisitos: Reuniões com stakeholders para entender detalhadamente os requisitos do projeto e criar um backlog



inicial de tarefas.

- Protótipos e Diagramas: Criação de protótipos e diagramas que são revisados continuamente para garantir que atendem às necessidades do cliente.

3) Desenvolvimento

Durante a fase de desenvolvimento, a equipe adota práticas ágeis como Scrum ou Kanban:

- Sprints: Divisão do trabalho em ciclos curtos e iterativos chamados Sprints, geralmente de duas a quatro semanas.
- Daily Stand-ups: Reuniões diárias para revisar o progresso, identificar obstáculos e planejar o trabalho do dia.
- Revisões de Sprint: Ao final de cada Sprint, a equipe revisa o que foi feito com os stakeholders e ajusta o backlog conforme necessário.
- Backlog de Produto e Sprint: O backlog é continuamente priorizado e refinado pelo PO, com base no feedback do cliente e nas mudanças de requisitos.

4) Entrega

Na fase de entrega, a Metodologia Ágil assegura que o software seja entregue de forma incremental e iterativa:

- Deploys Frequentes: Entregas frequentes de software funcional, com automação de pipelines de CI/CD no GitLab para garantir qualidade e rapidez.
- Testes e Feedback: Testes contínuos e coleta de feedback dos usuários para ajustes e melhorias rápidas.
- Treinamento e Suporte: Entregas incluem treinamento e suporte ao cliente, garantindo que eles estejam preparados para utilizar o software entregue.

A aplicação da Metodologia Ágil permite uma abordagem dinâmica e colaborativa ao desenvolvimento de software, garantindo que os projetos sejam entregues com qualidade, de forma eficiente e adaptável às necessidades do



cliente. A utilização de práticas ágeis em combinação com ferramentas como o GitLab fortalece a capacidade da equipe de fornecer soluções robustas e de alto valor.

4.3. SEI

O Sistema Eletrônico de Informações (SEI) é uma plataforma de gestão de processos administrativos eletrônicos que visa a transformação digital da administração pública. O SEI permite a tramitação eletrônica de documentos e processos, promovendo maior agilidade, transparência e segurança na gestão de informações.

Na metodologia de gerenciamento de projetos adotada, o SEI desempenha um papel fundamental em diversas fases do ciclo de vida do projeto. A seguir é detalhado como o SEI é utilizado em cada fase:

1) Abertura do Projeto

- **Recebimento do Pedido:** O processo de abertura do projeto começa com o recebimento do pedido pelo SEI. Cada projeto deve ter um processo próprio no SEI, onde todas as informações e documentos relacionados ao projeto serão centralizados.
- **Tramitação do Pedido:** O pedido de abertura do projeto é encaminhado pela SETIC à SUSIS através do SEI, onde a viabilidade é verificada e a decisão é documentada.
- **Comunicação de Decisões:** Todas as decisões tomadas pelo CGTIC, incluindo a aceitação ou negação do projeto, são comunicadas através do SEI. Isso garante que todas as partes interessadas estejam informadas e que o processo seja transparente.

2) Modelagem

- **Registro de Reuniões e Documentos:** Reuniões com o solicitante e levantamentos de requisitos são registrados no SEI, incluindo atas de reuniões, diagramas de caso de uso e cronogramas de projeto. Isso assegura que todas as informações críticas estejam centralizadas e acessíveis.
- **Colaboração e Aprovação:** Documentos como protótipos e diagramas de



classes são compartilhados e revisados no SEI, facilitando a colaboração entre a equipe de desenvolvimento e o solicitante.

3) Desenvolvimento

- Monitoramento do Progresso: O SEI é utilizado para monitorar o progresso do projeto, registrar entregas de Sprints e documentar feedbacks e mudanças solicitadas pelo cliente.
- Documentação de Alterações: Qualquer alteração no backlog do produto ou nas prioridades do projeto é registrada no SEI, garantindo que todas as mudanças sejam documentadas e aprovadas de forma transparente.

4) Entrega

- Agendamento e Comunicação: O agendamento da entrega do projeto, treinamentos e definições de período de teste são comunicados através do SEI, garantindo que todas as partes interessadas estejam cientes das datas e atividades planejadas.
- Relatórios de Testes e Aceite Final: Os resultados dos testes de homologação são documentados no SEI, juntamente com o Termo de Aceite Final, oficializando a entrega e a aceitação do projeto pelo solicitante.



5. PROCESSOS DE COMUNICAÇÃO

O processo de comunicação é uma parte crucial da Metodologia de Desenvolvimento de Software. A comunicação eficaz garante que todas as partes interessadas estejam informadas, engajadas e alinhadas com os objetivos e progresso do projeto. Este processo envolve a troca constante de informações entre os membros da equipe, stakeholders, solicitantes e outras partes envolvidas no projeto.

5.1. Objetivos do Processo de Comunicação

- 1) **Transparência:** Garantir que todas as informações relevantes sobre o projeto sejam compartilhadas de forma transparente e acessível.
- 2) **Engajamento:** Manter todas as partes interessadas engajadas e informadas sobre o progresso, mudanças e decisões do projeto.
- 3) **Alinhamento:** Assegurar que todos os envolvidos estejam alinhados com os objetivos, cronogramas e prioridades do projeto.
- 4) **Colaboração:** Promover a colaboração entre os membros da equipe e stakeholders, facilitando a resolução de problemas e a tomada de decisões.
- 5) **Velocidade:** Assegurar que as informações sejam trocadas rapidamente, permitindo respostas e ações rápidas em relação a problemas e mudanças no projeto.

5.2. Ferramentas de Comunicação

- 1) **SEI (Sistema Eletrônico de Informações):** Utilizado para registrar e tramitar documentos, reuniões e decisões de forma oficial e centralizada.
- 2) **E-mails:** Ferramenta principal para comunicações rápidas e formais entre a equipe, stakeholders e solicitantes.
- 3) **Reuniões:** Realizadas regularmente para discutir o progresso do projeto, resolver problemas e alinhar expectativas.
- 4) **GitLab:** Utilizado para a gestão de código e documentação técnica, facilitando a colaboração entre os desenvolvedores.
- 5) **Videoconferências:** Utilizadas para reuniões remotas, garantindo a participação de todos, independentemente da localização física.



- 6) Mensagens Instantâneas: Ferramentas como WhatsApp ou Discord são usadas para comunicações rápidas e informais.

5.3. Estrutura do Processo de Comunicação

- 1) Reuniões Iniciais
 - a) Objetivo: Alinhar expectativas, entender os requisitos do solicitante e definir o escopo inicial do projeto.
 - b) Participantes: Solicitante, Product Owner, membros chave da SUSIS.
 - c) Ferramenta: SEI para registro de atas e e-mails para confirmação de agendas.
- 2) Reuniões Regulares
 - a) Objetivo: Discutir o progresso do projeto, resolver problemas e ajustar o planejamento conforme necessário.
 - b) Participantes: Product Owner, equipe de desenvolvimento e, quando necessário, o solicitante.
 - c) Ferramenta: SEI para registro de atas, videoconferências para discussão e GitLab para acompanhamento de tarefas.
- 3) Relatórios de Progresso
 - a) Objetivo: Manter todas as partes interessadas informadas sobre o progresso do projeto, próximos passos e eventuais riscos.
 - b) Participantes: Product Owner e stakeholders.
 - c) Ferramenta: SEI para documentação formal e GitLab para registro do Backlog de cada Sprint.
- 4) Feedback e Ajustes
 - a) Objetivo: Coletar feedback contínuo do solicitante e fazer ajustes necessários no projeto.
 - b) Participantes: Product Owner, equipe de desenvolvimento e solicitante.
 - c) Ferramenta: Reuniões periódicas e SEI para documentação de feedback e alterações.
- 5) Reuniões de Revisão e Planejamento de Sprints
 - a) Objetivo: Revisar o trabalho realizado durante a Sprint, coletar feedback e planejar a próxima Sprint.
 - b) Participantes: Equipe de desenvolvimento e Product Owner.



- c) Ferramenta: GitLab para gerenciamento de backlog e tarefas, videoconferências e SEI para registro de atas.
- 6) Reuniões de Entrega e Treinamento
- a) Objetivo: Entregar o produto final ao solicitante, realizar o treinamento necessário e definir o período de teste.
 - b) Participantes: Product Owner e solicitante. A equipe de desenvolvimento pode optar por participar da entrega.
 - c) Ferramenta: SEI para documentação de entrega e treinamento, videoconferências para sessões de treinamento ou reunião presencial.

O processo de comunicação é fundamental para o sucesso dos projetos gerenciados pela SUSIS. Através de ferramentas como o SEI, e-mails, reuniões, GitLab e Discord, assegura-se que todas as informações sejam compartilhadas de forma eficiente e transparente, promovendo a colaboração, alinhamento e velocidade na resposta entre todos os envolvidos no projeto.



6. ANEXOS



Solicitação de Desenvolvimento

Informações da SETIC – Marque com um “X”

	Alto	Médio	Baixo		
Grau de alinhamento com a Estratégia Nacional do Poder Judiciário	()	()	()		
Alinhamento com o Prêmio CNJ de Qualidade	()	()	()		
Alinhamento com o PEI do Órgão	()	()	()		
Alinhamento com o PDTIC do Órgão	()	()	()		
Alinhamento com o Plano de Ação do Biênio	()	()	()		
Prazo necessário para execução do projeto de implantação do serviço	()	()	()		
Deve-se apontar os riscos da falta do sistema no TJAC, bem como as variantes de risco para seu desenvolvimento	()	()	()		
Custo necessário para o desenvolvimento	Até R\$ 80 mil ou sem custo ()	De R\$ 80 mil a R\$ 200 mil ()	Acima de R\$ 200 mil ()		
A pontuação será considerada conforme o risco da instituição ficar sem o sistema	Improvável ()	Pouco provável ()	Provável ()	Muito provável ()	Quase certo ()

Resultado - Informações da DITEC

Viável?	<input type="checkbox"/> Sim	<input type="checkbox"/> Não	Prazo:
Parecer da SETIC			

Aprovação pelo usuário solicitante

Responsável	Assinatura	Data

Priorizado pelo CGTIC - Informações da SETIC

Data	Pontuação recebida



7. HISTÓRICO DE REVISÕES

Versão	Data	Resumo das alterações	Autor
2026	03/2026	Atualização resultante da Portaria nº 1135, de 27/03/2026	Subsecretaria de Sistemas de Informação
2026	01/2026	Atualização do documento MDS para versão 0.0.2	Assessoria da SETIC
2024	07/2024	Revisão da versão inicial do documento MDS	Gerência de Sistemas
2023	07/2023	Elaboração da versão inicial do documento MDS	Gerência de Sistemas