

ESPECIFICAÇÃO DO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

Projeto: Sistema Distribuído (PDS Corporativo)

Equipe: 5 Desenvolvedores

Modelo: Ágil Híbrido (Kanban + XP + DevOps)

* VISÃO GERAL

Este processo foi desenhado para uma equipe enxuta com horários distintos (estágios/aulas), priorizando a comunicação assíncrona e a redução de retrabalho. O modelo adota um Fluxo Contínuo (Kanban) para gestão de tarefas, reforçado pela prática de "Design First" (documentação antes do código) e sustentado por uma esteira de CI/CD (Integração e Entrega Contínuas) que automatiza testes e implantação.

* DIAGRAMA DE ATIVIDADES (FLUXO DO CICLO DE VIDA)

Abaixo, a representação textual do fluxo para modelagem no diagrama (BPMN ou UML):

- * Início (Reunião Semanal) -> Definição de Prioridades
 - * Backlog -> Seleção de Tarefa
 - * Análise e Design Prévio (Obrigatório)
 - * Desenvolvimento (Branch Isolada + Testes Locais)
 - * Solicitação de Integração (Pull Request)
 - * Revisão de Código (Peer Review)
 - * Reprovado? -> Volta para Desenvolvimento.
 - * Aprovado? -> Segue para Merge.
 - * Automação CI/CD (O "Robô")
 - * Execução de Testes Automatizados -> Falha? Notifica equipe.
 - * Build e Deploy em Homologação (Staging) -> Sucesso? Segue.
 - * Conclusão (Done)
-

* DESCRIÇÃO DAS ATIVIDADES

A. Planejamento e Alinhamento (Gestão)

- * Frequência: Semanal (Reunião Única).
- * Descrição: Ocorre o alinhamento síncrono da equipe. A pauta é dividida em:
 - * Review: O que foi entregue e implantado na semana anterior.
 - * Planning: Priorização das tarefas do Backlog para a próxima semana.
 - * Impedimentos: Resolução conjunta de bloqueios técnicos ou de arquitetura.

B. Análise e Design (Engenharia)

- * Responsável: Desenvolvedor da tarefa.

- * Descrição: Antes de iniciar a codificação, o desenvolvedor deve documentar a solução técnica no card da tarefa ou na Wiki do projeto.

- * Objetivo: Validar a lógica, contrato de APIs e modelagem de dados antes de investir tempo em código, minimizando retrabalho (Design Up-Front Ágil).

C. Desenvolvimento e Teste Local (Execução)

- * Prática: Feature Branch Workflow.

- * Descrição: O desenvolvimento ocorre em uma ramificação isolada (ex: feature/login-usuario) para permitir concorrência sem conflitos na base principal.

- * Requisito: A implementação deve ser acompanhada da criação de testes unitários locais.

D. Revisão de Código (Garantia da Qualidade Humana)

- * Prática: Peer Review Assíncrono.

- * Descrição: Após finalizar a tarefa, o desenvolvedor abre um Pull Request (PR). Outro membro da equipe revisa se o código atende à documentação (Etapa B) e aos padrões do projeto.

E. Pipeline Automatizado (CI/CD)

- * Responsável: Automação ("O Robô").

- * Descrição: Ao aprovar o PR, o sistema de Integração Contínua é acionado automaticamente para:

- * Verificar conflitos de merge.

- * Rodar a bateria completa de testes (Unitários e Integração).

- * Realizar o deploy automático no ambiente de Homologação/Staging.

* PAPÉIS E RESPONSABILIDADES

Dada a natureza da equipe (5 integrantes), adota-se papéis rotativos ou acumulados, com foco na colaboração.

- * Product Owner / Líder: Gerencia o Backlog, define prioridades na reunião semanal e valida se a entrega final atende aos requisitos de negócio.

- * Desenvolvedor Full Stack: Atua em todo o ciclo: realiza a Análise/Design, codifica, cria testes automatizados e revisa o código dos pares (Code Review).

- * Engenheiro DevOps: Responsável pela manutenção da infraestrutura, configuração do Pipeline de CI/CD e monitoramento da saúde do sistema distribuído.

* PRÁTICAS ÁGEIS APLICADAS

Para atender aos critérios de PDS Corporativo, as seguintes práticas formalizadas são utilizadas:

- * Kanban (Fluxo Contínuo): Visualização do trabalho e limitação de WIP (Work in Progress), eliminando a rigidez de Sprints que não condizem com a rotina da equipe.

- * Design Up-Front (Ágil): Documentação técnica prévia obrigatória para garantir robustez arquitetural e evitar desperdício de horas de desenvolvimento.

* Continuous Integration & Deployment (CI/CD): Automação total de testes e deploy para garantir que o software esteja sempre em estado implantável.

* Peer Review (Revisão por Pares): Garantia de qualidade descentralizada e compartilhamento de conhecimento técnico entre os membros.

* Feature Branch Workflow: Estratégia de versionamento que permite o trabalho paralelo de múltiplos desenvolvedores sem comprometer a estabilidade da branch principal.