# Plano de Fluxo de Trabalho de Desenvolvimento e Ciclo de Vida do Bug

## 1. Introdução

Este documento descreve o fluxo de trabalho de desenvolvimento de software e o ciclo de vida de um bug, desde a sua identificação até a sua resolução e verificação. O objetivo é padronizar os processos para garantir eficiência e qualidade na entrega de software.

## 2. Fluxo de Trabalho de Desenvolvimento

O fluxo de trabalho de desenvolvimento segue as seguintes etapas:

### 2.1. Planejamento

- **Requisitos:** Coleta e análise dos requisitos do projeto.
- **Design:** Criação da arquitetura e design da solução.
- **Priorização:** Definição das prioridades das funcionalidades e tarefas.

### 2.2. Desenvolvimento

- **Codificação:** Implementação do código-fonte com base nos requisitos e design.
- **Testes Unitários:** Criação e execução de testes unitários para garantir a funcionalidade de componentes individuais.
- **Revisão de Código:** Revisão do código por pares para identificar erros e garantir a qualidade.

### 2.3. Testes e Qualidade

- **Testes de Integração:** Verificação da interação entre diferentes módulos do sistema.
- **Testes de Sistema:** Testes abrangentes para garantir que o sistema atenda aos requisitos funcionais e não funcionais.
- **Testes de Aceitação:** Validação do sistema pelos usuários finais ou stakeholders.

### 2.4. Implantação

- **Homologação:** Implantação em ambiente de homologação para testes finais.
- **Produção:** Implantação em ambiente de produção após aprovação.

### 2.5. Monitoramento e Manutenção

- **Monitoramento:** Acompanhamento do desempenho e comportamento do sistema em produção.
- **Manutenção:** Correção de bugs e implementação de melhorias contínuas.

## 3. Ciclo de Vida do Bug

O ciclo de vida de um bug é composto pelas seguintes fases:

### 3.1. Descoberta e Registro

- **Descoberta:** Um bug é identificado durante o desenvolvimento, testes ou em produção.
- **Registro:** O bug é registrado em uma ferramenta de gerenciamento de bugs (ex: Jira, Trello), contendo informações como:
  - Título claro e conciso.
  - Descrição detalhada (passos para reproduzir, comportamento esperado vs. obtido).
  - Severidade e prioridade.
  - Anexos (screenshots, logs).

### 3.2. Análise e Atribuição

- **Análise:** A equipe de desenvolvimento analisa o bug para entender sua causa e impacto.
- **Atribuição:** O bug é atribuído a um desenvolvedor responsável pela correção.

### 3.3. Correção

- **Implementação:** O desenvolvedor implementa a correção do bug.
- **Testes Unitários:** Testes unitários são criados ou atualizados para cobrir a correção.
- **Revisão de Código:** A correção é revisada por pares.

### 3.4. Verificação e Fechamento

- **Testes de Regressão:** Testes são executados para garantir que a correção não introduziu novos problemas.
- **Verificação:** A equipe de QA verifica a correção do bug no ambiente de testes.
- **Fechamento:** Após a verificação e confirmação da correção, o bug é fechado na ferramenta de gerenciamento.

### 3.5. Reabertura (se necessário)

- Se o bug for reincidente ou a correção não for eficaz, ele pode ser reaberto para nova análise e correção.