

SENAI CENTRO DE TREINAMENTO DA
TECNOLOGIA DA INFORMAÇÃO

DESENVOLVIMENTO DE SISTEMAS

CAIO UGO CAETANO GOMES
JOAO GUILHERME GOMES SANNA
ENZO FONSECA DE OLIVEIRA
BRUNO SOUZA CRUZ

CORREÇÃO DE BUGS E TESTES

BELO HORIZONTE 2025

1. Identificação de Bugs

A identificação de bugs é a primeira etapa crítica no processo de manutenção de um sistema. Detectar erros de forma eficiente permite uma correção mais rápida e evita impactos negativos no uso do sistema. As principais formas adotadas pela equipe são:

Fontes de Identificação:

Relatórios automáticos de erros

O sistema pode gerar relatórios em arquivos de log ou notificações em tempo real quando uma exceção é lançada, como falha de conexão, erro de banco de dados, ou falha de lógica de programação.

Feedback de usuários

Usuários finais são uma das principais fontes de identificação de problemas, pois utilizam o sistema em situações variadas. É importante manter canais abertos como e-mail de suporte, formulários ou chats de atendimento.

Logs do sistema (eventos e exceções)

Análise de logs gravados localmente ou enviados para um servidor. Eles mostram a linha do código que falhou, parâmetros enviados e mensagens de erro, ajudando na reprodução do problema.

2. Processo de Correção de Erros

Após o bug ser identificado, seguimos um processo organizado e seguro para garantir que a correção não afete outras áreas do sistema e que seja devidamente registrada para fins de controle e auditoria.

Etapas do Processo:

Reprodução do Erro

Simular o cenário no ambiente de testes (com dados semelhantes aos reais).

Tentar reproduzir o mesmo fluxo executado pelo usuário.

Confirmar o erro para não aplicar correções em falso-positivos.

Criação de uma Ramificação no Git

Criar uma branch específica para correção do erro, por exemplo:

hotfix/erro-tela-de-login.

Isso evita que a correção afete diretamente o código em produção.

Correção do Código

Análise da causa-raiz do problema (ex: falta de validação, erro de conversão de dados).

Aplicar a correção, seguindo as boas práticas de codificação da equipe.

Validação da Correção

Testes manuais são realizados em cenários semelhantes ao real.

Testes automatizados são executados se disponíveis, para garantir que o erro foi resolvido sem introduzir novos bugs.

Registro da Correção no Controle de Versão

Commit no Git com mensagem padronizada, exemplo:

sql

CopiarEditar

git commit -m "fix: corrigido erro de autenticação na tela de login"

Atualização no changelog ou sistema de tickets (ex: Trello, Jira).

3. Tipos de Testes Utilizados

Testes Manuais

Executados manualmente por um desenvolvedor ou tester, são úteis para:

Verificar formulários (envio, validações, mensagens).

Clicar em botões e links para garantir funcionalidade.

Navegação entre telas e verificação visual.

Testes Automatizados (se houver)

Executados por scripts de teste:

Validação automática de regras de negócio.

Execução de testes unitários com frameworks como NUnit ou xUnit no C#.

Garante rapidez e confiabilidade em alterações frequentes.

Testes de Regressão

Importantes após qualquer correção:
Verificam se outras partes do sistema continuam funcionando.
Impedem que uma correção “quebre” funcionalidades que já estavam estáveis.

4. Boas Práticas Adotadas pela Equipe

TechSolutions

Commits frequentes e significativos para rastreabilidade.
Ambiente de homologação antes de subir correções para produção.
Testes com múltiplos usuários simulados, para garantir estabilidade sob carga.
Checklist de correção e validação, usado em toda entrega de correções.
Documentação das correções no repositório oficial do sistema.

Exemplo Prático

Problema: Usuários relatam que não conseguem fazer login com senhas válidas.

Ação: Verificação dos logs mostra que o método de verificação está sensível a letras maiúsculas.

Correção: Adicionada conversão `.ToLower()` em comparações.

Teste: Reproduzido o erro, corrigido e testado com diferentes senhas.

Registro Git:

```
pgsql
```

CopiarEditar

```
git checkout -b hotfix/login-sensibilidade
```

```
git commit -m "fix: corrigido problema de login sensível a  
letras maiúsculas"
```

Conclusão

A manutenção de sistemas é uma etapa essencial para garantir a continuidade, qualidade e confiabilidade do software após sua entrega. Através da identificação rápida de erros, uso de processos bem definidos e aplicação de testes eficazes, conseguimos corrigir falhas com segurança e minimizar impactos aos usuários.

