

Plano de Manutenção Preventiva de Software

Objetivo

O objetivo deste plano de manutenção preventiva é garantir a continuidade na operação e a saúde do Software “XXXXXXX”

A manutenção preventiva tem como foco a prevenção de falhas, garantindo que o sistema opere de forma eficiente, segura e sem interrupções inesperadas.

Este plano abrange atividades regulares que visam otimizar o desempenho, melhorar a segurança e assegurar que o software seja capaz de suportar as necessidades de negócio a longo prazo.

Atividades de Manutenção Preventiva

1. Backup Regular de Dados: Realizar backups completos no fim de semana e backups incrementais durante a semana dos dados críticos do banco de dados do sistema durante a madrugada, com teste de integridade mensal. Garantir que os backups sejam armazenados em um local seguro e facilmente acessível para recuperação em caso de falha.

2. Atualizações de Software e Segurança: Realizar atualizações mensais de software, bibliotecas e dependências. Além disso, garantir que os patches de segurança sejam aplicados imediatamente após seu lançamento para proteger o sistema contra vulnerabilidades conhecidas.

3. Monitoramento de Performance: Monitoramento contínuo do desempenho do sistema com alertas configurados para detectar uso excessivo de CPU, memória, espaço em disco e tráfego de rede. Relatórios de desempenho devem ser gerados e analisados semanalmente para revisão e otimização. Além de dashboards informativos da saúde dos servidores

4. Rotina de Testes de Recuperação de Desastres: Realizar testes de recuperação de desastres semestralmente para garantir que o sistema pode ser restaurado rapidamente em caso de falha catastrófica. Utilizar backups armazenados para validar a recuperação.

5. Auditorias de Segurança: Conduzir auditorias de segurança semestrais para garantir que o sistema esteja protegido contra ataques e falhas. Isso inclui verificação de acesso, análise de logs de segurança, pentests e implementação de melhores práticas de segurança.

6. Limpeza e Organização do Código: Refatoração de código anual para garantir que o código-fonte esteja limpo, eficiente e bem documentado. Remover partes obsoletas e refatorar funções que possam ser mais eficientes.

Frequência das Atividades

- **Backup Regular de Dados:** Diário
- **Teste de Integridade de backup:** Mensal
- **Atualizações de Software e Segurança:** Mensal
- **Monitoramento de Performance:** Contínuo
- **Testes de Recuperação de Desastres:** Semestral
- **Auditorias de Segurança:** Semestral
- **Limpeza e Organização do Código:** Anual

Equipe Responsável

- **Administrador de Sistemas:** Responsável por atualizações de sistema e monitoramento de desempenho geral.
- **Desenvolvedor de Software:** Responsável por refatoração de código e organização do código-fonte.
- **Especialista em Segurança:** Responsável por auditorias de segurança e implementação de patches de segurança.
- **Engenheiro de Rede:** Responsável pelo monitoramento de desempenho de rede e otimização de recursos.
- **Administrador de Banco de Dados:** Responsável pelos backups do banco de dados, realização de testes de recuperação de desastres e monitoramento da integridade e performance do banco de dados, além de aplicar atualizações necessárias para garantir a continuidade e segurança do ambiente.

Ferramentas Utilizadas

- **Backup:** Bacula ou AWS Backup.
- **Monitoramento de Performance:** Grafana e Zabbix.
- **Segurança:** Nessus e OpenVAS.
- **Gerenciamento de Código:** Git + GitHub/GitLab/Bitbucket

Resultado

Seguir este plano de manutenção preventiva resultará em uma operação contínua, segura e eficiente do software “XXXXXXXX” ao longo do tempo. As principais vantagens incluem:

- **Maior Confiabilidade do Sistema:** Com a realização regular de backups e testes de recuperação, o sistema estará preparado para enfrentar falhas catastróficas, minimizando o impacto de possíveis desastres. A integridade dos dados será garantida, com fácil recuperação em caso de problemas.
- **Redução de Vulnerabilidades de Segurança:** A aplicação de atualizações mensais e a implementação imediata de patches de segurança protegerão o sistema contra vulnerabilidades conhecidas, garantindo a proteção dos dados e a conformidade com melhores práticas de segurança. As auditorias de segurança e os testes de penetração também contribuirão para identificar e corrigir potenciais pontos fracos.
- **Desempenho Otimizado:** O monitoramento contínuo e as análises de desempenho semanais, juntamente com o uso de dashboards informativos, permitirão detectar problemas de desempenho de forma proativa. Isso garantirá que o sistema opere dentro de parâmetros ideais, evitando gargalos e otimização contínua de recursos.
- **Eficiência e Facilidade de Manutenção:** A refatoração anual do código e a remoção de partes obsoletas contribuirão para um código-fonte mais limpo, eficiente e fácil de manter. A documentação adequada também facilitará o trabalho das equipes de desenvolvimento e operações, além de melhorar a legibilidade e a escalabilidade do sistema.
- **Proatividade na Identificação de Problemas:** Com a combinação de testes regulares de recuperação e auditorias de segurança, a equipe estará preparada para

lidar com problemas antes que se tornem críticos. A análise constante de dados de desempenho e segurança garantirá que possíveis falhas sejam identificadas e corrigidas rapidamente.

- **Segurança Jurídica e Conformidade:** A aplicação rigorosa de boas práticas de segurança, auditorias periódicas e testes de recuperação de desastres ajudará a garantir que o sistema esteja em conformidade com as exigências legais e regulatórias, protegendo a organização de possíveis penalidades ou perdas.

Em resumo, seguir este plano de manutenção preventiva garantirá que o sistema “XXXXXXXX” esteja sempre otimizado, seguro e confiável, atendendo às necessidades do negócio e proporcionando uma experiência sem interrupções para os usuários finais. A equipe responsável terá uma abordagem estruturada e eficiente para a manutenção contínua, contribuindo para o crescimento sustentável e a evolução do software.