

Engenharia de Software – Entrega 3

Estratégias de Teste de Software

Para garantir a qualidade e a confiabilidade do sistema, implementaremos uma abordagem abrangente de testes de software:

- **Testes de Integração:** Avaliaremos a interação entre diferentes módulos do sistema, como a comunicação entre o frontend, o backend e o banco de dados, para garantir que funcionem corretamente quando combinados.
- **Testes Funcionais:** Simularemos cenários de uso reais para verificar se o sistema atende aos requisitos funcionais especificados, assegurando que as funcionalidades correspondam às expectativas dos usuários.
- **Testes de Desempenho:** Mediremos a eficiência e a capacidade do sistema sob diferentes cargas, identificando possíveis gargalos ou problemas de escalabilidade.

Manutenção do Projeto

A manutenção contínua é essencial para a longevidade e adaptabilidade do projeto:

- **Documentação Detalhada:** Manteremos documentação atualizada do código, APIs e decisões de design, facilitando a compreensão e futuras modificações.
- **Código Limpo e Padrões de Codificação:** Seguiremos boas práticas de programação e padrões estabelecidos, promovendo a legibilidade e a consistência do código.
- **Controle de Versão:** Utilizaremos o Git para rastrear mudanças, facilitar a colaboração e permitir o gerenciamento eficaz de branches e merges.
- **Gerenciamento de Dependências:** Monitoraremos e atualizaremos regularmente as bibliotecas e pacotes utilizados, garantindo que o projeto utilize versões estáveis e seguras.
- **Gerenciamento de Issues e Tarefas:** Usaremos ferramentas como GitHub e Microsoft Planner para rastrear bugs, novas funcionalidades e tarefas de manutenção, priorizando conforme a necessidade.

Práticas de DevOps

Para acelerar a entrega de software e melhorar a colaboração entre as equipes de desenvolvimento e operações, adotaremos as seguintes práticas de DevOps:

- **Integração Contínua (CI):** Configuraremos pipelines de CI utilizando ferramentas como GitHub Actions, automatizando o processo de build e testes a cada commit. Isso garante que o código integrado esteja sempre em um estado funcional.
- **Entrega Contínua (CD):** Estabeleceremos processos de CD para automatizar a implantação de novas versões do software em ambientes de staging e

produção após a aprovação nos testes, reduzindo o tempo entre o desenvolvimento e a disponibilização para os usuários.