**Entrega 3**

**Estratégias de Teste de Software**

**1. Testes Unitários**

Os testes unitários serão escritos para verificar a funcionalidade individual de métodos e classes. Eles garantirão que cada componente do sistema, como a API, as funções de manipulação de dados e os modelos preditivos, funcione corretamente de forma isolada. Serão usadas bibliotecas de teste como pytest para automatizar esses testes.

* **Exemplo de Testes Unitários**:
  + Verificar se a duração das interrupções é calculada corretamente.
  + Testar a resposta da API para endpoints de consulta de dados.
  + Garantir que as operações de CRUD no banco de dados estão funcionando conforme esperado.

**2. Testes de Integração**

Os testes de integração verificarão se os diferentes módulos do sistema funcionam bem juntos. Isso incluirá a validação da comunicação entre a API, o banco de dados e os dashboards de visualização.

* **Objetivos dos Testes de Integração**:
  + Garantir que as requisições da API fornecem dados corretos ao frontend.
  + Testar a consistência e a precisão das visualizações geradas no dashboard.

**3. Testes de Performance**

Os testes de performance medirão a eficiência do sistema em termos de velocidade, tempo de resposta e capacidade de lidar com grandes volumes de dados. Serão utilizados para otimizar a API e as consultas ao banco de dados.

* **Cenários de Teste**:
  + Medir o tempo de resposta da API sob carga pesada.
  + Analisar o desempenho das consultas SQL em grandes conjuntos de dados.
  + Avaliar a renderização dos dashboards com um grande número de visualizações.

**5. Testes de Segurança**

Como o sistema lida com dados sensíveis, serão realizados testes para garantir que as informações sejam mantidas seguras. Isso inclui a validação de autenticação, autorização e proteção contra vulnerabilidades comuns, como injeção de SQL.

* **Exemplo de Testes de Segurança**:
  + Garantir que apenas usuários autorizados possam acessar determinadas rotas da API.
  + Verificar a segurança na manipulação de dados de entrada para evitar injeções maliciosas.

**Plano de Manutenção do Sistema**

**1. Manutenção Corretiva**

A manutenção corretiva será aplicada sempre que houver falhas ou bugs no sistema. Um sistema de monitoramento estará em vigor para capturar erros em tempo real, e patches de correção serão implementados com a maior brevidade possível.

* **Processo de Correção**:
  + Recebimento de relatórios de falhas.
  + Diagnóstico e correção do problema.
  + Testes para garantir que a correção não introduza novos erros.
  + Atualização do sistema em produção.

**2. Manutenção Preventiva**

Serão realizadas verificações periódicas para otimizar a performance do sistema e garantir que os componentes estejam atualizados. Isso inclui a atualização de dependências, melhorias na arquitetura e otimização das consultas ao banco de dados.

**Aplicação de Práticas DevOps**

**1. Integração Contínua (CI)**

A Integração Contínua automatiza o processo de teste e construção do sistema toda vez que uma nova mudança é inserida no repositório. Será configurado um pipeline de CI usando ferramentas como GitHub Actions:

* Executar testes unitários e de integração automaticamente em cada commit.
* Validar a qualidade do código e o cumprimento dos padrões de codificação.
* Automatizar a construção e preparação do sistema para o ambiente de produção.

**2. Entrega Contínua (CD)**

A Entrega Contínua garante que o código testado esteja sempre pronto para ser implantado em produção. O sistema será configurado para automatizar a implantação de mudanças aprovadas.

* **Fluxo de CD**:
  + Após uma revisão manual, as mudanças podem ser promovidas para o ambiente de produção.

**Conclusão**

A implementação de uma estratégia robusta de testes, combinada com práticas DevOps eficientes, garantirá que o sistema seja confiável, seguro e de alta performance. Com manutenção contínua e um processo de entrega ágil, a solução será capaz de evoluir rapidamente para atender às novas necessidades do setor elétrico, melhorando a análise e a gestão de dados de interrupções e reclamações. Isso trará um impacto positivo, permitindo que concessionárias de energia tenham mais controle sobre os serviços oferecidos e sejam mais responsivas às demandas dos consumidores.