**Nome do estudante: João de Deus Chichava Júnior.**

**Licenciatura em Informática Aplicada**

**Regime: Ensino à distância**

**Disciplina:** Garantia de Qualidade de Sistemas

Plano de GQS (Garantia de Qualidade de Software) abordando testes de unidade, testes de integração, testes de sistema, testes de aceitação e procedimentos de revisão de código para um CRUD em PHP:

1. Testes de unidade:

- Identificar as unidades do sistema: criar, ler, atualizar e excluir registros.

- Definir casos de teste para cada unidade, como:

- Caso de teste para criar um registro com dados válidos.

- Caso de teste para criar um registro com dados inválidos.

- Caso de teste para atualizar um registro existente.

- Caso de teste para excluir um registro existente.

- Estabelecer uma estrutura de testes usando uma biblioteca como o PHPUnit.

- Especificar os resultados esperados para cada caso de teste.

- Definir uma abordagem para testar exceções e erros.

2. Testes de integração:

- Identificar as partes do sistema que são interdependentes, como a camada de apresentação, a camada de negócios e o banco de dados.

- Definir casos de teste para validar as interações entre essas partes, como:

- Caso de teste para verificar se os dados inseridos na camada de apresentação são corretamente transmitidos para a camada de negócios.

- Caso de teste para verificar se os dados inseridos na camada de negócios são corretamente gravados no banco de dados.

- Caso de teste para verificar se os dados lidos do banco de dados são corretamente exibidos na camada de apresentação.

- Especificar os resultados esperados para cada caso de teste.

- Definir uma abordagem para testar as interações entre as partes do sistema.

3. Testes de sistema:

- Descrever casos de teste para validar o comportamento completo do sistema CRUD, cobrindo diferentes cenários e casos de uso, como:

- Caso de teste para criar, ler, atualizar e excluir registros com dados válidos.

- Caso de teste para criar, ler, atualizar e excluir registros com dados inválidos.

- Caso de teste para testar a funcionalidade de pesquisa no sistema CRUD.

- Especificar os resultados esperados para cada caso de teste.

- Definir uma abordagem para testar a interface do usuário e a funcionalidade geral do sistema.

4. Testes de aceitação:

- Determinar os requisitos de negócio e as expectativas do usuário para o sistema CRUD, como:

- O sistema deve permitir a criação de registros com campos obrigatórios preenchidos corretamente.

- O sistema não deve permitir a criação de registros com campos obrigatórios em branco.

- O sistema deve fornecer feedback claro ao usuário sobre as ações realizadas.

- Descrever casos de teste que cubram esses requisitos e expectativas.

- Especificar os resultados esperados para cada caso de teste.

- Definir uma abordagem para envolver os usuários finais para validação.

5. Procedimentos de revisão de código:

- Definir um processo formal para revisar o código do CRUD em PHP, como:

- Estabelecer um sistema de controle de versão (por exemplo, Git) para facilitar a revisão de código.

- Estabelecer critérios para revisão, como boas práticas de codificação, segurança, desempenho e legibilidade.

- Definir um processo de revisão de código por pares, com agendamentos regulares para revisar o código.

- Definir uma lista de verificação ou diretrizes para a revisão de código.

- Documentar os problemas encontrados durante a revisão de código e acompanhar as ações corretivas.

É importante ressaltar que este exemplo de plano de GQS é apenas uma sugestão e pode ser adaptado de acordo com as necessidades e requisitos específicos do projeto CRUD em PHP.