## **Grupo Garantia de qualidade de software:** Danilo Dos Santos Dias, Vitor Paz De Barros, Yago Henrique Dos Santos Lima.

## **Elaboração de um plano de teste para o app SIGA.**

### **1. Objetivo do Teste.**

Avaliar a funcionalidade, usabilidade, desempenho, segurança e compatibilidade do aplicativo de gestão acadêmica, garantindo que ele atenda aos requisitos especificados e ofereça uma experiência de usuário consistente e confiável.

### **2. Escopo.**

O aplicativo gerencia informações acadêmicas, incluindo:

* Cadastro de alunos, professores e disciplinas.
* Matrícula em disciplinas.
* Consulta de horários, notas e frequências.
* Geração de relatórios administrativos.
* Comunicação entre alunos, professores e a instituição.
* Mural de informações entre Fatecs.

### **3. Tipos de Teste.**

#### **3.1 Testes Funcionais.**

1. **Módulo de Cadastro.**
   * Cadastro, edição e exclusão de alunos, professores e disciplinas.
   * Validação de campos obrigatórios e formatos de entrada.
2. **Módulo de Matrícula.**
   * Inscrição de alunos em disciplinas com limite de vagas.
   * Verificação de conflitos de horário.
3. **Módulo de Consultas.**
   * Visualização de horários, notas e frequências.
   * Acesso correto às informações com base no perfil do usuário.
4. **Módulo de Relatórios.**
   * Geração de relatórios de desempenho por aluno e turma.
   * Exportação de relatórios.
5. **Comunicação Interna.**
   * Envio e recebimento de mensagens entre usuários.
   * Notificações de novas mensagens e eventos.
6. **Mural de informações.**
   * Visualização correta de todos os tipos de artigos.
   * Notificações de novos artigos e informações do mural.

#### **3.2 Testes de Usabilidade.**

* Navegação intuitiva.
* Interface responsiva para diferentes dispositivos.
* Legibilidade e clareza dos textos.

#### **3.3 Testes de Desempenho.**

* Tempo de resposta para ações como consulta de dados e geração de relatórios.
* Testes de carga com múltiplos usuários simultâneos.

#### **3.4 Testes de Segurança.**

* Controle de acesso.
* Proteção contra SQL Injection e outros ataques de injeção.
* Criptografia de dados sensíveis, como senhas.

#### **3.5 Testes de Compatibilidade.**

* Teste em diferentes dispositivos móveis com diferentes sistemas operacionais.

### **4. Estratégia de Teste.**

#### **4.1 Ferramentas Utilizadas.**

* **Gestão de Testes:** Jira ou TestRail.
* **Automação de Testes:** Selenium (testes funcionais), JMeter (testes de desempenho).
* **Segurança:** OWASP ZAP.

#### **4.2 Ambientes de Teste.**

* **Ambiente de Desenvolvimento:** Testes iniciais pelos desenvolvedores.
* **Ambiente de Homologação:** Testes realizados pela equipe de QA.
* **Ambiente de Produção:** Testes finais de aceitação.

#### **4.3 Cronograma.**

1. Semana 1: Testes funcionais.
2. Semana 2: Testes de usabilidade e desempenho.
3. Semana 3: Testes de segurança e compatibilidade.
4. Semana 4: Correção de defeitos e reteste.

### **5. Critérios de Aceitação.**

1. Taxa de sucesso nos testes funcionais acima de 95%.
2. Tempo de resposta médio inferior a 2 segundos em operações comuns.
3. Nenhuma vulnerabilidade de alta gravidade detectada.
4. Suporte a pelo menos 3 navegadores e 2 sistemas operacionais.

### **6. Riscos e Mitigações.**

* **Risco:** Falha na comunicação entre módulos.
  + **Mitigação:** Testes de integração exaustivos.
* **Risco:** Retrabalho devido a requisitos incompletos.
  + **Mitigação:** Revisão detalhada dos requisitos antes do desenvolvimento.

### **7. Relatório Final.**

Após a conclusão dos testes, será gerado um relatório detalhando:

* Defeitos encontrados.
* Ações corretivas realizadas.
* Conclusões sobre a viabilidade de implantação.