**MyMind**

**Plano de Teste**

**Versão 1.0**

**Histórico da Revisão**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| <29/08/2022> | <1.0> | Definição do plano de teste e do cronograma | Fernando, Giulia, André, Lucas e Luca |
| <05/09/2022> | <2.0> | Preenchimento dos papeis e do mapeamento de teste | Fernando, Giulia, André, Lucas e Luca |
| <14/05/2022> | <3.0> | Conclusão do plano de teste | Fernando, Giulia, André, Lucas e Luca |
|  |  |  |  |

**Plano de Teste**

1. **Introdução**
   1. **Finalidade**

A finalidade do Plano de Teste é reunir todas as informações necessárias para planejar e controlar o esforço de teste referente a uma iteração específica. Ele descreve a abordagem dada ao teste do software e é o plano de nível superior gerado e usado pelos gerentes para coordenar o esforço de teste.

Este documento *Plano de Teste* referente ao MyMind suporta os seguintes objetivos:

**Tela**

**Código fonte**

**Requisitos**

**Modelos UML**

***Abordagem utilizada é de caixa branca e caixa preta)***

***Recursos:***

* Telas funcionais referentes ao cadastro do usuário
* Diagramas tais como: sequencial, atividade, máquina de estado, classe objeto
* Documento da lista de requisitos
* Documento de especificação de requisitos
* Código fonte previamente desenvolvido com as funcionalidades, classes, métodos que abrangem os testes de unidade: Classe psicólogos, Classe paciente, Classe usuário, e seus respectivos métodos; bem como o código de integração com o banco de dados e suas funcionalidades que permitem o CRUD dos usuários

1. **Itens de Teste**

A listagem abaixo identifica os itens de software, de hardware e elementos de suporte do produto que foram identificados como objetivos dos testes. Esta lista representa os itens que serão testados.

**Testes caixa branca**

* Código fonte:
* Documentação requisitos:
* Documentação UML: A modelagem UML define as relações entre os itens de cada parte do sistema ou dos subsistemas

**Testes caixa preta:**

* Telas: As telas elaboradas para que o usuário final possa executar as ações necessárias no sistema
* Regras de negócio: As regras que definem a abrangência e as funcionalidades do sistema
* Funcionalidades: As funcionalidades que o sistema se propõe a fazer ou resolver
* Documentação requisitos: Os requisitos que definem o que o sistema deve fazer e quais regras devem seguir
* Documentação UML: A modelagem UML define as relações entre os itens de cada parte do sistema ou dos subsistemas

1. **Escopo**

O escopo de teste engloba unidade, teste de integração e teste de sistema; bem como funcionalidade, usabilidade e confiabilidade.

Das funcionalidades nos testes de sistema: (caixa preta-Lucas):

* Confiabilidade
* Responsividade
* Usabilidade

Das funcionalidades nos testes de integração (caixa branca - André)

* Funcionalidade
* Navegadores
* Teste unitário
* Segurança (pentest)

Unidade

1. **Não Escopo**

* Performance
* Volume
* Acessibilidade
* Manutenibilidade
* Confiabilidade
* Sanidade
* Regressão

1. **Abordagem**
   1. ***Teste de Unidade***

*Os testes de unidade avaliarão isoladamente o banco de dados, e outros componentes da aplicação; ressalva de que não será englobado nesses testes a interface gráfica.*

|  |  |
| --- | --- |
| Objetivo da Técnica: | *Testar os componentes do software de forma unitária* |
| Técnica: | *Criação de testes de unidade no código* |
| Estratégias: | *Criar testes de unidade para cada funcionalidade do sistema: CRUD do psicólogo, tornar um admin* |
| Ferramentas Necessárias: | *Código da aplicação, utilizando do Jest.js* |
| Critérios de Êxito: | *Todos os testes precisam passar* |
| Considerações Especiais: | não se aplica |

* 1. ***Segurança (pentest)***

*Os testes de pentest são para verificar a vulnerabilidade da segurança no site; propondo mitigações e melhorias em toda aplicação*

|  |  |
| --- | --- |
| Objetivo da Técnica: | *Avaliar a vulnerabilidade da aplicação no que tange o banco de dados SQL.* |
| Técnica: | *Pentest* |
| Estratégias: | *Inserir scripts maliciosos de sql injection nos campos de entrada da aplicação.* |
| Ferramentas Necessárias: | *Não se aplica, visto que o a única necessidade é a aplicação funcional* |
| Critérios de Êxito: | *Não deve ser retornado informações do banco de dados* |
| Considerações Especiais: | *Não se aplica* |

* 1. ***Navegadores***

*Os testes de navegadores visão avaliar a disponibilidade da aplicação em mais de um navegador.*

|  |  |
| --- | --- |
| Objetivo da Técnica: | *Verificar a disponibilidade da aplicação em diferentes navegadores bem como suas funcionalidades e error (caso ocorram)* |
| Técnica: | *Teste de responsividade* |
| Estratégias: | *Abrir diferentes navegadores com o link do site no github pages* |
| Ferramentas Necessárias: | *Navegadores chrome, opera, firefox e edge* |
| Critérios de Êxito: | *Aplicação deve estar funcional em 100% dos navegadores, bem como ter 10% de erros decorrente de cada navegador serão aceitáveis (ressalva erros críticos que indisponibilizem a utilização da aplicação)* |
| Considerações Especiais: | Não se aplica |

* 1. ***Teste de Usabilidade***

*Os testes de usabilidade visão entender o comportamento do usuário perante a interface*

|  |  |
| --- | --- |
| Objetivo da Técnica: | *Encontrar erros e oportunidade de melhoria no site* |
| Técnica: | *Exploratório* |
| Estratégias: | *Analisar se o usuário conseguia realizar as principais funções do sistema sem necessitar de ajudar* |
| Ferramentas Necessárias: | *Aplicação funcional* |
| Critérios de Êxito: | *O usuário conseguir realizar as principais funções do sistema sem necessitar de ajudar* |
| Considerações Especiais: | Não se aplica |

* 1. ***Teste de funcionalidade***

*Os testes de funcionalidade visão verificar as funcionalidades do sistema*

|  |  |
| --- | --- |
| Objetivo da Técnica: | *atestar se a aplicação é capaz de desempenhar as funções que se propõe a fazer.* |
| Técnica: | *caixa preta* |
| Estratégias: | *Realizar todas as atividades de cadastro, atualização de cadastro e deleção de cadastro* |
| Ferramentas Necessárias: | *Aplicação funcional* |
| Critérios de Êxito: | *As funcionalidades da aplicação dever estar todas funcionandos* |
| Considerações Especiais: | Não se aplica |

1. **Critérios de execução**

*Para a realização desses critérios é necessário que o sistema esteja funcional, nos testes de caixas branca que o testador conheça o sistema, e nos testes de caixa preta deve ser um usuário externo ao projeto.*

1. **Critérios de interrupção e continuação**

*Critério de interrupção: aplicação não conseguir se conectar ao servidor*

*Critério de continuação: aplicação estar conectada ao servidor*

1. **Entregáveis**

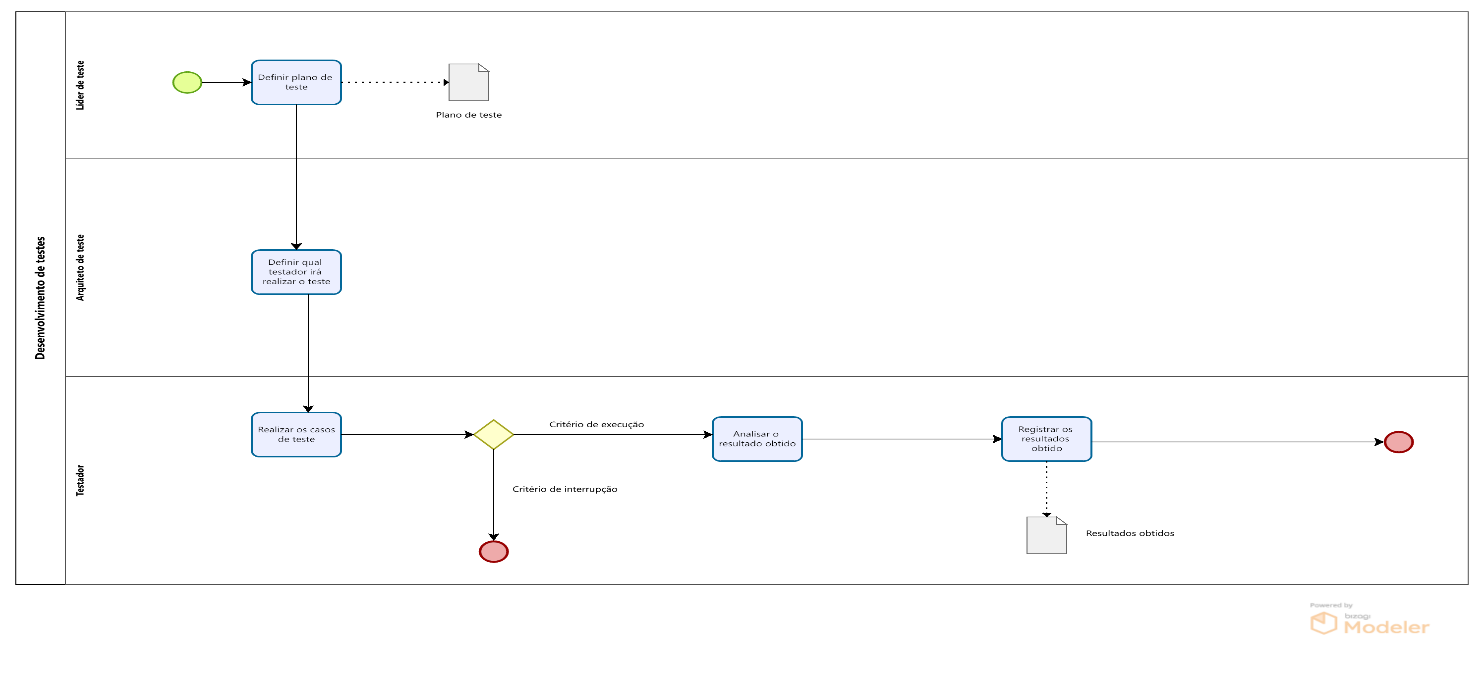
*Word com o plano de teste e a planilha com todos os casos de teste e as evidências*

1. **Ambiente de teste**

São necessários os seguintes elementos de software básicos no ambiente de teste deste *Plano de Teste*.

|  |  |  |
| --- | --- | --- |
| **Nome do Elemento de Software** | **Versão** | **Tipo e Outras Observações** |
| MyMind | 3.0 | Aplicação WEB |

1. **Processo adotado**



1. **Papéis e responsabilidades**

**11.1**     **Pessoas e Papéis**

Esta tabela mostra as suposições referentes ao perfil da equipe do esforço de teste.

|  |  |  |
| --- | --- | --- |
| **Recursos Humanos** | | |
| **Papel** | **Recursos Mínimos Recomendáveis  (número de papéis alocados em tempo integral)** | **Responsabilidades ou Comentários Específicos** |
| Testador/Arquitetura | Conhecer os tipos de teste e negócio | * Giulia Carvalho |
| Testador/Modelagem | Conhecer os tipos de teste | * Fernando Dias |
| Testador/Caixa branca | Conhecer o código, as tecnologias usadas pelo sistema e os tipos de teste. | * André Delfino |
| Testador/Caixa preta | Conhecer a plataforma, qual seria o funcionamento correto e os tipos de teste | * Lucas Eskudlark |
| Testador/Caixa Preta | Conhecer técnicas de SQL Injection | * Luca Maronezi |
|  |  |  |
|  |  |  |
|  |  |  |

1. **Cronograma**

|  |  |  |  |
| --- | --- | --- | --- |
| **Milestone** | **Data de Início** | **Data de Término** | **Responsável** |
| Definir plano de teste | 29/08/2022 | 02/09/2022 | Fernando, Giulia, Lucas e André |
| Especificar casos de teste | 05/08/2022 | 09/09/2022 | Fernando, Giulia, Lucas e André |
| Representar processo adotado | 06/09/2022 | 13/09/2022 | André e Fernando |
| Execução dos casos de teste | 12/09/2022 | 16/09/2022 | Fernando, Giulia, Lucas e André |
| Finalização do Artefato de entrega | 16/09/2022 | 20/09/2022 | Fernando, Giulia, Lucas e André |