Plano de Testes

DCC – UFBA

**Responsável pelo Documento**

Ive Tourinho

**Revisto por:**

Hortência Campos

**Validado por:**

Adriano Queiroz

**Histórico**

|  |  |  |
| --- | --- | --- |
| Versão | Descrição | Data |
| 0.1 | Estruturação e elaboração de Conteúdo, análise de necessidades, cronograma de ação e validação referentes a primeira interação do projeto. | 20/02/2013 |

**Resumo**:

O plano de testes foi desenvolvido no âmbito laboratórios do DCC, e tem como objetivo traçar planos e estratégias e as metodologias utilizadas para o desenvolvimento de testes eficientes ao longo do projeto.

**Índice**

**Introdução**

**Objetivo**

O plano de testes orienta as atividades de testes e define qual metodologia será utilizada ,quando deve ser aplicado, quais os recursos disponíveis e controlando os riscos.

Neste documento é definido o escopo cronograma e riscos.

**Visão Geral**

Este sistema serve para gerenciar e controlar o numero de refeições servidas e planejar a sua distribuição, como já descrito nos itens anteriores.

Por serem necessários cadastros de clientes e historiados dados para fins estatísticos, é necessário fazer-se uso de um banco de dados. O objetivo da implantação deste sistema é agilizar o serviço da distribuição de refeições tornando o processo que hoje é lento e causador de enormes filas em um processo veloz com menores ou nenhuma fila. As funcionalidades da aplicação a serem testadas de forma mais crítica são:

* Cadastro de clientes
* Cadastro de funcionários
* Sistema de Login
* Busca por matricula
* Classificação de tickets
* Contabilização de tickets fornecidos quanto ao tipo
* Controle de assiduidade de alunos bolsistas
* Armazenamento em banco de dados de quantidade de refeição por sessão

Para que tais testes obtenham sucesso, a estratégia de testes e alguns critérios devem ser definidos. Neste projeto utilizaremos testes estruturais e funcionais; cada um de seus critérios são descritos a seguir:

**Critérios e Cobertura de Testes**

Para a primeira interação levaremos em conta apenas os requisitos funcionais. Os testes devem ser classificados quanto a sua criticidade em: Alto, médio e baixo, para que sejam priorizados ou não, para orientar as ações visando a integridade e organização do processo de desenvolvimento e gerencia, e a importância da funcionalidade para cada versão de entrega ao cliente.

As funcionalidades e sua respectiva classificação são descritos a seguir:

|  |  |  |
| --- | --- | --- |
| Funcionalidade | Descrição | Nível |
| Cadastro de funcionários | O funcionário deve ser cadastrado com nome de usuário e senha. Seus dados e cadastro devem ser armazenados no banco de dados. Apenas o administrador previamente cadastrado pode cadastrar um funcionário. | Baixo |
| Cadastro de clientes | Cada cliente (aluno) deve ser cadastrado no sistema e seus dados armazenados no banco de dados. Apenas o administrador e funcionários cadastrados podem cadastrar um cliente. | Alto |
| Validação de Usuário e Senha | Ao entrar no sistema, o funcionário deve identificar-se com nome de usuário e senha que devem ser consultados no banco de dados para validação e acesso. | Médio |
| Validação de Aluno por matricula | Para ter aceso ao dados do cliente e liberação do ticket, o aluno deve se identificar através do numero de matricula que será consultado no banco de dados. | Alto |
| Liberação de Ticket | Após a consulta de status e identificação para aluno, ou apenas pagamento e liberação para não aluno, um ticket deve ser emitido os dados adequados. | Baixo |
| Contabilização Qualitativa | Cada ticket emitido deverá ser contabilizado de acordo com seu tipo. | Médio |
| Armazenamento em banco de dados | Todos os dados obtidos devem ser armazenados no banco de dados de acordo com seu tipo e sessão. | Alto |

**Testes Estruturais**

Por meio da análise do código fonte, para cada funcionalidade será examinada a utilização de boas praticas organização, estruturação, aplicação do paradigma, fidelidade a arquitetura e coerência. Através da ferramenta de automação JUnit cada funcionalidade será testada pontualmente por meio de testes de entrada/saída. Após estes testes será efetuado o teste de integração gradual testando cada funcionalidade inserida em seu respectivo módulo.

Todo passo de teste deve ser historiado para ser usado nos testes de regressão afim de detectar defeitos inseridos após cada interação. Mecanismos de automação para este fim devem ser utilizados para otimizar o processo.

**Testes Funcionais**

Os testes funcionais devem ser executados logo após cada modulo ter ser validado e verificado e cada funcionalidade sua ter passado pelos devidos testes estruturais e de integração no módulo. Novamente ferramentas de automação serão necessárias para a criação de casos de teste, desta vez utilizando @(??) o teste funcional será executado.

Deve ser definido pelo programador responsável pela aplicação dos testes, um subconjunto de entradas que gere um conjunto de saídas significativo quanto às possíveis ações do usuário do sistema e hipóteses de abrangência satisfatória (pior caso, melhor caso e caso médio).

Após todas as funções serem testadas e validades de acordo com os requisitos, os testes de integração com o sistema devem ser iniciados

!!!!!!!..(mais 2 fases: teste de uso( visão acima do ombro – supervisionada) e teste de operação( teste no ambiente real e nas condições reais)!!!!!!

Ao final dos testes deve ser gerado um relatório a fim de detalhar as atividades desenvolvidas e os resultados obtidos comparados dos os resultados esperados. O relatório deve conter as anomalias e incidentes ocorridos durante os testes.

**Necessidades**

Equipamentos

Para os testes equipamentos serão necessários, além das estações de trabalho para os programadores serão necessárias impressoras de pequeno porte para a geração de Tickets (Pesquisar modelos???)

Software

Além do software para desenvolvimento serão necessárias as ferramentas JUnit e (outro software?/??)

**Agenda**

O cronograma de testes deve estar alinhado com o cronograma geral apresentado no Plano de Projeto SiCRU(incluir no cronograma geral???)

**Análise de Riscos**

Nesta sessão são abordados os riscos durante a fase de testes e planos de contingência para contorná-los.

|  |  |
| --- | --- |
| **Risco** | **Plano de Contingência** |
| Perda, avaria ou destruição da impressora para teste | Negociar com o cliente reparo ou aquisição de novo dispositivo |
| Perda de funcionalidade de alguma estação de trabalho | Utilizar estações sobressalentes, caso não a possua, negociar com cliente dilatação de prazos, reparo ou aquisição de novo equipamento |
| Erros ou perda de qualquer funcionalidade no dispositivo de teste | Proceder com a tentativa de reparos usuais (reinstalação); aquisição de novo dispositivo de teste. Negociar com cliente novos prazos |
| Código ou arquitetura muito complexa dificultando testes | Revisar e se necessário refazer código e arquitetura buscando melhorar a testabilidade do produto |
| Requisitos requerem mais testes do que é possível executar | Redefinição dos requisitos ou das prioridades e critérios de teste |
| Dificuldades na preparação de ambientes de teste | Obter conhecimento e capacitação (cursos e treinamentos) necessárias à preparação do ambiente |
| Analise de testes mal definidas | Redefinição da análise de testes |
| Algumas funcionalidades não permitem teste automático | Alocar mais recursos para os testes e/ou redefinir metodologias. Se necessário/possível proceder o teste manual. |
| Testes apresentam erros recorrentes | Redefinir prioridades e critérios de testes, análise de arquitetura, análise estrutural e investigação dos erros encontrados |