**SGP - Sistemas de**

**Gerenciamento de Menu**

**Plano de Testes**

**1. Identificador do Plano de Testes**

PT\_SGM\_001

## 2. Introdução

Este documento tem como objetivo o planejamento de teste de partes do sistema SGM cuja a funcionalidade é gerenciar a produção de menus de um restaurante de forma a maximizar o lucro. O sistema deve calcular o lucro máximo possível de se obter para cada menu, levando em consideração o número de dias, quantidade de pratos que podem ser feitos, o lucro e custo de cada prato. As informações serão lidas de um arquivo .txt.

## 3. Itens de Teste

Artefatos utilizados pela equipe de Teste:

Requisitos identificados:

* Leitura do arquivo
* Criação dos pratos
* Criação dos menus
* Cálculo de lucro máximo através do método Guloso

### Camadas ou estrutura arquitetural

A aplicação será construída em console e utilizando o arquivo .txt como base de dados. Desta forma, os componentes da aplicação serão divididos de acordo com a sua responsabilidade, da seguinte maneira: Modelo: Classes que fazem a comunicação com o arquivo de dados e as entidades

do sistema. Console: Composto pelas regras de negócio do sistema e os elementos de interface com o usuário.

## 

## Visão Lógica

Visão que descreve e especifica a estrutura estática e dos colaboradores dinâmicos:

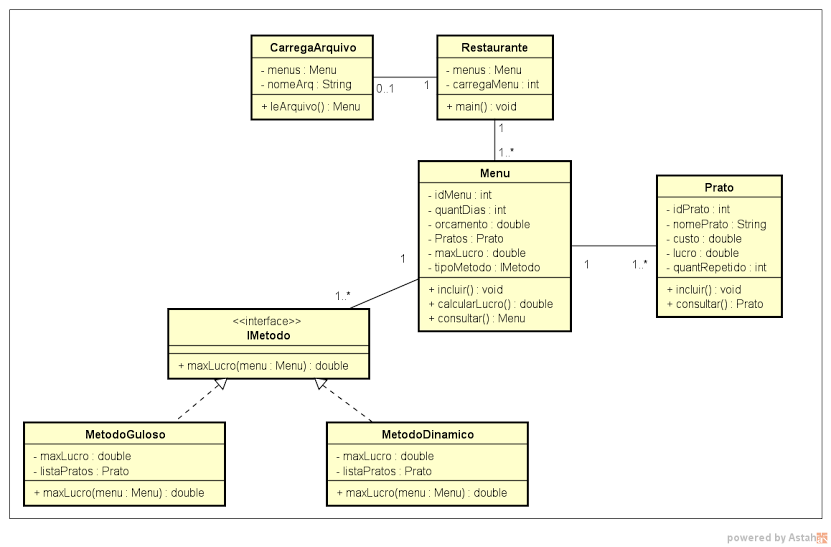


Figura 2: Diagrama de Classes

## 4. Funcionalidades a Serem Testadas

* Leitura do arquivo
* Criação dos pratos
* Criação dos menus
* Cálculo de lucro máximo através do método Guloso

## 5.Funcionalidades a não Serem Testadas

* Cálculo do lucro máximo através da tabela.

## 6. Abordagem do Teste

Os testes unitários serão realizados através da ferramenta JUnit e devem cobrir a leitura do arquivo, o carregamento de dados e o cálculo do lucro máximo (levando em consideração o prato e o menu).

Os testes poderão ser dados como terminados quando todos os casos de testes tiverem sido executados e não for encontrado tipo de defeito de severidade grave ou a sua correção ainda esteja pendente. No caso de não ser possível alcança este nível, os usuários e os desenvolvedores deverão chegar a um consenso de qual funcionalidade deverá ser priorizada nos testes.

Ao final de cada teste serão coletados os números de defeitos, o grau de severidade de cada defeito, o tempo de execução de cada caso de testes e o tempo gasto para executar cada etapa do processo de testes.

As comunicações do projeto deverão ser feitas através de reuniões de acompanhamento a qual irá gerar uma Ata com os principais tópicos da reunião e deverá ser validada pelos principais envolvidos.

## 7. Critérios de Liberação/Falha dos Itens

**Teste de Performance:**

Para o teste de performance, será utilizado o Gerenciador de Tarefas do Windows, a fim de verificar se o programa está utilizando no máximo 1% da capacidade de memória do hardware.

**Teste de Unidade:**

O teste deverá ser concluído quando não houverem falhas significativas com relação aos testes de caixa branca executados não retornarem falhas significativas ao sistema e atenderem aos requisitos levantados junto ao cliente.

Caso não seja possível alcançar o nível de integridade acima o teste deverá ser dado como concluído

quando apresentar nível inferior a 10% de falha.

## 8. Requisitos de Suspensão e Retomada

Os testes deverão ser interrompidos se apresentarem um nível de falha superior a 90%. Ou seja, a incidência de falhas permitidas é de 10% (por exemplo, a cada 10 testes realizados, no máximo 1 teste pode apresentar falha).

Os testes deverão ser retomados após a equipe de desenvolvimento ter efetuado as correções que foram solicitadas a partir dos testes efetuados anteriormente.

## 9. Entregas do Teste

|  |  |  |
| --- | --- | --- |
| **Documento** | **Responsável** | **Data de Entrega** |
| Plano de Testes - Leitura do arquivo | Mateus | 17/05/2016 |
| Plano de Teste - Criação dos pratos | Tiago | 17/05/2016 |
| Plano de Teste - Criação dos menus | Patrick | 17/05/2016 |
| Plano de Teste - Cálculo de lucro máximo através do método Guloso | Carmen | 17/05/2016 |
| Gerencia de Configuração | Carmen Maria  Mateus Diniz  Patrick  Tiago Antunes  Ludmila | 18/05/2016 |

## 10. Tarefas do Teste

* **Fornecer Requisitos**: Gerente de Requisitos
* **Fornecer Relatórios de Gerencia de Testes** - Gerente do Projeto de Teste
* **Elaborar Plano de teste:** Equipe de Testes
* **Preparação de Ambiente de Teste:** Administrador do Sistema de Teste;
* **Execução dos Testes:** Testador
* **Registro de Resultados:** Testador
* **Documento de Solicitação de Mudanças:** Testador

## 11. Necessidades de Ambientes

|  |
| --- |
| **Recursos do Sistema** |
| **Arquivo de dados** |
| Arquivo no formato .txt |
| **Terminais Clientes** |
| 1 Computador |
| **Repositório de Testes** |
| 1 PC |
| 3 PCs de Desenvolvimento de Teste |

|  |
| --- |
| **Recursos Humanos** |
| **Desenvolvimento dos testes** |
| Desenvolvedor de testes |
| **Aplicação dos testes** |
| Testador, no mínimo 3 |
| **Gerenciamento de testes** |
| Gerente de Testes |
|  |

|  |
| --- |
| **Recursos Softwares** |
| **Ferramentas** |
| Java 1.7 ou superior |
| JUnit |
| GitHub |

## 12. Responsabilidades

|  |  |  |  |
| --- | --- | --- | --- |
| **Recursos Humanos** |  |  |  |
| **Trabalhador** | Recursos  Recomendados | Mínimos | Responsabilidades Específicas ou Comentários |
| **Gerente de Teste** | Carmen Maria |  | Fornece supervisionamento gerencial. |
| **Gerente do Projeto de Teste** | Mateus Diniz |  | Responsabilidades:   * Provê direcionamento Técnico * Adquire recursos Apropriados * Fornece relatórios de gerenciamento |
| **Gerente de Requisitos** | Patrick |  | Fornecer os requisitos levantados para a equipe de testes, usados como entrada durante os testes, para garantir que a solicitação do cliente foi atendida. |
| **Plano de Teste** | Carmen Maria  Mateus Diniz  Patrick  Tiago Antunes |  | Identifica, prioriza, e implementa os casos de teste.  Responsabilidades:   * Gera o plano de teste * Cria o modelo de teste * Avalia a efetividade do   Esforço de teste |
| **Testador** | Tiago Antunes | | Executa os testes.  Responsabilidades:   * Executar os testes * Registrar os resultados * Reestabelecer-se dos erros * Documentar solicitações de mudança |
| **Administrador do Sistema de**  **Teste** | Ludmila | | Garante que o ambiente e os bens de teste sejam gerenciados e mantidos.  Responsabilidades:   * Administrar o sistema de gerenciamento teste * Instalar e gerenciar o acesso do trabalhador ao sistema de testes |

## 13. Necessidades de Equipe e de Treinamento

Para a execução das tarefas foi necessário efetuar treinamento da equipe para utilizar as seguintes ferramentas:

* JUnit
* GitHub

## 14. Cronograma

|  |  |  |
| --- | --- | --- |
| **Cronograma** |  |  |
| **Atividade de Testes** | Data Inicio | Data termino |
| **Fornecimento de Requisitos** | 1/05/2016 | 1/11/2016 |
| **Elaboração Plano de Testes** | 2/05/2016 | 4/05/2016 |
| **Preparação de Ambiente de Teste** | 4/05/2016 | 6/05/2016 |
| **Execução dos Testes** | 7/05/2016 | 8/05/2016 |
| **Registro de Resultados** | 7/05/2016 | 8/05/2016 |
| **Elaboração do Documento de Solicitação de Mudanças** | 8/05/2016 | 10/05/2016 |
| **Relatórios de Gerenciamento de Teste** | 11/05/2016 | 11/05/2016 |
| **Elaboração Gerencia de Configuração** | 10/05/2016 | 11/05/2016 |

## 15. Riscos e Contingências



**Risco: Máquinas não configuradas**

Plano de Mitigação: Agendamento de configuração das máquinas com 3 dias de antecedência; Plano de Contingência: Reservar tempo de 2 horas para verificação de configuração.

**Risco: Inconsistência no arquivo de dados**

Plano de Mitigação: arquivo de dados deverá ser verificado antes do início do processo de testes.

Plano de Contingência: As alterações das inconsistências deverão ser feitas em até 2 dias após identificação do erro

**Risco: Falta de Testador**

Plano de Mitigação: Alocar testado no início do projeto de testes

Plano de Contingência: Alocação de membro mais experiente.