

# God Challenge

## Plano de Testes da primeira entrega

Versão 1.1

### Histórico da Revisão

Data	Versão	Descrição	Autor
12/04/2014	1.0	Produção do documento de testes	Rodrigo Sadra
30/04/2014	1.1	Preenchimento do plano de testes	Jéssica Penna

# Índice Analítico

<b>1. Introdução .....</b>	<b>3</b>
1.1. Finalidade .....	3
1.2. O Sistema Carofour .....	3
1.3. Escopo.....	3
1.4. Público-alvo .....	3
1.5. Referências.....	3
<b>2. Missão de Avaliação e Motivação dos Testes .....</b>	<b>4</b>
2.1. Fundamentos.....	4
2.2. Missão de Avaliação .....	4
2.3. Motivadores dos Testes .....	4
<b>3. Itens de Teste-Alvo .....</b>	<b>4</b>
<b>4. Resumo dos Testes Planejados.....</b>	<b>5</b>
4.1. Testes funcionais.....	5
4.1.1. Compra.....	<b>Erro! Indicador não definido.</b>
4.2. Testes não funcionais .....	5
4.2.1. Tempo de resposta.....	5
4.2.2. Acessos simultâneos .....	5
4.3. Resumo das Inclusões dos Testes .....	5
<b>5. Abordagem dos Testes .....</b>	<b>5</b>
5.1. Tipos e Técnicas de Teste .....	5
5.1.1. Teste de Funcionamento .....	5
5.1.2. Teste de Interface do Usuário.....	6
<b>6. Critérios de Entrada e de Saída .....</b>	<b>6</b>
6.1. Plano de Teste.....	6
6.1.1. Critérios de Entrada de Plano de Teste .....	6
6.1.2. Critérios de Saída de Plano de Teste .....	6

# Plano de Teste God Challenge

## 1. Introdução

### 1.1. Finalidade

A finalidade do Plano de Teste de Iteração é reunir 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 jogo God Challenge desenvolvido pelo aluno Rodrigo Sadra que tem as seguintes características.

- Jogo em rede no estilo MOBA.
- Software multiplataforma.

### 1.2. O Jogo God Challenge

O software que está sendo testado é referente a um jogo utilizado como trabalho de conclusão de curso de jogos Digitais da Puc Minas. O objetivo do sistema é proporcionar entretenimento para os jogadores aplicando as diversas técnicas de jogos aprendidas no curso. Aplica-se inteligência artificial, computação gráfica, redes, banco de dados e programação para se obter um jogo atrativo para os jogadores.

### 1.3. Escopo

Serão realizados testes caixa preta, caixa branca e testes de sistema de duas classes significativas do sistema. Serão testadas apenas algumas funcionalidades do jogo (como por exemplo o Menu). Vale ressaltar que o software foi desenvolvido em uma plataforma diferente das que os membros do grupo estão habituados e a falta de contato entre os integrantes do grupo dificulta ainda mais a execução do trabalho.

### 1.4. Público-alvo

O presente plano de testes está sendo escrito para auxiliar testadores nas realizações dos testes do sistema e desenvolvedores para auxiliar na depuração e melhoramento do código.

### 1.5. Referências

Este documento é baseado no fluxo de testes do RUP e está diretamente relacionado com o Documento de Requisitos.

- Documento de requisitos:

## 2. Missão de Avaliação e Motivação dos Testes

### 2.1. Fundamentos

Os testes propostos nesse documento têm por objetivo adequar o sistema às necessidades dos usuários finais, fazendo com que seja produzido um sistema de maior qualidade com o menor número de defeitos possíveis.

O principal fundamento que justifica o esforço de teste a ser empregado em partes específicas do sistema é o fato do software não estar todo na plataforma .NET. Portanto apenas algumas funcionalidades serão testadas.

### 2.2. Missão de Avaliação

- Localizar o maior número de defeitos possível
- Localizar defeitos gerados por requisitos não funcionais
- Verificar o cumprimento da especificação de requisitos
- Verificar a precisão e confiabilidade dos dados gerados pelo game
- Avaliar a qualidade do produto que está sendo desenvolvido.

### 2.3. Motivadores dos Testes

O principal motivo dos esforços de testes é que o sistema deve ter a integração em diferentes tipos de plataforma portanto deve-se garantir a eficiência em ambas. Outro motivo importante é a preservação e fortificação da imagem do game perante os consumidores finais.

## 3. Itens de Teste-Alvo

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.

Número de ordem	Item
1	Teste de interface de menu,
2	Teste da classe Skill Zeus.
3	Teste da classe Skill Hades.

## 4. Resumo dos Testes Planejados

### 4.1. Testes funcionais

#### 4.1.1. Menu

<b>Objetivo do Tipo de Teste:</b>	Verificar se é possível selecionar os tópicos do menu no sistema
<b>Técnica:</b>	Realizar um clique de cliques nas opções, desde a seleção de um item até a finalização do menu.
<b>Critérios de Êxito:</b>	Visualizar as mensagens e verificar se o sistema carrega para a próxima fase.
<b>Considerações Especiais:</b>	Executar o teste em todos os itens de menu.

### 4.2. Testes não funcionais

#### 4.2.1. Tempo de resposta

<b>Objetivo do Tipo de Teste:</b>	Verificar o intervalo de tempo de resposta.
<b>Técnica:</b>	Realizar operações de entrada e saída (atendimento a requisições).
<b>Critérios de Êxito:</b>	O tempo de resposta deve ser no máximo, menor ou igual a 5 (TR <= 5).
<b>Considerações Especiais:</b>	Nenhuma.

#### 4.2.2. Acessos simultâneos

<b>Objetivo do Tipo de Teste:</b>	Verificar se é possível gerenciar o acesso simultâneo.
<b>Técnica:</b>	Enviar várias requisições ao mesmo tempo.
<b>Critérios de Êxito:</b>	O atendimento de cada requisição de forma escalonada.
<b>Considerações Especiais:</b>	Nenhuma.

### 4.3. Resumo das Inclusões dos Testes

- Menu;
- Tempo de resposta;
- Acessos simultâneos;

## 5. Abordagem dos Testes

Os testes serão executados de forma manual.

### 5.1. Tipos e Técnicas de Teste

#### 5.1.1. Teste de Funcionamento

Objetivo da Técnica:	Experimentar a funcionalidade do Jogo God Challenge, incluindo a navegação, a entrada, o processamento e a recuperação de dados a fim de observar e registrar o comportamento-alvo.
Técnica:	Experimentar os recursos e fluxos ou funções de cada um dos cenários de caso de uso, utilizando dados válidos e inválidos para verificar se: <ul style="list-style-type: none"><li>• os resultados esperados ocorrerão quando forem usados dados válidos;</li></ul>

	<ul style="list-style-type: none"> <li>as mensagens de erro ou de aviso apropriadas serão exibidas quando forem usados dados inválidos;</li> <li>cada regra de negócio será aplicada de forma adequada.</li> </ul>
Estratégias:	<ul style="list-style-type: none"> <li>Aplicar valores verdadeiros e falsos para verificar se a resposta pretendida é apresentada.</li> <li>Aplicar valores repetidas vezes para acompanhar o comportamento do software.</li> </ul>
Ferramentas Necessárias:	A técnica exige as seguintes ferramentas: <ul style="list-style-type: none"> <li>Ferramentas de backup e de recuperação</li> <li>Ferramentas de monitoramento de instalação (registro, disco rígido, CPU, memória etc)</li> </ul>
Critérios de Êxito:	A técnica suporta o teste de todos os principais cenários de caso de uso e todos os principais recursos do menu do sistema.
Considerações Especiais:	N/A.

### 5.1.2. Teste de Interface do Usuário

Objetivo da Técnica:	Testar: <ul style="list-style-type: none"> <li>A navegação pelo menu do jogo God Challenge para verificar se reflete os requisitos e funções de negócios, incluindo a navegação janela a janela e campo a campo, e o uso de métodos de acesso (teclas de tabulação, movimentos do mouse e teclas aceleradoras).</li> <li>As características e os objetos das janelas devem ser experimentados como, por exemplo, menus, tamanho, posição, estado e foco.]</li> </ul>
Técnica:	Efetuar testes para cada janela a fim de verificar a navegação adequada e os estados de objeto apropriados para cada janela e objeto do sistema.
Estratégias:	<ul style="list-style-type: none"> <li>Compilar o jogo em diferentes resoluções de telas.</li> <li>Aplicar comandos com os dispositivos de entrada e aguardar a resposta pretendida.</li> </ul>
Ferramentas Necessárias:	Dispositivos gráficos e de entrada de dados.
Critérios de Êxito:	A técnica suporta o teste de cada tela que será usada pelo usuário final.
Considerações Especiais:	N/A.

## 6. Critérios de Entrada e de Saída

### 6.1. Plano de Teste

#### 6.1.1. Critérios de Entrada de Plano de Teste

Assim que for concluída da Fase de Construção este Plano de Testes poderá ser executado.

#### 6.1.2. Critérios de Saída de Plano de Teste

Se os resultados do Plano de Teste forem satisfatórios este Plano de Testes terá sido concluído.

#### 6.1.3 Critérios de Suspensão e de Reinício

N/A.

## 7. Produtos Liberados

### 7.1 Sumários de Avaliação de Testes

N/A.

## 7.2 Geração de Relatórios sobre Cobertura de Teste

Para cada teste executado, um formulário de resultado de teste será criado. Isso deverá incluir o Nome do Teste, o Caso de Uso com qual o teste encontra-se relacionado, a Data do Teste, o Nome do Testador, as Condições de Pré-teste necessárias e os Resultados do Teste. O Microsoft Word será utilizado para registrar e reportar os resultados de teste.

## 7.3 Registros de Incidentes e Solicitações de Mudança

N/A.

## 7.4 Conjunto de Testes de Regressão e Scripts de Teste de Suporte

Serão elaborados scripts de teste que permitem a execução de instruções que serão elaboradas e documentadas em forma de instruções de texto para execução de testes.

## 7.5 Produtos de Trabalho Adicionais

### 7.5.1 Resultados Detalhados dos Testes

Será definido em um momento futuro.

### 7.5.2 Scripts de Teste Funcionais Automatizados Adicionais

Será definido em um momento futuro.

### 7.5.3 Guia de Teste

Será definido em um momento futuro.

### 7.5.4 Matrizes de Rastreabilidade

Será definido em um momento futuro.

## 9. Necessidades Ambientais

### 9.1 Hardware Básico do Sistema

Os conjuntos de tabelas a seguir apresentam os recursos do sistema necessários ao esforço de teste descrito neste *Plano de Teste*.

Recurso	Quantidade	Nome e Tipo
PCs de Teste Cliente	1	2GB RAM, HD 40GB e Vídeo 32 bits.
Repositório de Teste	1	Github (Repositório do trabalho)
PCs de Desenvolvimento de Teste	1	2GB RAM, HD 40GB e Vídeo 32 bits.

### 9.2 Elementos de Software Básicos do 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
Windows , MAC ou Linux	N/A	Sistema Operacional

### 9.3 Ferramentas de Produtividade e de Suporte

Serão utilizadas as seguintes ferramentas para suportar o processo de teste deste *Plano de Teste*.

<b>Categoria ou Tipo de Ferramenta</b>	<b>Nome da Marca da Ferramenta</b>	<b>Fornecedor ou Desenvolvida Internamente</b>	<b>Versão</b>
Gerenciamento de Teste	Rational	RequisitePro	2003
Design de Teste	Rational	Rose	2003
Testes Funcionais	Rational	Robot	2003
Testes de Performance	Rational	Visual Quantify	2003
Teste de Cobertura 2003	Rational	Visual PureCoverage	2003
Gerenciamento de Projeto	Microsoft	Word	2007
		Project	2000

### 9.4 Configurações do Ambiente de Teste

Devem ser fornecidas e suportadas as seguintes Configurações de Ambiente de Teste para este projeto.

<b>Nome da Configuração</b>	<b>Descrição</b>	<b>Implementada na Configuração Física</b>
Configuração do usuário comum		
Mínima configuração suportada		
Motivada por funções visuais e motoras		
Sistema Operacional Internacional de Dois Bytes		
Instalação de Rede (não cliente)		