

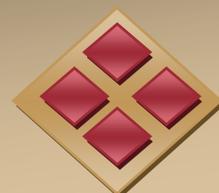
# **Sistema BuildParty para montagem e gerenciamento de eventos**

## **Plano de Testes**

**Versão <1.1>**

**DeltaInfo**

- ❖ **Soluções para web**
- ❖ **Soluções para o mundo**



## **Histórico de Revisões**

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autores</b>
16/10	1.0	Criação da primeira versão do plano de testes	Pâmela Thays
08/11	1.1	Revisão do documento e ajuste na descrição dos testes	Pâmela Thays
<b>08/11</b>	<b>1.1</b>		<b>Pâmela Thays</b>

## Conteúdo

1- Introdução-----	4
1.1- Objetivos-----	4
1.2- O sistema BuildParty-----	4
1.3- Escopo-----	4
1.4- Identificação do projeto-----	5
2- Requisitos a testar-----	5
2.1- Teste de banco de dados-----	5
2.2- Teste de Função-----	6
2.3- Teste da interface do usuário-----	6
2.4- Perfil do Desempenho-----	6
2.5- Teste de Carga-----	6
2.6- Teste de Stress-----	7
2.7- Teste de Volume-----	7
2.8- Teste de segurança e de controle de acesso-----	7
2.9- Teste de falha e recuperação-----	7
2.10- Teste de instalação-----	7
3- Estratégias de Teste-----	8
3.1- Tipos de testes-----	8
3.1.1- Testes de integridade de dados e do BD--	8
3.1.2- Testes de função-----	9
3.1.3- Testes de interface com o usuário-----	10
3.1.4- Testes de performance e carga-----	11
3.1.5- Testes de Carga-----	11
3.1.6- Testes de segurança e controle de acesso-	12
3.1.7- Testes de instalação-----	13
3.2- Ferramentas-----	13
4- Recursos-----	14
4.1- Trabalhadores-----	14
4.2- Sistema -----	15
5- Cronograma -----	16

## **I- Introdução**

### **I.1- Objetivos**

O objetivo deste documento é identificar e listar os requisitos do sistema que devem ser testados, bem como descrever as estratégias, ferramentas e recursos necessários para a execução destes testes.

### **I.2- O sistema BuildParty**

O sistema Buildparty tem como objetivo automatizar o procedimento de montar um evento bem como gerar o orçamento de tal evento. Um sistema que visa à comodidade do cliente, uma vez que ele não precisará mais ir a empresa de eventos para fazer o evento que deseja e avaliar os custos de fazê-lo, bem como agilizar o controle da empresa sobre os eventos que estão em execução e os clientes que contrataram seus serviços. Também por ser um sistema web, ele permite que um público maior de clientes, que devido a distância(moradia em outro município ou estado, por exemplo) ou falta de transporte até o local da empresa, possam contratar a empresa e se comunicar com a mesma sem sair de casa.

### **I.3- Escopo**

O sistema Buildparty vai passar pelos principais testes de qualquer sistema de software: unitários, de integração e de sistema. Como se trata de um sistema web, composto de uma parte em html e css(interface), php(funcionalidade) e banco de dados(integração de dados),os testes unitários e de integração são importantíssimos na análise da funcionalidade geral do sistema de acordo com os requisitos levantados anteriormente no documento de requisitos(acesso do usuário, emissão de contrato,geração do orçamento, etc). O teste de sistema vai tratar de problemas de performance, como ,por exemplo, o tempo de resposta do sistema. O hardware e o ambiente utilizado para a execução dos testes vão ser o mais simples possíveis: um computador normal com acesso a internet em um ambiente domiciliar ou de trabalho, de forma a ficar mais parecido com uma situação real onde um cliente, de seu computador pessoal, estando em casa, na escola, ou no trabalho, vai através da internet acessar o site da empresa.

Os testes mais importantes para o sistema obviamente são:

- Interface com o usuário: uma vez que o perfil do usuário a utilizar o sistema é muito amplo, desde daqueles que tem conhecimento técnico de informática até os de pouco costume com acesso a rede, a interface deve ser a mais simples e intuitiva possível para permitir que o cliente

monte o evento corretamente e faça o contrato de maneira segura.

- Banco de dados: o banco de dados é a base para o nosso sistema. É através dele que a empresa deve atualizar, remover preços e recursos , além de permitir o cadastro de novos usuários. O bom funcionamento deste garante a integridade de dados e, conseqüentemente o bom funcionamento do sistema.
- Programa de geração de orçamento, contrato e boleto: além da integridade dos dados, é necessário que o sistema possa acessar corretamente o banco de gerar os cálculos exatos para o orçamento do evento. bem como pegar os dados do cliente para gerar um contrato correto. A integridade desta parte com o banco de dados também é de suma importância para o sistema e deve ser bem testada.

## 1.4- Identificação do Projeto

Documento	Criado ou Disponível	Recebido ou revisado
Plano de projeto	<input type="checkbox"/> Sim <input type="checkbox"/> Não	<input type="checkbox"/> Sim <input type="checkbox"/> Não
Documento de requisitos	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não
Plano de Testes	<input type="checkbox"/> Sim <input checked="" type="checkbox"/> Não	<input type="checkbox"/> Sim <input checked="" type="checkbox"/> Não
Projeto de Testes	<input type="checkbox"/> Sim <input checked="" type="checkbox"/> Não	<input type="checkbox"/> Sim <input checked="" type="checkbox"/> Não
Documento de Análise e projeto	<input type="checkbox"/> Sim <input checked="" type="checkbox"/> Não	<input type="checkbox"/> Sim <input checked="" type="checkbox"/> Não
Protótipo	<input type="checkbox"/> Sim <input checked="" type="checkbox"/> Não	<input type="checkbox"/> Sim <input checked="" type="checkbox"/> Não

## 2- Requisitos a serem testados

Abaixo estão listados os componentes do sistema a serem testados.

### 2.1- Testes de Banco de Dados

- Verificar se as informações do cliente podem ser cadastradas, consultadas e removidas do sistema;
- Verificar se os preços dos materiais disponíveis no site podem ser inseridos e atualizados por um funcionário da empresa;

- Verificar caso o sistema busca corretamente os preços dos materiais;
- Verificar se o sistema guarda o dia do último evento feito pelos clientes;
- Verificar se o sistema guarda os eventos;
- Verificar se as opções escolhidas pelo cliente podem ser devidamente consultadas pela empresa;
- Verificar se os dados são atualizados corretamente (consistência de dados);

## **2.2- Testes de Função**

- Verificar acesso de usuários(cliente ou funcionários) através de login e senha;
- Verificar se a geração do orçamento está correto;
- Verificar se a geração do boleto ou contrato está correto;
- Verificar se cada tipo de usuário pode fazer as operações atribuídas ao mesmo através do sistema;

## **2.3- Testes de Interface do usuário**

- Verificar se as informações estão de forma clara e intuitiva para todos os clientes;
- Verificar se as fotos dos modelos estão apresentadas corretamente e são de fácil acesso;
- Verificar se todos os ícones estão dispostos de maneira simples e intuitiva;
- Verificar caso o português está claro e simples gramaticalmente;
- Verificar se os diferentes tipos de festas e seus recursos estão dispostos de maneira clara para todos os tipos de clientes;

## **2.4- Perfil do Desempenho**

- Verificar o tempo de resposta para cadastro de usuários;
- Verificar o tempo de resposta para operações de inserção, atualização e remoção de dados;
- Verificar o tempo de resposta do servidor web para disponibilizar arquivos multimídias(fotos);
- Verificar o tempo de resposta do sistema para a geração do orçamento, contrato e boleto bancário;

## **2.5- Testes de Carga**

- Verificar a resposta do sistema ao número grande de usuários logados e fazendo operações ao mesmo tempo(10, 50 ,100 usuários);

## **2.6- Testes de Stress**

Uma vez que nosso sistema é simples e não necessita de hardware ou software sofisticado para efetuar as operações, e levando em conta que as condições em que o mesmo é acessado nunca chegarão a um estado crítico onde a queda ou demora de resposta do sistema provocarão prejuízos e danos, o teste de stress não é necessário e, portanto, não será efetuado durante a fase de testes.

## **2.7- Testes de Volume**

Dado que não haverá uma grande quantidade de dados a serem armazenados no banco de dados, muito menos uma demasiada quantidade de mídia vai ser disponibilizada na interface gráfica, o teste de volume também é desnecessário para o nosso sistema.

## **2.8- Testes de Segurança e Controle de Acesso**

- Verificar se cada tipo de usuário(cliente ou funcionário), tem acesso a dados e operações necessárias ao mesmo;
- Verificar caso somente o usuário tem acesso aos seus dados;
- Verificar caso o funcionário possa apenas consultar o nome do cliente e o evento a ele associado. Qualquer outra informação pessoal do cliente não pode ser consultada ou alterada pelos funcionários da empresa;
- Verificar caso o cliente só consulte os materiais e seus preços, bem como fotos de cada evento;
- Verificar se apenas o funcionário da empresa pode inserir ou atualizar preços do banco de dados;
- Verificar se todas as operações no site só podem ser feitas por pessoas devidamente cadastradas;

## **2.9- Testes de falha e recuperação**

- Verificar caso o sistema mantém a consistência de dados mesmo que o servidor fique off-line de maneira inesperada;
- Verificar caso o sistema mantém a consistência de dados mesmo que o terminal de onde partem as requisições fique off-line inesperadamente;

## **2.10- Testes de Instalação**

- Verificar se o sistema foi devidamente instalado no servidor web a empresa de eventos CinEventos;

## 3- Estratégias de Testes

### 3.1- Tipos de Testes

#### 3.1.1- Teste de integridade de dados e Banco de Dados

Objetivo do Teste	Garantir que os métodos e processos de acesso ao banco de dados funcionam apropriadamente e se há consistência de dados.
Técnica	<ol style="list-style-type: none"> <li>1. Invocar cada método e processo de acesso ao banco de dados, alimentando cada um com dados ou requisições de dados válidos e inválidos.</li> <li>2. Inspeccionar o banco de dados para garantir que os dados foram devidamente povoados, que todos os eventos do banco de dados ocorreram apropriadamente, ou revisar os dados retornados para garantir que os dados corretos foram recuperados pelas razões corretas.</li> </ol>
Critério de Finalização	Todos os métodos e processos de acesso à base de dados funcionam como projetados e sem nenhuma corrupção de dados.
Considerações Especiais	<ol style="list-style-type: none"> <li>1. O teste pode necessitar de um ambiente de desenvolvimento ou drivers de SGBD para inserir ou modificar os dados diretamente nas base de dados.</li> <li>2. Processos devem ser invocados manualmente.</li> <li>3. Bases de dados pequenas ou minimizadas (número de registros limitados) devem ser usados para aumentar a visibilidade de eventos não-aceitáveis.</li> </ol>

### 3.1.2- Testes de Função

Objetivo do Teste	Testar todas as funcionalidades do sistema, desde o acesso do usuário através de login e senha até a geração de contratos e boletos bancários.
Técnica	Executar cada um dos casos de usos, respeitando seus fluxos, com entradas válidas e inválidas e fazer as seguintes observações: <ol style="list-style-type: none"><li>1. Os resultados esperados são observados quando dados válidos são inseridos;</li><li>2. As mensagens de erros e avisos apropriados são observados quando dados inválidos são usados.</li></ol>
Critério de Finalização	<ol style="list-style-type: none"><li>1. Todos os casos de usos foram testados;</li><li>2. Todos os erros encontrados foram corrigidos.</li></ol>
Considerações Especiais	Nenhum.

### 3.1.3- Testes de Interface de usuário

Objetivo do Teste	<ol style="list-style-type: none"><li>1. Verificar caso a navegação é fácil e intuitiva ,inclusive o acesso do usuário, bem como os menus, botões e tabelas estão dispostos de maneira explicativa e apropriada.</li><li>2. Verificar se a aparência do site segue um padrão e é atrativa para os diversos tipos de clientes.</li></ol>
Técnica	Examinar cada um dos possíveis fluxos de dados, se baseando pelos casos de uso, verificando se para os mesmos houve dificuldade na navegação e falta de entendimento da organização dos menus.
Critério de Finalização	Todos os fluxos possíveis foram testados e estão coerentes com os casos de uso.
Considerações Especiais	O teste de interface é bastante subjetivo, uma vez que uma mesma interface pode estar bem explicativa para uns e incompreensível para outros,o que faz com que este teste não seja completo.

### 3.14- Testes de Performance e Carga

Objetivo do Teste	Verificar o desempenho do sistema para: <ol style="list-style-type: none"> <li>1. Carga normal de trabalho;</li> <li>2. Carga de trabalho no pior caso.</li> </ol>
Técnica	<ol style="list-style-type: none"> <li>1. Testar o sistema como apenas um usuário acessando de uma única máquina;</li> <li>2. Testar para vários usuários, de diversas modalidades, fazendo requisições ao sistema ao mesmo tempo;</li> </ol>
Critério de Finalização	Para ambos os casos, finalização das operações com sucesso e com tempo de resposta adequado.
Considerações Especiais	<ol style="list-style-type: none"> <li>1. Criar carga de usuário "virtual" para simular muitos clientes, normalmente várias centenas. Ferramentas de Emulação de Terminal Remoto (RTE) são usadas para atingir essa carga. Essa técnica também pode ser usada para carregar uma rede com "tráfego".</li> <li>2. Usar diversos clientes físicos acessando o sistema por máquinas diferentes;</li> <li>3. O teste de performance deve ser executado em uma máquina dedicada ou em um tempo dedicado. Isso permite controle total e mensuração precisa.</li> <li>4. As bases de dados usadas para o Teste de Performance devem ser ou do tamanho real ou proporcionalmente iguais.</li> </ol>

### 3.1.5- Testes de Segurança e Controle

Objetivo do Teste	<ol style="list-style-type: none"><li>1. Segurança do Nível de Aplicação: Verifique que um usuário pode acessar apenas aquelas funções ou dados para os quais o seu tipo de usuário tem permissão.</li><li>2. Segurança do Nível de Sistema: Verifique que apenas aqueles usuários com acesso ao sistema e aplicações têm permissão de acessá-los.</li></ol>
Técnica	<ol style="list-style-type: none"><li>1. Segurança do Nível de Aplicação: Identifique e liste cada tipo de usuário e as funções ou dados para os quais cada tipo tem permissão.</li><li>2. Crie testes para cada tipo de usuário e verifique cada permissão criando transações específicas para cada tipo de usuário.</li><li>3. Modifique o tipo de usuário e repita os testes para os mesmos usuários. Em cada caso, verifique que funções ou dados adicionais estão corretamente disponíveis ou negados.</li></ol>
Critério de Finalização	Para cada tipo de usuário conhecido as funções ou dados apropriados estão disponíveis, e todas as transações funcionam como esperado e rodam nos Testes de Função anteriores.
Considerações Especiais	O Acesso ao sistema deve ser revisado ou discutido com o administrador de rede ou de sistema apropriado. Esse teste pode não ser necessário já que ele pode ser uma função da administração da rede ou sistema.

### 3.1.7- Testes de Instalação

Objetivo do Teste	Verificar caso houve instalação bem sucedida do sistema no servidor para os seguintes casos: <ol style="list-style-type: none"> <li>1. O servidor não possui o sistema BuildParty previamente instalado;</li> <li>2. O sistema já possui a mesma versão do sistema que deseja-se instalar.</li> <li>3. O servidor possui uma versão antiga do sistema que deseja-se instalar.</li> </ol>
Técnica	<ol style="list-style-type: none"> <li>1. Desenvolva scripts , manualmente ou automatizados, para validar a condição do servidor nos três casos acima listados.</li> <li>2. Começar ou executar a instalação.</li> <li>3. Usando um subgrupo predeterminado de scripts de teste de funções, rode as transações.</li> </ol>
Critério de Finalização	A instalação é bem sucedida nos três casos acima citados.
Considerações Especiais	Atentar para as condições usadas nos testes para que os mesmos sejam confiáveis.

### 3.2- Ferramentas

A seguir estão listadas as ferramentas usadas no processo de testes do projeto.

Etapa	Ferramenta	Vendedor
Gerenciamento de Teste	Rational RequisitePro	IBM
Projeto de Teste	Rational Rose	IBM
Gerenciamento de Projeto	Microsoft Word	Microsoft
Ferramentas do SGBD	MySQL Control Center	Free

## 4- Recursos

Essa seção apresenta os recursos recomendados para o projeto do sistema.

### 4.1- Recursos Humanos(Trabalhadores)

Trabalhador	Recursos Mínimos Recomendados	Responsabilidades específicas ou comentários
Gerente de testes	Pamela Thays	<p>Fornecer supervisão gerencial.</p> <p>Responsabilidades:</p> <ol style="list-style-type: none"> <li>1. provê direcionamento técnico a equipe;</li> <li>2. adquire recursos apropriados;</li> <li>3. fornece relatórios de gerenciamento;</li> </ol>
Designer de testes	Aline D'Oleron	<p>Identifica, prioriza, e implementa os casos de teste.</p> <p>Responsabilidades:</p> <ol style="list-style-type: none"> <li>1. gera o plano de teste</li> <li>2. cria o modelo de teste</li> <li>3. avalia a efetividade do esforço de teste</li> </ol>
Testador	Luiz Gustavo e Gustavo Souto	<p>Executa os testes.</p> <p>Responsabilidades:</p> <ol style="list-style-type: none"> <li>1. executar os testes</li> <li>2. registrar os resultados</li> <li>3. reportar os erros</li> <li>4. documentar solicitações de mudança</li> </ol>
Administrador do sistema de teste	Breno e Luiz Carlos	<p>Garante que o ambiente e os bens de teste sejam gerenciados e mantidos.</p> <p>Responsabilidades:</p> <ol style="list-style-type: none"> <li>1. Instalar ferramentas necessárias para as etapas de testes.</li> <li>2. administrar o sistema de gerenciamento teste</li> </ol>

		3. gerenciar o acesso dos trabalhadores ao sistema de testes
Administrador do banco de dados	Luiz Gustavo e Gustavo Souto	Garante que o ambiente e bens de teste de dados (banco de dados) sejam gerenciados e mantidos.  Responsabilidades:  Administrar os dados de teste (base de dados).
Designer	Pamela e Aline	Identifica e define as operações, atributos, e associações das classes de teste.  Responsabilidades: 1. identificar e definir as classes de teste 2. identificar e definir os pacotes de teste
Desenvolvedores	Breno e Luiz Carlos	Implementa e faz os testes unitários das classes e pacotes de teste.  Responsabilidades:  cria as classes e pacotes de teste implementados no modelo de teste

## 4.2- Recursos de Sistema

Recursos do Sistema
<b>Servidor de Banco de Dados</b>
MySQL DataBase Server
<b>Terminais Clientes</b>
4 notebooks conectados via internet
<b>Repositório de Testes</b>
Um computador com os testes guardados no HD Googledocs para compartilhamento de casos de testes 4 notebooks para o desenvolvimento de testes

## 5- Cronograma

<b>Milestone</b>	<b>Data de início</b>	<b>Data de entrega</b>
<b>Planejar Teste</b>	16/10/2010	18/10/2010
<b>Projetar Teste</b>	17/10/2010	19/10/2010
<b>Implementar Teste</b>	28/10/2010	04/11/2010
<b>Executar Teste</b>	05/11/2010	08/11/2010
<b>Avaliar Teste</b>	08/11/2010	10/11/2010