

Aurora

# AfterLife

Plano de testes

Felipe Duarte de Santana  
Guilherme José Carvalho Cavalcanti  
Igor Leonardo Rodrigues Gomes  
Jonathas Mendes Alves  
Mailson Daniel Lira Menezes  
Mateus Gondim Romão Batista  
14/10/2010



## Histórico das revisões

Data	Versão	Descrição	Autor
14/10/2010	1.0	Elaboração inicial do documento	Felipe Duarte Jonathas Alves Igor Gomes Guilherme Cavalcanti Mailson Lira Mateus Gondim

## Conteúdo

1.	Introdução .....	6
1.1.	Objetivos deste Documento .....	6
1.2.	Escopo .....	6
1.3.	Identificação do Projeto .....	7
2.	Requisitos a testar .....	8
2.1.	Teste de Banco de dados .....	8
2.2.	Teste Funcional .....	8
2.3.	Testes de Ciclo de Negócios .....	8
2.4.	Teste de Interface do Usuário .....	8
2.5.	Teste de performance .....	9
2.6.	Teste de estresse .....	9
2.7.	Teste de segurança e controle de acesso .....	9
2.8.	Teste de falha/recuperação .....	9
2.9.	Teste de instalação .....	9
3.	Estratégia de teste .....	10
3.1.	Tipos de teste .....	10
3.1.1.	Teste de Banco de Dados e Integridade de Dados: .....	10
3.1.2.	Teste Funcional .....	10
3.1.3.	Testes de Ciclo de Negócios .....	11
3.1.4.	Teste de Interface do Usuário .....	11
3.1.5.	Teste de performance .....	12
3.1.6.	Teste de estresse .....	12
3.1.7.	Teste de segurança e controle de acesso .....	13
3.1.8.	Teste de falha/recuperação .....	13
3.1.9.	Teste de instalação .....	13
3.2.	Ferramentas .....	14
4.	Recursos .....	15
4.1.	Sistema .....	15
4.2.	Recursos Humanos .....	15
5.	Cronograma .....	16



# 1. Introdução

## 1.1. Objetivos deste Documento

O presente documento tem como objetivo declarar os componentes do software AfterLife a serem testados, listando os requisitos a serem testados e as formas de testes empregadas durante o processo.

Também é importante listar os recursos alocados para os testes, fazendo uma estimativa de esforços de testes.

## 1.2. Escopo

Os testes estão programados para serem realizados em etapas, no total de quatro. São elas: teste de unidade, teste de integração, teste de sistema, teste de aceitação. O teste de unidade será realizado primeiro. Ele visa garantir que cada componente do programa está funcionando adequadamente de forma isolada. Gui, banco de dados, negócios serão testados nesta etapa. Passada esta etapa, será feito o teste de integração, onde os componentes individuais serão juntados e seu comportamento trabalhando em conjunto será analisado. Esta etapa é muito importante, pois componentes isolados podem funcionar perfeitamente e gerarem grandes erros ao trabalharem em conjunto.

Passadas essas duas etapas de “corretude” básica do sistema, terá início o teste de sistema, que tem como objetivo medir desempenho, tempo de resposta, confiabilidade e questões relativas a segurança de restrições de acesso.

Por último, será realizado o teste de aceitação. É nesta etapa onde o usuário final terá a oportunidade de testar o sistema. É muito importante dá atenção especial a esta etapa, pois é onde testaremos a recepção dos usuários/clientes ao sistema.

### 1.3. Identificação do Projeto

Na tabela abaixo, estarão contidos a situação corrente em termos de documentos disponíveis no momento de elaboração do plano de teste.

Documento	Criado ou Disponível	Recebido ou Revisado
Especificação de Requisitos	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não
Plano de Projeto	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não
Modelo de Análise	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não
Modelo de Projeto	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não
Documento de Arquitetura	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não
Protótipo	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não
Manual do Usuário	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não
Listas de Riscos	<input checked="" type="checkbox"/> Sim <input checked="" type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não

## 2. Requisitos a testar

Serão listadas as partes do sistema que devem ser testadas durante e após sua implementação. Busca-se a concordância do comportamento do sistema com a especificação.

### 2.1. Teste de Banco de dados

- Verificar se o sistema é capaz de cadastrar, remover e atualizar um atendente.
- Verificar se o sistema é capaz de cadastrar, remover e atualizar um falecido.
- Verificar se o sistema é capaz de cadastrar, remover e atualizar um responsável.
- Verificar se o sistema é capaz de cadastrar, remover e atualizar um gerente.
- Verificar se o sistema é capaz de cadastrar, remover e atualizar um cozeiro.
- Verificar se o sistema é capaz de atualizar um administrador.
- Verificar se o sistema é capaz de agendar um velório.
- Verificar se o sistema é capaz de agendar um enterro.
- Verificar se o sistema é capaz de buscar um falecido por CPF, localização, óbito, nome ou registro.
- Verificar se que o sistema é capaz de fornecer todas as informações necessárias para a geração dos relatórios requisitados.
- Testar se a persistência do banco de dados está funcionando como o esperado.
- Verificar se as informações do banco de dados são consistentes e não redundantes.

### 2.2. Teste Funcional

- Verificar se apenas o administrador, o gerente e o atendente podem se logar no sistema através de seu login e senha.
- Verificar se o nível de privilégio está correto e garantido a cada usuário que estiver habilitado a se logar no sistema.
- O sistema deverá lembrar quando fizer 2 anos que o falecido foi enterrado.
- Conferir se os relatórios estão sendo impressos segundo o molde desejado.

### 2.3. Testes de Ciclo de Negócios

- Verificar a correta execução de sequências de eventos logicamente associados, correspondentes a atividades comumente realizadas.

### 2.4. Teste de Interface do Usuário

- Verificar a dificuldade de instruir um usuário a utilizar o sistema.

- Navegar por todas as opções e menus e verificar se as transições são coerentes;
- Verificar se todas as palavras e expressões apresentadas estão em conformidade com as devidas normas sintáticas e gramaticais.
- Verificar se as interfaces estão padronizadas.
- Verificar a consistência das interfaces.

## 2.5. Teste de performance

- Verifica o tempo de resposta do enlace entre o servidor de banco de dados e os terminais do sistema;
- Verifica se o sistema é executado em tempo aceitável mesmo com a presença de outros processos rodando na máquina.

## 2.6. Teste de estresse

- Verifica o tempo de resposta do sistema ao utilizar o maior número de ferramentas possíveis;
- Verifica o tempo de resposta do sistema quando o processo estiver rodando concorrentemente com outras aplicações.

## 2.7. Teste de segurança e controle de acesso

- Verificar se apenas os usuários cadastrados podem ter acesso ao sistema;
- Verificar se todas as operações estão restritas para os usuários devidamente autorizados.

## 2.8. Teste de falha/recuperação

- Verifica se o sistema responde a todas as falhas e se consegue contorná-las.

## 2.9. Teste de instalação

- Verificar se o sistema é instalado corretamente em todas as máquinas suportadas;
- Verificar se o espaço em disco na máquina é suficiente para comportar todo o sistema instalado e os dados que serão inseridos;
- Verificar que, após a desinstalação, nenhum componente usado apenas pelo sistema continue na máquina.

## 3. Estratégia de teste

### 3.1. Tipos de teste

#### 3.1.1. Teste de Banco de Dados e Integridade de Dados:

<b>Objetivo do teste</b>	O teste de banco de dados deve garantir que tanto os processos que acessam o banco de dados, como os métodos que o manipulam funcionam de maneira correta e sem corromper os dados.
<b>Técnica</b>	<ul style="list-style-type: none"> <li>- Alimentar o banco de dados com dados similares aos que serão utilizados no dia-a-dia do software, mas que reflitam uma possível realidade para a aproximação com o ambiente real da aplicação.</li> <li>- Invocar todos os métodos que envolvem o banco de dados, sendo cada método alimentado com dados válidos, inválidos e incompletos.</li> <li>- Depois das chamadas dos métodos, verificar o banco de dados para checar a corretude das alterações, além disso, revisar os dados com os métodos de acesso ao banco de dados, para analisar se a recuperação também está funcionando de forma correta.</li> </ul>
<b>Critério de finalização</b>	<ul style="list-style-type: none"> <li>- Todos os métodos e processos de acesso à base de dados funcionam como projetados e sem nenhuma corrupção de dados.</li> <li>- Todos os eventos do banco de dados são acionados no momento correto.</li> <li>- A estrutura das entidades do banco de dados é mantida mesmo após diversas manipulações.</li> </ul>
<b>Considerações especiais</b>	<ul style="list-style-type: none"> <li>- O teste necessita de um ambiente de desenvolvimento de SGBD para inserir ou modificar os dados diretamente na base de dados, autorização dada pelo CIn.</li> <li>- Os testes serão executados sem automatização, assim, as invocações serão manuais.</li> <li>- Deve-se utilizar uma base de dados não muito grande para facilitar o processo de averiguação dos resultados após cada operação e ajudar com uma maior visibilidade de eventos.</li> </ul>

#### 3.1.2. Teste Funcional

<b>Objetivo do teste</b>	- O teste funcional visa garantir a corretude das diversas
--------------------------	--

	funcionalidades do sistema que envolve: cadastros, buscas, remoções, atualizações, gerações de relatórios. Assim, os testes funcionais garantirão a funcionalidade apropriada do alvo do teste, incluindo navegação, entrada de dados, processamento, e recuperação.
<b>Técnica</b>	<ul style="list-style-type: none"> <li>- Executar as funções implementadas inserindo dados válidos e inválidos para verificar se:</li> <li>- Os resultados corretos são constatados com dados válidos;</li> <li>- Mensagens de erro são exibidas para os dados inválidos;</li> <li>- Gerar todos os possíveis relatórios e verificar a corretude.</li> </ul>
<b>Critério de finalização</b>	<ul style="list-style-type: none"> <li>- Correção de todos os defeitos detectados durante os testes.</li> <li>- Todos os testes planejados são executados e obtêm resultados satisfatórios.</li> </ul>
<b>Considerações especiais</b>	- Abordagens bottom-up podem simplificar este teste, pois evitam que erros em níveis mais baixos de implementação resultem em falhas de maiores proporções e de difícil identificação de causa.

### 3.1.3. Testes de Ciclo de Negócios

<b>Objetivo do teste</b>	- Conferir se as regras de negócio funcionam corretamente.
<b>Técnica</b>	- Realizar os casos de uso definidos, analisando se estes se comportam de forma correta no sistema e verificar se o mesmo responde de forma adequada a ações não previstas.
<b>Critério de finalização</b>	- Após a realização dos testes, verifica-se que os casos de uso estão funcionando corretamente.
<b>Considerações especiais</b>	Nenhuma

### 3.1.4. Teste de Interface do Usuário

<b>Objetivo do teste</b>	<ul style="list-style-type: none"> <li>- Detectar possível falha do software na facilidade de interação com o cliente, visando minimizar custo com treinamento e aumentar a produção.</li> <li>- Analisar as disposições dos componentes nos painéis da interface gráfica, como barra de menu, posições de botões, caixa texto, entre outros, assim como dimensões e posicionamento das janelas, garantindo facilidade de utilização pelo usuário.</li> </ul>
<b>Técnica</b>	<ul style="list-style-type: none"> <li>- No primeiro momento, a própria equipe de testes irá realizar modificações nos testes ou criar novos testes para checar a navegação e os estados dos objetos em cada janela.</li> <li>- Validada, a interface será repassada para potenciais</li> </ul>

	usuários finais, os quais receberão um breve treinamento, para familiarizar-se com as operações que pode realizar e seu desempenho será avaliado em listas de tarefas de acordo com tempo e relatos de experiência dos mesmos.
<b>Critério de finalização</b>	- A interface deve ser aprovada por uma maioria considerável do grupo avaliado e problemas encontrados, após solucionados, passam pelo mesmo processo. Além desta aprovação, deve-se perceber que mesmo que ela ocorra, fatores como tempo devem ser levados em consideração, evitando que o resultado não seja condizente com a realidade encontrada pelos usuários.
<b>Considerações especiais</b>	Levar em consideração o grau de instrução dos usuários, familiaridades com tecnologias e mostrar ao usuário todas as atividades que o sistema pode fazer de modo a se ter conhecimento sobre todas as funcionalidades do sistema e os painéis da interface gráfica associados a cada uma delas.

### 3.1.5. Teste de performance

<b>Objetivo do teste</b>	Verificar se o tempo de resposta para funções do sistema são aceitáveis.
<b>Técnica</b>	Realizar múltiplas atividades com um único usuário e analisar o tempo de resposta. Realizar múltiplas consultas ao banco de dados variando no volume de informações transferidas.
<b>Critério de finalização</b>	O tempo de resposta para os testes foi considerado satisfatório.
<b>Considerações especiais</b>	O teste deve ser realizado em máquinas com o menor número de processos rodando, a fim de avaliar seu verdadeiro desempenho; como também em máquinas com vários processos rodando em concorrência, com o intuito de avaliar seu comportamento em situações extremas.

### 3.1.6. Teste de estresse

<b>Objetivo do teste</b>	Verificar o comportamento em situações limites, bem como avaliar se o sistema é executado corretamente em situações adversas.
<b>Técnica</b>	Usar aleatoriamente e de forma exaustiva todas as funcionalidades do sistema, a fim de verificar se alguma funcionalidade ocasiona o defeito de outra. Executar o sistema em uma máquina com os recursos de hardware mínimos exigidos.
<b>Critério de finalização</b>	Quando a performance do sistema estiver abaixo do limite esperado.

<b>Considerações especiais</b>	O teste de estresse deve ser feito em conjunto com o teste de performance, já que em ambos é estudado o tempo de respostas nas mais variadas situações.
--------------------------------	---

### 3.1.7. Teste de segurança e controle de acesso

<b>Objetivo do teste</b>	Verificar se é garantido, a cada tipo de usuário do sistema, acesso apenas às funcionalidades definidas na especificação.
<b>Técnica</b>	Listar os tipos de usuários do sistema. Verificar se cada um deles tem acesso ao que lhes é permitido. Tentar acessar funcionalidades não permitidas a cada tipo de usuário para verificar se as restrições de acesso estão sendo feitas.
<b>Critério de finalização</b>	Todos os diferentes tipos de usuário tem acesso apenas às funcionalidades devidamente autorizadas.
<b>Considerações especiais</b>	Os usuários devem informar um login e uma senha para acessar o sistema. Verificar, portanto, se apenas usuários logados podem acessá-los.

### 3.1.8. Teste de falha/recuperação

<b>Objetivo do teste</b>	Garantir que o sistema volte a um estado de estabilização após uma ocorrência de falha.
<b>Técnica</b>	Submeter a falhas como queda de energia, encerramento brusco do sistema, reboot. Após isso, verificar se o sistema conseguiu se recuperar mesmo que tenha sido fechado abruptamente durante a execução de uma tarefa. Garantir que informações inseridas, porém não enviadas ao banco de dados, possam ser recuperadas.
<b>Critério de finalização</b>	Se o sistema conseguir recuperar toda as informações, ou pelo menos boa parte dela, após uma falha.
<b>Considerações especiais</b>	É necessário que, após uma falha abrupta ocorrida intencionalmente, o sistema seja executado novamente, a fim de verificar a integridade dos dados.

### 3.1.9. Teste de instalação

<b>Objetivo do teste</b>	Verificar se o sistema é instalado corretamente, juntamente com todos os componentes necessários para sua execução. Garantir que todos os componentes exclusivos são excluídos após a remoção do sistema.
<b>Técnica</b>	Instalar e desinstalar o sistema em diversas máquinas e verificar se ele funciona corretamente, e se todos os

	componentes instalados foram devidamente removidos.
<b>Critério de finalização</b>	O sistema é instalado/desinstalado corretamente em todas as máquinas testadas.
<b>Considerações especiais</b>	A configuração das máquinas testadas deve ser semelhante a das máquinas do cliente.

## 3.2. Ferramentas

Ferramentas utilizadas pelos procedimentos descritos nesse documento.

	Ferramenta	Organização
<b>Gerenciamento de testes</b>	Rational RequisitePRO	Rational
	Rational Unified Process	
<b>Projeto de teste</b>	Rational Rose	Rational
<b>Realização de testes</b>	JUnit	JUnit.org
<b>Gerenciamento de projeto</b>	Microsoft Project	Microsoft
<b>Ferramentas do SGBD</b>	MySQL Query	MySQL AB
	PL/SQL Developer	Oracle

## 4. Recursos

### 4.1. Sistema

Recursos do Sistema	
<b>Desktop</b>	2 PCs conectados a internet com configuração mínima: Processador Intel Pentium 4 ou equivalente, 512 MB de memória ram, 500 MB de espaço livre em disco, Sistema Operacional Windows XP ou superior.
<b>Sistema de banco de dados</b>	É necessário um pc conectado a internet para teste com o banco de dados Oracle

### 4.2. Recursos Humanos

Recursos Humanos		
Função	Responsabilidades Específicas	Trabalhador
<b>Gerente de Teste</b>	Supervisiona o desenvolvimento e a aplicação dos testes. Elabora relatórios de gerenciamento. - Providenciar os recursos necessários aos testes.	Mateus Gondin
<b>Design de Testes</b>	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.	Mailson Daniel
<b>Testadores</b>	Executa os testes. Responsabilidades: Executar os testes; Registrar os resultados; Restabelecer-se dos erros; Documentar solicitações de mudança.	Igor Leonardo e Felipe

## 5. Cronograma

<b>Milestone</b>	<b>Data de início</b>	<b>Data de término</b>
<b>Planejar teste</b>	14/10/2010	18/10/2010
<b>Projetar teste</b>	14/10/2010	18/10/2010
<b>Implementar teste</b>	19/10/2010	03/11/2010
<b>Executar teste</b>	04/11/2010	08/11/2010
<b>Analisar teste</b>	04/11/2010	08/11/2010