
<Petlife>

<Petlife>
<Iteração/Mestre> Plano de Teste

Versão <1.0>

<Petlife>	Version: <1.0>
<Iteração/Mestre> Plano de Teste	Date: <dd/mm/yy>
<document identifier>	

Histórico da Revisão

Data	Versão	Descrição	Autor
15/08/2021	1.0	Teste de banco de dados (usuário/produto/pedido)	Helio
18/08/2021	1.0	Teste de banco de dados(imagem/comentário)	Helio
10/09/2021	1.0	Teste de login/verificação/status/rotas	Alice/Hélio
12/09/2021	1.0	Teste de publicação de imagem/ Cadastro de funcionário/produtos	Alice/Hélio
14/09/2021	1.0	Teste de gerenciamento de pedido/Gerenciamento de forma de pagamento/publicação de comentário	Alice/Hélio
16/09/2021	1.0	Carregar carrinho de compras	Alice/Hélio

<Petlife>	Version: <1.0>
<Iteração/Mestre> Plano de Teste	Date: <dd/mm/yy>
<document identifier>	

Índice Analítico

1. Introdução	5
1.1 Finalidade	5
1.2 Escopo	5
1.3 Público-alvo	5
1.4 Referências	5
2. Missão de Avaliação e Motivação dos Testes	6
2.1 Informações Detalhadas	6
2.2 Missão de Avaliação	6
2.3 Motivadores dos Testes	6
3. Itens-alvo dos Testes	6
4. Resumo dos Testes Planejados	6
4.1 Resumo das Inclusões dos Testes	7
4.2 Resumo dos Outros Candidatos a Possível Inclusão	7
4.3 Resumo das Inclusões dos Testes	7
5. Abordagem dos Testes	7
5.1 Técnicas e Tipos de Teste	7
5.1.1 Teste de Integridade de Dados e de Banco de Dados	8
5.1.2 Teste de Função	9
5.1.4 Teste da Interface do Usuário	10
5.1.6 Teste de Carga	10
5.1.9 Teste de Segurança e de Controle de Acesso	12
6. Critérios de Entrada e de Saída	13
6.1 Plano de Teste	13
6.1.1 Critérios de Entrada de Plano de Teste	13
6.1.2 Critérios de Saída de Plano de Teste	13
6.1.3 Critérios de Suspensão e de Reinício	13
6.2 Ciclos de Teste	13
6.2.1 Critérios de Entrada de Ciclo de Teste	13
6.2.2 Critérios de Saída de Ciclo de Teste	13
6.2.3 Término Anormal do Ciclo de Teste	13
7. Produtos Liberados	13
7.1 Sumários de Avaliação de Testes	13
7.2 Relatórios da Cobertura de Teste	13

<Petlife>	Version: <1.0>
<Iteração/Mestre> Plano de Teste	Date: <dd/mmm/yy>
<document identifier>	

7.3	Relatórios da Qualidade Perceptível	14
7.4	Registros de Incidentes e Solicitações de Mudança	14
8.	Fluxo de Trabalho de Teste	14
9.	Necessidades Ambientais	15
9.1	Hardware Básico do Sistema	15
9.2	Elementos de Software Básicos do Ambiente de Teste	15
9.3	Ferramentas de Produtividade e de Suporte	16
9.4	Configurações do Ambiente de Teste	16
10.	Responsabilidades, Perfil da Equipe e Necessidades de Treinamento	16
10.1	Pessoas e Papéis	16
11.	Riscos, Dependências, Suposições e Restrições	17
12.	Procedimentos e Processos de Gerenciamento	18
12.1	Estratégias de Rastreabilidade	18

<Petlife>	Version: <1.0>
<Iteração/Mestre> Plano de Teste	Date: <dd/mm/yy>
<document identifier>	

<Iteração/Mestre> Plano de Teste

1. Introdução

1.1 Finalidade

A finalidade do Plano de Teste de Iteração é reunir todas as informações necessárias ao planejamento e ao controle do 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 *Plano de Teste* referente ao <Petlife> suporta os seguintes objetivos:

- *Identificar informações de projeto existentes e os componentes de software que devem ser testados;*
- *Listar os requisitos a testar(Alto nível);*
- *Recomendar e descrever as estratégias de teste a serem empregadas;*
- *Identifica os recursos necessários e fornece uma estimativa dos esforços de teste;*
- *Lista os elementos liberados do projeto de teste.*

1.2 Escopo

A Petlife passará pelos seguintes testes: unidade, integração e de sistemas. Os testes unitários e de integração vão lidar com a qualidade funcional, das bases de dados, interface gráfica e do controle de acesso; enquanto que os testes de sistema trataram as questões de performance.

Testes utilizados: Teste de Integridade de Dados e de Banco de Dados

Teste de Função

Teste da Interface do Usuário

Teste de Carga

Teste de Stress

Teste de Volume

Teste de Segurança e de Controle de Acesso

Testes não utilizados, pois dependeria mais das funcionalidades do cliente: Teste de Ciclos de Negócios

Determinação do Perfil de Desempenho

Teste de Tolerância a Falhas e de Recuperação

Teste de Configuração

Teste de Instalação

1.3 Público-alvo

Nosso produto será destinado para as pessoas que buscam artigos do ramo pet, por exemplo: ração, petiscos, brinquedos, para seus animais de estimação.

E por meio de divulgações cativar pessoas que buscam animais para adoção às direcionando para as respectivas ONGs.

1.4 Referências

- Roteiro de teste: Interface
https://docs.google.com/spreadsheets/d/104NGrdXeSZZhqDXGKZO8G123-JdYjyI4kHS1_ULKg2I/edit#gid=549607045
- Roteiro de teste: Banco de dados
<https://docs.google.com/spreadsheets/d/1QDLAy2V7DWPgHUNxUl3bfRvlzIBgIpA8ISdu0G34YyY/edit#gid=818768411>

<Petlife>	Version: <1.0>
<Iteração/Mestre> Plano de Teste	Date: <dd/mm/yy>
<document identifier>	

2. Missão de Avaliação e Motivação dos Testes

Para melhor avaliação do produto e usabilidade, antes da liberação para o usuário.

2.1 Informações Detalhadas

- A importância da tecnologia sendo utilizada como uma estratégia para facilitar ao usuário a otimização da venda de seus produtos os organizando de forma eficiente, o principal problema está na dificuldade de agilidades nos processos, falta de segurança e acarretando em perda de tempo e podendo até causar perdas de informações

2.2 Missão de Avaliação

- *localizar o maior número de erros possível;*
- *localizar problemas importantes e avaliar os riscos da qualidade perceptível;*
- *advertir sobre os riscos de projeto percebidos;*
- *certificar um padrão;*
- *verificar uma especificação (requisitos, design ou alegações);*
- *advertir sobre a qualidade do produto e satisfazer os envolvidos;*
- *advertir sobre os testes;*
- *cumprir as determinações do processo.*

2.3 Motivadores dos Testes

Os principais elementos que motivaram o esforço deste teste nesta iteração. Como os riscos de qualidade, riscos técnicos, riscos do projeto, casos de uso, requisitos funcionais, requisitos não funcionais, elementos de design, falhas ou erros suspeitos, solicitações de mudança.

3. Itens-alvo dos Testes

A listagem abaixo identifica os itens _ software, hardware e elementos de suporte do produto _ que foram identificados como alvo dos testes. Essa lista representa os itens que serão testados.

- Casos de uso;
- Requisitos funcionais;
- Requisitos não funcionais;
- Falhas ou erros suspeitos;
- Solicitações de mudança.

4. Resumo dos Testes Planejados

Testes utilizados:

- Teste de Integridade de Dados e de Banco de Dados(Nível alto);
- Teste de Função(Nível alto);
- Teste da Interface do Usuário(Nível alto);
- Teste de Carga(Nível médio);
- Teste de Segurança e de Controle de Acesso(Nível alto).

Testes não utilizados, pois dependeria mais das funcionalidades do cliente:

- Teste de Ciclos de Negócios;
- Determinação do Perfil de Desempenho;
- Teste de Tolerância a Falhas e de Recuperação;

<Petlife>	Version: <1.0>
<Iteração/Mestre> Plano de Teste	Date: <dd/mm/yy>
<document identifier>	

- Teste de Configuração;
- Teste de Instalação;
- Teste de Stress;
- Teste de Volume.

4.1 Resumo das Inclusões dos Testes

- Teste de Integridade de Dados e de Banco de Dados(Nível alto);
 - Verificar informações, atualizar, inserir dados dos usuários e produtos;
- Teste de Função(Nível alto);
 - Verificar que qualquer usuário pode acessar sua própria conta através de login e senha;
- Teste da Interface do Usuário(Nível alto);
 - Verificar cada tela de interface gráfica pode ser rapidamente entendida e facilmente utilizada;
- Teste de Segurança e de Controle de Acesso(Nível alto);
 - Verificar que usuário não cadastrado, não podem acessar informações restritas aos cadastrados e usuários do sistema podem acessar apenas as funcionalidades e dados associados ao seu próprio tipo de usuário.

4.2 Resumo dos Outros Candidatos a Possível Inclusão

- Teste de Carga(Nível médio);
 - Verificar a resposta do sistema com “X” quantidade usuários.

4.3 Resumo das Inclusões dos Testes

- *O teste de configuração, pois este teste depende muito da configuração externa;*
- *O teste de instalação, como essa aplicação é executado em browser não há necessidade de realizar o teste;*
- *O teste de ciclos de negócios, como estamos utilizando o modelo de processo Scrum, isso já está sendo analisado;*
- *O Teste de Tolerância a Falhas e de Recuperação*
- *Determinação do Perfil de Desempenho, é desnecessária devido ao teste executado por teste de carga;*
- *O teste de stress é desnecessário devido ao teste executado por teste de carga.*

5. Abordagem dos Testes

- Teste de função:
 - Funções incorretas ou omitidas;
 - Erros de interface;
 - Erros de comportamento ou desempenho;
 - Erros de inicialização e término.
- a. Como será realizado: Analisar os dados resultantes e melhorar validações e verificações de entrada.
- Teste de Integridade de Dados e de Banco de Dados:
 - Acessando um banco de dados;
 - Buscando informações;
 - Inserção de dados;
 - Deletar dados;
 - Alterar dados;
- a. Como será realizado: Construir de forma sistemática, a estrutura do programa, ao mesmo tempo, executando testes para descobrir erros.
- Teste da interface do usuário:
 - Verificar a facilidade;
 - Aparência como, fundo, fontes e tamanhos da imagem;

<Petlife>	Version: <1.0>
<Iteração/Mestre> Plano de Teste	Date: <dd/mm/yy>
<document identifier>	

- a. Como será realizado: Levantamento em conjunto com os reais usuários.
- Teste de Segurança e de Controle de Acesso:
 - Apenas usuário cadastrado tem acesso às suas contas cadastradas;
 - Alguns usuários(administrador) pode utilizar algumas funções;
- a. Como será realizado: Verificar o acesso ao sistema com respectivos usuários.
- Teste de Carga:
 - Avaliar a resposta do sistema em condições extremas de carga de informações.
- a. Como será realizado: Verificar a resposta do sistema com “X” usuários.

5.1. Técnicas e Tipos de Teste

5.1.1 Teste de Integridade de Dados e de Banco de Dados

Os bancos de dados e os processos de banco de dados deverão ser testados como um subsistema independente. Esse teste deve testar os subsistemas sem que a interface do usuário do objetivo do teste faça interface com os dados.

Objetivo da Técnica:	<i>Testar processos e métodos de acesso a banco de dados independentes da UI para que você possa observar e registrar comportamentos-alvo incorretos ou a existência de dados corrompidos.</i>
Técnica:	<ul style="list-style-type: none"> · <i>Dispare cada processo e método de acesso a banco de dados, propagando solicitações de dados ou dados válidos e inválidos em cada um deles.</i> · <i>Inspecione o banco de dados para assegurar que os dados foram distribuídos conforme o planejado e que todos os eventos de banco de dados ocorreram de forma adequada, ou revise os dados retornados para assegurar que os dados corretos foram recuperados pelas razões corretas.</i>
Estratégias:	<ul style="list-style-type: none"> - <i>Verifique que as informações do usuário podem ser cadastradas, consultadas e removidas.</i> - <i>Verifique que as informações do produto podem ser cadastradas, consultadas e removidas.</i> - <i>Verifique que as informações do pedido podem ser cadastradas, consultadas e removidas.</i> - <i>Verifique que as informações sobre o produto/usuário possam ser cadastradas, removidas e atualizadas pelo administrador do sistema.</i>
Ferramentas Necessárias:	<p><i>A técnica exige as seguintes ferramentas:</i></p> <ul style="list-style-type: none"> • <i>Ferramenta de Automação de Scripts de Teste</i> • <i>restaurador e reprodutor de imagem da configuração básica</i> • <i>ferramentas de backup e de recuperação</i> • <i>ferramentas de monitoramento de instalação (registro, disco rígido, CPU, memória, etc.)</i> • <i>ferramentas e utilitários SQL de banco de dados</i> • <i>ferramentas de geração de dados</i>
CrITÉRIOS de Êxito:	<i>Todos os métodos e processos de acesso à base de dados funcionam como projetados e sem nenhuma corrupção de dados.</i>

<Petlife>	Version: <1.0>
<Iteração/Mestre> Plano de Teste	Date: <dd/mm/yy>
<document identifier>	

Considerações Especiais:	<ul style="list-style-type: none"> Os processos deverão ser disparados manualmente. Deverão ser usados bancos de dados pequenos ou de tamanho mínimo (com um número limitado de registros) para aumentar a visibilidade de quaisquer eventos não aceitáveis.
--------------------------	--

5.1.2. Teste de Função

O teste de função do objetivo do teste deve concentrar-se em todos os requisitos de teste que possam ser diretamente associados a casos de uso ou funções e regras de negócios. A meta desse teste é verificar a adequada aceitação, o processamento e a recuperação dos dados, e a implementação apropriada das regras de negócios. Esse tipo de teste baseia-se em técnicas de caixa preta; ou seja, verificar o aplicativo e seus processos internos interagindo com o aplicativo através da Interface Gráfica do Usuário (GUI) e analisar a saída ou os resultados.

Objetivo da Técnica:	Testar a funcionalidade do objetivo do teste, incluindo a navegação, a entrada, o processamento e a recuperação de dados a fim de observar e registrar o comportamento-alvo.
Técnica:	<p>Execute os recursos e os fluxos ou as funções de cada um dos cenários de caso de uso, utilizando dados válidos e inválidos para verificar se:</p> <ul style="list-style-type: none"> os resultados esperados ocorrerá quando forem usados dados válidos as mensagens de erro ou de aviso apropriadas serão exibidas quando forem usados dados inválidos cada regra de negócio será aplicada de forma adequada
Estratégias:	<ul style="list-style-type: none"> Verifique que as informações úteis obtidas pelo subsistema responsável são automaticamente e periodicamente atualizadas; Verifique que qualquer usuário pode acessar sua própria conta através de login e senha;
Ferramentas Necessárias:	<p>A técnica exige as seguintes ferramentas:</p> <ul style="list-style-type: none"> Ferramenta de Automação de Scripts de Teste restaurador e reproduzidor de imagem da configuração básica ferramentas de backup e de recuperação ferramentas de monitoramento de instalação (registro, disco rígido, CPU, memória, etc.) ferramentas de geração de dados
CrITÉrios de Êxito:	<p>Todos os testes planejados foram executados.</p> <p>Todos os defeitos identificados foram tratados.</p>
Considerações Especiais:	Nenhum

<Petlife>	Version: <1.0>
<Iteração/Mestre> Plano de Teste	Date: <dd/mm/yy>
<document identifier>	

5.1.2 Teste da Interface do Usuário

O Teste da Interface do Usuário (UI) verifica a interação do usuário com o software. A meta do teste de UI é assegurar que a UI forneça ao usuário o acesso e a navegação adequados através das funções do objetivo do teste. Além disso, o teste de UI assegura que os objetos contidos na UI funcionem conforme o esperado e estejam em conformidade com padrões corporativos ou da indústria.

Objetivo da Técnica:	<p>Testar os seguintes pontos para observar e registrar a conformidade com padrões e o comportamento-alvo:</p> <ul style="list-style-type: none"> • A navegação pelo objetivo do teste para verificar se reflete os requisitos e as 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). • Os objetos e as características das janelas poderão ser experimentados como, por exemplo, menus, tamanho, posição, estado e foco.
Técnica:	Criar ou modificar testes para cada janela a fim de verificar a navegação adequada e os estados de objeto apropriados para cada janela e objeto do aplicativo
Estratégias:	- Navegue através de todos os use cases, verificando que cada tela de interface gráfica pode ser rapidamente entendida e facilmente utilizada;
Ferramentas Necessárias:	Necessita da Ferramenta de Automação de Scripts de Teste.
Crterios de Êxito:	A técnica suporta o teste de cada tela ou janela principal que será muito usada pelo usuário final.
Considerações Especiais:	Nem todas as propriedades referentes a objetos personalizados e de terceiros poderão ser acessadas.

5.1.3 Teste de Carga

O teste de carga é um teste de desempenho que revela o objetivo do teste a diferentes cargas de trabalho para medir e avaliar as habilidades e os comportamentos de desempenho dele, a fim de verificar se este continua a funcionar adequadamente com essas diferentes cargas de trabalho. A meta desse teste de carga é determinar e assegurar que o sistema funcione adequadamente com uma carga de trabalho superior à carga máxima esperada. Além disso, esse teste avalia as características do desempenho como, por exemplo, tempos de resposta, taxas de transação e outros aspectos que mudam com o tempo.

Objetivo da Técnica:	Testar casos de negócio ou transações designadas em várias condições de carga de trabalho, a fim de observar e registrar o comportamento-alvo e os dados de desempenho do sistema.
----------------------	--

<Petlife>	Version: <1.0>
<Iteração/Mestre> Plano de Teste	Date: <dd/mm/yy>
<document identifier>	

Técnica:	<ul style="list-style-type: none"> · Utilizar os Scripts de Teste de Transação desenvolvidos para os Testes de Ciclos de Negócios ou de Funcionamento como uma base, mas lembrar de remover as iterações e os atrasos desnecessários. · Modificar os arquivos de dados a fim de aumentar o número de transações ou modificar os testes a fim de aumentar o número de vezes que cada transação ocorre. · As Cargas de Trabalho devem incluir cargas de Pico, por exemplo, Diárias, Semanais, Mensais, etc. · As Cargas de Trabalho devem representar cargas Médias assim como cargas de Pico. · As Cargas de Trabalho devem representar Picos Instantâneos e Picos Sustentados. · As Cargas de Trabalho devem ser executadas com diferentes configurações do Ambiente de Teste.
Estratégias:	- Verificar a resposta do sistema com “X” usuários.
Ferramentas Necessárias:	<p>A técnica exige as seguintes ferramentas:</p> <ul style="list-style-type: none"> • Ferramenta de Automação de Scripts de Teste • ferramenta de controle e de Programação de Carga de Transações • ferramentas de monitoramento de instalação (registro, disco rígido, CPU, memória, etc.) • ferramentas de restrição de recursos (por exemplo, Enlatados) • ferramentas de geração de dados
CrITÉRIOS de Êxito:	A técnica suporta o teste de Emulação de Carga de Trabalho, que é a emulação bem-sucedida da carga de trabalho sem que nenhuma falha ocorra devido a problemas de implementação do teste.
Considerações Especiais:	<ul style="list-style-type: none"> · O teste de carga deverá ser executado em uma máquina dedicada ou em um período de tempo dedicado. Isso permitirá o controle total e a medição exata. · Os bancos de dados usados para o teste de carga deverão ter um tamanho real ou deverão ser dimensionados igualmente em escala.

<Petlife>	Version: <1.0>
<Iteração/Mestre> Plano de Teste	Date: <dd/mm/yy>
<document identifier>	

5.1.4 Teste de Segurança e de Controle de Acesso

O Teste de Segurança e de Controle de Acesso concentra-se em duas áreas de segurança principais:

- Segurança no nível do aplicativo, incluindo o acesso aos Dados ou às Funções de Negócios
- Segurança no nível do sistema, incluindo efetuar login.

Com base no nível de segurança desejado, a segurança no nível do aplicativo assegura que os atores estejam restritos a funções ou casos de uso específicos, ou que tenham acesso limitado aos dados disponíveis. Por exemplo, todos têm permissão para inserir dados e criar novas contas, mas apenas os gerentes poderão excluí-los. Se houver segurança no nível dos dados, o teste assegurará que o “tipo de usuário um” possa ver todas as informações de um cliente.

Objetivo da Técnica:	<p>Testar o objetivo do teste nas seguintes condições para observar e registrar o comportamento-alvo:</p> <ul style="list-style-type: none"> • Segurança no nível do aplicativo: um ator poderá acessar somente as funções ou os dados para o qual seu tipo de usuário tenha recebido permissão. • Segurança no nível do sistema: somente os atores com acesso ao sistema e aos aplicativos têm permissão para acessá-los.
Técnica:	<ul style="list-style-type: none"> • Identifique e liste cada tipo de usuário e as funções ou os dados para os quais cada um deles tem permissão de acesso.] <ul style="list-style-type: none"> ○ Crie testes para cada tipo de usuário e verifique cada permissão criando transações específicas para cada tipo de usuário. ○ Modifique o tipo de usuário e execute novamente os testes para os mesmos usuários. Em cada caso, verifique se as funções ou dados adicionais estão corretamente disponíveis ou se têm seu acesso negado. • Acesso no nível do sistema: [Consulte Considerações Especiais abaixo]
Estratégias:	<ul style="list-style-type: none"> - Verificar que usuários não cadastrados não podem acessar informações restritas aos cadastrados; - Verificar que além do administrador, ninguém mais pode remover dados do sistema; - Verificar que os usuários do sistema podem acessar apenas as funcionalidades e dados associados ao seu próprio tipo de usuário.
Ferramentas Necessárias:	<p>A técnica exige as seguintes ferramentas:</p> <ul style="list-style-type: none"> • Ferramenta de Automação de Scripts de Teste • Ferramentas de investigação e contra a violação da segurança por “hackers” • Ferramentas de Administração da Segurança do Sistema Operacional
CrITÉRIOS de Êxito:	A técnica suporta, para cada tipo de ator conhecido, o teste das funções

<Petlife>	Version: <1.0>
<Iteração/Mestre> Plano de Teste	Date: <dd/mm/yy>
<document identifier>	

	<i>apropriadas ou dos dados afetados pelas configurações de segurança.</i>
Considerações Especiais:	<i>O acesso ao sistema deverá ser revisto ou discutido com o administrador de sistemas ou de rede adequada. Talvez esse teste não seja necessário, já que poderá ser uma das funções da administração de sistemas ou de rede.</i>

6. Critérios de Entrada e de Saída

6.1 Plano de Teste

6.1.1 Critérios de Entrada de Plano de Teste

O plano de teste se inicializará ao término da implementação dos requisitos especificados no documento de requisitos.

6.1.2 Critérios de Saída de Plano de Teste

A conclusão dos testes se dará quando todos os requisitos forem testados, e não podendo ser encontrado erros. Em casos de algum bugs no processo, os testes serão suspensos até que os problemas sejam solucionados e assim dar continuidade aos testes.

6.1.3 Critérios de Suspensão e de Reinício

No caso de problemas que impossibilitem a execução da lista de requisitos a serem testados, serão suspensos os testes até que sejam solucionados, só assim os demais requisitos serão novamente testados e validados. Caso tenha um problema que possa continuar com teste, terá continuidade, mas quando solucionar o problema o teste será realizado novamente. Quando todos os requisitos forem testados sem nenhum erro, os mesmos serão dados como concluídos, e assim será feito até que termine toda lista com sucesso.

6.2 Ciclos de Teste

6.2.1 Critérios de Entrada de Ciclo de Teste

Os ciclos de teste seguirão a ordem especificada do plano de testes.

6.2.2 Critérios de Saída de Ciclo de Teste

Os testes deverão passar por ciclos, sendo que todos os defeitos e exceções terão que estar corrigidos.

6.2.3 Término Anormal do Ciclo de Teste

Só ocorrerá se tiver falhas no planejamento do plano de teste.

7. Produtos Liberados

7.1 Sumários de Avaliação de Testes

- Teste de Integridade de Dados e de Banco de Dados;
Frequência: A cada alteração do código relacionado ao banco de dados.
- Teste de Segurança e de Controle de Acesso;
Frequência: Quando é implementado um novo componente (Página).
- Teste da Interface do Usuário.
Frequência: Cada alteração e cada adição de um novo componente na interface
- Teste de Carga;
Frequência: A cada implementação nova ou alteração.

7.2 Relatórios da Cobertura de Teste

- Teste de Carga foi utilizado loadView-TedotComMonitor, exibindo o relatório em forma de gráfico.
- Teste unitário(BD e Controle de acesso) foi utilizado a Jest, testando cada entrada e a saída de valores. Os dados são exibidos no terminal, com essa informação são feitos os relatórios.

<Petlife>	Version: <1.0>
<Iteração/Mestre> Plano de Teste	Date: <dd/mm/yy>
<document identifier>	

7.3 Relatórios da Qualidade Perceptível

- Ao longo da cobertura de teste, houve uma solicitação de mudança na interface, realizamos as mudanças e realizamos os testes e não teve nenhum problema;
- Ao longo da cobertura de teste, houve alguns incidentes que no banco de dados, as informações digitadas pelo usuário não estavam sendo gravadas corretamente, foi necessário realizar algumas correções.

7.4 Registros de Incidentes e Solicitações de Mudança

- Na interface do usuário não existe um botão para retornar/voltar à

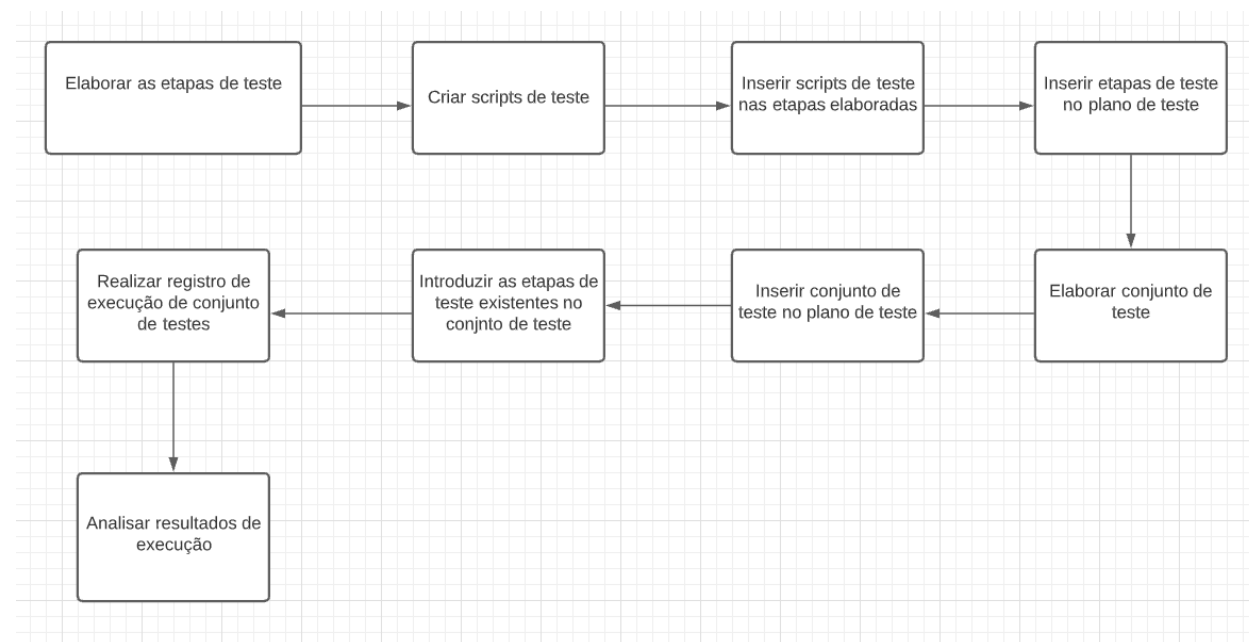
tela. - Tela de Login não é situada na prototipação.

- Na figura 6, o botão de finalizar a compra (vermelho) não está intuitivo pois a cor dá a entender que seria o botão de cancelar.

- Não existe o botão de cancelamento de um produto.

- Não existe o botão de registro no site.

8. Fluxo de Trabalho de Teste



1- Elaborar as etapas de teste;

2 - Criar scripts de teste;

3- Inserir scripts de teste nas etapas elaboradas;

4 - Inserir etapas de teste no plano de teste;

5 - Elaborar conjunto de teste;

<Petlife>	Version: <1.0>
<Iteração/Mestre> Plano de Teste	Date: <dd/mm/yy>
<document identifier>	

6 - Inserir conjunto de teste no plano de teste;

7 - Introduzir as etapas de teste existentes no conjunto de teste;

8 - Realizar registro de execução de conjunto de teste;

9 - Analisar resultados de execução.

9. Necessidades Ambientais

São necessários os seguintes elementos de softwares básicos no ambiente de teste deste Plano de Teste:

- Windows 10 / Linux;
- Navegador Mozilla Firefox, Google Chrome, Safari e Internet Explorer.

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.

Recursos do Sistema	
Recurso	Nome e Tipo
Servidor de Banco de Dados	MongoDB
—Rede ou Sub-rede	DHCP
—Nome do Servidor	nodejs
Nome do Banco de Dados	petlife_db
PCs de Teste Cliente	Processador de 400MHz ou mais recente compatível com SSE3
—Rede ou Sub-rede	DHCP
—Nome do Servidor	nodejs
PCs de Desenvolvimento de Teste	Processador de 400MHz ou mais recente compatível com SSE3

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 10	recente	Sistema Operacional
Linux	recente	Sistema Operacional
Internet Explorer	recente	Navegador da Internet
Google Chrome	recente	Navegador da Internet
Gmail/Outlook/	recente	Software Cliente de E-Mail
Windows defender	recente	Software de Detecção e Recuperação de Vírus

<Petlife>	Version: <1.0>
<Iteração/Mestre> Plano de Teste	Date: <dd/mm/yy>
<document identifier>	

9.3 Ferramentas de Produtividade e de Suporte

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

Categoria ou Tipo de Ferramenta	Nome da Marca da Ferramenta	Versão
Gerenciamento de Teste	Jest	27.2
Gerenciamento de Teste	loadView-TedotComMonitor	

9.4 Configurações do Ambiente de Teste

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

Nome da Configuração	Implementada na Configuração Física
Configuração do usuário comum	Processador de 400MHz ou mais recente compatível com SSE3
Mínima configuração suportada	Processador de 400MHz ou mais recente compatível com SSE3

10. Perfil da Equipe

10.1 Pessoas e Papéis

Esta tabela mostra as suposições referentes ao perfil da equipe do esforço de teste.

Recursos Humanos	
Papel	Responsabilidades ou Comentários Específicos
Analista de Teste	<p>Identifica e define os testes específicos a serem conduzidos.</p> <p>Estas são as responsabilidades:</p> <ul style="list-style-type: none"> identificar idéias de teste definir detalhes dos testes determinar os resultados dos testes documentar solicitações de mudança avaliar a qualidade do produto
Designer de Teste	<p>Define a abordagem técnica referente à implementação do esforço de teste.</p> <p>Estas são as responsabilidades:</p> <ul style="list-style-type: none"> definir a abordagem dos testes

<Petlife>	Version: <1.0>
<Iteração/Mestre> Plano de Teste	Date: <dd/mm/yy>
<document identifier>	

	<ul style="list-style-type: none"> • definir a arquitetura de automação de teste • verificar as técnicas de teste • definir os elementos de testabilidade • estruturar a implementação dos testes
Testador	<p>Implementa e executa os testes.</p> <p>Estas são as responsabilidades:</p> <ul style="list-style-type: none"> • implementar os testes e os conjuntos de testes • executar os conjuntos de testes • registrar os resultados • analisar as falhas dos testes e possibilitar a recuperação posterior • documentar incidentes
Implementador	<p>Implementa as classes de teste e os pacotes de teste e efetua testes de unidade nos mesmos.</p> <p>Estas são as responsabilidades:</p> <ul style="list-style-type: none"> • cria os componentes de teste necessários para suportar os requisitos de testabilidade conforme definido pelo designer

11. Riscos, Dependências, Suposições e Restrições

Risco	Estratégia de Diminuição	Contingência (O risco se concretizou)
Os pré-requisitos dos critérios de entrada não serem atendidos.	<p>O <Testador> definirá os pré-requisitos que deverão ser atendidos antes que o Teste de Carga possa ter início.</p> <p>O <Cliente> tentará atender aos pré-requisitos indicados pelo <Testador>.</p>	<ul style="list-style-type: none"> • Atender a pré-requisitos importantes • Considerar a possibilidade de Falha do Teste de Carga
Os dados de teste se mostrarem inadequados.	<p>O <Cliente> assegurará a disponibilidade de um conjunto completo de dados de teste apropriados e protegidos.</p> <p>O <Testador> indicará o que é necessário e verificará a adequação dos dados de teste.</p>	<ul style="list-style-type: none"> • Redefinir dados de teste • Revisar o Plano de Teste e modificar os componentes (ou seja, os scripts) • Considerar a possibilidade de Falha do Teste de Carga
O banco de dados necessitar de uma	O <Administrador do Sistema> tentará assegurar que o Banco de Dados seja atualizado	<ul style="list-style-type: none"> • Restaurar os dados e reiniciar • Limpar o Banco de Dados

<Petlife>		Version: <1.0>
<Iteração/Mestre> Plano de Teste		Date: <dd/mmm/yy>
<document identifier>		
atualização.	regularmente conforme exigido pelo <Testador>.	

12. Procedimentos e Processos de Gerenciamento

- Na execução, o Analista de teste se encarrega de coordenar os recursos disponíveis para executar o que foi planejado, se baseando no plano de teste. Aqui, as tarefas são delegadas para os integrantes da equipe, conforme as suas funções.

No decorrer dessa fase, o Analista de teste verifica se as entregas estão em harmonia com o plano de teste, o defende de possíveis mudanças e reafirma o nível da qualidade prevista para o trabalho que está em processo de execução.

Por fim, a equipe entra em contato com o Analista de teste para informá-lo sobre a evolução dos testes, que em seguida vai relatar tal desempenho.

No período da execução/gerenciamento do escopo, o analista de teste também pode rever os riscos. Caso tenha aparecido um obstáculo que não foi antevisto, há que se readequar o planejamento, alterando os planos de teste que estão sendo afetados.

12.1 Estratégias de Rastreabilidade

- *Cobertura de Teste em relação às Especificações — possibilita a medição da extensão do teste*
- *Motivações de Teste — possibilitam a avaliação da relevância dos testes a fim de ajudar a determinar se eles deverão ser mantidos ou não*
- *Elementos de Design de Software — possibilitam o rastreamento das mudanças de design subsequentes que exigirão que os testes sejam executados novamente ou sejam cancelados*
- *Solicitações de Mudança Resultantes — fazem com que os testes que descobriram a necessidade da mudança sejam identificados e executados novamente para verificar se a solicitação de mudança foi efetuada com êxito*