

Carefour
Plano de Teste de <Iteração/Mestre>

Versão 1.0

Histórico da Revisão

Data	Versão	Descrição	Autor
<17/04/14>	<1.0>	Release Inicial	Carefour

Índice Analítico

1. Introdução

1.1 Finalidade

1.2 Escopo

1.3 Público-alvo

1.4 Terminologia e Acrônimos do Documento

1.5 Referências

1.6 Estrutura do Documento

2. Missão de Avaliação e Motivação dos Testes

2.1 Fundamentos

2.2 Missão de Avaliação

2.3 Motivadores dos Testes

3. Itens de Teste-Alvo

....

Plano de Teste de <Iteração/Mestre>

1. Introdução

1.1 Finalidade

A finalidade do Plano de Teste de Iteração é reunir todas as 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 Projeto Carefour suporta os seguintes objetivos:

- Identificar os itens que devem ser inspecionados pelos testes.
- Identificar a motivação e as idéias subjacentes às áreas de teste a serem abrangidas.
- Descrever a abordagem de teste que será usada.
- Identificar os recursos necessários para a execução dos testes

1.2 Escopo

Este documento aborda os testes de unidade, de integração e de sistema do Projeto Carefour desenvolvido por alunos do sétimo período do curso de Sistemas de Informação da PUC Minas. Testes de aceitação não estão inclusos no escopo deste *Plano de Testes* pelo fato do projeto não passar pela fase de validação com o cliente.

1.3 Público-alvo

Este Plano de Testes tem como público alvo alunos da disciplinas de Testes e Manutenção de Software, ministradas pela Prof. Dra. Maria Augusta Vieira Nelson no 1.o semestre de 2014 na PUC Minas. O presente documento também poderá ser utilizado como base para outros alunos do curso de Sistemas de Informação ou correlatos.

1.4 Terminologia e Acrônimos do Documento

CT – Caso de Teste

DAS – Documento de Arquitetura de Software

ERSw – Especificação de Requisitos de Software

GUI – Interface Gráfica com o Usuário

1.5 Referências

[01] DAS

[02] ERSw do Projeto Carefour

1.6 Estrutura do Documento

Este documento apresenta a definição das metas e dos objetivos dos testes no escopo da iteração, os itens-alvo, a abordagem adotada, os recursos necessários e os produtos que serão liberados.

2. Missão de Avaliação e Motivação dos Testes

Neste documento serão contemplados os principais testes a serem executados no Projeto Carefour, desenvolvido na disciplina Tecnologias WEB ministrada pelo professor Bruno Alves Leite no 1.o semestre de 2014.

2.1 Fundamentos

Vide Documento de Arquitetura de Software elaborado na disciplina de Tecnologias WEB

2.2 Missão de Avaliação

Localizar problemas importantes e avaliar os riscos da qualidade perceptível.

2.3 Motivadores dos Testes

Detectar riscos de qualidade, riscos técnicos, de projeto, casos de uso, requisitos funcionais e não funcionais.

3. Itens de Teste-Alvo

Os principais alvos a serem testados contemplados nesse plano serão:

1. Unidades de código – O código utilizado para implementar o software é o principal alvo do processo de teste. Os códigos desenvolvidos em uma linguagem de programação devem estar de acordo com requisitos e projeto do sistema. De maneira indireta também são testados os scripts unitários, de integração automáticos utilizados para validar o sistema de acordo com a arquitetura. De maneira geral, qualquer implementação que interaja com o sistema;
2. Sistema completo – Nos testes de integração e de aceitação o software permitirá o teste da interconexão entre os módulos e a funcionalidade final do software de acordo com os requisitos.
3. Requisitos e arquitetura – Os documentos utilizados como referência para os testes, comumente requisitos e arquitetura, também são indiretamente alvos dos testes, pois inconsistências relacionadas a esses documentos são naturalmente identificadas durante os testes.

4. Resumo dos Testes Planejados

Esta seção apresenta os recursos recomendados para o projeto Carefour, suas principais responsabilidades e seu conjunto de conhecimentos ou de habilidades.

4.1 Resumo das Inclusões dos Testes

Neste software serão executados testes voltados para a funcionalidade do sistema. Serão executadas testes de caixa preta e testes de caixa branca.

Testes a serem executados:

- Testes relacionados ao CRUD de algum registro de algum modelo de entidade;
- Testes relacionados a consulta de algum de modelo de entidade;
- Testes relacionados a visualização de informações pro usuário.

5. Abordagem dos Testes

As estratégias de testes adotados no Carefour são descritas a seguir:

Teste de Unidade

Os testes de unidade serão realizados pelos próprios desenvolvedores e serão rodados a cada compilação do sistema de forma automática. O ideal, é que mesmo

antes mesmo de gerar o código serão construídos testes de unidade para o componente desenvolvido utilizando uma ferramenta adequada para a linguagem de programação utilizada na codificação do software.

Os casos de teste de unidade podem ser realizados usando uma caso de teste para cada método ou um caso de teste que envolva vários métodos. Pelo menos um caso de teste por componente desenvolvido deve ser gerado que procure realizar uma maior cobertura do código.

Tabela descritiva do teste

Objeto	Classes (métodos)
Estágio	Teste - Unitário
Tipo	Funcionalidade
Abordagem	Caixa Branca
Técnica	Automática

Testes de Integração

Os testes de integração consistem basicamente do conjunto dos testes unitários cumulativos e outros construídos para testar especificamente o funcionamento de um componente individualmente após integração de componentes e/ou subsistemas. Na integração podem ser empregadas ferramentas de automação dos testes de GUI.

Tabela descritiva do teste

Objeto	Componentes e Módulos
Estágio	Teste - Integração
Tipo	Funcionalidade / Ciclo do Negócio
Abordagem	Caixa Branca
Técnica	Automática

Testes de Sistema

Os testes de sistema devem focar na interação entre os componentes do sistema, para isso podem ser utilizadas ferramentas de automação de testes de interface e ferramentas de testes de unidade que explorem a interação dos componentes nos trechos de código dependentes do funcionamento de vários componentes e

subsistemas integrados. Os testes de sistema são tipicamente da camada de fachada do sistema.

Para os teste cuja automação seja complexa, testes manuais podem ser bastante úteis para testar o software ao nível de sistema.

Tabela descritiva do teste

Objeto	Fachada
Estágio	Teste - Sistema
Tipo	Funcionalidade / Ciclo do Negócio
Abordagem	Caixa Preta
Técnica	Automática / Manual

5.1 Catálogos Iniciais de Idéias de Teste e Outras Fontes de Referência

Não se aplica

5.2 Tipos e Técnicas de Teste

5.2.1 Teste de Integridade de Dados e de Banco de Dados

Objetivo da Técnica:	Garantir que os métodos e processos de acesso ao banco de dados funcionam apropriadamente e sem corrupção dos dados
Técnica:	<ul style="list-style-type: none">• Invocar cada método e processo de acesso ao banco de dados, alimentando cada um com dados ou requisições de dados válidos e inválidos.• Inspeccionar o banco de dados para garantir que os dados foram populados como pretendido, 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.
Estratégias:	Utilização de classes específicas de testes, para a realização dos casos de testes. Criar métodos específicos para testar cada processo de acesso ao banco de dados, tendo já definido os resultados esperados de cada teste.
Ferramentas Necessárias:	SGBD MySQL
Critérios de Êxito:	Todos os métodos e processos de acesso à base de dados funcionam

	como projetados e sem nenhuma corrupção de dados.
Considerações Especiais:	<ul style="list-style-type: none"> • O teste pode necessitar de um ambiente de desenvolvimento ou drivers de SGBD para inserir ou modificar os dados diretamente nas base de dados • Processos devem ser invocados manualmente • Bases de dados pequenas ou minimizadas (número de registros limitados) devem ser usados para aumentar a visibilidade de eventos não-aceitáveis.

5.2.2 Teste de Funcionamento

Objetivo da Técnica:	Experimentar 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>Executar cada caso de uso, fluxo de caso de uso, usando dados válidos e inválidos, para verificar o seguinte:</p> <ul style="list-style-type: none"> ▪ Os resultados esperados ocorrem quando dados válidos são usados ▪ As mensagens de erro ou aviso apropriadas são exibidas quando dados inválidos são usados. <p>Cada regra de negócio é aplicada apropriadamente</p>
Estratégias:	Verificar valores dentro e fora das classes de equivalência de cada teste, além de verificar valores de fronteira quando aplicáveis no teste.
Ferramentas Necessárias:	1 PC com acesso à Internet
CrITÉRIOS de Êxito:	<ul style="list-style-type: none"> • Todos os testes planejados foram executados. • Todos os defeitos identificados foram tratados.
Considerações Especiais:	Não se aplica

5.2.3 Teste de Ciclos de Negócios

Não se aplica

5.2.4 Teste de Interface do Usuário

O Teste de Interface do Usuário (UI) verifica a interação do usuário com o software. O teste de UI tem por fim 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>Avaliar no critério da usabilidade e da conformidade, a interface oferecida pelo sistema. Esta avaliação tem por objetivo verificar se:</p> <p>A interface do usuário reflete as necessidades do público alvo do sistema, bem como os requisitos e funções de negócios definidos. Também será contemplado a navegação através das janelas do sistema e também campo a campo.</p>
Técnica:	<p>Utilizando de uma simulação dos casos de uso, deverão ser executados as funcionalidades esperadas e avaliar o comportamento da interface de acordo com as funções executadas. Cada janela deverá apresentar um comportamento aceitável baseando-se no Template de telas do projeto</p>
Estratégias:	<ul style="list-style-type: none"> • Utilização de grupo de usuários testes • Definir cenário que aborde as diferentes interfaces do sistema • Medir através de testes de usabilidade, o tempo necessário para executar funções do sistema levando em consideração a interface necessária.
Ferramentas Necessárias:	Não se aplica
Crítérios de Êxito:	A interface ser aprovada em 90% levando em consideração as necessidades e avaliação dos usuários.
Considerações Especiais:	Não se aplica

5.2.5 Teste de Segurança e de Controle de Acesso

Não se aplica

5.3 Casos de Teste

CT 01 – Teste de Funcionamento: Adicionar itens ao carrinho de compras

Tipo do teste: Caixa preta

Descrição: Verificar se o sistema permite a inclusão de itens no carrinho de compras de forma correta.

Casos de Uso: Selecionar categoria e Adicionar produto ao carrinho

Entradas:

Quantidade: número inteiro

Critério de saída: Usuário deve selecionar a categoria de produto que deseja, marcar o *checkbox* referente ao produto desejado da lista de produtos disponíveis, informar a quantidade deste mesmo produto e clicar em *comprar*.

Saídas esperadas:

Entradas válidas: “Item adicionado ao carrinho”

Entradas inválidas: Caso o *checkbox* ou a quantidade não for informada ou for inválida o sistema deve exibir a mensagem: “Verifique se o produto foi selecionado e a quantidade foi informada”

CT 02 – Teste de Funcionamento: Remover itens do carrinho de compra

Tipo do teste: Caixa preta

Descrição: Verificar se o sistema permite remover itens do carrinho de compras de forma correta

Caso de Uso: Ver carrinho de compras

Entradas: Clique em *remover*.

Critério de saída: Usuário deve escolher o item no carrinho e clicar em *remover* ao lado do campo quantidade referente ao produto que deseja remover

Saídas esperadas: “Item removido com sucesso”

CT 03 – Teste de Funcionamento: Atualizar quantidade de itens no carrinho de compras

Tipo do teste: Caixa preta

Descrição: Verifica se quando o usuário compra um item que já está no carrinho de compras o sistema atualiza a quantidade desse item.

Caso de Uso: Ver carrinho de compras

Entradas:

Quantidade: número inteiro

Critério de saída: Usuário deve modificar o valor da quantidade de um produto que esteja no carrinho de compras e clicar em *atualizar*.

Saídas esperadas:

Entradas válidas: “Valor do item atualizado!”

Entradas inválidas: “Verifique a quantidade informada!”

CT 04 – Teste de Funcionamento: Visualizar resumo de itens no carrinho de compras

Tipo do teste: Caixa preta

Descrição: Verifica se o sistema permite a visualização de resumo de itens no carrinho de compras

Caso de Uso: Ver carrinho de compras

Entradas: Clique no botão *Ver Carrinho*.

Critério de saída: Usuário deve clicar no botão *Ver Carrinho*.

Saídas esperadas: Sistema recupera e exibe na tela a descrição, o preço e a quantidade de todos os itens que o usuário adicionou no carrinho de compras

CT 05 – Teste de Funcionamento: Esvaziar carrinho de compras

Tipo do teste: Caixa Preta

Descrição: Verifica se o sistema permite ao usuário limpar todos os itens do carrinho de compras

Caso de Uso: Ver carrinho de compras

Entradas: Clique no botão *Limpar Carrinho*.

Critério de saída: Usuário deve clicar no botão *Limpar Carrinho*.

Saídas esperadas: “Seu carrinho de compras está vazio!”

CT 06 – Teste de Funcionamento: Continuar compras

Tipo do teste: Caixa preta

Descrição: Verifica se o sistema permite ao usuário continuar as compras sem que se perca os itens já adicionados ao carrinho.

Caso de Uso: Ver carrinho de compras

Entradas: Clique no botão *Continuar Compras*.

Critério de saída: Usuário deve clicar no botão *Continuar Compras*.

Saídas esperadas: Sistema retorna ao usuário a página de categoria de produtos sem remover itens já adicionados ao carrinho de compras.

CT 07 – Teste de Funcionamento: Fechar compra

Tipo do teste: Caixa preta

Descrição: Verifica se o sistema permite ao usuário fechar a compra de forma correta.

Caso de Uso: Fechar compra

Entradas: Clique no botão *Fechar Compra*.

Critério de saída: Usuário deve clicar no botão *Fechar Compra*.

Saídas esperadas: Sistema retorna ao usuário a página com o resumo da compra, exibe formas de pagamento e o formulário para que o cliente entre com seus dados pessoais. Caso o usuário não possua itens no carrinho de compras o sistema exibe a mensagem: “Seu carrinho de compras esta vazio!”.

CT 08 – Teste de Funcionamento: Verificar Registro dos dados do cliente

Tipo do teste: Caixa preta

Descrição: Verificar se o sistema valida os dados do formulário de cadastro do cliente de forma que todos os dados sejam fornecidos de forma correta.

Caso de Uso: Registrar dados do cliente

Entradas:

Nome: somente letras até 60 caracteres

e-mail: e-mail válido

Senha: de 6 a 10 caracteres alfanuméricos

Data de Nascimento: data válida

Sexo: “Feminino” ou “Masculino”

Endereço: Até 100 caracteres alfanuméricos

Telefone: Somente 10 números

Forma de pagamento: Boleto ou cartão

Critério de saída: Todos os campos do formulário devem ser preenchidos e o botão *Enviar* deve ser acionado.

Saídas esperadas:

Entradas válidas: “Cadastro efetuado com sucesso”

Entradas inválidas: Caso um dos campos não tenham sido preenchidos corretamente ou exista campos não preenchidos o sistema deverá exibir a mensagem na frente do campo: “Valor inválido”.

CT 09 – Teste de Funcionamento: Confirmação da compra

Tipo do teste: Caixa preta

Descrição: Verifica se o sistema confirma a compra e exibe o resumo da compra para o cliente de forma correta.

Caso de Uso: Fechar compra e Registrar dados do cliente

Entradas: Clique no botão *Enviar*.

Critério de saída: O usuário deve possuir itens no carrinho de compras, ter clicado em *Fechar Compra*, ter preenchido o formulário de identificação, escolhido a forma de pagamento e acionado o botão *Enviar*.

Saídas esperadas: O sistema exibe a mensagem ao usuário “Pedido realizado com sucesso” e retorna ao usuário a página com o endereço de entrega, número e resumo do pedido e a previsão de entrega.

CT 10 – Teste de Funcionamento: Idioma do sistema

Tipo do teste: Caixa preta

Descrição: Verifica a funcionalidade de exibição dos idiomas português e espanhol do sistema

Caso de Uso: Nenhum

Entradas: Clique nas opções de idioma *Português* ou *Espanhol*.

Critério de saída: Usuário deve clicar nas opções de idioma *Português* ou *Espanhol*.

Saídas esperadas: Todos os textos e mensagens do sistema devem ser exibidos na linguagem escolhida pelo usuário (português ou espanhol).

CT 11– Teste de Integridade: public ActionResult Produtos(string produto)

Tipo do teste: Caixa branca

Descrição: verifica se a função apontada retorna o objeto esperado pelo sistema para alternar entre as categorias de produtos.

Caso de Uso: Nenhum

Entradas: String chamada de produto contendo um dos valores disponíveis para categoria: Carnes, Laticínios, Padaria ou Hortifrutigranjeiros

Critério de saída: possuir pelo menos um produto representando a Model Produto, existe um Produto associado a uma categoria representado pelo Model Categoria.

Saídas esperadas: O metodo deve retornar um objeto através do comando *return View(_produto.listaProdutos);* . Este objeto (_Produto.ListaProduto) deve conter a lista de produtos que foi selecionada através dos operadores Swicth(produto). O parametro passado para o comando seleciona a lista de produtos que estão associados a minha categoria e os adiciona na minha resposta da função.

CT 12– Teste de Integridade: public ActionResult Carrinho()

Tipo do teste: Caixa branca

Descrição: Função responsável por exibir o estado atual do Objeto Pedido. Nele contem os itens de produtos selecionados atualmente para a compra. E verificar valores representados pela minha classe Pedido.

Caso de Uso: Nenhum

Entradas: Não possui entradas esperadas.

Critério de saída: possuir pelo menos um produto representando a Model Produto, Existe uma instancia de Pedido.

Saídas esperadas: O metodo deve retornar um objeto através do comando *return View(_produto.listaProdutos);* . Este objeto (*_Produto.ListaProduto*) deve conter a lista de produtos que foi selecionada adicionada a minha instancia de Pedido, para ser exibida na tela. Da mesma forma, o método deve apresentar através de uma variável chamada *ViewBag.SubTotal*, a soma dos valores de cada produto da minha lista de produtos *_produto.listaProdutos*.

6. Critérios de Entrada e de Saída

6.1 Plano de Teste

6.1.1 Critérios de Entrada de Plano de Teste

Após o primeiro marco do projeto Carefour este Plano de Testes poderá começar a 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

Caso os requisitos do projeto venham a ser alterados o Plano de Testes deverá ser

interrompido de acordo com as mudanças propostas.

7. Produtos Liberados

7.1 Sumários de Avaliação de Testes

Será definido posteriormente.

7.2 Geração de Relatórios sobre Cobertura de Teste

Será definido posteriormente.

7.3 Registros de Incidentes e Solicitações de Mudança

Não se aplica

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

Não se aplica

7.5 Produtos de Trabalho Adicionais

7.5.1 Resultados Detalhados dos Testes

Será definido posteriormente.

7.5.2 Scripts de Teste Funcionais Automatizados Adicionais

Será definido posteriormente.

7.5.3 Guia de Teste

Será definido posteriormente.

7.5.4 Matrizes de Rastreabilidade

Será definido posteriormente.

9. Necessidades Ambientais

9.1 Hardware Básico do Sistema

A tabela a seguir apresentam os recursos do sistema necessários ao esforço de teste descrito neste *Plano de Teste*.

Recursos do Sistema
Servidor de Banco de Dados: MySQL Server
Terminais Clientes: 3 PCs (conectados à Internet)
Repositório de Testes: 3 PCs de Desenvolvimento de Teste

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	Tipo e Outras Observações
Windows 7	Sistema Operacional
Internet Explorer	Navegador da Internet
Google Chrome	Navegador da Internet
Mozilla Firefox	Navegador da Internet
Microsoft Visual Studio	Ferramenta de desenvolvimento

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
Framework de Teste	JUnit

9.4 Configurações do Ambiente de Teste

A ser definido posteriormente.