Software Soluções e Desenvolvimento

Projeto de testes para Cálculo de IMC

Histórico da Revisão

Data	Versão	Descrição	Autor	Área/Setor
<24/11/2022>	<1.0>	<Desenvolvimento do plano de testes >	<Everton MDS>	<Desenvolvimento>

Identificação do Plano de Testes: Projeto de testes para Cálculo de IMC

1. Introdução

1.1. Objeto

A NutriVitta é uma empresa atuante no ramo da nutrição e presta atendimento clínico para seus clientes e contratou a Testful para a elaboração de um sistema de cálculo de IMC (Índice de Massa Corporal). O Índice de Massa Corporal (IMC) é reconhecido como padrão internacional para avaliar o grau de obesidade. O IMC é calculado dividindo o peso (em kg) pela altura ao quadrado (em metros).

1.2. Objetivo

O software é composto pelas seguintes regras pré-estabelecidas:

- Realização do cálculo do IMC;
- Classificação do IMC de acordo com a tabela da “Abeso”;
- Fazer o planejamento e desenho do teste, considerando as especificações técnicas do projeto;
- Testar o sistema, aplicando correções de acordo com os resultados obtidos no teste.

2. Escopo

Este plano de teste aplicado aos testes de integração e de sistema que serão conduzidos nos Releases do software Índice de Massa Corporal (IMC).
O teste de unidade fornecerá por meio do “Black Box”, uma cobertura extensiva de todas as interfaces do módulo.
O teste de caixa preta envolve a verificação de uma determinada funcionalidade ou de um componente da aplicação, sem que exista a preocupação com a estrutura interna do programa, ou seja, desconsidera-se a estrutura do código-fonte, os detalhes de implementação ou os cenários de execução.

3. Abordagem

Serão empregados testes dinâmicos também conhecidos como Caixa Preta para a avaliação do software de cálculo do IMC.
O teste é particularmente útil para revelar problemas, tais como:

- funções incorretas ou omitidas;
- erros de interface;
- erros de comportamento ou desempenho;
- erros de iniciação e término.

4. Missão de Avaliação e Motivação dos Testes

Este Plano de Teste será aplicado aos testes de integração e de sistema que serão conduzidos nos Releases do Software de Cálculo do Índice de Massa Corporal (IMC).

4.1 Fundamentos

O plano de testes, buscará delinear os passos para a verificação das funcionalidades e também dos principais itens que poderão dificultar o desempenho do Software de Cálculo do Índice de Massa Corporal (IMC), bem como propor soluções tornando-o assim mais otimizado.

4.2 Missão de Avaliação

O teste tem por missão avaliar e identificar as principais características do Software de Cálculo do IMC:

- Localizar o maior número de erros possível durante a execução do teste;
- Localizar problemas importantes que se manifestem durante a avliação;
- Avaliar os riscos da qualidade perceptível;
- Informar sobre os riscos perceptíveis do projeto;
- Verificar uma especificação (requisitos, design ou alegações);
- Informar sobre a qualidade do produto.

4.3 Motivadores dos Testes

A Motivação para a elaboração foi motivada mediante a identificação dos riscos técnicos e de projetos que venha a interferir na qualidade do programa durante a execução dos cálculos do IMC.

5. Itens de Teste-Alvo

A listagem abaixo identifica os principais itens de software, hardware e elementos de suporte do produto que serão identificados como objetivos dos testes.
O hardware de processamento básico para a execução do teste exigirá como requisitos mínimos para a execução do software, processador Intel(R) Core (TM) i5-3470 CPU @ 3.20GHz, 4GB-RAM para Desktop.

Sistemas operacionais mínimos compatíveis para a execução do teste desktop Windows 32x ou 64x, para os dispositivos móveis mínimo Android 7.

O teste buscará:

- Localizar o maior número de erros possível durante a execução do teste;
- Localizar os problemas importantes que se manifestem durante a avaliação;
- Avaliar os riscos da qualidade perceptível;
- Informar sobre os riscos perceptíveis do projeto;
- Verificar uma especificação (requisitos, design ou alegações);
- Informar sobre a qualidade do produto.

6. Resumo dos Testes Planejados

6.1 Resumo das Inclusões dos Testes

Teste de Integridade de Dados e de Banco de Dados

Teste de Funcionamento

Teste de Ciclos de Negócios

Teste de Interface do Usuário

Teste de Carga

Teste de Stress

6.2 Resumo dos Outros Candidatos a Possível Inclusão

Não foram identificadas áreas extras ao software de cálculo do IMC que necessitem de uma investigação mais profunda.

6.3 Resumo das Exclusões dos Testes

Os testes de caixa Branca não serão contemplados neste plano de testes, visto que não serão avaliados o código fonte do software de cálculo do IMC

7. Abordagem dos Testes

O teste dinâmico vê o software como uma (Caixa-Preta) nome popular “Blackbox” e trabalha, principalmente, com as informações que são inseridas nas rotinas de entrada e saída de dados. Além disso, são verificados itens como:

- O tempo de resposta;
- A performance da aplicação;
- A capacidade do software se adaptar a diferentes ambientes;
- O comportamento funcional.

Muitas empresas adotam a análise dinâmica por ela permitir que problemas mais sutis sejam identificados, dessa forma, o teste dinâmico consegue dar mais segurança e confiabilidade ao produto final.

7.1 Catálogos Iniciais de Ideias de Teste e Outras Fontes de Referência

Dentre as principais ideias que serão levadas em conta para a verificação do software de cálculo do IMC podem ser citadas as seguintes:

- Os principais riscos a qualidade o software estão contemplados no teste;
- Os ataques com maior potencialidade de danos serão verificados;
- Algum modelo de falha relevante está incluso no teste.

7.2 Tipos e Técnicas de Teste

7.2.1 Teste de Integridade de Dados e de Banco de Dados

Os bancos de dados e os processos de banco de dados devem ser testados como sistemas separados.

Objetivo da Técnica:	O teste buscará analisar os processos e métodos de acesso a banco de dados independentes da UI. Visando observar e registrar comportamentos alvo incorretos ou a existência de dados corrompidos do sistema de cálculo de IMC (Índice de Massa Corporal)**.**

Técnica:	Integridade da entidade - examina se cada linha de uma tabela consiste em uma chave primária não nula, onde cada uma deve ser específica. O teste pode ser obtido definindo valores duplicados ou nulos nos dados de teste. Integridade do domínio – verifica-se cada conjunto de valores de dados. A coluna é exibida com um intervalo especifico permitido. O teste pode ser realizado usando valores nulos, padrão e inválidos. Integridade referencial - verifica o relacionamento entre uma chave estrangeira e a chave primária de várias tabelas. Este teste é realizado eliminando-se a linha pai ou filho em uma tabela.
Estratégias:	• Verificar acesso ao Banco de Dados do Catálogo de Cursos. • Verificar acessos de leitura simultâneos ao registro. • Verificar interrupção durante atualizações do Catálogo de Cursos. • Verificar a recuperação correta de atualizações dos dados do banco de dados.
Ferramentas Necessárias:	• Ferramenta de Automação de Scripts de Teste (Selenium); • Ferramenta de backup e de recuperação (Cobian Backup); • Ferramentas e utilitários SQL de banco de dados (SQL Server Management Studio (SSMS), (SQL Server), Visual Studio Code.
Critérios de Êxito:	As técnicas que serão empregadas conforme análise suportam todos os testes referentes aos principais processos e métodos de acesso a banco de dados.
Considerações Especiais:	• Para a execução do teste os processos poderão ser disparados manualmente ou de forma automatizada. • 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.

7.2.2 Teste de Funcionamento

Esse teste tem por objetivo verificar a adequada aceitação, processamento e 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:	Os testes funcionais, também conhecidos como testes de caixa-preta, são uma validação de software na qual determinada funcionalidade é verificada, sem levar em conta a estrutura do código-fonte, os detalhes de implementação ou os cenários de execução.

Técnica:	Casos de uso: Os casos de uso descrevem as interações entre os atores, que podem ser usuários ou outros sistemas, os quais possuem pré-condições que devem ser atendidas para atingir o funcionamento correto do software. Técnicas baseados na experiência: As técnicas baseadas na experiência são aquelas em que os testes são derivados da habilidade e intuição do testador, bem como de sua experiência com aplicativos e tecnologias semelhantes. Este método é amplamente utilizado na previsão de erros.
Estratégias:	O teste funcional permite identificar: • Funções incorretas ou ausentes; • Erros de interface; • Erros em acessos a bancos de dados externos; • Erros de desempenho; • Erros de inicialização e término.
Critérios de Êxito:	As técnicas escolhidas obtêm êxito para suportar o teste de todos os principais cenários de caso de uso e também para a identificação dos principais erros.
Considerações Especiais:	Sem definição de impedimentos.

7.2.3 Teste de Ciclos de Negócios

O Teste de Ciclos de Negócios deverá emular as atividades executadas no Projeto de testes para Cálculo de IMC ao longo do tempo.

Durante um período identificado de um trimestre, deverão ser executadas as transações e as atividades que ocorreriam durante esse período englobando todos os ciclos diários, semanais e mensais, assim como os eventos que mudam com as datas como, por exemplo, lembretes.

Objetivo da Técnica:	Garantir que o sistema funcione apropriadamente durante um ciclo de atividades relativas ao negócio e que ao final desse ciclo todos os resultados esperados sejam obtidos.
Técnica:	O teste incluirá o uso de casos válidos e inválidos para verificar se: ● Os resultados esperados irão ocorrer 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; ● Todas as funções sensíveis a datas ou tempo serão executadas usando datas ou períodos de tempo válidos. ● Os testes de funções do aplicativo serão modificados / melhorados para aumentar o número de vezes que cada função é executada, a fim de simular vários

	usuários diferentes ao longo de um período de tempo especificado.
Estratégias:	O teste incluirá o uso de casos válidos e inválidos para verificar se: ● Os resultados esperados irão ocorrer 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; ● Todas as funções sensíveis a datas ou tempo serão executadas usando datas ou períodos de tempo válidos. ● Os testes de funções do aplicativo serão modificados / melhorados para aumentar o número de vezes que cada função é executada, a fim de simular vários usuários diferentes ao longo de um período de tempo especificado.
Ferramentas Necessárias:	• Ferramenta de Automação de Scripts de Teste (Selenium) • Ferramenta de backup e de recuperação (Cobian Backup) • Ferramentas e utilitários SQL de banco de dados (SQL Server Management Studio (SSMS), (SQL Server), Visual Studio Code.
Critérios de Êxito:	Serão verificados os seguintes critérios de êxito: • Todos os testes planejados foram executados. • Todos os defeitos identificados foram tratados.
Considerações Especiais:	Os testes terão suas demandas ocorridas sem considerações especiais.

7.2.4 Teste de Interface do Usuário

O Teste de Interface do Usuário (UI) tem por fim assegurar que a UI forneça ao usuário o acesso e a navegação adequados através das funções do objetivo do teste.

Objetivo da Técnica:	Os seguintes itens serão observados durante a execução do teste: • A navegação pelo aplicativo reflete os requisitos e funções de negócios, incluindo a navegação janela a janela, campo a campo e o uso de métodos de acesso (teclas de tabulação, movimentos do mouse e teclas aceleradoras) • Objetos e características da janela, tais como menus, tamanho, posição, estado e foco estão em conformidade com os padrões definidos.

Técnica:	Os testes serão criados/modificados para cada janela a fim de verificar a navegação adequada e os estados de objeto para cada janela e objeto do aplicativo.
Estratégias:	A estratégia combina elementos do método através do qual a observação pode ser feita e também das características dos resultados específicos que indicam um provável êxito ou falha do teste. Serão empregados testes automatizados para que façam uma avaliação inicial do êxito ou falha do teste.
Ferramentas Necessárias:	• Ferramenta de Automação de Scripts de Teste (Selenium)
Critérios de Êxito:	Verificação do êxito de cada janela conseguir permanecer consistente dentro do padrão projetado.
Considerações Especiais:	Os testes terão suas demandas ocorridas sem considerações especiais.

7.2.5 Determinação do Perfil de Desempenho

A meta do teste de Desempenho é verificar e validar se os requisitos de desempenho foram alcançados.

Objetivo da Técnica:	Validar o Tempo de Resposta do Sistema para funções de negócios ou transações designadas sob as duas condições a seguir. • Volume normal previsto • Volume de pior caso previsto.
Técnica:	A técnica visa modificar arquivos de dados (a fim de aumentar o número de transações) / modificar scripts a fim de aumentar o número de iterações ocorrido em cada transação. Os scripts devem ser executados em uma máquina avaliando o desempenho de um único usuário, uma única transação e posteriormente repetidor com vários clientes.
Estratégias:	A estratégia buscará ser um processo auto verificado, permitindo que os testes automatizados façam uma avaliação inicial do êxito ou falha do teste.
Ferramentas Necessárias:	• Ferramenta de Automação de Scripts de Teste (Selenium) • Ferramenta de backup e de recuperação (Cobian Backup) • Ferramentas e utilitários SQL de banco de dados (SQL Server Management Studio (SSMS), (SQL Server), Visual Studio Code.

Critérios de Êxito:	Para êxito dos testes estes devem apresentar as seguintes características: • Transação Única / usuário único: Conclusão com êxito dos scripts de teste sem nenhum defeito e na alocação de tempo esperada / requerida (por transação). • Várias Transações / vários usuários: Conclusão com êxito dos scripts de teste sem nenhum defeito e dentro de alocação de tempo aceitável.
Considerações Especiais:	• O teste de desempenho 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 utilizados para teste de Desempenho deverão ter tamanhos reais ou ser igualmente escalados.

7.2.6 Teste de Carga

O teste de carga tem por fim determinar e assegurar que o sistema funcione adequadamente com uma carga de trabalho superior à carga máxima esperada.

Objetivo da Técnica:	Teste de Carga: • Usado para simular condições normais. • Quantidade de dados utilizada é a mesma estimada para o dia a dia de trabalho. • Objetivo não é quebrar o sistema.
Técnica:	• Empregar os testes desenvolvidos para o Teste do Ciclo de Negócio. • As cargas de trabalho devem incluir cargas de pico — por exemplo, diárias, semanais e mensais. • As cargas de trabalho devem representar cargas médias assim como cargas de pico. • As cargas de trabalho devem representar picos instantâneos e picos sustentados.
Estratégias:	A principal estratégia utilizará os testes automatizados afim de que façam uma avaliação inicial do êxito ou falha do teste.

Ferramentas Necessárias:	• A Ferramenta de controle e de programação de carga de transações empregada será “LoadView “plataforma de teste de carga baseada em nuvem sob demanda que determina como um site, aplicativo web ou móvel, ou API responde a vários tráfegos. • A ferramenta CPU-Z será empregada para o monitoramento de disco rígido, CPU, memória etc. • A Ferramenta de Automação de Scripts de Teste (Selenium)
Critérios de Êxito:	As Várias Transações / vários usuários: o êxito dos testes ocorrerá se nenhum defeito emergir dentro da alocação de tempo aceitável.
Considerações Especiais:	Os testes de carga deverão ser executados em uma máquina dedicada e em um período de tempo dedicado. Isso permitirá o controle total e a medição exata. Os bancos de dados utilizados para teste de carga deverão ter tamanhos reais ou ser igualmente escalados.

7.2.7 Teste de Stress

O teste de estresse foi projetado para localizar erros devidos à falta de recursos ou competição por recursos. Pouca memória ou espaço em disco poderão revelar defeitos no software que não são aparentes sob condições normais.

Objetivo da Técnica:	Teste de Estresse: • Testa o comportamento do sistema simulando condições extremas. • O teste só encerra após ter gerado uma falha no sistema. • Objetivo é quebrar o sistema, gerar falhas.

Técnica:	• Para testar recursos limitados, os testes devem ser executados em uma única máquina e a RAM e DASD no servidor devem ser reduzidos (ou limitados).
	• Para os testes de estresse restantes, deverão ser utilizados vários clientes, executando-se os mesmos testes ou testes complementares a fim de produzir o conjunto / volume de transações no pior caso.
	• Teste de Estresse Exploratório: Este é um tipo de teste de estresse que pode ser usado para testar o sistema com parâmetros incomuns ou condições difíceis de acontecer em cenários reais. É usado para encontrar erros em cenários incomuns como:
	  • Grande número de usuários logados ao mesmo tempo;
	  • Se um antivírus começou a rodar em todas as máquinas simultaneamente;

	• Se um banco de dados ficou offline quando acessado por um website; • Quando um grande volume de dados é inserido no banco de dados simultaneamente.
Estratégias:	As estratégias serão estruturadas de forma auto verificadas, permitindo que os testes automatizados façam uma avaliação inicial do êxito ou falha do teste.
Ferramentas Necessárias:	• A Ferramenta de controle e de programação de carga de transações empregada será “LoadView “plataforma de teste de carga baseada em nuvem sob demanda que determina como um site, aplicativo web ou móvel, ou API responde a vários tráfegos, sem o incômodo de gerenciar, investir ou manter qualquer infraestrutura de teste adicional; • A ferramenta CPU-Z será empregada para o monitoramento de disco rígido, CPU, memória etc; • A Ferramenta de Automação de Scripts de Teste (Selenium).
Critérios de Êxito:	A obtenção de êxito ocorrerá quando todos os testes planejados forem executados e quando os limites do sistema especificados forem alcançados / excedidos sem a falha do software.
Considerações Especiais:	A geração do estresse na rede poderá exigir ferramentas de rede para que ocorra o carregamento a rede com mensagens / pacotes.