



Plano de Testes do Aplicativo

Liv (Logística inteligente de vendas)

Plano de Testes apresentado ao projeto da Squad II –
Rifeiros, pela equipe dos Cursos de Tecnologias.
Centro Universitário Una Bom Despacho.

Orientadores: Prof^ª. Dayana Keitty Carmo Gonçalves
Prof. Edmilson Ferreira da Silva
Prof. Me. Hewerton Luís P. Santiago.

Equipe: Izabella Fernanda Silva Veloso
João Pedro Oliveira Santos
Luiz Guilherme Mendes Pedrosa
Luís Gustavo Braga Bernardes
Marcelo Rodrigues Araújo Filho
Miguel Elias Dias
Paulo Roberto Bernardes Pereira
Sarah Ferreira Lemes de Souza



**Bom Despacho – MG.
novembro / 2021.**

Histórico de Revisões

Data	Versão	Descrição	Autor
22/11/2021	1.0	Release Inicial	Izabella Fernanda

Índice

1. INTRODUÇÃO.....	3
1.1 OBJETIVOS.....	3
1.2 O RECYCLE+.....	3
1.3 ESCOPO.....	3
2. REQUISITOS A TESTAR.....	3
2.1 TESTE DO BANCO DE DADOS.....	3
2.2 TESTE FUNCIONAL.....	4
2.3 TESTE DA INTERFACE DO USUÁRIO.....	4
2.4 PERFIL DA PERFORMANCE.....	4
2.5 TESTE DE CARGA.....	4
2.6 TESTE DE SEGURANÇA E DE CONTROLE DE ACESSO.....	4
2.7 TESTE DE INSTALAÇÃO.....	5
3. ESTRATÉGIA DE TESTE.....	5
3.1 TIPOS DE TESTE.....	5
3.1.1 <i>Teste de Integridade de Dados e do Banco de Dados</i>	5
3.1.2 <i>Teste de Função</i>	6
3.1.3 <i>Teste da Interface do Usuário</i>	6
3.1.4 <i>Teste de Performance</i>	7
3.1.5 <i>Teste de Carga</i>	8
3.1.6 <i>Teste de Segurança e Controle de Acesso</i>	8
3.1.7 <i>Teste de Instalação</i>	9
3.2 FERRAMENTAS.....	9
4. RECURSOS.....	10
4.1 TRABALHADORES.....	10
4.2 SISTEMA.....	11
5. CRONOGRAMA.....	12
6. CONSIDERAÇÕES FINAIS.....	12

1. INTRODUÇÃO

1.1 Objetivos

Esse documento do Plano de Testes do Liv (Logística inteligente de vendas) compõe-se dos seguintes objetivos:

- Identificar informações de projeto existentes e os componentes de software que devem ser testados.
- Listar os Requisitos a Testar recomendados (alto nível).
- Recomendar e descrever as estratégias de teste a serem empregadas.
- Identificar os recursos necessários e prover uma estimativa dos esforços de teste.
- Listar os elementos resultantes do projeto de testes.

1.2 Liv (Logística inteligente de vendas)

O sistema Liv (Logística inteligente de vendas) é um aplicativo para smartphones que vem para orientar aos donos de rifas, colocadores, fornecedores e outros no banco de dados de suas vendas e produtos no estoque. Possibilitando assim um melhor planejamento para tanto os donos de rifas, colocadores, fornecedores e outros.

1.3 Escopo

O LIV(Logística inteligente de vendas) passará pelos testes unitário, de integração e de sistema. 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 tratarão as questões de performance. Serão realizados também, os testes de configuração.

Para a execução dos testes serão utilizadas máquinas as mais idênticas possível, em termos de hardware, àquelas que serão implantadas no ambiente real, a fim de garantir a previsibilidade de performance e compatibilidade.

Os testes mais críticos serão os testes de banco de dados, que compõe a maior parte do sistema:

1. Testaremos o tempo de resposta para operações que envolvam os dados de todos os usuários.
2. E também a correta atualização do banco de dados para as funções de cadastrar, atualizar e remover.

2 REQUISITOS A TESTAR

A lista abaixo identifica aqueles itens – *use cases*, requisitos funcionais e não funcionais – que foram identificados como alvos de teste. Essa lista representa o que será testado.

2.1 Teste do Banco de Dados

- Verifique que as informações dos usuários cadastrados possam ser editadas ou removidas através do painel administrativo.

- Verifique que as informações do historico de vendas podem ser inseridas, atualizados ou consultados.
- Verifique que as informações e dados de usuários podem ser inseridos e modificados adequadamente pelos devidos administradores no momento de implantação do sistema.
- Verifique que os dados de login e autenticação estão sendo aceitos e validados de forma correta.

2.2 Teste Funcional

- Verifique que as informações úteis obtidas pelo subsistema responsável são automaticamente e periodicamente atualizadas.
- Verifique que os usuários da cidade podem acessar sua conta através de login e senha.
- Verifique que os dados do usuários no app estão corretos.

2.3 Teste da Interface do Usuário

- Navegue através de todos os use cases, verificando que cada tela de interface gráfica pode ser rapidamente entendida e facilmente utilizada.
- Verifique que toda ajuda online funciona.

2.4 Perfil da Performance

- Verifique o tempo de resposta da rede interna, do servidor em relação aos terminais.
- Verifique o tempo de consulta/atualização do subsistema de informações úteis.
- Verifique que o tempo de resposta para operações que envolvam dados dos produtos em estoque não ultrapassem 5 segundos.

2.5 Teste de Carga

- Verificar a resposta do sistema com 10 usuários.
- Verificar a resposta do sistema com 50 usuários.
- Verificar a resposta do sistema com 100 usuários.
- Verificar a resposta do sistema com 200 usuários.
- Verificar a resposta do sistema com 500 usuários.

2.6 Teste de Segurança e de Controle de Acesso

- Verificar que usuários não cadastrados não podem acessar informações restritas aos cadastrados.
- Verificar que além do administrador, ninguém mais pode inserir, atualizar ou remover dados do sistema.

2.7 Teste de Instalação

- Verifique que a instalação do sistema ocorre normalmente em celulares com Android acima de 5.1.
- Verifique que os smartphones com os requisitos mínimos a aplicação funcione normalmente.
- Verifique que o sistema é capaz de obter e atualizar as informações úteis a que se propõe a disponibilizar.
- Verifique que o espaço disponível em disco para informações deve ser capaz de armazenar todos os dados/atualizações que forem cadastrados.

3 ESTRATÉGIA DE TESTE

3.1 Tipos de Teste

Nota: As transações abaixo se referem às “transações lógicas de negócio”. Essas transações são definidas como funções específicas que um usuário final do sistema é suposto de executar ao usar a aplicação, tais como adicionar ou modificar uma dada informação.

3.1.1 Teste de Integridade de Dados e do Banco de Dados

Objetivo do Teste:	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.
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:	<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 bases 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.
--------------------------	--

3.1.2 Teste de Função

Objetivo do Teste:	Garantir a funcionalidade apropriada do alvo do teste, incluindo navegação, entrada de dados, processamento e recuperação.
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. ▪ Cada regra de negócio é aplicada apropriadamente
Critério de Finalização:	<ul style="list-style-type: none"> ▪ Todos os testes planejados foram executados. ▪ Todos os defeitos identificados foram tratados.

3.1.3 Teste da Interface do Usuário

Objetivo do Teste:	<p>Verificar o seguinte:</p> <ul style="list-style-type: none"> ▪ A navegação através dos alvos de teste reflete as funções e os requisitos do negócio apropriadamente, incluindo janela-a-janela, campo-a-campo, e o uso de métodos de acesso (digitação, rolagem e toques na tela) ▪ Objetos e características da janela, tais como menus, tamanho, posição, estado e foco conformam-se aos padrões.
Técnica:	Criar ou modificar os testes para cada janela para verificar a navegação e os estados de objeto apropriados para cada janela e objetos da aplicação.
Critério de Finalização:	É verificado que cada janela permanece consistente com a versão de comparação ou dentro de padrões aceitáveis.
Considerações Especiais:	Nem todas as propriedades para objetos personalizados e terceirizados podem ser acessadas.

3.1.4 Teste de Performance

Objetivo do Teste:	<p>Verificar que os comportamentos de performance para as transações designadas ou funções de negócio sob as seguintes condições:</p> <ul style="list-style-type: none"> ▪ Carga de trabalho normal antecipada ▪ Carga de trabalho no pior caso antecipada
Técnica:	<ul style="list-style-type: none"> ▪ Usar Procedimentos de Teste desenvolvidos para Teste da Função ou Ciclo de Negócio ▪ Modificar os arquivos de dados para aumentar o número de transações ou os scripts para aumentar o número de iterações que ocorre a cada transação. ▪ Scripts devem ser rodados em uma máquina (melhor caso para comparar um único usuário, uma única transação) e ser repetidas com múltiplos clientes (virtual ou real, ver Considerações Especiais abaixo).
Critério de Finalização:	<p>Único usuário ou transação: finalização com sucesso dos scripts de testes sem nenhuma falha e dentro da alocação de tempo por transação esperada ou requisitada.</p> <p>Múltiplas transações ou usuários: finalização bem sucedida dos scripts de teste sem qualquer falha e dentro da alocação de tempo aceitável.</p>
Considerações Especiais:	<p>Um teste abrangente de performance inclui ter uma carga de trabalho no servidor.</p> <p>Há vários métodos que podem ser usados para executar isso, incluindo:</p> <ul style="list-style-type: none"> ▪ “Direcionar transações” diretamente para o servidor, usualmente na forma de chamadas SQL. ▪ 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”. ▪ Usar múltiplos clientes físicos, cada um rodando scripts de teste para gerar uma carga no sistema. <p>O teste de performance deve ser executado em uma máquina dedicada ou em um tempo dedicado. Isso permite controle total e mensuração precisa.</p> <p>As bases de dados usadas para o Teste de Performance devem ser ou do tamanho real ou proporcionalmente iguais.</p>

3.1.5 Teste de Carga

Objetivo do Teste:	Verifique o tempo de resposta para as transações designadas ou casos de negócio sob condições variantes de carga de trabalho.
Técnica:	<ul style="list-style-type: none"> Use testes desenvolvidos para o Teste do Ciclo de Negócio ou Função. Modifique os arquivos de dados para aumentar o número de transações ou os testes para aumentar o número de vezes que cada transação ocorre.
Critério de Finalização:	Múltiplas transações ou usuários: finalização bem sucedida dos testes sem qualquer falha e dentro da alocação de tempo aceitável.
Considerações Especiais:	<ul style="list-style-type: none"> O teste de carga deve ser executado em uma máquina dedicada ou em um tempo dedicado. Isso permite o controle total a e mensuração precisa. As bases de dados usadas para os testes de carga devem ou ser do tamanho real ou igualmente dimensionadas.

3.1.6 Teste de Segurança e Controle de Acesso

Objetivo do Teste:	<ul style="list-style-type: none"> Segurança do Nível de Aplicação: Verifique que um ator pode acessar apenas aquelas funções ou dados para os quais o seu tipo de usuário tem permissão. Segurança do Nível de Sistema: Verifique que apenas aqueles atores com acesso ao sistema e aplicações têm permissão de acessá-los.
Técnica:	<ul style="list-style-type: none"> Segurança do Nível de Aplicação: Identifique e liste cada tipo de usuário e as funções ou dados para os quais cada tipo tem permissão. Crie testes para cada tipo de usuário e verifique cada permissão criando transações específicos para cada tipo de usuário. 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. Acesso de Nível de Sistema: Ver Considerações Especiais abaixo.
Critério de Finalização:	Para cada tipo de ator 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 Teste de Instalação

Objetivo do Teste:	<p>Verifique que os alvos de teste instalam apropriadamente em cada configuração de hardware necessária sobre as seguintes condições:</p> <ul style="list-style-type: none"> • Uma nova instalação, em um novo smartphone, que nunca fora anteriormente instalada com o TIG. • atualização, numa máquina onde o TIG já fora previamente instalado, para a mesma versão • atualização, numa máquina que já disponha do TIG instalado, de uma versão mais velha
Técnica:	<p>Manualmente ou desenvolva scripts automatizados, para validar a condição da máquina alvo – novo TIG nunca instalado; TIG na mesma versão ou versão mais velha já instalada.</p> <p>Começar ou executar a instalação</p> <p>Usando um subgrupo predeterminado de scripts de teste de funções, rode as transações.</p>
Critério de Finalização:	As transações do TIG executam de forma bem sucedida, sem falha.
Considerações Especiais:	Quais transações do TIG devem ser selecionadas para abranger um teste de confiança de que a aplicação TIG foi instalada de forma bem sucedida e que nenhum componente importante de software está faltando?

3.2 Ferramentas

As seguintes ferramentas serão empregadas para esse plano de testes:

	Ferramenta
Gerenciamento de Teste	Trello
Projeto de Teste	Microsoft Word
Gerenciamento de Projeto	Trello
Ferramentas do SGBD	Firebase

4 RECURSOS

Essa seção apresenta os recursos recomendados para o projeto Liv (Logística inteligente de vendas) suas principais responsabilidades, e seus conhecimentos ou conjunto de habilidades.

4.1 Trabalhadores

O quadro abaixo, representa as necessidades de recursos humanos para o projeto.

Recursos Humanos	
Trabalhador	Responsabilidades Específicas ou Comentários
Gerente de Teste, Gerente do Projeto de Teste	Fornece supervisionamento gerencial. Responsabilidades: <ul style="list-style-type: none"> • provê direcionamento técnico • adquire recursos apropriados • fornece relatórios de gerenciamento
Test Designer	Identifica, prioriza, e implementa os casos de teste. Responsabilidades: <ul style="list-style-type: none"> • gera o plano de teste • cria o modelo de teste • avalia a efetividade do esforço de teste
Testador	Executa os testes. Responsabilidades: <ul style="list-style-type: none"> • executar os testes • registrar os resultados • reestabelecer-se dos erros • documentar solicitações de mudança
Administrador do Sistema de Teste	Garante que o ambiente e os bens de teste sejam gerenciados e mantidos. Responsabilidades: <ul style="list-style-type: none"> • administrar o sistema de gerenciamento teste • instalar e gerenciar o acesso do trabalhador ao sistema de testes

Gerente do Banco de Dados, Administrador do Banco de Dados	Garante que o ambiente e bens de teste de dados (banco de dados) sejam gerenciados e mantidos. Responsabilidades: <ul style="list-style-type: none"> • administrar os dados de teste (base de dados)
Designer	Identifica e define as operações, atributos, e associações das classes de teste. Responsabilidades: <ul style="list-style-type: none"> • identificar e definir as classes de teste • identificar e definir os pacotes de teste
Implementador	Implementa e faz os testes unitários das classes e pacotes de teste. Responsabilidades: <ul style="list-style-type: none"> • cria as classes e pacotes de teste implementados no modelo de teste

4.2 Sistema

O quadro seguinte, expõe os recursos do sistema para o projeto de teste.

Recursos do Sistema
Servidor de Banco de Dados — Firebase
Terminais Clientes — 5 Smartphones Android conectados à Internet
Repositório de Testes — 2 smartphones

5 CRONOGRAMA

Atividade	Data de Início	Data de Término
Planejar Teste	xx/xx/xxxx	xx/xx/xxxx
Projetar Teste		
Implementar Teste		
Executar Teste		
Avaliar Teste		

6 CONSIDERAÇÕES FINAIS

Após aplicar os testes coletando seus resultados e aplicando as correções se necessário, busca trazer uma aplicação com o menor índice de erros e falhas possíveis, além de respeitar todas as suas funcionalidades e processos de forma coesa, com tudo uma aplicação onde o usuário venha usufruir de todas as suas funcionalidades sem problemas.