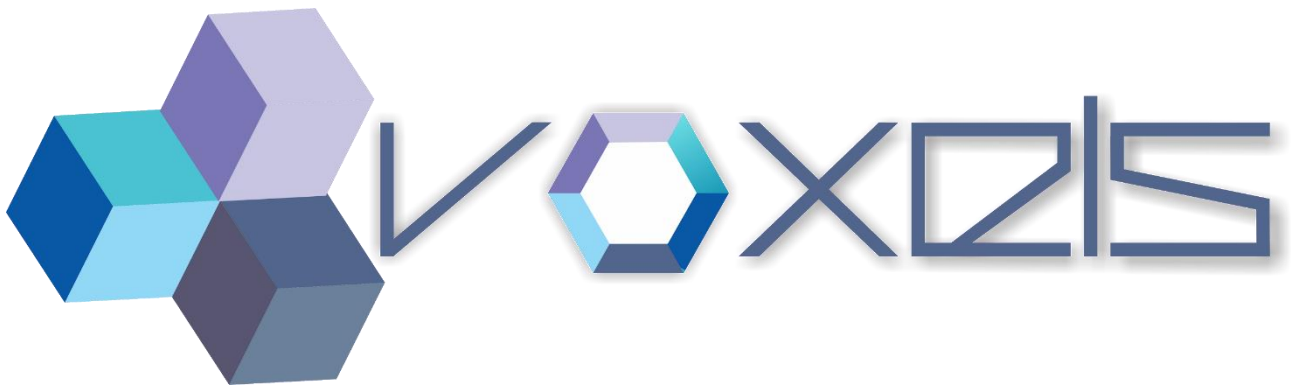


# **PLANO DE TESTE DE SOFTWARE VOXELS**



MUB - Mobilidade Urbana

### HISTÓRICO DE REVISÃO

DATA	VERSÃO	DESCRIÇÃO	AUTOR
05/11/2019	1.2	Fechamento de Versão	Cassiano Silva / Dayvison Oliveira
05/11/2019	1.1	Alteração dos programas necessárias e inclusão de novas ferramentas	Ruan Rodrigues
04/11/2019	1.0	Alteração das Ferramentas Necessárias	Ruan Rodrigues
10/10/2019	0.6	Organização do Histórico de Revisão	Cassiano Silva
10/10/2019	0.5	Fechamento de versão	Cassiano Silva
17/05/2019	0.4	Necessidades Ambientais	Geovanny Santos
13/05/2019	0.3	Abordagem Dos Testes	Ruan Rodrigues/ Hiago Mendes
10/05/2019	0.2	Resumo Dos Testes Planejados	Dayvison Oliveira
07/05/2019	0.1	Itens-alvo Dos Testes	Cassiano Silva

FICHA TÉCNICA / EQUIPE

<b>Membro</b>	<b>Função</b>	<b>Atribuições</b>
Cassiano Silva	Gerente de Projeto	Gerenciar os membros da equipe designando suas funções, repassando os prazos e identificar dificuldades encontradas pelos mesmo. Deve-se documentar, gerenciar e solucionar todos os problemas que possam surgir, realizar o controle de qualidade e assim proporcionar um nível de qualidade aceitável, exigir sempre o máximo de cada membro da equipe para que esteja sendo realizada com sucesso todas as etapas já pré definidas no cronograma, verificar cada etapa do projeto e assim, podendo passar para a fase seguinte, verificar a finalização do projeto e realizar um levantamento dos erros e acertos, sendo responsável pelo sucesso final de cada projeto.
Dayvison Oliveira	Analista de Sistemas	Acompanhar todo o processo de desenvolvimento junto ao Programador para que o mesmo não fuja de suas atividades ao ponto de atrapalhar ou se diferenciar do que já existe documentado, e caso haja essa mudança que necessite de uma nova correção da documentação o responsável (Analista de Sistemas) deverá realizar essas correções para que não exista diferença entre o que está documentado e o que está sendo desenvolvido. Responsável também por manter todas as documentações sempre disponíveis e manter o contato direto com o GP da equipe para um melhor desenvolver das mesmas.
Geovanny Santos	Projetista	Projetar uma arquitetura adequada de modo que se adapta ao sistema com foco em crescimento exponencial.Responsável por manter uma arquitetura de software ágil e fácil, com o objetivo de uma melhor interação dos usuários a aplicação.
Hiago Mendes	Programador	Responsável por manter contato com toda a estruturação do software,escolher a melhor linguagem para o projeto, escrever o código fonte de forma mais adequada e entregá-lo para o perfeito funcionamento, também responsável pela execução do projeto comunicando-se com outros setores do

		projeto.
Ruan Rodrigues	Teste	<p>O responsável por todas as atividades de testes dentro do processo de desenvolvimento que garantem a eficiência do sistema que está sendo desenvolvido.</p> <p>precisa analisar as aplicações para que possíveis bugs sejam corrigidos enquanto estão sendo desenvolvidos. Com isso, o objetivo geral é corrigir as falhas antes que o produto final fique pronto.</p>

## Sumário

DBMG - Database Management Systems	6
ITENS-ALVO DOS TESTES	9
RESUMO DOS TESTES PLANEJADOS	10
RESUMO DAS INCLUSÕES DOS TESTES	10
ABORDAGEM DOS TESTES	11
TÉCNICAS E TIPOS DE TESTE	11
Teste De Integridade De Dados E De Banco De Dados	11
Teste de Função	12
Teste de Ciclos de Negócios	12
Teste da Interface do Usuário (UI)	13
Determinação do Perfil de Desempenho	14
Teste de Carga	15
Teste de Stress	16
Teste de Volume	17
Teste de Segurança e de Controle de Acesso	18
Teste de Configuração	19
NECESSIDADES AMBIENTAIS	19
HARDWARE BÁSICO DO SISTEMA	19
ELEMENTOS DE SOFTWARE BÁSICOS DO AMBIENTE DE TESTE.	20
CONFIGURAÇÕES DO AMBIENTE DE TESTE	21

## Abreviaturas

TI	-	Tecnologia da Informação
	▪	DBMG - Database Management Systems
GUI	-	Interface Gráfica do Usuário
UI	-	Interface do Usuário
RAM	-	Random Access Memory (Memória de Acesso Aleatório)
DASD	-	Direct-Access Storage Device (dispositivo de armazenamento de acesso direto)

## Lista de Tabelas

Tabela 1 - Itens-alvo dos testes.	8
Tabela 2 - Teste De Integridade De Dados	10
Tabela 3 – Teste de Função	11
Tabela 4 - Teste de Ciclos de Negócios	12
Tabela 5 - Teste da Interface do Usuário	12
Tabela 6 - Determinação do Perfil de Desempenho	14
Tabela 7 - Teste de Carga	15
Tabela 8 - Teste de Stress	16
Tabela 9 - Teste de Volume	17
Tabela 10 - Teste de Segurança e de Controle de Acesso	18
Tabela 11 - Teste de Configuração	18
Tabela 12 - Recursos do Sistema	19
Tabela 13 - Elementos de Software	19
Tabela 14 - Configurações de Ambiente	20

## Lista de Quadros

## Lista de Figuras



## 1 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.

DISPOSITIVO	COMPONENTES DESENVOLVIDOS	COMPONENTES DE TERCEIROS	NAVEGADORES
Dispositivos móveis:	Módulo de Perfil (Usuário pode realizar cadastro, editar e desativar conta. Usuário após cadastro pode realizar login normalmente com	Módulo API Google Maps SDK (Para geolocalização)	

Centro Universitário Dr. Leão Sampaio  
Campus Crajubar

Celulares de várias resoluções e com Android superior a 4.0 (Ice Cream) Tablet de 7" a 11.5" com Android também superior a 4.0 (Ice Cream)	dados cadastrados inicialmente)		
	Módulo de Ocorrências (Agente de trânsito se utiliza desse módulo para atender, realizar e finalizar ocorrência, no final gerando relatório. Usuário se utiliza desse módulo para inserir suas ocorrências e Usuário pode verificar todas as ocorrências inseridas no sistema)	Níveis de API do Android 4.0 API 14 ou superior (Para uso de câmera e armazenamento)	
		API MUB (Para comunicação entre aplicativo Android e base dados)	
Outros Dispositivos:  Computadores e Notebooks	Módulo de Ocorrências (Órgão municipal faz de uso para monitoramento das ocorrências)	Web Módulo Google Maps Java Script API	Google Chrome, Mozilla Firefox, Edge e Opera
	Módulo de Relatórios (Para emitir relatórios semanais, mensais, trimestrais, semestrais ou anuais das ocorrências)		
	Módulo de Perfil (Administradores possui controle sobre a base de dados)		

**Tabela 1** – Itens-alvo dos testes.

## 2 RESUMO DOS TESTES PLANEJADOS

Os testes que irão ser realizados têm como objetivo garantir e assegurar o funcionamento da aplicação, de modo que as funcionalidades estejam em pleno funcionamento. Para isso utilizaremos TDD, testes de integração, teste de carga, teste de usabilidade, teste funcional e teste de regressão.

### 2.1 RESUMO DAS INCLUSÕES DOS TESTES

**Test-Driven Development (Desenvolvimento orientado a testes):** A partir do desenvolvimento da aplicação e do ciclo de desenvolvimento, os requisitos definidos serão transformados em casos de testes específicos, e ao decorrer do desenvolvimento os testes serão aplicados e avaliados se estão de acordo com os requisitos especificados.

**Teste de Integração e Banco de dados:** Garantir que os dados dos usuários sejam gravados no banco de dados de modo que as informações não sofram modificações às alterações na inserção dos dados cadastrais do usuário. Onde serão avaliados a integridade dos dados.

**Teste de Carga:** Avaliar o comportamento da aplicação em conjunto com o banco de dados, sob um alto índice de requisições ao sistema. A fim de encontrar gradualmente o nível máximo de requisições possíveis que a aplicação suporta, identificando a performance do mesmo e analisando os resultados obtidos.

**Teste de Usabilidade:** Com o objetivo de visualizar como o aplicativo desempenha o seu papel sob uso real de usuários. Visando o desempenho, precisão, lembrança e resposta emocional, respectivamente lidando com: Medição de tempo e quantidade de passos para realização da tarefa do aplicativo, quantos erros o usuário cometeu usando o aplicativo e o quão fatal pode ser esses erros, o quanto o usuário se lembra do uso após tempos sem usar e como o mesmo se sentiu após completar a tarefa que o aplicativo propõe.

**Teste Funcional:** Avalia se o aplicativo de fato atende aos requisitos funcionais que foram definidos no projeto, visando as funções principais que estão no aplicativo e verificando se a aplicação está apta para realizar as funções que foram definidas à elas.

**Teste de Regressão:** Avalia a aplicação de versões mais recentes após atualizações e inserções de componentes para que haja uma garantia de que os componentes antigos e os novos não tenham defeitos em sua versão final, a fim de entregar a aplicação em pleno funcionamento.

### 3 ABORDAGEM DOS TESTES

Devido ao nível da aplicação e seus componentes foram definidas algumas técnicas que foram julgadas importantes para o plano de testes da referida aplicação.

Foram descritas em conjuntos com as técnicas, suas ferramentas, estratégias e seu fator de sucesso que foi descrito nas tabelas como critério de êxito.

#### 3.1 TÉCNICAS E TIPOS DE TESTE

##### 3.1.1 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. Esses sistemas devem ser testados sem os aplicativos (como a interface para os dados). É necessário executar pesquisas adicionais referentes ao DBMS a fim de identificar as ferramentas / técnicas que poderão existir para suportar os testes.

<b>Objetivo da Técnica:</b>	Assegurar que os processos e/ou métodos de acesso ao Banco de Dados funcionem corretamente, sem corrupção de dados.
<b>Técnica:</b>	<ul style="list-style-type: none"><li>● Chamar cada processo e método de acesso a banco de dados, propagando cada um com dados válidos e inválidos (ou pedidos de dados).</li><li>● Inspeccionar o banco de dados para assegurar que os dados foram preenchidos conforme planejado e que todos os eventos do banco de dados ocorreram adequadamente ou revisar os dados retornados para assegurar que os dados corretos foram recuperados (pelas razões corretas)</li></ul>
<b>Estratégias:</b>	Teste de integridade de banco de dados, utilizando a estratégia de montar a estrutura dos pacotes no JMeter de modo que na sua construção. As técnicas de utilizar casos de usos para analisar as entradas e saídas durante a integração.
<b>Ferramentas Necessárias:</b>	A técnica requer ferramentas, podendo ser citadas: <ul style="list-style-type: none"><li>● JMeter</li></ul>

<b>Critérios de Êxito:</b>	Todos os processos e métodos de acesso ao banco de dados funcionam conforme projetado e sem nenhuma corrupção/perda de dados.
<b>Considerações Especiais:</b>	<ul style="list-style-type: none"><li>Os testes podem exigir drivers ou um ambiente de desenvolvimento DBMS para digitar ou modificar dados diretamente nos bancos de dados.</li><li>Os processos devem ser chamados manualmente.</li></ul>

**Tabela 2 - Teste De Integridade De Dados**

### 3.1.2 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. A tabela a seguir identifica um resumo do teste recomendado para cada aplicativo.

Ter certeza que as funções que foram utilizadas no ato de construção do sistema, estejam de acordo com as especificações citadas nos casos de uso. Com o objetivo de analisar os processos internos e cada interação, para validar se os testes coincidem com a regra de negócio aplicada. Utilizando-se técnicas do Teste de caixa preta.

<b>Objetivo da Técnica:</b>	Assegurar a navegação correta do aplicativo, além da entrada, processamento e recuperação de dados.
<b>Técnica:</b>	Executar cada caso de uso funcional através de seu fluxo principal e secundário, usando dados válidos e inválidos, para verificar o seguinte: <ul style="list-style-type: none"><li>Os resultados esperados ocorrem quando dados válidos são usados.</li><li>As mensagens de erro ou aviso apropriadas são exibidas quando dados inválidos são usados.</li><li>Cada regra de negócio é aplicada apropriadamente.</li></ul>
<b>Estratégias:</b>	Uma forma de verificar se a técnica utilizada estar de acordo com o que foi especificado, utilizaremos um smartphone com sistema operacional android igual ou superior a 4.0 para que se possa analisar se a funcionalidade realizadas no cadastro por exemplo, esteja gravando os dados no banco de dados de maneira correta sem perda ou corrupção das informações.
<b>Ferramentas Necessárias:</b>	A técnica exige ferramentas podem ser mencionadas: <ul style="list-style-type: none"><li>JMeter</li></ul>
<b>Critérios de Êxito:</b>	<ul style="list-style-type: none"><li>Todos os testes planejados foram executados.</li><li>Todos os defeitos identificados foram tratados.</li></ul>
<b>Considerações Especiais:</b>	Pois a análise dos testes, deverá ser verificados se as funcionalidades do sistema atende aos requisitos especificados.

**Tabela 3 – Teste de Função**

### 3.1.3 Teste de Ciclos de Negócios

Garantir que o sistema funcione apropriadamente durante um ciclo de atividades relativas ao negócio e que no final desse ciclo, todos os resultados esperados sejam obtidos.

Verificar se os relatórios de consultas realizadas estão sendo geradas corretamente.  
Verificar se o formulário de inserção de ocorrências está sendo preenchido com informações corretas  
Verificar se o tratamento de exceções está correto  
Verificar se os campos obrigatórios estão sendo preenchidos em cada formulário  
Verificar se os campos estão sendo preenchidos com informações no formato correto em cada formulário.

<b>Objetivo da Técnica:</b>	Assegurar que os processos de segundo plano e do aplicativo corretos funcionem de acordo com os planejamentos e os modelos de negócios requeridos.
<b>Técnica:</b>	<ul style="list-style-type: none"> <li>Os testes utilizados para o teste 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.</li> <li>Todas as funções sensíveis a datas ou tempo serão executadas usando datas ou períodos de tempo válidos e inválidos.</li> <li>Todas as funções que ocorrerem segundo um planejamento periódico serão executadas / iniciadas no momento adequado.</li> <li>As mensagens de erro / aviso apropriadas sejam exibidas quando dados inválidos forem utilizados.</li> <li>Cada regra de negócio será adequadamente aplicada.</li> </ul>
<b>Estratégias:</b>	Para verificar se as informações referentes as datas ou tempo estão de forma verídica. Uma forma seria adicionar uma informações com valores inconsistentes para que a aplicação possa analisar os dados informados que através das mesmas possam da continuidade na aplicação.
<b>Ferramentas Necessárias:</b>	<p>A técnica pode ser aplicada com as seguintes ferramentas:</p> <ul style="list-style-type: none"> <li>Testlink</li> <li>JMeter</li> </ul>
<b>Crítérios de Êxito:</b>	<ul style="list-style-type: none"> <li>Todos os testes planejados foram executados.</li> <li>Todos os defeitos identificados foram tratados.</li> </ul>
<b>Considerações Especiais:</b>	<ul style="list-style-type: none"> <li>Os eventos e as datas do sistema podem exigir atividades de suporte especiais</li> <li>É necessário um modelo de negócios para identificar requisitos e procedimentos de teste adequados.</li> </ul>

**Tabela 4 - Teste de Ciclos de Negócios**

### 3.1.4 Teste da Interface do Usuário (UI)

O teste da Interface com o Usuário verifica a interação de um usuário com o software. A meta do Teste de UI é assegurar que a Interface com o Usuário forneça ao usuário o acesso e a navegação adequados por meio das funções dos aplicativos. Além disso, o Teste de UI assegura que os objetos contidos na UI funcionem conforme esperado e estejam em conformidade com padrões corporativos ou do segmento de mercado.

<b>Objetivo da Técnica:</b>	<ul style="list-style-type: none"> <li>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)</li> <li>Objetos e características da janela, tais como menus, tamanho, posição, estado e foco estão em conformidade com os padrões.</li> </ul>
<b>Técnica:</b>	<ul style="list-style-type: none"> <li>Criar / modificar testes para cada janela a fim de verificar a navegação adequada e os estados de objeto para cada janela e objeto do aplicativo.</li> </ul>

<b>Estratégias:</b>	Escolha de um seletor grupo de 100 usuário de aplicativos Android direcionados a mobilidade urbana para melhor averiguação sobre a usabilidade da Interface do Usuário (UI) se realmente está de fácil manuseio, e também para ter uma melhor ideia em escala de quantas pessoas aprovam o aplicativo e quantas não gostaram para futuras melhorias.
<b>Ferramentas Necessárias:</b>	A técnica exige as seguintes ferramentas: <ul style="list-style-type: none"><li>• Smartphones de vários tamanhos de telas de 4” a 6,7” polegadas.</li><li>• Os testes serão realizados em versões do Android a partir do 4.0 (IceCream) com API level 14 até a versão 9.0 com API level 28.</li><li>• Várias localidades com distintas conexões de rede, desde 3G ao Wireless 5.8.</li><li>• Plataforma do Android Studio.</li></ul>
<b>Critérios de Êxito:</b>	<ul style="list-style-type: none"><li>• Verificação com êxito de cada janela permanecer consistente com a versão de benchmark ou dentro do padrão aceitável</li></ul>
<b>Considerações Especiais:</b>	<ul style="list-style-type: none"><li>• Nem todas as propriedades de objetos personalizados e de terceiros podem ser acessadas.</li></ul>

**Tabela 5 - Teste da Interface do Usuário**

### 3.1.5 Determinação do Perfil de Desempenho

O teste de desempenho mede tempos de resposta, taxas de transação e outros requisitos sensíveis ao tempo. A meta do teste de Desempenho é verificar e validar se os requisitos de desempenho foram alcançados. O teste de desempenho normalmente é executado várias vezes, cada uma utilizando uma "carga de segundo plano" diferente no sistema. O teste inicial deve ser executado com uma carga "nominal", semelhante à carga normal observada (ou prevista) no sistema de destino. Um segundo teste de desempenho é executado utilizando uma carga de pico. Além disso, os testes de desempenho podem ser utilizados para traçar o perfil e ajustar o desempenho de um sistema como uma função de condições, como a carga de trabalho ou configurações de hardware.

**Obs:** As transações da tabela a seguir são “transações de negócios lógicas”. Essas transações são definidas como casos de uso específicos que se espera que um ator do sistema execute utilizando o objetivo do teste como, por exemplo, adicionar ou modificar um determinado contrato.

<b>Objetivo da Técnica:</b>	Validar o Tempo de Resposta do Sistema para funções de negócios ou transações designadas sob as duas condições a seguir: <ul style="list-style-type: none"><li>• Volume normal previsto</li><li>• Volume de pior caso previsto</li></ul>
<b>Técnica:</b>	<ul style="list-style-type: none"><li>• Utilizar Scripts de Teste desenvolvidos para Teste de Modelo de Negócio (Teste do Sistema).</li><li>• Modificar arquivos de dados (a fim de aumentar o número de transações) ou modificar scripts a fim de aumentar o número de iterações ocorrido em cada transação.</li></ul>
<b>Estratégias:</b>	Será adicionado de forma gradativa informações a base de dados para verificar o desempenho da aplicação, com o objetivo de analisar o tempo de gravação para determinar o índice de desempenho do sistema.
<b>Ferramentas Necessárias:</b>	A técnica exige as seguintes ferramentas:

	<ul style="list-style-type: none"> <li>● Apache JMeter</li> <li>● JMETER</li> </ul>
<b>Crítérios de Êxito:</b>	<ul style="list-style-type: none"> <li>● Transação Única / usuário único: Conclusão com êxito dos scripts de teste sem nenhum defeito e na alocação de tempo esperada / requerida (por transação)</li> <li>● Várias Transações / vários usuários: Conclusão com êxito dos scripts de teste sem nenhum defeito e dentro de alocação de tempo aceitável.</li> </ul>
<b>Considerações Especiais:</b>	<ul style="list-style-type: none"> <li>● O teste abrangente do desempenho inclui ter uma carga "em segundo plano" no servidor. Há vários métodos que podem ser usados para executar esse teste, incluindo: <ul style="list-style-type: none"> <li>– "Encaminhar as transações" diretamente para o servidor, geralmente na forma de chamadas SQL.</li> <li>– Criar carga "virtual" de usuários para simular muitos (geralmente várias centenas) de clientes. Para se obter essa carga, geralmente são usadas ferramentas de Emulação de Terminal Remoto. Essa técnica também pode ser utilizada para carregar a rede com "tráfego".</li> <li>– Utilizar vários clientes físicos, cada qual executando scripts de teste para inserir carga no sistema.</li> </ul> </li> <li>● 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.</li> <li>● Os bancos de dados utilizados para teste de Desempenho deverão ter tamanhos reais ou ser igualmente escalados.</li> </ul>

**Tabela 6 - Determinação do Perfil de Desempenho**

### 3.1.6 Teste de Carga

Esse tem como objetivo analisar como o sistema se comporta diante de um grande processamento de dados, com o objetivo de atingir o limite de requisições até o sistema parar o seu funcionamento. Com isso será possível obter a carga máxima de processamento dos dados que a aplicação suporta simultaneamente, sem que haja qualquer falha por um determinado período de tempo.

**Obs:** As transações da tabela a seguir são “transações de negócios lógicas”. Essas transações são definidas como funções específicas que se espera que um usuário final do sistema execute utilizando o aplicativo como, por exemplo, adicionar ou modificar um determinado contrato.

<b>Objetivo da Técnica:</b>	Verificar o Tempo de Resposta do Sistema para casos de negócios ou transações designadas sob condições de carga de trabalho variáveis.
<b>Técnica:</b>	<ul style="list-style-type: none"> <li>● Utilizar os testes desenvolvidos para o Teste do Ciclo de Negócio.</li> <li>● Modificar os arquivos de dados (a fim de aumentar o número de transações) ou os testes a fim de aumentar o número de vezes que cada transação ocorre.</li> </ul>
<b>Estratégias:</b>	Como o teste de carga consiste na inserção de dados e/ou requisições simultaneamente fazendo assim com que chegue ao seu limite, utilizaremos de um script que fará as requisições com tamanhos gradativos até encontrar o limite

	ou até que as informações sejam parcialmente guardadas, também será feito um teste com o máximo de usuários que conseguirmos para que tenhamos uma base real de como a aplicação poderá responder em diferentes dispositivos e conexões.
<b>Ferramentas Necessárias:</b>	<ul style="list-style-type: none"> <li>● Apache JMeter</li> <li>● GenerateData</li> <li>● TestLink</li> </ul>
<b>Crítérios de Êxito:</b>	<ul style="list-style-type: none"> <li>● Várias Transações / vários usuários: Conclusão com êxito dos testes sem nenhum defeito e dentro de alocação de tempo aceitável.</li> </ul>
<b>Considerações Especiais:</b>	<ul style="list-style-type: none"> <li>● Os testes de carga devem ser executados em uma máquina dedicada e em um período de tempo dedicado. Isso permitirá o controle total e a medição exata.</li> <li>● Os bancos de dados utilizados para teste de carga deverão ter tamanhos reais ou ser igualmente escalados.</li> </ul>

**Tabela 7 - Teste de Carga**

### 3.1.7 Teste de Stress

O teste de estresse foi projetado para localizar erros devidos a falta de recursos ou competição por recursos. Pouca memória ou espaço em disco podem revelar defeitos no software que não são aparentes sob condições normais. Outros defeitos podem resultar da competição por recurso compartilhado, como bloqueios de banco de dados ou largura da banda de rede. O teste de estresse identifica a carga de pico que o sistema pode manipular.

**Obs:** As transações mencionadas na tabela a seguir são transações de negócios lógicas.

<b>Objetivo da Técnica:</b>	<p>Verificar se o sistema e o software funcionam corretamente e sem erros sob as seguintes condições de estresse:</p> <ul style="list-style-type: none"> <li>● Pouca ou nenhuma memória disponível no servidor (RAM e DASD).</li> <li>● Número máximo (real ou fisicamente capaz) de clientes conectados (ou simulados).</li> <li>● Vários usuários executando as mesmas transações nos mesmos dados / contas.</li> <li>● Conjunto / volume de transações no pior caso (consulte o teste de desempenho acima).</li> </ul>
<b>Técnica:</b>	<ul style="list-style-type: none"> <li>● Utilizar os testes desenvolvidos para o Teste de Desempenho.</li> <li>● 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).</li> <li>● 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.</li> </ul>
<b>Estratégias:</b>	Será feito em um ambiente controlado diversos acessos e diversas requisições ao servidor aderido para esta aplicativo, para verificar o real limite de acessos que podem ser realizados simultaneamente e tomar decisões de acordo com resultados obtidos.
<b>Ferramentas Necessárias:</b>	<p>A técnica exige as seguintes ferramentas::</p> <ul style="list-style-type: none"> <li>● TestLink;</li> <li>● Apache JMeter;</li> </ul>



<b>Crítérios de Êxito:</b>	<ul style="list-style-type: none"><li>• Todos os testes planejados são executados e os limites do sistema especificados são alcançados / excedidos sem o software ou falha do software (ou as condições sob as quais a falha do sistema ocorre estão fora das condições especificadas).</li></ul>
<b>Considerações Especiais:</b>	<ul style="list-style-type: none"><li>• Gerar estresse na rede pode exigir ferramentas da rede para carregar a rede com mensagens / pacotes.</li><li>• O DASD usado para o sistema deverá ser reduzido temporariamente a fim de restringir o espaço disponível para que o banco de dados se desenvolva.</li><li>• Sincronização do acesso simultâneo dos clientes aos mesmos registros / contas de dados.</li></ul>

**Tabela 8 - Teste de Stress**

### 3.1.8 Teste de Volume

O Teste de Volume sujeita o software a grandes quantidades de dados para determinar se serão atingidos limites que farão com que o software falhe. O teste de volume também identifica o volume ou a carga máxima contínua que o sistema pode manipular durante um determinado período de tempo. Por exemplo, se o software estiver processando um conjunto de registros de banco de dados para gerar um relatório, um Teste de Volume utilizará um grande banco de dados de testes e verificará se o software se comportou normalmente e gerou o relatório correto.

Analisar como o sistema se comporta diante de um grande processamento de dados, com o objetivo de atingir o limite até o sistema parar o seu funcionamento. Com isso será possível obter a carga máxima de processamento dos dados que o sistema suporta simultaneamente, sem que haja qualquer falha por um determinado período de tempo.

<b>Objetivo da Técnica:</b>	Verifica se o aplicativo / sistema funciona com êxito sob os seguintes cenários de alto volume: <ul style="list-style-type: none"><li>• Número máximo (real ou fisicamente capaz) de clientes conectados (ou simulados), todos executando a mesma função de negócio em pior caso (desempenho) por um período extenso.</li><li>• N tamanho máximo do banco de dados foi alcançado (real ou escalado) e várias consultas / transações de relatório são executadas simultaneamente.</li></ul>
<b>Técnica:</b>	<ul style="list-style-type: none"><li>• Utilizar os testes desenvolvidos para o Teste de Desempenho.</li><li>• Deverão ser usados vários clientes, executando-se os mesmos testes ou testes complementares a fim de produzir o conjunto / volume de transações no pior caso (consulte teste de estresse acima) durante um longo período de tempo.</li><li>• O tamanho máximo do banco de dados é criado (real, escalado ou preenchido com dados representativos) e vários clientes são utilizados para executar consultas / transações de relatório simultaneamente por longos períodos de tempo.</li></ul>
<b>Estratégias:</b>	A estratégia utilizada será realizar varia requisições ao banco de dados para verificar o volume suportado para se situar a capacidade máxima que a base pode desempenhar sem uma perda gradativa de desempenho.
<b>Ferramentas Necessárias:</b>	A técnica exige as seguintes ferramentas: <ul style="list-style-type: none"><li>• Apache JMeter</li><li>• JMETER</li></ul>

	<ul style="list-style-type: none"> <li>• TestLink</li> </ul>
<b>Crítérios de Êxito:</b>	<ul style="list-style-type: none"> <li>• Todos os testes planejados foram executados e os limites do sistema especificados são alcançados / excedidos sem o software ou falha do software.</li> </ul>
<b>Considerações Especiais:</b>	Verificar qual o período de tempo será considerado aceitável em condições de alto volume

**Tabela 9 - Teste de Volume**

### 3.1.9 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 ou acessar remotamente o sistema.

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, incluindo dados financeiros. No entanto, o “tipo de usuário dois” somente verá os dados demográficos referentes ao mesmo cliente.

A segurança no nível do sistema assegura que somente os usuários a que tenha sido concedido acesso ao sistema serão capazes de acessar os aplicativos e somente através dos gateways apropriados.

<b>Objetivo da Técnica:</b>	<ul style="list-style-type: none"> <li>• Segurança de Função / Dados: Verificar se o usuário pode acessar apenas as funções / dados para os quais seu tipo de usuário tenha recebido permissão.</li> <li>• Segurança do Sistema: Verificar se apenas os usuários com acesso ao sistema e aplicativo(s) têm permissão para acessá-los.</li> </ul>
<b>Técnica:</b>	<ul style="list-style-type: none"> <li>• Segurança de Função / Dados: Identificar e listar cada tipo de usuário e as funções / dados para os quais cada tipo tem permissão.</li> <li>• Criar testes para cada tipo de usuário e verificar a permissão criando transações específicas para cada tipo de usuário.</li> <li>• Modificar o tipo de usuário e executar novamente os testes para os mesmos usuários. Em cada caso, verificar se as funções / dados adicionais estão corretamente disponíveis ou se têm seu acesso negado.</li> <li>• Acesso ao Sistema</li> </ul>
<b>Estratégias:</b>	A princípio será testado o Login em si alterando as letras entre maiúsculas e minúsculas nos campos de users e passwords, tendo em vista que o sistema será Case Sensitive portanto fará a diferenciação entre letras maiúsculas e minúsculas, após verificação de que esse tipo de acesso está em devido funcionamento partiremos para a navegação do sistema e seu devido acesso àquele login.
<b>Ferramentas Necessárias:</b>	<ul style="list-style-type: none"> <li>• Apache JMeter</li> <li>• TestLink</li> </ul>
<b>Crítérios de Êxito:</b>	<ul style="list-style-type: none"> <li>• Para cada tipo de usuário conhecido, a função / dados apropriados estão disponíveis e todas as transações funcionem como esperado e sejam executadas nos testes de Função de Aplicativo anteriores</li> </ul>

<b>Considerações Especiais:</b>	<ul style="list-style-type: none"><li>• O acesso ao sistema deve ser revisado / discutido com o administrador da rede ou do sistema apropriado. Talvez esse teste não seja necessário, pois pode ser uma função de administração da rede ou do sistema.</li></ul>
---------------------------------	---

**Tabela 10 - Teste de Segurança e de Controle de Acesso**

### 3.1.10 Teste de Configuração

O teste de tolerância a falhas e de recuperação assegura que o objetivo do teste possa tolerar e se recuperar, de maneira eficaz, de uma série de falhas de hardware, software ou de rede com perda indevida de dados ou da integridade dos dados.

Para os sistemas que devem ser mantidos em execução, o teste de tolerância a falhas assegura que, ao ocorrer uma condição de tolerância a falhas, os sistemas alternativos ou de backup “assumirão” adequadamente o papel do sistema danificado sem qualquer perda de dados ou transações.

O teste de recuperação é um processo de teste antagonista em que o aplicativo ou o sistema é exposto a condições extremas, ou condições simuladas, para gerar falhas como, por exemplo, falhas de Entrada/Saída (E/S) de Dispositivo, ou ponteiros e chaves de banco de dados inválidos. Os processos de recuperação são disparados e o aplicativo ou o sistema é monitorado e inspecionado para verificar se foi efetuada a recuperação adequada do aplicativo ou do sistema e de dados.

<b>Objetivo da Técnica:</b>	Validar e verificar se os Aplicativos cliente funcionam corretamente nas estações de trabalho cliente prescritas.
<b>Técnica:</b>	<ul style="list-style-type: none"><li>• Utilizar scripts de Teste de Integração e do Sistema</li><li>• Abrir / fechar diversos aplicativos do PC, como parte do teste ou antes do início do teste.</li><li>• Executar transações selecionadas para simular atividades do usuário para dentro e fora de diversos aplicativos de PC.</li><li>• Repita o processo acima, minimizando a memória convencional disponível no cliente.</li></ul>
<b>Estratégias:</b>	A estratégia a ser utilizada será realizar analisar através da integração do sistema com o banco de dados, para analisar a falhas que possam ocorrer, com o intuito de determinar um índice de tolerância para a possíveis falhas da aplicação.
<b>Ferramentas Necessárias:</b>	A técnica exige as seguintes ferramentas: <ul style="list-style-type: none"><li>• Apache JMeter</li><li>• TestLink</li></ul>
<b>Crítérios de Êxito:</b>	<ul style="list-style-type: none"><li>• Para cada combinação, as transações são concluídas com êxito e sem falhas.</li></ul>
<b>Considerações Especiais:</b>	<ul style="list-style-type: none"><li>• Quais Aplicativos de PC estão disponíveis e acessíveis nos clientes?</li><li>• Quais os aplicativos normalmente usados?</li></ul>

**Tabela 11 - Teste de Configuração**

## 4 NECESSIDADES AMBIENTAIS

Neste tópico está contido todos os recursos não humanos necessários para o desenvolvimento dos testes, que são: Hardware, softwares, plugins, servidores, plataformas de testes;

### 4.1 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		
DISPOSITIVO	QUANTIDADE	NOME E TIPO
Servidor de Banco de Dados	1	MySQL
Domínio	1	<a href="http://www.voxelsbrasil.com">www.voxelsbrasil.com</a>
Nome do servidor de Hospedagem	1	Hostgator
Celulares de teste	2	Celular de usuários comuns e agentes de trânsito
Configuração Especial		Dispositivos móveis com acesso direto a internet (internet 3G/4G), GPS e versão Android superior a 4.0.3
Repositório de teste	1	<a href="http://www.voxelsbrasil.com/teste">www.voxelsbrasil.com/teste</a>
PC de desenvolvimento de teste	1	Computador com sistema superior ou igual ao Windows 7, mínimo de 4GB e RAM e processador superior ou igual ao i3 de 4ª geração
Configuração ambiental	1	Link de internet de no mínimo 1MB de internet e roteadores com redes 2.4 e 5.8Ghz

Tabela 12 - Recursos do Sistema

○

## 4.2 ELEMENTOS DE SOFTWARE BÁSICOS DO AMBIENTE DE TESTE.

Abaixo segue a lista de cada programa que será necessário para configurar ambiente de teste e realizar os mesmos; Seguindo a versão necessária e contendo observações a respeito de cada;

ELEMENTO DE SOFTWARE	VERSÃO	TIPO E OUTRAS OBSERVAÇÕES
Android Studio	Última Versão	Software de Desenvolvimento
Windows 10	Última Versão	Sistema Operacional
Google Chrome	Última Versão	Navegador da Internet
Firefox	Última Versão	Navegador da Internet
XAMPP	Última Versão	Software de Desenvolvimento
Tomcat	9.0.27	Software para executar API
Postman	Última Versão	Software para executar requisições na API
Visual Studio Code	Última Versão	Software para alterações necessárias no código e/ou configurações dos programas acima

Tabela 13 - Elementos de Software

### 4.3 CONFIGURAÇÕES DO AMBIENTE DE TESTE

Aqui está listada cada configuração necessária para configurar o ambiente de teste, ambientes físicos e suas descrições para que haja um melhor entendimento da metodologia utilizada

CONFIGURAÇÃO	DESCRIÇÃO	IMPLEMENTADA NA CONFIG. FÍSICA
Configuração do usuário comum	O usuário precisará possuir um aparelho celular.	Sistema Operacional Android
Mínima configuração suportada	A configuração mínima trata-se da versão 4.0.3 do android.	Sistema Operacional Android
Notebooks e/ou Computadores	Computadores com processadores core i3, sistema operacional Windows.	Hardware
Celulares	Celulares com sistema operacional android com versões superiores a 4.0.3.	Hardware
Android Studio	Plataforma de desenvolvimento para produção do aplicativo, e correções de bugs.	Plataforma de Desenvolvimento
Google Chrome	Será utilizado para verificar a funcionalidade do sistema.	Navegador da Internet

**Tabela 14 - Configurações de Ambiente**