

# Automobile Market

## Plano de Testes

### 1. Introdução

#### 1.1. Objetivos

Esse documento de plano de testes tem os seguintes objetivos:

- Identificar informações de projeto existentes e os componentes de software que devem ser testados.
- Listar os requisitos a serem testados.
- Descrever as estratégias de teste a serem empregadas.

#### 1.2. Escopo

O plano completo engloba testes unitários, de qualidade de código, de integração, de segurança e de carga/performance.

Os testes unitários garantirão que cada método e classe do sistema se comporte da maneira esperada.

Os testes de qualidade de código terão a responsabilidade de manter um padrão de qualidade para o código escrito preservando a manutenibilidade.

Os testes de integração testarão o comportamento funcional do sistema, ao se comunicarem entre si e com o banco de dados.

Os testes de segurança garantem que apenas os usuários autorizados podem executar ações protegidas no sistema.

Os testes de carga e performance garantirão que o sistema irá suportar a volumetria de usuários esperado e manter um tempo de resposta adequado.

#### 1.3. O Sistema

Automobile Market é um sistema com o objetivo de fornecer uma plataforma para comércio de automóveis. Apresenta funcionalidades de gerenciamento de automóveis e gerenciamento de boletos. O sistema é composto de 5 peças, sendo elas:

- Auth - Servidor de Autenticação
- Eureka - Seridor de Service Discovery
- Zul - Gateway da API
- Automobile - Microserviço para gerenciamento de automóveis
- Bill - Microserviço para gerenciamento de boletos

### 2. Requisitos a Testar

#### 2.1. Testes de Qualidade de Código

- Duplicação de código
- Cobertura de testes
- Avaliação de manutenibilidade
- Avaliação de confiança
- Avaliação de segurança

## **2.2. Testes de Integração**

- Verificar que o sistema permite cadastrar um automóvel.
- Verificar que o sistema valida que todos os campos necessários para o cadastro de automóvel estão preenchidos.
- Verificar que o sistema retorna uma lista com os automóveis cadastrados.
- Verificar que o sistema permite cadastrar um boleto referente a um dos automóveis cadastrados.
- Verificar que o sistema valida que todos os campos necessários para o cadastro de um boleto estão preenchidos.
- Verificar que o sistema valida que o automóvel referente ao boleto cadastro existe.
- Verificar que o sistema retorna os detalhes completos de um boleto cadastrado.

## **2.3. Testes de Segurança**

- Verificar que apenas usuários autenticados podem cadastrar automóveis.
- Verificar que apenas usuários autenticados podem cadastrar boletos.
- Verificar que apenas usuários autenticados podem buscar detalhes de boletos.

## **2.4. Testes de Carga e Performance**

- Verificar que o tempo de resposta permanece abaixo de 50 ms com 100 usuários.
- Verificar que o tempo de resposta permanece abaixo de 100 ms com 500 usuários.
- Verificar que o tempo de resposta permanece abaixo de 200 ms com 1000 usuários.

## **3. Estratégia de Teste**

### **3.1. Testes Unitários**

- Objetivo do Teste: Garantir que cada método e classe do sistema se comporte da maneira esperada
- Técnica: Os testes serão desenvolvidos utilizando JUnit e Mockito.
- Critério de Finalização: Cobertura de código superior a 90% com 100% de sucesso.

### **3.2. Testes de Qualidade de Código**

- Objetivo: Manter um padrão de qualidade para o código escrito preservando a manutenibilidade.
- Técnica: Será utilizado a ferramenta SonarQube para análise de código completa.
- Critério de Finalização: Duplicação de Código inferior a 3%; Cobertura de Testes superior a 90%; Avaliação de Manutenibilidade igual a A; Avaliação de confiança igual a A; Avaliação de segurança igual a A.

### **3.3. Testes de Integração**

- Objetivo: Testar o comportamento funcional do sistema, ao se comunicarem entre si e com o banco de dados.
- Técnica: Será utilizado a ferramenta Postman para fazer requisições em um ambiente de testes.
- Critério de Finalização: Todos os requisitos especificados neste documento para este teste devem ser atendidos.

### **3.4. Testes de Segurança**

- Objetivo: Garantir que apenas os usuários autorizados podem executar ações protegidas no sistema.
- Técnica: Será utilizado a ferramenta Postman para fazer requisições em um ambiente de testes.
- Critério de Finalização: Todos os requisitos especificados neste documento para este teste devem ser atendidos.

### **3.5 Testes de Carga e Performance**

- Objetivo: Garantir que o sistema irá suportar a volumetria de usuários esperado e manter um tempo de resposta adequado.
- Técnica: Será utilizado JMeter para execução de scripts que irão simular usuários virtuais acessando simultaneamente.
- Critério de Finalização: Todos os requisitos especificados neste documento para este teste devem ser atendidos.