

PLANO DE TESTES

EXO - API

Plano de testes da API Restfull da EXOAPI, para sistema de biblioteca. Teste de Back-End, para sistema de cadastro de usuário e consulta de livros.



EXO-API
TECHNOLOGY

Gerente:	Matheus Holannda gerente do projeto
Líder:	Gisele Nisimura líder do projeto
Membros:	<ul style="list-style-type: none">● Equipe A<ul style="list-style-type: none">○ -○ -● Equipe B<ul style="list-style-type: none">○ -○ -
Linha do Tempo:	22/03/2023
Projeto:	LibraryAPi

HISTÓRICO DE REVISÕES

DATA	VERSÃO	DESCRIÇÃO	AUTOR
11/03/2023	1.0	Projeção Inicial	LibApi

Detalhes do Projeto

Introdução

A Testful é uma empresa de desenvolvimento de software focada em testes para garantir a segurança, confiabilidade e consistência de softwares e implementações.

Implementação de testes em Softwares garante a integridade , a consistência dos dados e as informações por ele disponibilizadas.

Objetivos e Propósito

OBJETIVOS

Esse documento de Planos de Teste compõe dos seguinte objetivos:

- Identificar informações do projeto existentes e testar seus componentes
- Listar os requisitos e Testar áreas recomendadas
- Recomendar e descrever as estratégias de teste a serem empregadas
- Identificar os recursos necessários
- Listar os resultados do projeto de teste

ExoAPI (Contratante)

A ExoApi é uma empresa de desenvolvimento de softwares focada em desenvolver aplicações para empresas de grande e médio porte, além de construção de Web Applications, Web Sites, Local Systems , focando sempre em dar um produto seguro.

Para isso, o sistema desenvolvido precisa de uma análise de teste em sua aplicação desenvolvida com base numa aplicação de biblioteca, no qual usuários farão busca da disponibilidade de um livro, para isso o usuário necessita ter cadastro de acesso, tendo como segurança de log o uso de token.

O software desenvolvido é composto pelas seguintes regras pré-estabelecidas:

- Identificar informações do projeto existente e os componentes de software que devem ser testados
- Listar os requisitos e testar o s recomendados
- Recomendar e descrever as estratégias de teste a serem empregadas
- Identificar recursos necessários
- Listar os resultados do projeto de teste

Escopo

A LibApi passará por testes unitários, testes de integração e de sistema. Os testes unitários e de integração vão lidar com a qualidade funcional, das bases de dados, de controle de acesso, enquanto os teste de sistemas tratarão as questões de performance.

Os testes de performance entre o banco de dados será :

- na correta atualização de um cadastro no banco de dados, tanto ao cadastrar como ao atualizar (em todas entidades)
- criar token ao realizar um login, bem como erro com dados inválidos

1 - Especificações do Projeto de Teste

A tabela abaixo identifica a documentação e disponibilidade usados para desenvolver o plano de teste:

Documento	Disponível/Criado	Recebido/Revisado
Especificações de Requisitos	<input checked="" type="checkbox"/> sim <input type="checkbox"/> não	<input checked="" type="checkbox"/> sim <input type="checkbox"/> não
Plano de Projeto	<input checked="" type="checkbox"/> sim <input type="checkbox"/> não	<input checked="" type="checkbox"/> sim <input type="checkbox"/> não
Modelo e Análise	<input checked="" type="checkbox"/> sim <input type="checkbox"/> não	<input checked="" type="checkbox"/> sim <input type="checkbox"/> não
Modelo de Projeto	<input checked="" type="checkbox"/> sim <input type="checkbox"/> não	<input checked="" type="checkbox"/> sim <input type="checkbox"/> não
Documento de Arquitetura	<input checked="" type="checkbox"/> sim <input type="checkbox"/> não	<input checked="" type="checkbox"/> sim <input type="checkbox"/> não
Protótipo	<input checked="" type="checkbox"/> sim <input type="checkbox"/> não	<input checked="" type="checkbox"/> sim <input type="checkbox"/> não
Manual do Usuário	<input type="checkbox"/> sim <input checked="" type="checkbox"/> não	<input type="checkbox"/> sim <input checked="" type="checkbox"/> não
Lista de Riscos	<input checked="" type="checkbox"/> sim <input type="checkbox"/> não	<input checked="" type="checkbox"/> sim <input type="checkbox"/> não

Estratégia de Teste

1 - Requisitos a testar

Identificação de itens - casos de uso, requisitos funcionais - o que será testado

1 - Teste de Banco de dados

- verificar se o sistema pode realizar o cadastro , consultadas ou remover um usuário
- verificar se o sistema pode realizar um cadastrado, atualizar dados, fazer consulta ou remover um livro cadastrado

2 - Teste Funcional

- verificar se quando o usuário logar o sistema irá gerar um token de acesso
- verificar se o usuário consegue acessar com o uso do token

3 - Performance

- verifique o tempo de resposta

4 - Teste de segurança e controle de Acesso

- verificar se o acesso de usuário será somente com autorização com uso do token
- verificar hierarquia de acesso, no caso se somente o admin pode realizar alterações nas informações

5 - Teste de falha

2 - Tipos de Teste

2.1 - Teste de Integridade de banco de dados

Objetivo do Teste:	Garantir que os métodos e processos de acesso ao banco de dados funcionem apropriadamente e sem corrupções de dados
Técnica:	<ul style="list-style-type: none">● Invocar cada método e processo e de acesso ao banco de dados, alimentando cada um com dados ou requisições de dados válidos e inválidos● Inspecionar o banco de dados para garantir que os dados foram preenchidos como deveria, que os eventos do banco de dados ocorram apropriadamente , ao realizar consultas que os dados retornem corretamente
Critério de Finalização:	Assim que todos os métodos, e processos de acesso a base de dados funcionem como projetados, e sem corrupção
Considerações Especiais:	<ul style="list-style-type: none">● O teste pode necessitar de um ambiente de desenvolvimento ou drivers de SGBD para inserção ou modificação dos dados● Processos devem ser invocados manualmente

2.2 - Teste de Função

Objetivo do Teste:	Garantir a funcionalidade apropriada ao alvo teste, incluindo entrada de dados e processamento
Técnica:	<p>Executar cada caso de uso, fluxo de caso de uso, usando dados válidos e inválidos, verificar o seguinte:</p> <ul style="list-style-type: none">• ver os resultados esperados fazendo uso de dados válidos• se as mensagens de erro ou aviso apropriados são exibidas ao inserir dados inválidos• verificar se cada regra de negócio está sendo aplicada apropriadamente
Critério de Finalização:	<ul style="list-style-type: none">• se todos os testes planejados foram realizados• caso haja erros de funcionamento, se esses foram identificados e tratados
Considerações Especiais:	----- nenhuma-----

2.3 - Teste de Segurança e Controle de Acesso

Objetivo do Teste:	verificar se o acesso está restrito apenas a pessoas autorizadas
Técnica:	<ul style="list-style-type: none">• identifique e liste cada tipo de usuário e as funções ou dados para os quais o seu tipo de usuário tem permissão• crie testes para cada tipo de usuário e verifique cada permissão criando transações específicas para cada tipo de usuário• modifique o tipo de usuário e repita os testes para os mesmos usuários, verificando em cada usuário se os dados adicionados estão corretamente disponíveis ou negados
Critério de Finalização:	se todas as transações estão funcionando corretamente
Considerações Especiais:	Necessário ser revisado com o administrador da rede do contratante. (esse teste pode não ser necessário)

Recursos

A tabela abaixo mostra os recursos do sistema para o projeto:

Banco de dados:	<ul style="list-style-type: none">Azure DataBase StudioDocker (container/ persistencia em volume)servidor SQLServer
Terminais de testes:	2 PC
Ferramenta de desenvolvimento:	Visual Studio 2022

Cronograma

Histórico	Data de Início	Data de término
Planejar Teste	13/03/2023	14/03/2023
Projetar Teste	15/03/2023	16/03/2023
Implementar Teste	17/03/2023	17/03/2023
Executar Teste	17/03/2023	17/03/2023
Avaliar Teste	17/03/2023	17/03/2023

Aprovação do Projeto

Preparado por:

Aprovado por:

CARGO

CARGO

DD/MM/AAAA

DD/MM/AAAA