# Nome do projeto

Verificando as funções de usuários da API ServeRest e, também, a User Story.

# Resumo

Foi requisitado um planejamento de testes em uma seção de usuário de uma API REST chamada ServeRest. Cabe ao QA planejar e executar todos os testes relacionados possíveis.

# Pessoas envolvidas

Apenas o testador (QA) e o avaliador dos testes estarão envolvidos, em um primeiro momento, neste projeto.

# Funcionalidades ou Módulos a serem testados

Apenas o módulo de usuários da API será testado, em um primeiro momento, neste projeto.

# Local dos testes

Os testes serão realizados na residência do testador (QA) em modelo de Home-Office.

# Recursos e Ferramentas necessárias

- Conexão com Internet

- Computador ou aparelho similar

- API designada

- Softwares de planejamento de teste, tais como: Miro, GitLab, LibreOffice, etc.

- Softwares de execução de teste, tais como: Postman e Node JS.

# Critérios usados

- Crie uma requisição Get para validar o retorno de usuários através da API;

- Crie uma requisição Post para cadastrar um novo usuário através da API;

- Crie uma requisição Get para validar o retorno de um usuário apenas através da API;

- Crie cenários alternativos no cadastro de usuários, explore possíveis erros que podem ocorrer e mapeie as requisições através do Postman;

- Crie cenários alternativos na atualização de usuários, explore possíveis erros que podem ocorrer e mapeie as requisições através do Postman;

- Crie cenários alternativos na exclusão de usuários, explore possíveis erros que podem ocorrer e mapeie as requisições através do Postman.

# Riscos

Caso o testador não possa realizar suas atividades, seja por quaisquer motivos, é necessário que o avaliador (Rafael Vescio) saiba disso o quanto antes para que seja definido o que fazer em seguida.

A interpretação dos dados e criação do relatório é uma tarefa do testador (QA). Caso essa tarefa seja atribuída erroneamente a qualquer outra pessoa, é necessário que o avaliador (Rafael Vescio) saiba disso o quanto antes para que seja definido o que fazer em seguida.

# Como os resultados do teste serão divulgados

Os resultados dos testes, junto a todas as informações requisitadas sobre seu planejamento e execução, devem ser apresentados em cerca de 3 minutos durante o challenge, no dia 10 da sprint atual, conforme definido e avisado pelo organizador Rafael Vescio.

1. **Cronograma**

O cronograma de planejamento e execução de testes deve seguir o cronograma da sprint atual, com duração de 10 dias.

1. **Suíte e casos de teste**

Testes de API

* Lista de usuários
* Usuário encontrado por ID
* Usuário não encontrado por ID
* Cadastro de usuário
* Cadastro com e-mail já cadastrado
* Alteração de usuário
* Cadastro de usuário com PUT
* Cadastro com e-mail já cadastrado com PUT
* Exclusão de usuário

Testes de História de Usuário:

* Ciclo de cadastro de usuário com campos obrigatórios: Nome, e-mail, senha e administrador
* GET e PUT de usuários inexistentes, ou seja, não cadastrados
* POST de usuário com e-mail já usado
* Cadastro de usuário com PUT
* PUT de cadastro de usuário com e-mail já usado
* Cadastro de usuário com Gmail e Hotmail
* Cadastro de usuário com vários padrões de e-mail
* Cadastro de usuário com senhas abaixo de 5 caracteres e acima de 10 caracteres

Cenários e fluxos alternativos de teste:

* Cadastro de usuário (POST e PUT) com nome preenchido por caractere invisível
* Cadastro de usuário (POST e PUT) com nome contendo caracteres especiais
* Cadastro de usuário (POST e PUT) com nome curto ou longo demais

1. **Testes selecionados para serem candidatos a automação**

Testes que validam o objetivo da aplicação, nesse caso, fluxo de usuário (Vendedor):

* [GET] Lista de usuários cadastrados
* [POST] Cadastro de usuário com sucesso
* [POST] Cadastro de usuário sem um dos campos
* [POST] Cadastro de usuário com e-mail já usado
* [POST] Cadastro de usuário com gmail
* [POST] Cadastro de usuário com hotmail
* [POST] Cadastro de usuário com e-mail padrão inválido
* [POST] Cadastro de usuário com senha menor que 5 caracteres
* [POST] Cadastro de usuário com senha maior que 10 caracteres
* [POST] Cadastro de usuário com nome contendo caracteres especiais
* [POST] Cadastro de usuário com campo preenchido por caractere invisível
* [PUT] Cadastro de usuário com sucesso
* [PUT] Cadastro de usuário com e-mail já registrado
* [PUT] Cadastro de usuário com campo preenchido por caractere invisível
* [PUT] Cadastro de usuário com nome contendo caracteres especiais

1. **Cobertura de teste**

Operator Coverage (Imput) de [API] Usuários da ServeRest  
  
 9 Operações/Métodos da Swagger  
  
 4 Operações/Métodos para automação  
  
 9/4 = 44,44% de Cobertura de teste