



Plan De Pruebas Automatizadas

Reqres

Historia de Revisiones

Version	Autor	Descripcion	Fecha
1.0	Juan Esteban Londoño Ruiz	Creacion del documento de pruebas	Marzo-2022

Alcance De Las Pruebas

Para estas pruebas se llevaran a cabo las validaciones de las respuestas de los servicios internos que se consumen en el aplicativo web **Reqres**.

Estas pruebas seran efectuadas de manera automatizada, y estaran enfocadas en su totalidad en la verificacion de las funcionalidades consumidas a travez de microservicios(**API**)...

A continuacion se hara mención a las historias de usuario planteadas para cada prueba

1.Register

Yo como usuario del sistema
Necesito validar la operacion de registro de usuario
Para poder tener control de los usuarios

2.Update

Yo como usuario del portal
Necesito verificar la posibilidad de actualizar mis atributos de usuario
Para poder realizar cambios dentro de mi perfil cuando sea necesario

Fuera Del Alcance

1. No se tendrán en cuenta las demás funcionalidades del aplicativo web que no estén definidas dentro de las historias de usuario en las historias de usuario.
2. No se realizaran pruebas **manuales** para testear las funcionalidades del aplicativo Web.

3. No se realizaran pruebas a nivel de **WEB UI**
4. No se realizaran pruebas **no funcionales** sobre el aplicativo web.

Prerrequisitos para las Pruebas/Supuestos

- Tener un ambiente de pruebas estable.
- Las pruebas se llevaran acabo PC, asi que es necesario tener un equipo de compute que cuente con acceso a internet y con todas las herramientas necesarias previamente instaladas para ejecutar las prueba
- El equipo de desarrollo cuenta con la disponibilidad de apoyo para las contextualización y solución de bugs encontrados.

Estrategia

Lo principal que se hara sera realizar el **Smoke Test** que nos ayudara a verificar que se tenga acceso y disponibilidad al aplicativo desde nuestro sistema.

Antes de ser ejecutados los escenarios de manera automatica, se utilizara un programa de terceros que permita el consumo de microservicios(**Postman en este caso**) que nos ayude a validar la funcionalidad del aplicativo de manera manual, para despues proceder ahora si con la ejecucion de estos escenarios de manera automatica.

Las pruebas deberan ser repetibles para asegurar que la funcionalidad de los modulos probados cuenten con la fiabilidad y el estandar de calidad esperado.

Aca seran listados los escenarios planteados para las pruebas

1.Register

```
@RegisterSuccesfully
Scenario: Registro Exitoso
Given el usuario se encuentra en la pagina de registro con el correo "eve.holt@reqres.in" y password "pistol"
When el usuario hace una peticion de respuesta
Then el sistema debera mostrarle un codigo de respuesta exitoso junto un id y un token
```

2.Update

```
@JobActualization
Scenario: Actualizacion job
Given el usuario se encuentra en el apartado de actualizacion con el nombre "morpheus" y el job "zion resident"
When el usuario envia una peticion de respuesta
Then el sistema debera mostrarle un codigo de respuesta exitoso junto a sus atributos actualizados
```

Basandonos en nuestros resultados, verificaremos que no se presente ninguna anomalia con los resultados obtenidos de las pruebas, si esto llegara a pasar procederiamos entonces a documentar la

anomalia presentada y escalarla al equipo correspondiente para que sea realizada la gestion y solución de esta. Para ya despues ejecutar pruebas de regresion sobre el modulo o las funciones corregidas.

Riesgos

1. Se podria tener un retraso en la pruebas a realizar por contratiempos en las entregas de desarrollo, esto llevaria a tener perdidas economicas por parte del cliente y tambien se veria afectada la calidad y confiabilidad en el producto final.

Historia de usuario	Probabilidad	Impacto	Riesgo	Observacion
General:Retraso en la implementacion de las funcionalidades	2	3	6	Se podria tener un retraso en la pruebas a realizar por contratiempos en las entregas de desarrollo