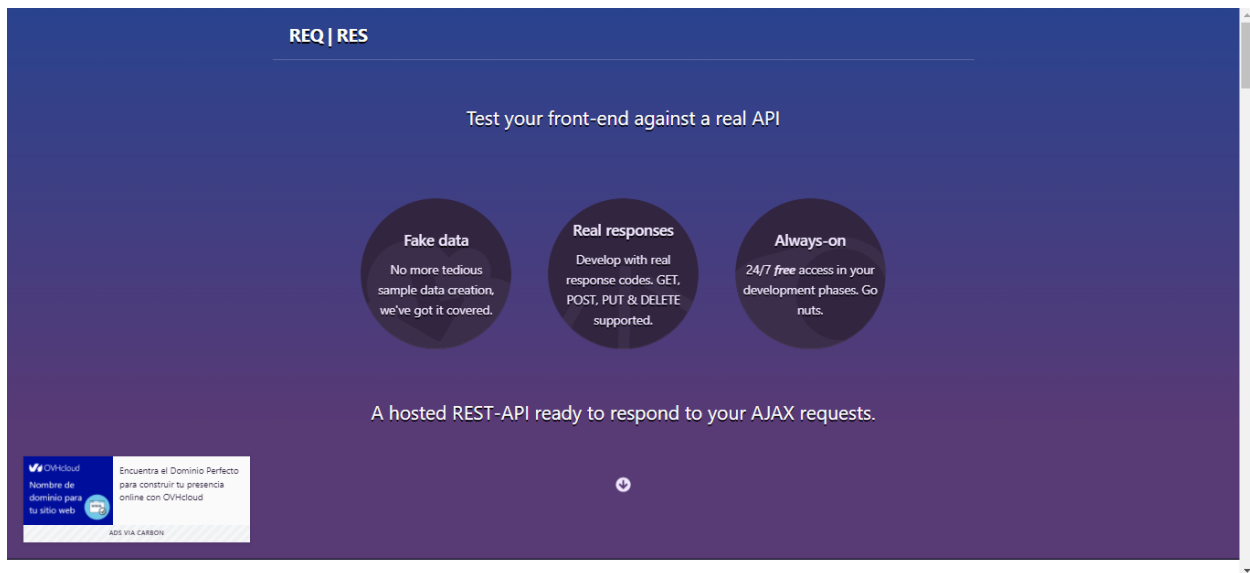


## Proyecto: Sitio Web Reqres.in



## Estrategia de Prueba Automatizadas

## Historia de revisiones

Versión	Autor(es)	Descripción	Fecha
1.0	Juan Felipe Garcia Duque	Documento evidencias	Diciembre 31 de 2021

# Tabla de Contenidos

## Contenido

1.Introducción.....	4
2. Alcance .....	5
<b>2.1 Alcance Sprint 1</b> .....	5
<b>2.2 Alcance Sprint 2.</b> .....	5
<b>2.3 No está en el alcance</b> .....	6
3. Roles y Responsabilidades .....	6
4. Riesgos .....	6
4.1 Matriz de Riesgos. ....	6
4.1.1 Probabilidad.....	6
4.1.2 Impacto.....	7
4.2 Riesgos Sprint 1 .....	7
4.3 Riesgos Sprint 2 .....	7
5. Diseño de casos de prueba .....	8
6. Ambiente y Herramientas de Pruebas .....	8
6.1 Herramientas de Pruebas .....	8
6.2 Ambiente de Pruebas .....	9
7. Criterios de Entrada y Salida .....	9
7.1 Criterios de Entrada .....	9
7.2 Criterios de Salida .....	9
8. Planificación de ejecución de las pruebas .....	9
9. Reporte de Pruebas.....	10
9.1 Reporte Exitoso.....	10
9.2 Reporte con error Generado.....	12

# 1.Introducción

En este documento se describe el plan para la realización de pruebas automatizadas se describe el alcance de las mismas, el ambiente de pruebas, los recursos necesarios, las herramientas a utilizar y los riesgos de las pruebas del proyecto Regres.in.

**Nota:**

- Se interpreta la palabra cliente como cualquier persona que ingrese a la página web de reqres.in y no solo a las personas que hagan un registro o login.

## 2. Alcance

Se probarán las siguientes historias de usuario distribuidas de la siguiente manera teniendo en cuenta que los sprint serán de 1 semanas y 1 iteración durará 2 sprint.

Se realizarán pruebas automatizadas a las funcionalidades seleccionada durante la planificación de cada sprint usando RestAssured.

Primer sprint la verificación de la HU "**HU001** – GET - Single user not found"

Segundo sprint la verificación de la HU "**HU002** – POST - Register - successful"

### 2.1 Alcance Sprint 1

Las pruebas del primer Sprint incluyen la verificación de la siguiente historia de usuario.

- **HU001** – GET - Single user not found

#### ❖ Criterios de Aceptación CA.

- |        |   |
|--------|---|
| CA001. | Generar GET - Single user not found       |
| CA002. | Obtener respuesta 404                     |
| CA003. | Obtener en body el vacío.                 |
| CA004. | Verificar los datos esperados y obtenidos |

### 2.2 Alcance Sprint 2.

Las pruebas del primer Sprint incluyen la verificación de la siguiente historia de usuario.

- **HU002** – POST - Register - successful

#### ❖ Criterios de Aceptación CA

- |        |   |
|--------|---|
| CA001. | Generar POST - Register – successful.           |
| CA002. | Obtener respuesta 200                           |
| CA003. | Obtener en body con la información del registro |
| CA004. | Verificar los datos esperados y obtenidos       |

## 2.3 No está en el alcance

- La ejecución de otras transacciones que no estén incluidas en este plan de pruebas.
- Cualquier otra validación o funcionalidad no incluida dentro de la estrategia de pruebas.

## 3. Roles y Responsabilidades

Roles	Responsabilidades
Manager de QA	Verificar que lo realizado corresponda.
Ingeniero QA de Automatización/ Analista QA	Planificación y monitoreo de las pruebas automatizadas Reporte de Defectos Reporte de progreso de las pruebas Diseño e implementación de las pruebas. Ejecución de las pruebas automatizadas. Reporte de resultados de las pruebas.

## 4. Riesgos

### 4.1 Matriz de Riesgos.

#### 4.1.1 Probabilidad

5	Frecuente
4	Moderado
3	Ocasional
2	Remoto
1	Improbable

### 4.1.2 Impacto

5	Catastrófico
4	Mayor
3	Moderado
2	Menor
1	Insignificante

## 4.2 Riesgos Sprint 1

- **HU001** – Selección de vuelo Ida y Vuelta para 2 adultos

CA	Probabilidad	Impacto	Riesgo	Observaciones
CA001	1	2	2	
CA002	1	2	2	
CA003	1	2	2	
CA004	1	3	3	

## 4.3 Riesgos Sprint 2

- **HU002** – Selección de vuelo Ida 1 adulto y un menor

CA	Probabilidad	Impacto	Riesgo	Observaciones
CA001	1	2	2	
CA002	1	2	2	
CA003	1	2	2	
CA004	1	2	2	

## 5. Diseño de casos de prueba

A continuación, se enuncian los diseños de casos de prueba:

**Feature:** Usuario no Encontrado

Yo como un nuevo cliente de la página de la dirección <https://reqres.in/> quiero registrarme exitosamente

**Scenario:** Usuario no encontrado

**Given** el Cliente está en la Url <https://reqres.in/> y Busca el usuario Veintitres.

**When** cuando el cliente realiza la petición de búsqueda de usuario

**Then** el cliente deberá ver un código de respuesta de usuario no encontrado

**Feature:** Registro Exitoso

Yo como un nuevo cliente de la página de la dirección <https://reqres.in/> quiero registrarme exitosamente

**Scenario:** Registro con éxito

**Given** el Cliente está en la Url <https://reqres.in/> y digita su usuario "eve.holt@reqres.in" y contraseña "pistol"

**When** cuando el cliente realiza la petición de creación de usuario

**Then** el cliente deberá ver un código de respuesta exitoso y los datos creados

## 6. Ambiente y Herramientas de Pruebas

### 6.1 Herramientas de Pruebas

Herramienta	Función
Selenium WebDriver	API para automatizar sistemas Web
JUnit testing framework	Ejecución y Reporte de las pruebas
Gradle	Creación de la estructura de proyectos y uso e importación de librerías
Chromedriver	Crea una instancia del navegador Chrome
Geckodriver	Crea una instancia del navegador Firefox
Cucumber	Gestor de features en lenguaje Gherkin
LOG4J	Generación de mensajes por consola



## 6.2 Ambiente de Pruebas

Navegadores	Chrome, Mozilla Firefox
Sistemas Operativos	Windows

## 7. Criterios de Entrada y Salida

### 7.1 Criterios de Entrada

- Las funcionalidades deben estar desplegadas en el ambiente de QA y haber sido probadas manualmente.
- El framework de pruebas está instalado y listo para la ejecución
- El ambiente de QA está disponible.

### 7.2 Criterios de Salida

- Ejecución de todos los casos de pruebas automatizados
- Se ha logrado la suficiente cobertura de los requerimientos y funcionalidades bajo pruebas
- Ningún defecto de severidad alta se encuentra abierto.

## 8. Planificación de ejecución de las pruebas

Lista de funcionalidades a ser automatizadas por Sprint

Sprint número	Funcionalidades	Comentarios
1	Realizar la petición GET	
2	Realizar la petición POST	

Es necesario que las funcionalidades a automatizar se desarrollen, implementen y prueben manualmente para que tengan un nivel determinado de estabilidad cuando comienzan las tareas de automatización.

## 9. Reporte de Pruebas

El Reporte automático de pruebas se obtendrá a través de Cucumber. Este Reporte informará sobre los resultados de la ejecución de la automatización. Incluirá las pruebas que pasaron y las que fallaron, los errores encontrados, la tasa de éxito y el tiempo transcurrido.

### 9.1 Reporte Exitoso

The screenshot displays the Cucumber Reports interface. At the top, a dark blue header contains the 'Cucumber Reports' logo and a 'Log in with GitHub' button. Below the header, a yellow warning banner states: 'This report will self-destruct in a day. Keep your future reports forever.' The main content area features a green bar indicating '2 PASSED'. Below this, a table summarizes the test results:

<b>100 % passed</b> 2 executed	<b>17 minutes ago</b> Last Run	<b>4.29 seconds</b> Duration
 Windows 10	 Java HotSpot(TM) 64-Bit Server VM 25.281-b09	 cucumber-jvm 6.11.0

Below the table, a red button labeled 'Delete Report' is visible. The main section of the report lists two features:

- file:///C:/WorkspaceJava/RetoRestAssured/src/test/resources/features/Get.feature**
  - Report Edit
  - Feature: Usuario no Encontrado
  - Yo como un nuevo cliente de la página de la dirección <https://reqres.in/> quiero registrarme exitosamente
  - Scenario: Usuario no encontrado
  - Given el Cliente está en la url <https://reqres.in/> y Busca el usuario Veintitres.
  - When cuando el cliente realiza la petición de búsqueda de usuario
  - Then el cliente deberá ver un código de respuesta de usuario no encontrado
- file:///C:/WorkspaceJava/RetoRestAssured/src/test/resources/features/Post.feature**
  - Report Edit
  - Feature: Registro Exitoso
  - Yo como un nuevo cliente de la página de la dirección <https://reqres.in/> quiero registrarme exitosamente
  - Scenario: Registro con éxito
  - Given el Cliente está en la url <https://reqres.in/> y digita su usuario "eve.holt@reqres.in" y contraseña "pistol"
  - When cuando el cliente realiza la petición de creación de usuario
  - Then el cliente deberá ver un código de respuesta exitoso y los datos creados

```
✓ Tests passed: 2 of 2 tests - 4 sec 241 ms
✓ co.com.sofka.runner.RRA 4 sec 241 ms
  ✓ Usuario no Encontrad 3 sec 495 ms
    ✓ Usuario no encont 3 sec 495 ms
  ✓ Registro Exitoso 746 ms
    ✓ Registro con exito 746 ms

{"id": 4,
 "token": "QpwL5tke4Pnpja7X4"}
}

HTTP/1.1 200 OK
Date: Mon, 03 Jan 2022 03:52:51 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 36
Connection: keep-alive
x-powered-by: Express
access-control-allow-origin: *
etag: W/"24-4iP0za1geN2he+ohu8F0FhCjLks"
via: 1.1 vegur
CF-Cache-Status: DYNAMIC
Expect-CT: max-age=604800, report-uri="https://report-uri.cloudflare.com/cdn-cgi/beacon/expect-ct"
Report-To: {"endpoints":[{"url":"https://a.nel.cloudflare.com/report/v3?s=8vpmKmEbHldwZjWDPJLiEbTC1vUbdgalnVWE"}]}
NEL: {"success_fraction":0,"report_to":"cf-nel","max_age":604800}
Server: cloudflare
CF-RAY: 6c7946da3bf663db-ATL
alt-svc: h3=":443"; ma=86400, h3-29=":443"; ma=86400, h3-28=":443"; ma=86400, h3-27=":443"; ma=86400

{
  "id": 4,
  "token": "QpwL5tke4Pnpja7X4"
}

2022-01-02 22:53:37 INFO RRAStepdefinitions:89 Verificacion Exitosa

View your Cucumber Report at:
https://reports.cucumber.io/reports/42ce9a92-fcdd-456c-8b53-90e418cbd982

This report will self-destruct in 24h.
Keep reports forever: https://reports.cucumber.io/profile

Process finished with exit code 0
```

## 9.2 Reporte con error Generado.

Cucumber Reports

Log in with GitHub

This report will self-destruct in a day. Keep your future reports forever.

2 FAILED

<b>0 % passed</b> 2 executed	<b>a few seconds ago</b> Last Run	<b>4.48 seconds</b> Duration
Windows 10	Java HotSpot(TM) 64-Bit Server VM 25.281-b09	cucumber-jvm 6.11.0

Delete Report

file:///C:/WorkspaceJava/RetoRestAssured/src/test/resources/features/Get.feature

Report Edit

Feature: Usuario no Encontrado

Yo como un nuevo cliente de la página de la direccion <https://reqres.in/> quiero registrarme exitosamente

Scenariio: Usuario no encontrado

- Given el Cliente está en la url <https://reqres.in/> y Busca el usuario Veintitres.
- When cuando el cliente realiza la peticion de busqueda de usuario
- Then el cliente deberá ver un codigo de respuesta de usuario no encontrado

```
org.opentest4j.AssertionFailedError: expected: <{}> but was: <{}>
    at org.junit.jupiter.api.AssertionUtils.fail(AssertionUtils.java:55)
    at org.junit.jupiter.api.AssertionUtils.failNotEqual(AssertionUtils.java:62)
    at org.junit.jupiter.api.AssertEquals.assertEquals(AssertEquals.java:152)
    at org.junit.jupiter.api.AssertEquals.assertEquals(AssertEquals.java:177)
    at org.junit.jupiter.api.Assertions.assertEquals(Assertions.java:1124)
    at co.com.sofka.stepdefinition.BRAStepDefinitions.elClienteDeberaVerUnCodigoDeRespuestaDeUsuarioNoEncontrado(BRAStepDefinitions.java:47)
    at #.el cliente deberá ver un código de respuesta de usuario no encontrado(file:///C:/WorkspaceJava/RetoRestAssured/src/test/resources/features/Get.feature:8)
```

file:///C:/WorkspaceJava/RetoRestAssured/src/test/resources/features/Post.feature

© 2022 SmartBear Software. All Rights Reserved.

Cucumber Reports

Log in with GitHub

This report will self-destruct in a day. Keep your future reports forever.

2 FAILED

<b>0 % passed</b> 2 executed	<b>a few seconds ago</b> Last Run	<b>4.48 seconds</b> Duration
Windows 10	Java HotSpot(TM) 64-Bit Server VM 25.281-b09	cucumber-jvm 6.11.0

Delete Report

file:///C:/WorkspaceJava/RetoRestAssured/src/test/resources/features/Get.feature

file:///C:/WorkspaceJava/RetoRestAssured/src/test/resources/features/Post.feature

Report Edit

Feature: Registro Exitoso

Yo como un nuevo cliente de la página de la direccion <https://reqres.in/> quiero registrarme exitosamente

Scenariio: Registro con exito

- Given el Cliente está en la url <https://reqres.in/> y digita su usuario "eve.holt@reqres.in" y contraseña "pistol"
- When cuando el cliente realiza la peticion de creacion de usuario
- Then el cliente deberá ver un codigo de respuesta exitoso y los datos creados

```
org.opentest4j.AssertionFailedError: expected: <{"id":"4","token":"QpwU5tke4Pnpja7X4"}> but was: <{"id":"4","token":"QpwU5tke4Pnpja7X4"}>
    at org.junit.jupiter.api.AssertionUtils.fail(AssertionUtils.java:55)
    at org.junit.jupiter.api.AssertionUtils.failNotEqual(AssertionUtils.java:62)
    at org.junit.jupiter.api.AssertEquals.assertEquals(AssertEquals.java:152)
    at org.junit.jupiter.api.AssertEquals.assertEquals(AssertEquals.java:177)
    at org.junit.jupiter.api.Assertions.assertEquals(Assertions.java:1124)
    at co.com.sofka.stepdefinition.BRAStepDefinitions.elClienteDeberaVerUnCodigoDeRespuestaExitosoYLosDatosCreados(BRAStepDefinitions.java:88)
    at #.el cliente deberá ver un código de respuesta exitoso y los datos creados(file:///C:/WorkspaceJava/RetoRestAssured/src/test/resources/features/Post.feature:8)
```

```
Run: RRATestCucumber <
Tests failed: 2 of 2 tests - 4 sec 427 ms
co.com.sofka.runner.RRA 4 sec 427 ms
  Usuario no Encontrad 3 sec 650 ms
  Usuario no encont 3 sec 650 ms
  Registro Exitoso 777 ms
  Registro con exito 777 ms

Cache-Control: max-age=14400
CF-Cache-Status: EXPIRED
Expect-CT: max-age=604800, report-uri="https://report-uri.cloudflare.com/cdn-cgi/beacon/expect-ct"
Report-To: {"endpoints":[{"url":"https://a.nel.cloudflare.com/report/v3?s=TYz0nAIEo6v5GhHijTLRbmD1CM3VixY6UNCBr"}]}
NEL: {"success_fraction":0,"report_to":"cf-nel","max_age":604800}
Vary: Accept-Encoding
Server: cloudflare
CF-RAY: 6c796bf56e4f8d40-ATL
alt-svc: h3=":443"; ma=86400, h3-29=":443"; ma=86400, h3-28=":443"; ma=86400, h3-27=":443"; ma=86400

{
}

org.opentest4j.AssertionFailedError: expected: <{ }> but was: <{}>
Expected :{ }
Actual   :{}
<Click to see difference>
```

```
Run: RRATestCucumber <
Tests failed: 2 of 2 tests - 4 sec 427 ms
co.com.sofka.runner.RRA 4 sec 427 ms
  Usuario no Encontrad 3 sec 650 ms
  Usuario no encont 3 sec 650 ms
  Registro Exitoso 777 ms
  Registro con exito 777 ms

Cache-Control: max-age=14400
CF-Cache-Status: EXPIRED
Expect-CT: max-age=604800, report-uri="https://report-uri.cloudflare.com/cdn-cgi/beacon/expect-ct"
Report-To: {"endpoints":[{"url":"https://a.nel.cloudflare.com/report/v3?s=TYz0nAIEo6v5GhHijTLRbmD1CM3VixY6UNCBr"}]}
NEL: {"success_fraction":0,"report_to":"cf-nel","max_age":604800}
Vary: Accept-Encoding
Server: cloudflare
CF-RAY: 6c796bf56e4f8d40-ATL
alt-svc: h3=":443"; ma=86400, h3-29=":443"; ma=86400, h3-28=":443"; ma=86400, h3-27=":443"; ma=86400

{
}

org.opentest4j.AssertionFailedError: expected: <{ }> but was: <{}>
Expected :{ }
Actual   :{}
<Click to see difference>
```