



Proyecto: Servicios SOAP

Plan de pruebas de automatización

## Historia de revisiones

Versión	Autor(es)	Descripción	Fecha
1.0	Nilton Rodríguez	Creación del documento	5 enero de 2022

## Tabla de contenido

<b>1. Introducción .....</b>	<b>4</b>
<b>2. Alcance .....</b>	<b>5</b>
<b>2.1. Casos de prueba.....</b>	<b>5</b>
2.1.1. TempConvert.....	5
2.1.2. CapitalCity .....	6
<b>3. Roles y Responsabilidades.....</b>	<b>7</b>
<b>4. Riesgos y Planes de contingencia .....</b>	<b>8</b>
<b>5. Ambiente y Herramientas de prueba. ....</b>	<b>9</b>
5.1. Herramientas de prueba .....	9
5.2. Arquitectura del framework de automatización .....	9
5.3. Ambiente de pruebas .....	10
<b>6. Criterios de entrada y salida.....</b>	<b>11</b>
6.1. Criterios de entrada .....	11
6.2. Criterios de salida .....	11
<b>7. Planificación de la ejecución de las pruebas.....</b>	<b>12</b>
<b>8. Reporte de pruebas .....</b>	<b>13</b>

## 1. Introducción

Simple Object Access Protocol (SOAP) es un protocolo estándar diseñado para que aplicaciones construidas en diferentes lenguajes y plataformas pudieran comunicarse. El protocolo SOAP continúa usándose principalmente en sistemas Legacy. El alcance para el actual plan de pruebas es el servicio de TempConvert, un servicio convertidor de grados Celsius a Fahrenheit, y CapitalCity, un sistema de consulta de ciudades capitales a partir del código ISO de cada país. Las pruebas se desarrollarán en sistemas operativos Windows y MacOS. La automatización de las pruebas se desarrollará con Java, usando la librería Serenity BDD, el patrón Screenplay y Cucumber. Entre los riesgos se encuentra la disponibilidad de los servicios. Ante dicho riesgo no se tiene plan de contingencia, puesto que las pruebas dependen directamente de la implementación de los servicios.

## 2. Alcance

Se realizarán pruebas de caja negra (automatizadas) a los servicios TempConvert(<https://www.w3schools.com/xml/tempconvert.asmx?wsdl>) y CapitalCity(<http://webservices.oorsprong.org/websamples.countryinfo/CountryInfoService.wso?wsdl>) . A demás de la correcta funcionalidad de los servicios, se realizarán pruebas con escenarios para verificar la respuesta ante consultas no soportadas, como letras en TempConvert y Códigos de países inexistentes den CapitalCity.

### 2.1. Casos de prueba

#### 2.1.1. TempConvert

**Feature:** Convert from Celsius to Fahrenheit

As

user of the converter

I WANT TO

validate that the functionality of the Celsius to Fahrenheit converter works correctly

SO THAT

I can make correct physics operations

**Scenario:** Convert Celsius degrees to Fahrenheit

**Given** a user of the converter that wants to convert "37" Celsius

**When** the user executes the converter to Fahrenheit

**Then** the user obtains the result "98.6"

**Scenario:** Convert an incorrect Celsius value to Fahrenheit

**Given** a user of the converter that set "A" as Celsius

**When** the user attempt to convert it to Fahrenheit

**Then** the user obtains the message "Error"

### 2.1.2. CapitalCity

**Feature:** Search the capital city of a country

AS

user of the system

I WANT TO

search the capital city of a country

SO THAT

I can visit the city

**Background:**

**Given** a user that wants to know the capital city of a country

**Scenario:** Consult the capital city of a country

**When** the user send the ISO code of Colombia "CO"

**Then** the user obtains the capital city "Bogota"

**Scenario:** Consult the capital city of a country that doesn't exist

**When** the user send the ISO code "CQ" that doesn't exists

**Then** the user obtains the message "Country not found in the database"

### 3. Roles y Responsabilidades

ROLES	RESPONSABILIDADES
Manger de QA	Planificación y monitoreo de las pruebas automatizadas. Reporte de defectos
Analista QA	Diseño e implementación de las pruebas. Ejecución de las pruebas automatizadas. Reporte de resultados de las pruebas.
Product Owner / Stakeholders	Toma de Decisiones.

#### 4. Riesgos y Planes de contingencia

No	Riesgos	Probabilidad de ocurrencia (1-4)	Impacto (1-4)	Severidad	Plan de contingencia
1	Los servicios no se encuentran disponibles.	1	4	4	



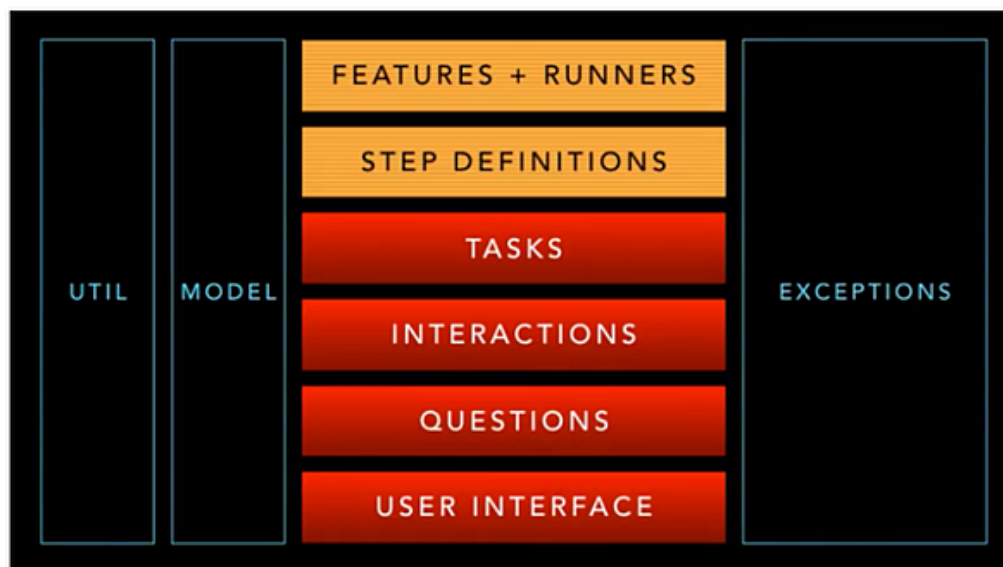
## 5. Ambiente y Herramientas de prueba.

### 5.1. Herramientas de prueba

Herramienta	Función
Serenity BDD	Librería para escribir pruebas automatizadas y producir documentación viva.
Screenplay	Patrón para escribir pruebas de aceptación automatizadas.
Cucumber	Herramienta de BDD para desarrollar casos de prueba y el reporte de las pruebas.
Gradle	Creación de la estructura de proyectos y uso e importación de librerías.

### 5.2. Arquitectura del framework de automatización

#### Arquitectura de Screen Play



Serenity BDD es una librería open source que ayuda a construir pruebas automatizadas limpias y fáciles de mantener sin necesidad de construir un framework propio. Serenity utiliza los resultados de las pruebas para generar informes que documentan la aplicación y su funcionamiento. Esta librería proporciona soporte para pruebas web con Selenium, pruebas API REST con RESTAssured, pruebas SOAP que son fáciles de escalar y mantener gracias al patrón Screenplay.

El patrón Screenplay se enfoca en escribir pruebas de aceptación automatizadas basadas en los principios SOLID, fomentando buenas prácticas de pruebas y conjuntos de pruebas fáciles de leer y mantener. Este patrón se centra en escribir pruebas como si se tratara de un guion de una película o una obra de teatro, donde *actores* hacen *preguntas*, ejecutan *tareas* y tienen *habilidades* que les permiten ejecutar *acciones* en su interacción con las páginas web.

Gracias a su integración con Behaviour Driven Development (BDD) Serenity se integra con otras librerías y frameworks como Selenium y Cucumber, lo cual permite que el código sea fácil de leer y pueda aplicar lenguajes como Gherkin de manera prácticamente nativa.

### 5.3. Ambiente de pruebas

Sistemas Operativos	Windows, MacOS.
---------------------	-----------------

## 6. Criterios de entrada y salida

### 6.1. Criterios de entrada

Las funcionalidades e y las funcionalidades han sido probadas manualmente.

El framework se encuentra instalado y listo para la operación.

El website se encuentra disponible.

Los defectos críticos encontrados durante pruebas manuales han sido resueltos y cerrados.

### 6.2. Criterios de salida

Ejecución de todos los casos de prueba automatizados.

Se logra suficiente cobertura de los requerimientos y funcionalidades bajo las pruebas.

Ningún defecto de severidad alta se encuentra abierto.

## 7. Planificación de la ejecución de las pruebas

Lista de funcionalidades a ser automatizadas.

No.	Funcionalidad	Comentarios
1	Convertir grados Celsius a Fahrenheit	A demás de su correcto funcionamiento se probará el comportamiento ante inputs no soportados para la operación.
2	Consultar ciudades capitales.	A demás de su correcto funcionamiento se probará el comportamiento ante códigos de países inexistentes.

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.

Se realizan pruebas con inputs no soportados puesto que los servicios a probar están diseñados para funcionar correctamente y generar una respuesta ante estos sin detener su ejecución.

## 8. Reporte de pruebas

El reporte automático de las pruebas se obtendrá a través de SerenityBDD y será publicado en un link generado directamente por esta herramienta. El reporte informará sobre los resultados de la ejecución de cada caso de prueba, incluirá las pruebas que pasaron y las pruebas que fallaron, la tasa de éxito y el tiempo transcurrido.

