

# Comparativa de Frameworks de Pruebas de APIs

Bryan Puma Villa, Álvaro Gallegos, Yumin Gomez, Nicolas Telleria, Oswaldo Chino

03 de Junio de 2021

## Resumen

El API REST permite el acceso y el control remoto de la consola Security Controls. La función proporciona una interfaz sencilla RESTful con respuestas de formato JSON ligeras que le permiten leer y escribir datos a/desde el programa. Le permite automatizar muchas de las operaciones diarias, lo que le ahorra mucho tiempo y esfuerzo. El API REST le permite integrar completamente los controles de seguridad de Ivanti en sus sistemas de orquestación y de automatización.

## Abstract

The REST API allows remote access and control of the Security Controls console. The function provides a simple RESTful interface with lightweight JSON format responses that allow data to be read and written to / from the program. It allows you to automate many of your daily operations, saving you a lot of time and effort. The REST API allows you to fully integrate Ivanti security controls into your automation and orchestration systems.

## 1. Introduccion

Una interfaz de programación de aplicaciones (API por sus siglas en inglés) es definida por la Universidad de Oxford como “un conjunto de funciones y procedimientos que permiten el acceso a las funcionalidades o

los datos de un sistema operativo u otros servicios.” Una explicación más simple: imaginémonos que estamos en un restaurante, donde interactuamos con el mesero. Él tomará nuestra orden, responderá nuestras preguntas y nos traerá la comida a la mesa cuando esté lista. El mesero está a cargo de todos los procedimientos complicados que suceden detrás de escena; nosotros no nos preocupamos por la cocina, los platos, las bebidas y demás. De cierta manera, el mesero funciona como la API del restaurante.

## 2. Desarrollo

### 2.1 ¿Qué son las APIS?

Las Interfaces de Programacion de Aplicaciones (APIs por sus siglas en inglés) son construcciones disponibles en los lenguajes de programación que permiten a los desarrolladores crear funcionalidades complejas de una manera simple. Estas abstraen el código más complejo para proveer una sintaxis más fácil de usar en su lugar.

Como ejemplo, piensa en el suministro de electricidad de tu casa, apartamento, o cualquier otro edificio. Si quieres usar un electrodoméstico, simplemente lo conectas en un enchufe y funciona. No intentas conectarlo directamente a la fuente de alimentación — hacerlo sería muy ineficiente y, si no eres electricista, difícil y peligroso.

## 2.2 Para qué sirve una API

Una de las principales funciones de las API es poder facilitarles el trabajo a los desarrolladores y ahorrarles tiempo y dinero. Por ejemplo, si estás creando una aplicación que es una tienda online, no necesitarás crear desde cero un sistema de pagos u otro para verificar si hay stock disponible de un producto. Podrás utilizar la API de un servicio de pago ya existente, por ejemplo, PayPal, y pedirle a tu distribuidor una API que te permita saber el stock que ellos tienen.

Con ello, no será necesario tener que reinventar la rueda con cada servicio que se crea, ya que podrás utilizar piezas o funciones que otros ya han creado. Imagínate que cada tienda online debiese tener su propio sistema de pago, para los usuarios normales es mucho más cómodo poder hacerlo con los principales servicios que casi todos utilizan.

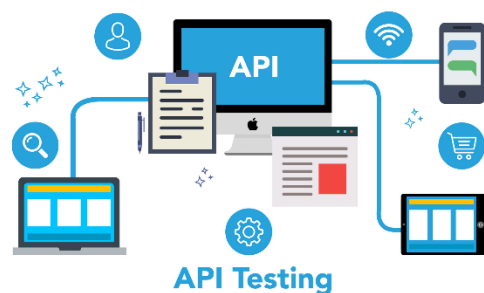
También son útiles para cuando lo único que se quiere es utilizar deliberadamente las funciones de determinado servicio para ofrecer ventajas a sus usuarios o atraer a los usuarios de ese servicio a que utilicen tu aplicación.

Por ejemplo, piensa que quieres crear una aplicación que se conecte a las publicaciones que hay en Twitter, pues para ello tendrás que conectar tu aplicación al servicio mediante la API que Twitter tiene disponible para los desarrolladores. Y si quieres que esta aplicación se comunique con la barra de notificaciones de un sistema operativo, pues también necesitará otra API del sistema operativo.

Piensa si no en cuando vas a comprar una entrada a través de la web de una sala de cine. Cuando pones la información de tu tarjeta, la web utiliza una API para enviarle esa información de forma remota a otro programa que verifica si tus datos son correctos o es una

tarjeta inventada. Una vez se verifica, este programa remoto le dice a la web que todo está en orden, y está ya te emite tus entradas.

Otro ejemplo clásico es el de las aplicaciones de terceros para conectarse a un servicio. Por ejemplo, redes sociales como Twitter o Reddit permiten que se creen aplicaciones diferentes a las oficiales para conectarse a ellas. Estas aplicaciones necesitarán las API de las redes sociales para poder mostrarte la información y los mecanismos internos que las hacen funcionar.



## 2.3 Configuración del entorno de prueba de API

Las pruebas de API son diferentes de otros tipos de pruebas de software, ya que la GUI no está disponible y, sin embargo, debe configurar el entorno inicial que invoca la API con un conjunto de parámetros requerido y, finalmente, examina el resultado de la prueba.

Por lo tanto, configurar un entorno de pruebas para las pruebas de automatización de API parece un poco complejo.

La base de datos y el servidor deben configurarse según los requisitos de la aplicación.

Una vez realizada la instalación, se debe llamar a la función de API para comprobar si esa API funciona.

## 2.4 Casos de prueba para pruebas de API:

Los casos de prueba de las pruebas de API se basan en

Valor devuelto basado en la condición de entrada: es relativamente fácil de probar, ya que la entrada se puede definir y los resultados se pueden autenticar

No devuelve nada: Cuando no hay ningún valor devuelto, un comportamiento de API en el sistema que se va a comprobar

Desencadenar alguna otra

API/evento/interrupción: Si una salida de una API desencadena algún evento o interrupción, se debe realizar un seguimiento de esos eventos y escuchas de interrupción.

Actualizar estructura de datos: La actualización de la estructura de datos tendrá algún resultado o efecto en el sistema, y eso debe autenticarse

Modificar ciertos recursos: Si la llamada a la API modifica algunos recursos, debe validarse accediendo a los recursos respectivos

## 2.5 Cómo probar la API

Las pruebas de automatización de API deben cubrir al menos los siguientes métodos de prueba aparte del proceso SDLC habitual

Pruebas de detección: El grupo de prueba debe ejecutar manualmente el conjunto de llamadas documentadas en la API, como la comprobación de que un recurso específico expuesto por la API se puede enumerar, crear y eliminar según corresponda

Pruebas de usabilidad: Esta prueba comprueba si la API es funcional y fácil de usar. Y api se integra bien con otra plataforma, así

Pruebas de seguridad: Esta prueba incluye qué tipo de autenticación se requiere y si los datos confidenciales se cifran a través de HTTP o ambos

Pruebas automatizadas: Las pruebas de API deben culminar en la creación de un conjunto de scripts o una herramienta que se pueda usar para ejecutar la API regularmente

Documentación: El equipo de pruebas tiene que asegurarse de que la documentación es adecuada y proporciona suficiente información para interactuar con la API. La documentación debe formar parte de la entrega final

## 2.6 Prácticas recomendadas de pruebas de API:

Los casos de prueba de API deben agruparse por categoría de prueba

Además de cada prueba, debe incluir las declaraciones de las API a las que se llama.

La selección de parámetros debe mencionarse explícitamente en el propio caso de prueba

Priorizar las llamadas a funciones de API para que sea fácil de probar para los evaluadores

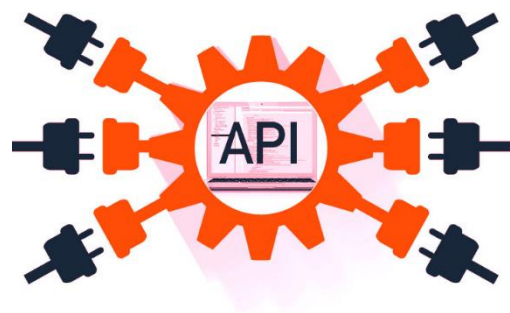
Cada caso de prueba debe ser lo más autónomo e independiente posible de las dependencias

Evite el "encadenamiento de pruebas" en el desarrollo

Se debe tener especial cuidado al manejar funciones de llamada de una sola vez como - Delete, CloseWindow, etc.

La secuenciación de llamadas debe realizarse y planificarse bien

Para garantizar una cobertura de prueba completa, cree casos de prueba de API para todas las combinaciones de entrada posibles de la API.



### 3. KATALON

Katalon es una herramienta de automatización de pruebas de código abierto de KMS Technology. Katalon Es una GUI muy simple y fácil de usar que permite que los usuarios no técnicos también escriban la prueba.

Katalon está construido sobre Selenium y Appium, también eliminará las complejidades que hacen que el comprobador sea más cómodo. La plataforma básica de Katalon es solo Eclipse, por lo que tendrás la misma sensación al trabajar con Katalon.



#### 3.1 Características de Katalon Studio

1- Katalon nos permite automatizar aplicaciones web, aplicaciones móviles y pruebas API.

2- Código abierto - Descargue y comience a usar en este momento.

3- En Build Library que ahorra mucho tiempo para un tester.

4- Grabación y reproducción para aplicaciones web y móviles.

5- Katalon tiene varias características como:

- Creación de Test Case : creación de Test Suite, Test Suite y captura de objetos utilizando Object Spy como QTP.
- Ejecución de prueba: la ejecución es bastante fácil con los botones rápidos. También nos permite ejecutar nuestra prueba en una configuración diferente. Podemos activar la prueba

en modo secuencial o en modo paralelo.

- Mantenimiento : crea automáticamente un depósito de objetos y mantiene nuestro objeto de prueba, lo que facilita el mantenimiento.
- Podemos segregar datos de prueba, repositorio de objetos, biblioteca, casos de prueba, suites de prueba.
- Integración con Git y GitHub, integración de Jenkins, etc.
- Informe: Katalon crea informes automáticamente y podemos exportar el informe en diferentes formatos, como CSV, HTML, PDF, etc.

### 4. SELENIUM

#### 4.1 ¿Qué es Selenium?

Selenium es un entorno de pruebas que se utiliza para comprobar si el software que se está desarrollando funciona correctamente. Esta herramienta permite: grabar, editar y depurar casos de pruebas que se pueden automatizar.

Lo interesante de Selenium es que se pueden editar acciones o crearlas desde cero.

También ayuda mucho en las pruebas de regresión porque consigue pruebas automatizadas que luego se pueden reutilizar cuando se necesite.

Comenzó a desarrollarse en 2004 por Jason Huggins y poco a poco se fueron uniendo varios especialistas. Un gran detalle es que este software es de código abierto (bajo licencia apache 2.0) y puede ser descargado y usado sin pagar.

Como curiosidad hay que indicar que el significado de Selenium surgió a raíz de un comentario que Jason Huggins le hizo a un competidor de aquella época llamado Mercurio: el envenenamiento por mercurio se puede curar con Selenio. Esto provocó que la mayoría de las personas que lo escucharon

empezasen a llamar al proyecto de Huggins, "Selenium".

#### 4.2 Principales características de Selenium

- Las acciones serán ejecutadas punto a punto, si así se considera.
- A la hora de escribir el código tiene la opción de autocompletar.
- Se puede referenciar a objetos DOM: nombre, ID o con XPath.
- Ejecutar test complejos que ahorran muchas horas de trabajo.
- Gran depuración y puntos de verificación
- Almacenamiento en varios formatos los test realizados



#### 5. Comparación Selenium vs Katalon

Features	Selenium	Katalon
Soporte para pruebas de Android y iPhone	✓	
Encuentra las coordenadas de cualquier objeto	✓	
Simule las interacciones del usuario, como pulsaciones de teclas y clics	✓	
la capacidad de escuchar eventos de los navegadores web (navegación, actualizaciones, aceptar alertas, etc.)	✓	✓
Generación de casos de prueba listos para usar a través de Web Recorder	✗	
Una interfaz de usuario intuitiva dedicada a las pruebas de API que admite WSDL, Swagger	✗	
Sistema de gestión de pruebas avanzado que facilita la ejecución en paralelo y por lotes	✗	
Integre fácilmente con otras herramientas de CI / CD como Jenkins, Bamboo, CircleCi, etc.	✗	

#### Conclusiones

Las pruebas en un proyecto de APIs son fundamentales, nos garantizan que podemos hacer cambios de versiones, actualizaciones, corrección de bugs o nuevas implementaciones garantizando que la especificación coincide con el método y que cualquier cliente que va a consumir nuestras APIs lo puede hacer sin ningún problema.

Selenium es una de las mejores herramientas de testing actual. Es muy avanzada y permite gestionar casi cualquier necesidad que tenga

un especialista. No es un programa sencillo y necesita un amplio conocimiento de programación para poder utilizarlo. La única desventaja, que se puede considerar como tal, es la dificultad de aprendizaje, pero ocurre lo mismo en otras herramientas o lenguajes de programación.

#### Recomendaciones

1. planificación previa desde el inicio de la estimación de casos de prueba, es aconsejable asignar prioridades de ejecución Alta, Media, Baja y palabras clave como regresión, Sanidad, etcétera para los casos de prueba. De esta manera se puede hacer un filtrado rápido por prioridades y etiquetas, que permitan diferenciar rápidamente cuales deber ser los casos más importantes.

2. Uso de técnicas de diseño de casos de prueba Entidades como International Software Testing Qualifications Board y American Society for Quality mencionan varias técnicas para la creación de casos de prueba que permiten una mejora, de tal manera que al aplicarlas se obtienen menos casos de prueba y un mayor factor de cobertura o bien una base de creación empírica. Al tener un menor número de casos de prueba, se reducen las pruebas exhaustivas y tiempos de ejecución.

3. Clasificación adecuada mediante suites. La división de casos de prueba según los módulos de un sistema facilitara la selección de casos al momento de construir un Plan de Pruebas para regresión. En las carpetas o suites deberían evitarse nombres ambiguos o diferentes a lo definido en el requerimiento, o en los nombres que se muestran en el sistema. Si la herramienta de manejo de casos de prueba lo permite, lo ideal es organizar las suites mediante jerarquías según la misma aplicación para que la selección de casos de prueba por componente pueda ser natural. Por ejemplo, si el módulo "Transferencias" tiene como opciones "Transferencias locales", "Transferencias a terceros", "Transferencias internacionales", debería existir un folder

padre llamado "TRANSFERENCIAS", con 3 carpetas hijas, de manera tal que la carpeta llamada "Transferencias" tenga como opciones "Transferencias locales", "Transferencias a terceros", "Transferencias internacionales". Este tipo de organización es útil para mantener los casos de prueba ordenados y agilizar la búsqueda de casos de prueba. Sin embargo, no se debe abusar de la jerarquía, pues llegar a tener más de 4 o 5 niveles de profundidad se torna poco eficiente.

## **Bibliografía**

[1] [Tutorial de pruebas de API: ¿Qué es la automatización de pruebas de API? Cómo probar \(guru99.com\)](#)

[2] [API REST de partners de Microsoft - Partner | Microsoft Docs](#)

[3] [Repositorio Digital Universidad Técnica del Norte: Análisis de frameworks de desarrollo de api rest y su impacto en el rendimiento de aplicaciones web con arquitectura Spa \(utn.edu.ec\)](#)

[4] [Articles > Page #1 \(infoq.com\)](#)

[5] [Introducción a Selenium: Cómo funciona, Características y Opciones \(digital55.com\)](#)

[6] [Tutorial para usar Selenium: conceptos básicos y primeros pasos - IONOS](#)

[7] <https://javiermartinalonso.github.io/drafts/testing/katalon/2018-12-31-katalon.html>

[8] <https://www.softwaretestinghelp.com/selenium-vs-katalon-studio/>