

EJERCICIO 3 (QAA-03) EXPLICACIÓN DE TÉCNICAS

Natalia Andujar.

Para ambos ejercicios, el primer paso previo a la definición de los casos de prueba consistió en el análisis de los criterios especificados para los fines.

Concluido el análisis, se procedió a estructurar la matriz con los elementos imprescindibles a especificar en los casos tales como el ID de la prueba, el tipo de prueba (HappyPath/CasoAlterno), descripción, precondiciones, datos a utilizar, resultado esperado, resultado obtenido, estado de satisfacción y defectos asociados.

Para ambos ejercicios, los tipos de pruebas que estuve realizado corresponden a **pruebas funcionales y de regresión** que en conjunto me permiten validar los requerimientos del usuario (Criterios) y comprobar en todo momento el estado de los diferentes escenarios para cualquier alerta de inyecciones o reproducción de errores que causen defectos y provoquen situaciones incómodas ante fallos.

Así mismo, entre las técnicas utilizadas para generar los escenarios se encuentra principalmente las pruebas combinatorias, que dado las diferentes entradas y sus dominios de valores, me permitieron cubrir una gran cantidad de escenarios entre estos, encontrándose de camino feliz para casos aceptables y caminos alternos que bien pueden ser errores humanos o errores del servidor.

Como buena práctica, se estableció el siguiente estándar en el nombre de los casos automatizados para su identificación factible:

- ⇒ **CAMINO FELIZ** = [RECURSOS – CAMPO/S – HAPPYPATH]
- ⇒ **CASOS ALTERNOS** = [RECURSOS – CAMPO/S – DESCRIPCION]

En el caso del **Ejercicio 2 (QAA-02)**, aunque el requerimiento solo contemplaba automatizar por lo menos 6 casos de los levantados, se contempló un poco más para validar la reacción y respuesta HTTP de los recursos del API, identificándose algunos de los siguientes hallazgos:

- Respuestas HTTP incorrectas según el escenario de prueba.
- Endpoint sin validaciones de formato, longitud, tipo y demás en los parámetros de entrada.
- Al incluir dos parámetros sin relación en el request, trae la data relacionada a uno de estos dos cuando debería no haber encontrado el registro.
- Sensibilidad a minúsculas y mayúsculas.

Conforme a lo anterior, entre los puntos que deben ser tomados en cuenta al momento de probar una API se pueden citar:

- 1) Documentación de la función del API y cada método.
- 2) Uso de métodos de petición correctos.
- 3) Qué tipo de API es (SOAP/REST).
- 4) Endpoints que componen el API.
- 5) Parámetros de entrada obligatorios y opcionales.
- 6) Validaciones de los parámetros de entrada.
- 7) Si requiere autenticación o esta expuesta a uso público.
- 8) Formato y estructura de salida, si mantienen un estándar para todas las respuestas.
- 9) Uso de códigos de estado de respuesta HTTP correctos:

- ✓ Respuestas informativas (100–199).
- ✓ Respuestas satisfactorias (200–299).
- ✓ Redirecciones (300–399).
- ✓ Errores de los clientes (400–499).
- ✓ y errores de los servidores (500–599).

- 10) Consistencia e integridad de la data retornada conforme a lo solicitado.
- 11) Registro de información de errores.
- 12) Disponibilidad y tiempo de respuesta habitual.
- 13) Carga soportada y variación en el tiempo de respuesta.
- 14) Uso de controladores para agrupar los endpoint por funcionalidad.