Prueba - Automation

Complete hasta donde sus conocimientos le permitan, la idea de la prueba es obtener un perfil completo como automator.

# Teórico

1. Si te piden “automatizar todo”, ¿Cuál sería tu respuesta?

**Primero: Entender el desarrollo realizado, validando que flujos se pueden automatizar en base a sistema único o depende de otros (integración).**

**Segundo: Identificar escenarios para ambos casos (manual y automation)**

1. ¿Qué tipos de funcionalidades consideras que se deben automatizar y cuáles no?

**Cuando no se puede interactuar con otros sistemas por temas de seguridad y permisos.**

1. ¿Qué funcionalidad de Selenium utilizarías para localizar y interactuar con un elemento de la página? ¿Consideras que hay algún orden en el que deberían ser utilizados? ¿Cuál?

**El orden depende del proyecto dado BDD, como funcionalidad de Selenium WebDriver ofrece varios tipos de selectores integrados como por ejemplo findelement a través de un xpath, css, id, name, etc..**

1. ¿Conoces el paradigma Record & Playback? ¿Podrías explicarlo?

**Grabar las pruebas para posterior revisar (play), a diferencia de scripting donde la revisión puede ser mas en detalle.**

1. ¿Qué tipos de esperas soporta Selenium? ¿Cuál es la diferencia entre ellas? Qué opinas sobre el uso de ***Thread.sleep()*** para esperar por algún elemento o comportamiento de la aplicación?

**Las que he ocupado es webdriverwait y sleep; me parace mejor la primera ya que es mas eficiente al esperar el tiempo preciso.**

1. ¿Qué partes debe contener un reporte de incidente?

**Gráficas, resultados de test (ok, nook, no ejecutados), detalle de errores, tester, id de la prueba o escenario.**

1. ¿Cómo estimarías el esfuerzo de automatizar 100 casos de prueba?

**Depende del tiempo de cada caso y después se puede estimar completo aunque desde el punto de vista de Scrum se le puede dar un peso X; para cierta cantidad, por ejemplo cada 10 casos seria un peso XS.**

1. A partir del siguiente código: Que puntos de mejora encuentras y qué cambios harías?



**\*url se puede incluir en otro package de links**

**\*los elementos y acciones sobre estos los dejaría en otro package como en el ejemplo de mercadolibre**

1. ¿En qué consiste el patrón Page Object? ¿Qué beneficios trae consigo el uso de este patrón en la automatización de pruebas?

**Es un patrón de diseño de objetos en Selenium, donde las páginas web se representan como clases y los diversos elementos de la página se definen como variables en la clase. Todas las interacciones de usuario posibles se pueden implementar como métodos en la clase. Reduce el índice de errores en las correcciones de la codificación y actualizaciones en el código de desarrollo.**

1. ¿Cómo manejarías la interacción con alertas en Selenium?

**org.openqa.selenium.Alert para manejar las alertas en Selenium**

1. ¿Qué técnica utilizarías para lograr la ejecución de un mismo test con distintos juegos de datos?

**Manejando un Excel de entrada con distintos set por hoja de la planila.**

1. ¿Cómo se seleccionan todos los registros de una tabla genérica?

**Select \***

1. ¿A qué se denomina Join y que tipos existen?

**sirven para combinar filas de dos o más tablas basándose en un campo común entre ellas, devolviendo por tanto datos de diferentes tablas**

**Los más importantes:**

**INNER JOIN: Devuelve todas las filas cuando hay al menos una coincidencia en ambas tablas.**

**LEFT JOIN: Devuelve todas las filas de la tabla de la izquierda, y las filas coincidentes de la tabla de la derecha.**

**RIGHT JOIN: Devuelve todas las filas de la tabla de la derecha, y las filas coincidentes de la tabla de la izquierda.**

**OUTER JOIN: Devuelve todas las filas de las dos tablas, la izquierda y la derecha. También se llama FULL OUTER JOIN.**

1. ¿Cómo se agregan registros a una tabla genérica?

**INSERT INTO "nombre\_tabla" ("columna1", "columna2", ...) ...**

1. ¿Cómo se cambia el valor del campo de una tabla genérica?

**Update tabla Set campo1='José' Where campo2='Pepe'**

1. ¿Qué herramientas utilizarías para probar tus test en distintos navegadores?

**Manejo de webdriver (.exe) para cada navegador**

1. ¿Qué es un framework de automatización y qué beneficios consideras que trae consigo el desarrollo de uno?

**Es una plataforma base de pruebas con una estructura especifica de trabajo; tiene beneficios de manejo de repositorios de data entrada/salida, evidencias, código de automation.**

1. ¿Cómo eliminarías el código duplicado de tus pages object? ¿En el caso de los test se podría aplicar la misma técnica?

**Para eso se maneja nomenclatura por tipos por ejemplo botonpag1, botonpag2, labelframe1, labelframe2, y en los test se puede aplicar lo mismo.**

1. ¿Qué otros patrones de diseño conoces que pueden ser aplicados en la automatización?

**Solo conozco PO**

1. ¿Qué diferencia hay entre cerrar el driver utilizando la función ***Close()*** o mediante la función ***Quit()***?

**El primero cierra el actual navegador y el siguiente cierra todas las conexiones.**

1. ¿Es posible ejecutar código Javascript sobre la página que estás probando utilizando Selenium?

**Si, lo he utilizado para moverme a los extremos de la pagina**

1. ¿Tienes conocimiento sobre CI/CD? Si trabajaste con esto, ¿podrías hacer un diagrama básico?

**Repositorio GIT -> Pipeline (Jenkins / gitlab - ci) -> Imagenes docker para deploy -> test unitarios(tdd) /automatizados (bdd)**

1. ¿Conoces el concepto de BDD? ¿En qué consiste?

**Pruebas guiada por criterios de usuario y si lo he usado en proyectos con Scrum en base a Criterios de aceptación voy creando los escenarios de prueba y luego los paso a features de cucumber.**

1. ¿Podrías poner un ejemplo de un comportamiento usando BDD en texto plano?

**Scenario : Pruebas portal comptras**

**Given : Hacer primera compra**

**When: Ingreso al portal y selecciono articulo**

**Then Compro lo seleccionado**

**El ejemplo práctico esta hecho con BDD gracias a cucumber (feature-java-runner)**

1. ¿Has realizado Automation para Mobile? ¿Con qué herramientas?

**No, solo temas personales con Appium pero no en proyectos**

# Práctico

## Web Automation:

Utilizando Selenium desarrolle un test que tenga como objetivo navegar a la página

de MercadoLibre, buscar “autos” y generar un archivo de texto que contenga:

* Los nombres de artículos.
* Los links para acceder a ellos.

Se valorará:

* La creación de un framework.
* El uso de Page Objects.
* Buenas prácticas de programación.

El entregable debe ser el framework o proyecto construido.

## API Automation:

La tienda de mascotas “***PerfDog***” se encuentra desarrollando una nueva versión para su sitio web.

Como los propietarios de la tienda saben que no hay mejor manera de vender una mascota que teniendo un sitio web de calidad, deciden que deben realizar pruebas sobre el mismo.

Dado que el front end no está terminado se decide comenzar a probar las funcionalidades utilizando la API de la aplicación, que será entregada al final del próximo sprint.

La tienda nos provee la documentación de la API para que podamos crear las pruebas:

<https://petstore.swagger.io>

Para esta primera versión se nos pide validar que el siguiente escenario va a funcionar correctamente:

* Paso 1 - Login
* Paso 2 - Listar mascotas disponibles
* Paso 3 - Consultar datos de una mascota disponible
* Paso 4 - Crear una orden para una mascota
* Paso 5 - Actualizar Usuario
* Paso 6 - Logout

Para ellos se pide crear una colección de Postman que contenga las invocaciones necesarias para completar dicho flujo.

Se pide agregar al menos un test por invocación ([aquí](https://blog.getpostman.com/2014/03/07/writing-automated-tests-for-apis-using-postman/) se puede ver como).