

Reto final BootCamp Automatizacion QA-Sophos

Presentado por:

Yasser Leonardo Pacheco Cañizares

Presentado a:

Sophos Solutions S.A.S

Sophos Solutions S.A.S

Colombia

2022

Prueba General BootCamp

1. Las siguientes preguntas por favor responderlas con sus propias palabras y/o ejemplos.

Preguntas sobre de Java.

- ¿Qué versión de java ha trabajado?

Rta: He trabajado con la versión de Java 8 y el JDK 1.8 en el desarrollo de este BootCamp.



- ¿Qué es Java Lang?

Rta: Java Lang es un paquete de la biblioteca de Java que contiene varias clases que dan funciones básicas en Java como por ejemplo operaciones matemáticas básicas, manejar tipos primitivos de datos, concatenar cadenas de texto entre otras cosas. Este paquete es importado automáticamente en todos los programas de Java por esto no es necesario importarlo de forma explícita por el programador. También incluye la clase object que es la base de todos los objetos en Java.

En relación a las clases estas están en el paquete de java.lang destacando algunas principales como: Boolean, Double, Float, Integer, Math, String, System.

- ¿Qué entiende por polimorfismo?

Rta: El polimorfismo en Java es una característica que permite a una clase o método tener diferentes significados o comportamientos según el contexto en el que se utilice. Gracias a esta característica los programadores pueden crear un código más flexible y reutilizable.

Por ejemplo, en Java se puede utilizar el operador de suma (+) para sumar dos números enteros o dos números decimales, pero también se puede utilizar para concatenar dos cadenas de texto. En este caso, el operador de suma tiene un significado diferente en función del tipo de datos que se utiliza.

- Diferencia entre una clase y un objeto

Rta: La principal diferencia entre una clase y un objeto en Java es que una clase es una plantilla o un molde que se utiliza para definir un tipo de datos, mientras que un objeto es una instancia de esa clase.

Un ejemplo sencillo para entender la diferencia entre una clase y un objeto, sería: Supongamos que se tiene una clase llamada "Perro" que define un tipo de datos para representar a un perro. La clase "Perro" tendrá atributos como el nombre, la raza, la edad del perro, y métodos como ladrar() y comer().

Luego, cuando se quiera crear un tipo de raza de perro en específico, se puede crear un objeto de la clase "Perro" y asignarles valores a sus atributos. Por ejemplo, se puede crear un objeto "miPerro" de la clase "Perro" y asignarle el nombre "Lucas", la raza "Chihuahua" y la edad "5 años". Así, "miPerro" sería un objeto concreto que representa a un perro específico.

- Que es herencia

Rta: La herencia en Java es una característica que permite a una clase (llamada clase hijo o subclase) heredar los atributos y métodos de otra clase (llamada clase padre o superclase). Esto permite a los programadores crear una jerarquía de clases y aprovechar el código existente en lugar de tener que escribirlo de nuevo.

Por ejemplo, si se tiene una clase "Animal" que define atributos común para todos los animales, como el peso y la edad, y un método "mover()" que se implementa de forma genérica, se puede crear una clase "Perro" que herede de "Animal" y tenga atributos y métodos específicos para los perros, como la raza y el método "ladrar()". De esta forma, la clase "Perro" tendrá acceso a todos los atributos y métodos de la clase "Animal" y podrá aprovechar el código existente en lugar de tener que escribirlo de nuevo.

- Palabras claves para encapsulamiento y sobrecarga

Encapsulamiento: Get, set, private, protected, public.

Con el encapsulamiento podemos proteger la información contenida en una clase asegurándonos que solo pueda ser modificada de forma controlada.

El encapsulamiento en Java se utiliza para proteger la información contenida en una clase y asegurar que solo pueda ser modificada de forma controlada. Por ejemplo, si se tiene una clase "CuentaBancaria" que tiene un atributo "saldo" que almacena el saldo actual de una cuenta bancaria, se puede definir el atributo "saldo" como "private" y solo permitir que sea modificado mediante métodos específicos como "depositar()" o "retirar()". De esta forma, se puede asegurar que el saldo de la cuenta solo sea modificado de forma controlada y no sea posible modificarlo directamente desde otras partes del código.

Sobrecarga: overload

Con la sobrecarga damos la capacidad a una clase de tener múltiples métodos con el mismo nombre, pero con diferentes parámetros, permitiendo a los programadores crear métodos con diferentes comportamientos según los parámetros que reciban. La palabra clave para implementar la sobrecarga en Java es "overload".

Por ejemplo, si se tiene una clase "Calculadora" que tiene un método "sumar()" que recibe dos parámetros enteros, se puede sobrecargar el método "sumar()" para que también reciba dos parámetros de tipo flotante o dos parámetros de tipo cadena. De esta forma, se pueden utilizar diferentes versiones del método "sumar()" según los parámetros que reciban.

- ¿Qué es programación orientada a objetos?

Rta: Es una forma de programar específica porque se organiza el código en clases en las cuales se crean objetos que se relacionan entre sí y permiten lograr los objetivos de las aplicaciones o programas además utiliza 4 pilares fundamentales que son el polimorfismo, herencia, abstracción y encapsulamiento.

Como una de las ventajas principales permite a los programadores crear código reutilizable y organizado, porque las clases pueden heredar atributos y métodos de otras clases y se pueden crear múltiples objetos de una misma clase.

Preguntas Sobre automatización

- ¿Qué es SOLID?

Es una sigla en inglés el cual se compone de 5 principios de la programación orientada a objetos y de buenas prácticas que se deben tener como base antes de proponer una arquitectura de software para el desarrollo de nuestras aplicaciones.

- Para que sirve SOLID

Se utilizan para mejorar la calidad del software y hacerlo más fácil de mantener y ampliar. Al seguir los principios de SOLID, los programadores pueden crear código que sea más reutilizable, organizado, fácil de entender y flexible. Esto permite ahorrar tiempo y recursos en el desarrollo y el mantenimiento del software, y facilita la colaboración entre diferentes equipos de desarrollo.

- Que Significa cada letra de SOLID

S: Single Responsibility (Responsabilidad única)

O: Open/Closed (Abierto/Cerrado)

L: Liskov substitution (Sustitución de Liskov)

I: Interface segregation (Segregación de interfaz)

D: Dependency inversion (Inversión de la dependencia)

- Que patrones de diseño de automatización conoce

Screenplay aprendido en el desarrollo de este Bootcamp.

- ¿Qué es un xpath relativo?

Es un tipo de localizador que nos permite navegar sobre los elementos presentes haciendo referencia al que se desea e ir desde allí y podemos comenzar desde el medio de la estructura HTML y no es necesario escribir la ruta larga, especificando el camino desde un nodo en el documento hasta el nodo que se desea seleccionar.

Xpath relativo: `//img[@alt='Objeto a mapear']`

- ¿Qué es un xpath absoluto?

Este tipo de xpath nos permite encontrar de forma directa el elemento, pero no es recomendable usarlo o se debe tener como ultima alternativa, pues no es estable y varia con cualquier cambio en la página. Especifica el camino desde el nodo raíz del documento hasta el nodo que se desea seleccionar

Xpath absoluto: /html/body/div/div/div/article[7]/div/div/a/img

- Herramientas que ha usado para automatizar

Cucumber y Selenium usada en el desarrollo de este Bootcamp.

- Lenguajes que ha usado para automatizar

Java usado en el desarrollo de este Bootcamp.

- ¿Para que se usa los tags?

Los tags son las etiquetas que nos permiten organizar el contenido de una página web además con ellas podemos mostrar un texto, imagen, formulario, tabla y demás elementos presentes en la página y que se asignan a las task utilizadas en las pruebas de automatización a realizar en dichas páginas.

- Cómo realizar un feature para buscar en Google

Crear un file que tenga la extensión en.feature, esto se puede hacer en IntelliJ IDEA

Darle el nombre al Feature

Feature: Búsqueda en Google

Hacer una descripción del paso a paso de la estructura siendo así:

Yo como...

Necesito...

Para...

Quedando de esta forma:

Yo como usuario de Google

Necesito poder buscar en Google

Para encontrar información relevante

Ahora se crea un Scenario con las precondiciones que tendrá la estructura

Scenario: Buscar una cadena de texto en Google

Given que estoy en la página de Google

When busco la cadena de texto "java" en Google

Then se me muestran los resultados de la búsqueda

- ¿Qué frameworks ha utilizado para automatizar?

Selenium utilizado en el desarrollo de este Bootcamp.

- ¿Cuál es la diferencia entre BDD, cucumber y gherkin?

BDD es un enfoque de desarrollo de software que se enfoca en identificar de forma clara el comportamiento deseado de una función, escribiendo las pruebas antes de escribir el código fuente.

Gherking al ser un lenguaje entendible por humanos y ordenadores, por medio de este describimos funcionalidades, definiendo el comportamiento del software y resolviendo un problema

Cucumber es un framework que permite como herramienta a utilizar para automatizar las pruebas BDD como tal. Ejecutando descripciones funcionales en el gherkin como pruebas de software automatizadas.

Preguntas sobre Pruebas

- ¿Qué es calidad de pruebas?

Son los procesos que al ejecutarlos nos permitirá conocer la calidad del software que estamos revisando. Para la revisión de calidad del producto contamos con varios tipos de pruebas como son de funcionalidad, rendimiento, compatibilidad, usabilidad, fiabilidad, seguridad, mantenibilidad y portabilidad.

- ¿Para qué sirve la calidad de pruebas?

Para detectar fallas de los datos, de las interfaces y relaciones entre componentes, también en la implementación de requerimientos para cubrirlos además de detectar fallas en la implementación del sistema.

- ¿Qué es la pirámide de pruebas?

Es un concepto para estructurar los test a realizar por medio de una variedad de niveles para cada tipo. Teniendo tests: unitario, integración, funcional, sistema y aceptación.

- ¿Qué casos se deben automatizar?

La decisión de automatizar o no un caso de prueba depende de varios factores, como el costo y el tiempo que se requiere para automatizar la prueba, la frecuencia con la que se ejecuta la prueba, y el valor que aporta la automatización de la prueba en términos de eficiencia y confiabilidad.

En general, se puede decir que es conveniente automatizar los casos de prueba que se ejecutan con frecuencia y que son repetitivos, ya que la automatización permite ahorrar tiempo y esfuerzo en la ejecución de estas pruebas. También es conveniente automatizar los casos de prueba que son críticos para el sistema, ya que la automatización permite asegurarse de que estos casos de prueba se ejecutan de forma consistente y confiable.

2. Indique los pasos para versionar en GitHub.

1. Se crea una carpeta que contendrá el proyecto quedando así:

```
$ mkdir reto-git
```

2. Luego se ingresa a la carpeta del proyecto creada

```
$ cd reto-git
```

```
$ git init
```

3. Se crea un archivo index.html

```
$ touch index.html
```

4. Se realiza un seguimiento a los cambios que hemos hecho al agregar el fichero index.html

```
$ git add. Así añadimos los cambios a guardar.
```

5. Para guardar estos cambios agregaremos un commit a ejecutar

```
$ git commit -m "add version 1.1 index.html"
```

Prueba General Practica

Nota: Tenga en cuenta que el proyecto debe estar desarrollado en java con Gradle y Patrón screenplay, y reporte con serenity reports. Al finalizar el proyecto se debe crear un .zip el cual no debe de pesar más de 30 (MB), No olvidar aplicar buenas prácticas.

Automatizar la página de: <https://demoqa.com/>

Esta prueba consistirá en automatizar tres flujos:

1. Se debe de ingresar a la opción elements->web tables y agregar un nuevo registro y que me permita eliminar cualquier registro que este en la tabla, verificar que quede ingresado el registro.
2. Se debe ingresar a la opción Widgets->Date Picker y seleccionar cualquier fecha en las opciones (Select Date y Date And Time).
3. Se debe ingresar a la opción Alerts,Frame & Windows->Alerts se debe de interactuar con las 4 tipos de ventanas emergentes tanto para activarlas como para cerrarlas, verificar que en el último aparezca el nombre.

Nota: Se debe de utilizar un log o reporte donde se pueda evidenciar los resultados de las ejecuciones.

La automatización debe poder leer los datos (desde un archivo plano u otro medio NO quemado en el código) y que permita varias ejecuciones.

Nota: Algunos códigos se dejaron comentareados como evidencia de diferentes maneras utilizadas para el desarrollo del reto.

A continuación, adjunto capturas de pantalla de los test pasados en las pruebas de automatización.

