NOMBRE: Luis Hoyos\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Caso de negocio**:

La empresa de producción de prendas de vestir “**New experience**” está desarrollando una solución para la ampliación de sus canales de ventas, los clientes podrán comprar los productos mediante una plataforma virtual (Tienda virtual), en el que seleccionarán las diferentes prendas de temporada, agregarlas a su compra, navegar las diferentes secciones, registrarse, detallar sus datos para el envió de sus productos a domicilio (es posible que sea diferente para cada compra), tener una vista previa de los productos seleccionados y la posibilidad de pagar en línea. Es necesario registrarse para probar algunas funcionalidades, como por ejemplo completar los flujos de la compra.

Acuden a usted para realizar las pruebas ya que el responsable de este no está disponible. Asimismo, le entregan los diferentes criterios que se tienen que cumplir:

Ruta de la aplicación: <http://automationpractice.com/index.php>

1. **Criterios tácticos:**
2. Determine que debe probar (Para esto puede explorar la sección como guste y parta de la condición de que existen errores) para asegurar que se cumple con el mínimo de calidad.

Se deben realizar las siguientes pruebas a la tienda virtual:

1. Registro de usuarios en la tienda virtual
2. Iniciar sesión en la tienda virtual con un usuario registrado
3. Recuperar contraseña
4. Agregar productos al carrito de compras
5. Eliminar productos del carrito
6. Buscar productos en la tienda por medio del buscador
7. Filtros de búsqueda por atributos
8. Agregar nuevas direcciones de envío
9. Pagar un pedido
10. Descargar facturas de pedidos realizados
11. Ver lista de deseos
12. Diseño Responsive de la tienda
13. Funcionamiento de la tienda en diferentes navegadores
14. Considere la creación de casos de prueba. (Considerar pasos si es necesario)
15. De estos casos, clasifique que casos son automatizables y no automatizables.
16. Indique si existen y cuáles son las precondiciones para ejecutar los casos de prueba.
17. Presente estadísticas de prueba que permitan describir la conclusión de un indicador que defina.
18. Sugiera mejoras al módulo probado.

Se sugieren las siguientes mejoras al módulo:

1. Reorganización de la barra de navegación ubicada en el header de la página
2. Implementar botón para volver al home de la tienda
3. Implementar botón para acceder a la lista de deseos desde cualquier página de la tienda
4. Crear sección de descuentos de la tienda
5. Modificar tallas de productos agregados al carrito
6. Modificar label de porcentajes de descuento ofrecidos
7. **Criterios técnicos:**
8. Realice la automatización (programación) de los casos de prueba que seleccionó como automatizables. Se requiere implementar el uso de las siguientes Herramientas:
   * + Lenguaje: Java
     + Tipo de proyecto: Deseable Gradle
     + Framework de ejecución: Junit
     + Framework de pruebas: Serenity/Selenium
     + Driver o navegador: Chrome
     + Arquetipo: Page Object Model o Screenplay
     + Escribir HU en: Cucumber
     + Ejecución: Junit o Jenkins Local
9. Realice la ejecución de lo automatizado, con sus respectivas evidencias.
10. Colocar el código automatizado en **github** y enviarlo. (Repositorio público)
11. **Agilismo**
12. De las sugerencias de mejora propuestas, defina historias de usuario que plantearía para un nuevo desarrollo, con sus respectivos criterios de aceptación.
13. Teniendo en cuenta que estas historias de usuario se seleccionaron en un sprint, liste que ***tareas de prueba y de desarrollo*** son necesarias realizar estas historias.
14. **Conceptos Programación / CI / CD**

Describa con sus propias palabras cada uno de los siguientes conceptos:

1. **Principios SOLID****:** Son los cinco principios básicos y necesarios de la POO y el diseño de un software, estos principios son los siguientes:
2. **S-Single Responsibility:** Establece que cada clase debe responsable de una única cosa
3. **O-Open/Closed:** Las entidades que hacen parte del software se deben permitir extender si el software necesita crecer y cerrar cuando requiera modificaciones, es decir que cuando debe crecer no afecte otros segmentos del software y de cierta forma se pueda hacer reutilizable.
4. **L-Liskov substitution:** El software debe tener la capacidad de que una subclase se pueda sustituir por su clase padre.
5. **I-Interface segregation:** Ningún cliente puede tener dependencias de métodos del software que no utiliza
6. **D-Dependency integration:** No deben existir dependencias entre los módulos de alto y bajo nivel.
7. **Patrón Singleton:** Es un patrón de diseño de software que se utiliza en caso de que se necesite instanciar únicamente un objeto de una clase.
8. **Patrón FIRST:**
9. **Patrón AAA:** Es un patrón de las Unit Testing (Pruebas unitarias) que se utiliza para dividir dichas pruebas en tres pasos, Inicialización, Actuación y Confirmación. Se llama patrón AAA por sus siglas en inglés Arrange/Act/Assert
10. **Pull Request:** Es una solicitud que se realiza para que el dueño de un repositorio autorice cambios que se realizaron o se están proponiendo después de realizar un fork a la rama del dueño, para que sean incluidos en la rama original.
11. **Release Train:** Es una forma de planificar las entregas que se van a realizar de un software, donde se debe especificar de forma exacta las fechas de las entregas
12. **Quality Gates:** Es una aplicación que se enfoca en el monitoreo de calidad de los datos que se manejen en ella, por medio de reglas para validar los datos, pruebas con sus respectivos escenarios planteados y el análisis de los procesos realizados por medio de dashboard.
13. **Diferencias servicios SOAP / REST:** Los servicios SOAP son Webservices que solo soportan intercambio de datos en formato XML, además que son servicios más robustos que los servicios API REST, mientras que los servicios REST se comunican por medio del protocolo HTTP pero de forma más diversa utilizando los métodos GET, POST, PUT, DELETE y a su vez en los response se pueden obtener los códigos de respuesta estándar de HTTP. También son más ligeros con respecto a los servicios SOAP, además que nos da la posibilidad de utilizar diferentes tipos de archivos como JSON.

¡Este es un reto! Por ende, puede consultar toda la información que sea necesaria u otra opción es preguntar las dudas que tenga a personas conocidas.