

**NOMBRE: Airlenis Gaitán****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 envío 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>

**I. Criterios tácticos:**

1. Determine que debe probar (Para esto puede explorar la sección como guest y parte de la condición de que existen errores) para asegurar que se cumple con el mínimo de calidad.
  - Registro
  - Inicio de Sesión
  - Agregar artículos al carrito
  - Eliminar artículos del carrito
  - comprar
2. Considere la creación de casos de prueba. (Considerar pasos si es necesario)
3. De estos casos, clasifique que casos son automatizables y no automatizables.
  - Registro (NO AUTOMATIZABLE)
  - Inicio de Sesión (AUTOMATIZABLE)
  - Agregar artículos al carrito (AUTOMATIZABLE)
  - Eliminar artículos del carrito (AUTOMATIZABLE)
  - Comprar (NO AUTOMATIZABLE)
4. Indique si existen y cuáles son las precondiciones para ejecutar los casos de prueba.
  - Registro
    - Que el correo nuevo a registrar no se encuentre en uso por otro usuario y que sea un correo valido
  - Inicio de Sesión
    - Que este creado el usuario para realizar el ingreso al sitio
  - Agregar artículos al carrito
    - Que exista disponibilidad de los artículos deseados
  - Eliminar artículos del carrito
    - Que existan artículos no deseados en el carrito de compras
  - Comprar
    - Que el carrito de compras tenga al menos un artículo agregado.



5. Presente estadísticas de prueba que permitan describir la conclusión de un indicador que defina.
6. Sugiera mejoras al módulo probado.
  - Registro
    - Mejora sugerida: al finalizar el proceso de registro, mostrar una pantalla de registro exitoso y enviar un correo electrónico a la persona para notificarle el registro en el portal.
  - Inicio de Sesión
    - Considerar colocar un id o elemento reconocible luego del inicio de sesión para facilitar la ejecución de la prueba.
  - Agregar artículos al carrito
  - Eliminar artículos del carrito
  - comprar

## II. Criterios técnicos:

1. 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
  - Driver o navegador: Chrome
  - Arquetipo: Page Object Model o Screenplay
  - Escribir HU en: Cucumber
  - Ejecución: Junit o Jenkins Local
2. Realice la ejecución de lo automatizado, con sus respectivas evidencias.
3. Colocar el código automatizado en **github** y enviarlo. (Repositorio público)

## III. Agilismo

1. De las sugerencias de mejora propuestas, defina historias de usuario que plantearía para un nuevo desarrollo, con sus respectivos criterios de aceptación.
  - Yo como nuevo usuario del portal de compras, me gustaría recibir un correo notificándome mi registro exitoso.
2. 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.
  - Tareas desarrollo:
    - Definir un servicio que genere un correo electrónico automático informando al nuevo usuario su registro.
  - Tareas de pruebas:
    - Validar que en efecto luego del registro en la bandeja de entrada del correo suministrado, se encuentre el mensaje de registro exitoso.



**IV. Conceptos Programación / CI / CD**

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

1. Principios SOLID
2. Patrón Singleton
3. Patrón FIRST
4. Patrón AAA
5. Pull Request
6. Release Train
7. Quality Gates
8. Diferencias servicios SOAP / REST

**Principios SOLID:** Los principios SOLID son un conjunto de buenas prácticas recomendadas para desarrollar un software, ya que con este logramos obtener un código fácil y reutilizable, entendible para nosotros, así como para los desarrolladores que lleguen después. Este principio nos permite abarcar la fase de mantenimiento de una manera más sencilla, así como conseguir crear nuevas funcionalidades sin afectar lo que se tenga.

**Patrón Singleton:** este patrón permite restringir la creación de los objetos pertenecientes a una clase, se utiliza en caso de que el sistema tenga una clase que solo debe ser instanciada una vez o que la misma requiera un punto de acceso global o conocido.

**Patrón AAA:** el patrón AAA se organiza de tres formas: la primera es Arrange(organización o inicialización) en esta se crean las variables y se asignan los valores que se van a utilizar durante el test a realizar, luego continúa Act(actuar) en esta se realiza el llamado al método con los parámetros que ya se habían establecidos y por último está el Assert(confirmar o comprobar) con este se valida que el método está haciendo la acción que se tenía prevista que hiciera. con este patrón podemos lograr que nuestros test sean entendibles para cualquier desarrollador.

**Pull Request:** entiendo por pull request el proceso mediante el que un desarrollador que termina de crear y probar alguna nueva funcionalidad o alguna corrección de algo ya existente solicita al resto del equipo revisar el desarrollo y proceder a la integración del mismo al repositorio principal.

**Quality Gates:** es un formato de inspección que nos ayudan a mejorar la calidad de los procesos mediante el monitoreo de las fallas encontradas. Esto nos da la oportunidad de tomar mejores decisiones.

**Diferencia servicios SOAP/REST:**

Soap es un protocolo mientras que REST es un estilo arquitectónico, además SOAP como protocolo define las normas que se deben seguir estrictamente y su propia seguridad en cambio REST no define tantos estándares y hereda medidas de seguridad.

El protocolo SOAP solo permite formatos de datos XML mientras que el REST permite diferentes formatos como por ejemplo HTML, XML, JSON entre otros.

¡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.



Evaluación de pruebas candidatos Choucair  
*Tipo 1 opción 1*

