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.

<table>
<tr><th>N°</th><th>Nombre</th><th>Nombre del proyecto</th></tr>
<tr><td>N-001</td><td>Realizar compras sin registro</td><td>ChoucauirTest</td></tr>
<tr><th colspan="2">Módulo:</th><th>Submódulo:</th></tr>
<tr><td colspan="2">Ventas</td><td>Online</td></tr>
<tr><th colspan="3">Texto Descriptivo</th></tr>
<tr><td colspan="3">Yo como usuario deseo realizar compras de las diferentes prendas de la temporada sin tener que registrarme en la página para no asociarme a la plataforma.</td></tr>
<tr><th colspan="2">Datos de Entrada</th><th>Datos de salida</th></tr>
<tr><td colspan="2">• Datos de los ítems seleccionados (Cantidad, color, tamaño)
• Datos del comprador
o Nombres String Max 15 car
o Apellidos, String Max 15 car
o Email, Varchar Max 50 car
o Tipo de documento
o Número de documento
• Dirección de envío
• Lista de operadores logísticos
• Datos de método de pago seleccionado</td><td>• Resumen de compra indicando:
o Nombre y apellido del comprador
o Método de pago
o Número de pago
o Estado de pago
o Detalle de los ítems comprado
o Cantidad total de ítems comprados
o Número de factura
o Valor unitario de cada ítem comprado
o Costo del delivery.
o Valor del IVA
o Costo total
o Dirección de entrega
o Nombre del operador logístico
o Fecha estimada de entrega
o Fecha y hora de la transacción</td></tr>
<tr><th colspan="3">Criterios de aceptación</th></tr>
<tr><td colspan="3">• Todos los datos de entradas son de carácter obligatorio
• En la opción tipo de documento debe ser una lista desplegable de selección única donde se encuentre las opciones de:
o Cedula de ciudadanía, int, máximo máx 10 caracteres min 8 caracteres.
o Cedula de extrangeria, int, máximo máx 7 caracteres min 5 caracteres.
o Pasaporte, varchar, máximo 10 caracteres min 8 caracteres.
• Los valores de Nombres y apellidos no pueden tener valores numéricos ni caracteres especiales.
• El sistema debe mostrar la opción de compra online sin registro.
• El sistema debe realizar en tiempo real una consulta en base de datos y mostrar en pantalla la disponibilidad del producto seleccionado por talla y color de los ítems seleccionados.</td></tr>
</table>

- Si el usuario excede la cantidad disponible de un ítem el sistema debe arrojar un mensaje en pantalla indicando que excede la cantidad disponible y no podrá avanzar en la compra.
- La lista de operadores logísticos debe ser de selección única
- La lista de operadores logísticos debe tener la opción de retiro en tienda.
- Si el usuario elige la opción de retiro en tienda, en los datos de salida las opciones de Dirección de entrega, Nombre del operador logístico y Fecha estimada de entrega mostrará “No aplica”.
- El usuario debe ingresar la dirección de entrega, sólo si se seleccionó algún operador logístico, de lo contrario el campo queda deshabilitado.
- El sistema debe generar una alerta al usuario si el operador logístico no tiene cobertura en la dirección ingresada, sólo si se seleccionó algún operador logístico, de lo contrario el campo queda deshabilitado.
- Si el operador seleccionado por el usuario no tiene cobertura, el sistema debe mostrar la opción de seleccionar otro operador y la opción de retiro en tienda.
- El sistema debe mostrar una fecha estimada de entrega, sólo si se seleccionó algún operador logístico, de lo contrario el campo queda deshabilitado.
- El sistema realizara un resumen de la transacción detallando todos los datos de salida.
- El usuario debe seleccionar método de pago
- Los métodos de pagos pueden ser Tarjeta de crédito o PSE.
- El sistema debe redireccionar al usuario a la página del método de pago seleccionado.
- El Usuario debe recibir un email con el resumen de la compra
- Sistema debe tener la opción de consultar al usuario si desea recibir notificaciones de promociones, descuentos y nuevos productos en venta.
- El sistema debe volver a el homepage una vez se haya confirmado la compra.

N°	Nombre	Nombre del proyecto
N-002	Filtros opciones avanzadas en el HomePage	ChoucauirTest

Módulo:	Submódulo:
Ventas	Online

Texto Descriptivo

Yo como **Sistema** deseo **Mostrar un filtro de artículos de opciones avanzadas** para poder **mostrar ítems por rangos de precios, temporada, tallas, color entre otros.**

Datos de Entrada	Datos de salida

• Filtro de los articulos por:
  ○ precios (Min y max).
  ○ Colecciones
  ○ Temporadas
  ○ Lo más vendido
  ○ En descuento
	• Artículos que cumplan con la condición del filtro seleccionado.

Criterios de aceptación

- El sistema debe poseer un paginador para mostrar los ítems que coincidan con la búsquedas de artículos.
- El paginador debe ser de máximo 8 artículos por vista.
- Se debe poseer una opción de limpiar filtro.

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 de desarrollo	Tareas de QA
Implementar cambios necesarios en la Base de datos	Elaborar test cases
	Generación de data
Implementar API´s mockeadas	Verificación de nivel de cobertura de pruebas unitarias
Implementar API´s integradas	Elaborar scripts de automatización frontend (BDD)
Implementar backend (TDD)	Elaborar scripts automatización backend (Automatización de servicios API's)
Implementar diseño (Front-end)	Elaborar pruebas de carga, estres
Ejecutar pruebas unitarias	Ejecutar scripts de pruebas automatizadas de frontend
Verificar nivel de cobertura exigido	Ejecutar scripts de pruebas automatizadas de backend
Subir al repositorio	Ejecutar scripts de pruebas carga,estrés
	Ejecución pruebas manuales
Aprobar Pull Request	Generación de reportes de pruebas manuales
Desplegar en ambiente de QA	Generación de reportes de pruebas automatizadas de frontend
	Generación de reportes de pruebas automatizadas de backend
	Generación de reportes de pruebas de carga, estres
	Subir al repositorio scripts pruebas automatizadas frontend
	Subir al repositorio scripts pruebas automatizadas backend
	Subir al repositorio scripts pruebas carga estres
	Ejecutar scripts de pruebas automatizadas para regresión de frontend
	Ejecutar scripts de pruebas automatizadas para regresión de backend
	Ejecutar scripts de pruebas de regresión para pruebas de carga estres
	Aprobar pull request scripts automatizadas frontend
	Aprobar pull request scripts automatizadas backend
	Aprobar pull request scripts automatizadas carga estres
	Generar documentación de certificación
	Generar certificación

Nota: Es importante destacar que en el cuadro anterior donde se listan las tareas que debe ejecutar el área de desarrollo y QA, no están relacionadas directamente entre ellas… Es decir, cuando desarrollo este ejecutando la primera tarea listada, no necesariamente QA va ejecutar la primera tarea listada en su sección. Al igual que el orden entre las tareas de cada área no es sucesivo, todo esto depende de la estrategia de pruebas y desarrollo que se haya definido desde el inicio del proyecto.

IV. Conceptos Programación / CI / CD

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

1. **Principios SOLID:**

Single Responsibility Principle (Responsabilidad única): cada clase que se desarrolla ya sea una tarea, interacción, una pregunta debe realizarse para ejecutar una función en específica, sin embargo, por buenas practicas de desarrollo deben ser reutilizables para no repetir código.

Open Closed Principle (Principio abierto cerrado): los módulos en nuestro desarrollo deben ser abiertos para apartarse al cambio que puedan presentarse durante el proyecto, sin embargo, debemos diseñarlos de la manera para que los cambios se puedan realizar en determinadas instancias o módulos de manera que nuestro código no sea afectado.

Liskov Substitution Principle (Principio de sustitución de Liskov): Se basa en la abstracción de las clases los suficientemente bien diseñado como para poder ser reutilizada la clase padre en las clases hijas y no se vean afectadas las clases hijas en cuanto a su resultado al momento de implementarlas.

Interface Substitution Principle (Principio de sustitución de interfaz): este principio nos habla de que se debe definir cada interfaz de manera individual cuando estamos desarrollando nuestro proyecto, de esta manera podemos vincularla con la tarea de necesite implementarla.

Dependency Inversion Principle(Principio de inversión de dependencia), habla de que las clases de alto nivel no deberían depender de las clases de bajo nivel, es decir, las interacciones ejecutadas por tareas y las tareas por Steps.

2. Patrón Singleton:

Es utilizado para garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella. No se encarga de la creación de objetos en sí, sino que se enfoca en la restricción en la creación de un objeto.

3. Patrón FIRST

Los tests unitarios para ser eficientes deben ser, Fast (rápido), Independent (independiente), Repeatable (repetible), Self-validating (auto evaluable), Timely (oportuno).

4. Patrón AAA

Se estructura de la siguiente manera: **Arrange** (Organizar/Inicializa), **Act** (Actuar) **Assert (Confirmar/Comprobar).** Generalmente es aplicado a pruebas unitarias.

5. Pull Request: es la solicitud que hace un desarrollador para que su código sea incorporado (Merge) a el proyecto en el que se está trabajando

6. Release Train: es un enfoque para alinear la visión, la planificación y las interdependencias de muchos equipos al proporcionar sincronización entre equipos basada en una cadencia común.

7. Quality Gates: Son los las reglas o condiciones que establece la organización para determinar que un proyecto cumple con los estándares de calidad de software deseados, para la puesta en producción del producto.

8. Diferencias servicios SOAP / REST
SOAP: responde con un XML.

REST: responde con un JSON.