**REQUERIMIENTOS**

***1.- Escribe un prompt que te ayude a generar los requerimientos funcionales y no funcionales de tu proyecto de software***

**Prompt**

Eres un experto en pruebas de software. Tienes certificaciones ISTQB. Vas a realizar los requerimientos funcionales y no funcionales para el proceso de venta en la página de Home Depot (<http://www.homedepot.com.mx> )

**Requerimientos Funcionales**

1. **Registro y autenticación de usuarios**:
   * Creación de cuentas de usuario y autenticación segura.
   * Recuperación de contraseña y verificación de correo electrónico.
2. **Búsqueda y navegación de productos**:
   * Búsqueda avanzada con filtros por categoría, precio, marca, etc.
   * Navegación intuitiva por categorías y subcategorías.
3. **Visualización de productos**:
   * Detalles del producto, incluyendo imágenes, descripción, especificaciones técnicas y reseñas de clientes.
   * Disponibilidad de productos en diferentes tiendas y opciones de envío.
4. **Carrito de compras**:
   * Agregar y eliminar productos del carrito.
   * Actualización de cantidades y visualización del total de la compra.
5. **Proceso de pago**:
   * Integración con múltiples métodos de pago (tarjetas de crédito, débito, PayPal, etc.).
   * Seguridad en las transacciones y almacenamiento de datos sensibles.
6. **Gestión de pedidos**:
   * Confirmación de pedidos y envío de correos electrónicos de seguimiento.
   * Visualización del historial de pedidos y estado de los mismos.
7. **Opciones de envío**:
   * Selección de métodos de envío y cálculo de costos.
   * Seguimiento de envíos y notificaciones de entrega.
8. **Promociones y descuentos**:
   * Aplicación de códigos de descuento y promociones.
   * Visualización de ofertas especiales y productos en promoción.
9. **Servicio al cliente**:
   * Chat en vivo y soporte por correo electrónico.
   * Sección de preguntas frecuentes y políticas de devolución.
10. **Cumplimiento de normativas**:
    * Cumplimiento con las leyes y regulaciones locales e internacionales[[1]](https://visuresolutions.com/es/gu%C3%ADa-de-trazabilidad-de-gesti%C3%B3n-de-requisitos/requerimientos-no-funcionales/).
    * Implementación de prácticas de seguridad y protección de datos.

**Requerimientos No Funcionales**

1. **Rendimiento**:
   * El sitio web debe cargar en menos de 3 segundos para garantizar una experiencia de usuario óptima[[2]](https://www.pmoinformatica.com/2015/05/requerimientos-no-funcionales-ejemplos.html).
   * Capacidad para manejar picos de tráfico durante eventos especiales o promociones.
2. **Seguridad**:
   * Protección contra ataques de seguridad como SQL injection, cross-site scripting (XSS), etc.
   * Encriptación de datos sensibles y cumplimiento con estándares de seguridad.
3. **Usabilidad**:
   * Interfaz de usuario intuitiva y fácil de navegar.
   * Accesibilidad para usuarios con discapacidades, cumpliendo con las normas WCAG.
4. **Escalabilidad**:
   * Capacidad para escalar horizontalmente y verticalmente según la demanda.
   * Soporte para múltiples idiomas y monedas.
5. **Disponibilidad**:
   * Alta disponibilidad con un tiempo de actividad del 99.9%.
   * Implementación de redundancia y recuperación ante desastres.
6. **Mantenibilidad**:
   * Código limpio y documentado para facilitar el mantenimiento y las actualizaciones.
   * Uso de prácticas de desarrollo ágil y pruebas automatizadas.
7. **Compatibilidad**:
   * Compatibilidad con diferentes navegadores y dispositivos móviles.
   * Adaptabilidad a diferentes resoluciones de pantalla.

***2.- Escribe un prompt para generar todos los casos de prueba de aceptación sobre el documento de requerimientos funcionales y no funcionales***

**Prompt**

Eres un experto en pruebas de software. Tienes certificaciones ISTQB. De acuerdo a los requerimientos funcionales realizados para el proceso de venta en la página de Home Depot (http://www.homedepot.com.mx ) realiza todos los casos de prueba que cubran todos los requerimientos funcionales y no funcionales realizados. Los casos de prueba deben tener ID de caso de prueba, nombre de caso de prueba, precondiciones, pasos del caso de prueba y resultado esperado

**Casos de Prueba**

| **ID** | **Nombre del Caso de Prueba** | **Precondiciones** | **Pasos del Caso de Prueba** | **Resultado Esperado** |
| --- | --- | --- | --- | --- |
| F1 | Creación de cuenta de usuario | Ninguna | 1. Navegar a la página de registro 2. Ingresar datos requeridos 3. Enviar formulario | El usuario puede crear una cuenta exitosamente |
| F2 | Autenticación de usuario | Cuenta de usuario creada | 1. Navegar a la página de inicio de sesión 2. Ingresar credenciales 3. Iniciar sesión | El usuario puede iniciar sesión con sus credenciales |
| F3 | Recuperación de contraseña | Cuenta de usuario creada | 1. Navegar a la página de recuperación de contraseña 2. Ingresar correo electrónico 3. Seguir instrucciones del correo | El usuario puede recuperar su contraseña |
| F4 | Búsqueda avanzada de productos | Productos disponibles en la base de datos | 1. Navegar a la página de búsqueda 2. Ingresar criterios de búsqueda 3. Aplicar filtros | El usuario puede buscar productos utilizando filtros |
| F5 | Navegación por categorías | Categorías y subcategorías definidas | 1. Navegar a la página de inicio 2. Seleccionar una categoría 3. Navegar por subcategorías | El usuario puede navegar por categorías y subcategorías |
| F6 | Visualización de detalles del producto | Productos disponibles en la base de datos | 1. Navegar a la página de producto 2. Ver detalles del producto | El usuario puede ver imágenes, descripción y reseñas del producto |
| F7 | Ver disponibilidad de productos | Productos disponibles en diferentes tiendas | 1. Navegar a la página de producto 2. Ver disponibilidad en tiendas | El usuario puede ver la disponibilidad en diferentes tiendas |
| F8 | Agregar productos al carrito | Productos disponibles en la base de datos | 1. Navegar a la página de producto 2. Agregar producto al carrito | El usuario puede agregar productos al carrito |
| F9 | Eliminar productos del carrito | Productos agregados al carrito | 1. Navegar al carrito 2. Eliminar producto del carrito | El usuario puede eliminar productos del carrito |
| F10 | Actualización de cantidades en el carrito | Productos agregados al carrito | 1. Navegar al carrito 2. Actualizar cantidad de productos | El usuario puede actualizar las cantidades de productos en el carrito |
| F11 | Integración de métodos de pago | Métodos de pago configurados | 1. Navegar a la página de pago 2. Seleccionar método de pago 3. Realizar pago | El usuario puede realizar pagos con diferentes métodos |
| F12 | Seguridad en transacciones | Métodos de pago configurados | 1. Navegar a la página de pago 2. Realizar pago | Las transacciones son seguras y los datos sensibles están protegidos |
| F13 | Confirmación de pedidos | Pedido realizado | 1. Realizar pedido 2. Verificar correo de confirmación | El usuario recibe una confirmación de pedido por correo electrónico |
| F14 | Visualización de historial de pedidos | Pedido realizado | 1. Navegar a la página de historial de pedidos 2. Ver historial de pedidos | El usuario puede ver el historial y estado de sus pedidos |
| F15 | Selección de métodos de envío | Métodos de envío configurados | 1. Navegar a la página de envío 2. Seleccionar método de envío | El usuario puede seleccionar diferentes métodos de envío |
| F16 | Seguimiento de envíos | Pedido enviado | 1. Navegar a la página de seguimiento 2. Ver estado del envío | El usuario puede seguir el estado de sus envíos |
| F17 | Aplicación de códigos de descuento | Códigos de descuento disponibles | 1. Navegar a la página de pago 2. Ingresar código de descuento 3. Aplicar descuento | El usuario puede aplicar códigos de descuento en su compra |
| F18 | Visualización de ofertas especiales | Ofertas disponibles | 1. Navegar a la página de ofertas 2. Ver productos en promoción | El usuario puede ver productos en promoción |
| F19 | Acceso al chat en vivo | Servicio de chat configurado | 1. Navegar a la página de soporte 2. Iniciar chat en vivo | El usuario puede acceder al chat en vivo para soporte |
| F20 | Acceso a preguntas frecuentes | Preguntas frecuentes disponibles | 1. Navegar a la página de preguntas frecuentes 2. Ver preguntas frecuentes | El usuario puede acceder a la sección de preguntas frecuentes |
| F21 | Cumplimiento de leyes y regulaciones | Normativas definidas | 1. Navegar a la página de términos y condiciones 2. Verificar cumplimiento | El sitio cumple con las leyes y regulaciones locales e internacionales |
| NF1 | Tiempo de carga del sitio | Ninguna | 1. Navegar a la página de inicio 2. Medir tiempo de carga | El sitio carga en menos de 3 segundos |
| NF2 | Manejo de picos de tráfico | Ninguna | 1. Realizar pruebas de carga 2. Verificar rendimiento | El sitio maneja picos de tráfico sin problemas |
| NF3 | Protección contra ataques | Ninguna | 1. Realizar pruebas de seguridad 2. Verificar protección | El sitio está protegido contra SQL injection, XSS, etc. |
| NF4 | Encriptación de datos sensibles | Ninguna | 1. Navegar a la página de pago 2. Verificar encriptación | Los datos sensibles están encriptados |
| NF5 | Interfaz intuitiva | Ninguna | 1. Navegar por el sitio 2. Verificar usabilidad | La interfaz es fácil de navegar |
| NF6 | Accesibilidad | Ninguna | 1. Navegar por el sitio 2. Verificar accesibilidad | El sitio cumple con las normas WCAG |
| NF7 | Capacidad de escalar | Ninguna | 1. Realizar pruebas de escalabilidad 2. Verificar capacidad | El sitio puede escalar según la demanda |
| NF8 | Soporte para múltiples idiomas | Idiomas configurados | 1. Navegar a la página de configuración 2. Seleccionar idioma | El sitio soporta múltiples idiomas y monedas |
| NF9 | Alta disponibilidad | Ninguna | 1. Realizar pruebas de disponibilidad 2. Verificar tiempo de actividad | El sitio tiene un tiempo de actividad del 99.9% |
| NF10 | Redundancia y recuperación | Ninguna | 1. Realizar pruebas de recuperación 2. Verificar redundancia | El sitio tiene redundancia y recuperación ante desastres |
| NF11 | Código limpio y documentado | Ninguna | 1. Revisar código fuente 2. Verificar documentación | El código es limpio y está bien documentado |
| NF12 | Uso de prácticas ágiles | Ninguna | 1. Revisar proceso de desarrollo 2. Verificar prácticas ágiles | Se utilizan prácticas de desarrollo ágil y pruebas automatizadas |
| NF13 | Compatibilidad con navegadores | Ninguna | 1. Navegar por el sitio en diferentes navegadores 2. Verificar compatibilidad | El sitio es compatible con diferentes navegadores |
| NF14 | Adaptabilidad a dispositivos móviles | Ninguna | 1. Navegar por el sitio en diferentes dispositivos 2. Verificar adaptabilidad | El sitio se adapta a diferentes resoluciones de pantalla |

**PRUEBAS DE ACEPTACIÓN**

***1.- Invalida algunos requerimientos para que se contradigan.***

**Prompt**

De esos requisitos, invalida algunos para que se contradigan

**Requerimientos Funcionales Contradictorios**

1. **Registro y autenticación de usuarios**:
   * **Requerimiento original**: Creación de cuentas de usuario y autenticación segura.
   * **Requerimiento contradictorio**: Permitir acceso sin necesidad de autenticación.
2. **Proceso de pago**:
   * **Requerimiento original**: Integración con múltiples métodos de pago (tarjetas de crédito, débito, PayPal, etc.).
   * **Requerimiento contradictorio**: Solo permitir pagos en efectivo al momento de la entrega.
3. **Gestión de pedidos**:
   * **Requerimiento original**: Confirmación de pedidos y envío de correos electrónicos de seguimiento.
   * **Requerimiento contradictorio**: No enviar ninguna confirmación de pedido ni correos electrónicos.

**Requerimientos No Funcionales Contradictorios**

1. **Rendimiento**:
   * **Requerimiento original**: El sitio web debe cargar en menos de 3 segundos.
   * **Requerimiento contradictorio**: No importa el tiempo de carga del sitio.
2. **Seguridad**:
   * **Requerimiento original**: Protección contra ataques de seguridad como SQL injection, cross-site scripting (XSS), etc.
   * **Requerimiento contradictorio**: No implementar ninguna medida de seguridad.
3. **Disponibilidad**:
   * **Requerimiento original**: Alta disponibilidad con un tiempo de actividad del 99.9%.
   * **Requerimiento contradictorio**: No garantizar ningún tiempo de actividad específico.

***2- Escribe un prompt para que aplique todos los casos de prueba de aceptación que hiciste contra los documentos de requerimientos funcionales y no funcionales.***

**Prompt para Aplicar Casos de Prueba de Aceptación**

**Objetivo**: Validar que los requerimientos funcionales y no funcionales del proceso de venta en la página de Home Depot México se cumplen mediante la ejecución de casos de prueba de aceptación.

**Documentos de Referencia**:

* Requerimientos Funcionales
* Requerimientos No Funcionales

**Casos de Prueba**:

1. **TC001 - Registro de Usuario**
   * **Precondiciones**: La página de registro está accesible.
   * **Pasos**:
     1. Navegar a la página de registro.
     2. Ingresar información válida en los campos requeridos (nombre, correo electrónico, contraseña).
     3. Hacer clic en "Registrar".
   * **Resultado Esperado**: El usuario es registrado exitosamente y se muestra un mensaje de confirmación.
2. **TC002 - Autenticación de Usuario**
   * **Precondiciones**: El usuario está registrado.
   * **Pasos**:
     1. Navegar a la página de inicio de sesión.
     2. Ingresar correo electrónico y contraseña válidos.
     3. Hacer clic en "Iniciar sesión".
   * **Resultado Esperado**: El usuario es autenticado y redirigido a la página principal.
3. **TC003 - Búsqueda de Productos**
   * **Precondiciones**: La página principal está accesible.
   * **Pasos**:
     1. Ingresar un término de búsqueda en la barra de búsqueda.
     2. Hacer clic en el botón de búsqueda.
   * **Resultado Esperado**: Se muestran los productos relacionados con el término de búsqueda.
4. **TC004 - Agregar Producto al Carrito**
   * **Precondiciones**: El usuario está autenticado y en la página de detalles del producto.
   * **Pasos**:
     1. Seleccionar cantidad de producto.
     2. Hacer clic en "Agregar al carrito".
   * **Resultado Esperado**: El producto es agregado al carrito y se muestra un mensaje de confirmación.
5. **TC005 - Proceso de Pago**
   * **Precondiciones**: El usuario tiene productos en el carrito.
   * **Pasos**:
     1. Navegar al carrito de compras.
     2. Hacer clic en "Proceder al pago".
     3. Seleccionar método de pago.
     4. Ingresar información de pago.
     5. Confirmar la compra.
   * **Resultado Esperado**: La compra es procesada exitosamente y se genera un recibo electrónico.
6. **TC006 - Seguridad de la Información**
   * **Precondiciones**: La página está accesible.
   * **Pasos**:
     1. Intentar acceder a información sensible sin autenticación.
     2. Realizar una transacción de pago.
   * **Resultado Esperado**: La información sensible está protegida y la transacción cumple con los estándares de seguridad.
7. **TC007 - Rendimiento de la Página**
   * **Precondiciones**: La página está accesible.
   * **Pasos**:
     1. Medir el tiempo de carga de la página principal.
     2. Realizar búsquedas y navegar entre diferentes páginas.
   * **Resultado Esperado**: La página carga en menos de 3 segundos y maneja picos de tráfico sin problemas.
8. **TC008 - Usabilidad de la Interfaz**
   * **Precondiciones**: La página está accesible.
   * **Pasos**:
     1. Navegar por la página en diferentes dispositivos y navegadores.
     2. Realizar tareas comunes (registro, búsqueda, compra).
   * **Resultado Esperado**: La interfaz es intuitiva y fácil de usar en todos los dispositivos y navegadores.
9. **TC009 - Escalabilidad del Sistema**
   * **Precondiciones**: La página está accesible.
   * **Pasos**:
     1. Simular un aumento en el número de usuarios y transacciones.
   * **Resultado Esperado**: El sistema soporta el aumento sin afectar el rendimiento.
10. **TC010 - Mantenimiento del Sistema**
    * **Precondiciones**: La página está accesible.
    * **Pasos**:
      1. Realizar una actualización del sistema.
      2. Verificar que el servicio no se interrumpa.
    * **Resultado Esperado**: La actualización se realiza sin interrupciones y la documentación es clara.

**DISEÑO DEL SISTEMA**

1. *Escribe un prompt que te ayude a generar los casos de uso posibles sobre el documento de requerimientos funcionales y no funcionales.*

**Prompt**

Eres un experto en pruebas de software con certificaciones ISTQB. Tienes un documento de requerimientos funcionales y no funcionales para el proceso de venta en la página de Home Depot (http://www.homedepot.com.mx). Utiliza este documento para generar los casos de uso posibles. Cada caso de uso debe incluir los siguientes elementos:

1. **Nombre del Caso de Uso**: Un título claro y descriptivo.
2. **Actor Principal**: El usuario o sistema que interactúa con el caso de uso.
3. **Precondiciones**: Las condiciones que deben cumplirse antes de ejecutar el caso de uso.
4. **Flujo Principal**: Los pasos detallados que el actor principal debe seguir para completar el caso de uso.
5. **Resultado Esperado**: El resultado que se espera al completar el caso de uso.

**Resultado**

Caso de Uso 1: Registro de Usuario

* Nombre del Caso de Uso: Registro de Usuario
* Actor Principal: Usuario
* Precondiciones: La página de registro está accesible.
* Flujo Principal:
  1. El usuario navega a la página de registro.
  2. El usuario ingresa información válida en los campos requeridos (nombre, correo electrónico, contraseña).
  3. El usuario hace clic en "Registrar".
* Resultado Esperado: El usuario es registrado exitosamente y se muestra un mensaje de confirmación.

Caso de Uso 2: Autenticación de Usuario

* Nombre del Caso de Uso: Autenticación de Usuario
* Actor Principal: Usuario
* Precondiciones: El usuario está registrado.
* Flujo Principal:
  1. El usuario navega a la página de inicio de sesión.
  2. El usuario ingresa correo electrónico y contraseña válidos.
  3. El usuario hace clic en "Iniciar sesión".
* Resultado Esperado: El usuario es autenticado y redirigido a la página principal.

Caso de Uso 3: Búsqueda de Productos

* Nombre del Caso de Uso: Búsqueda de Productos
* Actor Principal: Usuario
* Precondiciones: La página principal está accesible.
* Flujo Principal:
  1. El usuario ingresa un término de búsqueda en la barra de búsqueda.
  2. El usuario hace clic en el botón de búsqueda.
* Resultado Esperado: Se muestran los productos relacionados con el término de búsqueda.

Caso de Uso 4: Agregar Producto al Carrito

* Nombre del Caso de Uso: Agregar Producto al Carrito
* Actor Principal: Usuario
* Precondiciones: El usuario está autenticado y en la página de detalles del producto.
* Flujo Principal:
  1. El usuario selecciona la cantidad de producto.
  2. El usuario hace clic en "Agregar al carrito".
* Resultado Esperado: El producto es agregado al carrito y se muestra un mensaje de confirmación.

Caso de Uso 5: Proceso de Pago

* Nombre del Caso de Uso: Proceso de Pago
* Actor Principal: Usuario
* Precondiciones: El usuario tiene productos en el carrito.
* Flujo Principal:
  1. El usuario navega al carrito de compras.
  2. El usuario hace clic en "Proceder al pago".
  3. El usuario selecciona el método de pago.
  4. El usuario ingresa la información de pago.
  5. El usuario confirma la compra.
* Resultado Esperado: La compra es procesada exitosamente y se genera un recibo electrónico.

Caso de Uso 6: Ver Historial de Pedidos

* Nombre del Caso de Uso: Ver Historial de Pedidos
* Actor Principal: Usuario
* Precondiciones: El usuario está autenticado.
* Flujo Principal:
  1. El usuario navega a la sección de "Mi cuenta".
  2. El usuario selecciona "Historial de pedidos".
* Resultado Esperado: Se muestra el historial de pedidos del usuario con detalles de cada compra.

Caso de Uso 7: Cancelar Pedido

* Nombre del Caso de Uso: Cancelar Pedido
* Actor Principal: Usuario
* Precondiciones: El usuario tiene un pedido en proceso.
* Flujo Principal:
  1. El usuario navega a la sección de "Mi cuenta".
  2. El usuario selecciona "Historial de pedidos".
  3. El usuario selecciona el pedido que desea cancelar.
  4. El usuario hace clic en "Cancelar pedido".
* Resultado Esperado: El pedido es cancelado y se muestra un mensaje de confirmación.

Caso de Uso 8: Calcular Costos de Envío

* Nombre del Caso de Uso: Calcular Costos de Envío
* Actor Principal: Usuario
* Precondiciones: El usuario tiene productos en el carrito.
* Flujo Principal:
  1. El usuario navega al carrito de compras.
  2. El usuario ingresa su ubicación para calcular el costo de envío.
* Resultado Esperado: Se muestra el costo de envío basado en la ubicación del usuario.

Caso de Uso 9: Recogida en Tienda

* Nombre del Caso de Uso: Recogida en Tienda
* Actor Principal: Usuario
* Precondiciones: El usuario tiene productos en el carrito.
* Flujo Principal:
  1. El usuario navega al carrito de compras.
  2. El usuario selecciona la opción de "Recogida en tienda".
  3. El usuario selecciona la tienda de recogida.
  4. El usuario confirma la compra.
* Resultado Esperado: La compra es procesada exitosamente y se genera un recibo electrónico con detalles de la recogida en tienda.

*2-Escribe un prompt que te ayude a generar las interfaces de usuario posibles sobre el documento de requerimientos funcionales y no funcionales*

**Prompt para Generar Interfaces de Usuario**

Diseña las interfaces de usuario necesarias para el proceso de venta en la página de Home Depot México, basándose en los requerimientos funcionales y no funcionales.

**Resultado**

**1. Página de Registro**

* **Objetivo**: Permitir a los usuarios crear una cuenta.
* **Componentes Principales**:
  + Formulario de registro con campos para nombre, correo electrónico y contraseña.
  + Botón de "Registrar".
  + Mensaje de confirmación.
* **Flujo de Usuario**:
  + El usuario ingresa información válida en los campos del formulario.
  + El usuario hace clic en el botón de "Registrar".
  + Se muestra un mensaje de confirmación.
* **Requerimientos Asociados**: RF001 (Registro de Usuarios), RNF001 (Seguridad de la Información).

**2. Página de Inicio de Sesión**

* **Objetivo**: Permitir a los usuarios autenticarse.
* **Componentes Principales**:
  + Formulario de inicio de sesión con campos para correo electrónico y contraseña.
  + Botón de "Iniciar sesión".
  + Enlace para recuperar contraseña.
* **Flujo de Usuario**:
  + El usuario ingresa correo electrónico y contraseña válidos.
  + El usuario hace clic en el botón de "Iniciar sesión".
  + El usuario es redirigido a la página principal.
* **Requerimientos Asociados**: RF002 (Autenticación de Usuarios), RNF001 (Seguridad de la Información).

**3. Página de Catálogo de Productos**

* **Objetivo**: Mostrar una lista de productos disponibles.
* **Componentes Principales**:
  + Barra de búsqueda.
  + Filtros por categorías, marcas, precios, etc.
  + Lista de productos con imágenes, descripciones, precios y disponibilidad.
* **Flujo de Usuario**:
  + El usuario ingresa un término de búsqueda en la barra de búsqueda.
  + El usuario aplica filtros según sus preferencias.
  + Se muestran los productos relacionados con el término de búsqueda y los filtros aplicados.
* **Requerimientos Asociados**: RF003 (Catálogo de Productos), RNF003 (Usabilidad).

**4. Página de Detalles del Producto**

* **Objetivo**: Mostrar información detallada de un producto específico.
* **Componentes Principales**:
  + Imagen del producto.
  + Descripción del producto.
  + Precio y disponibilidad.
  + Botón de "Agregar al carrito".
* **Flujo de Usuario**:
  + El usuario selecciona un producto del catálogo.
  + Se muestra la página de detalles del producto.
  + El usuario puede agregar el producto al carrito.
* **Requerimientos Asociados**: RF003 (Catálogo de Productos), RF004 (Carrito de Compras).

**5. Página de Carrito de Compras**

* **Objetivo**: Permitir a los usuarios revisar y gestionar los productos en su carrito.
* **Componentes Principales**:
  + Lista de productos en el carrito con cantidades y precios.
  + Botón de "Proceder al pago".
  + Opciones para modificar cantidades o eliminar productos.
* **Flujo de Usuario**:
  + El usuario navega al carrito de compras.
  + El usuario revisa los productos en el carrito.
  + El usuario puede modificar cantidades o eliminar productos.
  + El usuario hace clic en "Proceder al pago".
* **Requerimientos Asociados**: RF004 (Carrito de Compras), RF005 (Proceso de Pago).

**6. Página de Pago**

* **Objetivo**: Permitir a los usuarios completar la compra.
* **Componentes Principales**:
  + Formulario de pago con opciones de método de pago.
  + Campos para ingresar información de pago.
  + Botón de "Confirmar compra".
* **Flujo de Usuario**:
  + El usuario selecciona el método de pago.
  + El usuario ingresa la información de pago.
  + El usuario hace clic en "Confirmar compra".
  + Se genera un recibo electrónico.
* **Requerimientos Asociados**: RF005 (Proceso de Pago), RNF001 (Seguridad de la Información).

**7. Página de Historial de Pedidos**

* **Objetivo**: Permitir a los usuarios ver el historial de sus compras.
* **Componentes Principales**:
  + Lista de pedidos con detalles de cada compra.
  + Opciones para ver el estado del pedido, cancelar o modificar.
* **Flujo de Usuario**:
  + El usuario navega a la sección de "Mi cuenta".
  + El usuario selecciona "Historial de pedidos".
  + Se muestra el historial de pedidos del usuario.
* **Requerimientos Asociados**: RF006 (Gestión de Pedidos).

**8. Página de Envío y Entrega**

* **Objetivo**: Permitir a los usuarios calcular costos de envío y seleccionar opciones de entrega.
* **Componentes Principales**:
  + Formulario para ingresar ubicación.
  + Opciones de entrega (envío estándar, envío rápido, recogida en tienda).
  + Botón de "Calcular costo de envío".
* **Flujo de Usuario**:
  + El usuario ingresa su ubicación.
  + El usuario selecciona la opción de entrega.
  + El usuario hace clic en "Calcular costo de envío".
  + Se muestra el costo de envío basado en la ubicación y opción seleccionada.
* **Requerimientos Asociados**: RF007 (Envío y Entrega).

*3- Escribe un promtp para generar todos los casos de prueba de sistema sobre el documento de requerimientos funcionales y no funcionales y casos de uso.*

**Prompt**

Crea casos de prueba de sistema para validar que los requerimientos funcionales y no funcionales, así como los casos de uso del proceso de venta en la página de Home Depot México, se cumplen correctamente.

**Documentos de Referencia**:

* Requerimientos Funcionales
* Requerimientos No Funcionales
* Casos de Uso

**Resultado**

**Caso de Prueba de Sistema 1: Registro de Usuario**

* **ID del Caso de Prueba**: TC001
* **Nombre del Caso de Prueba**: Registro de Usuario
* **Precondiciones**: La página de registro está accesible.
* **Pasos del Caso de Prueba**:
  1. Navegar a la página de registro.
  2. Ingresar información válida en los campos requeridos (nombre, correo electrónico, contraseña).
  3. Hacer clic en "Registrar".
* **Resultado Esperado**: El usuario es registrado exitosamente y se muestra un mensaje de confirmación.

**Caso de Prueba de Sistema 2: Autenticación de Usuario**

* **ID del Caso de Prueba**: TC002
* **Nombre del Caso de Prueba**: Autenticación de Usuario
* **Precondiciones**: El usuario está registrado.
* **Pasos del Caso de Prueba**:
  1. Navegar a la página de inicio de sesión.
  2. Ingresar correo electrónico y contraseña válidos.
  3. Hacer clic en "Iniciar sesión".
* **Resultado Esperado**: El usuario es autenticado y redirigido a la página principal.

**Caso de Prueba de Sistema 3: Búsqueda de Productos**

* **ID del Caso de Prueba**: TC003
* **Nombre del Caso de Prueba**: Búsqueda de Productos
* **Precondiciones**: La página principal está accesible.
* **Pasos del Caso de Prueba**:
  1. Ingresar un término de búsqueda en la barra de búsqueda.
  2. Hacer clic en el botón de búsqueda.
* **Resultado Esperado**: Se muestran los productos relacionados con el término de búsqueda.

**Caso de Prueba de Sistema 4: Agregar Producto al Carrito**

* **ID del Caso de Prueba**: TC004
* **Nombre del Caso de Prueba**: Agregar Producto al Carrito
* **Precondiciones**: El usuario está autenticado y en la página de detalles del producto.
* **Pasos del Caso de Prueba**:
  1. Seleccionar cantidad de producto.
  2. Hacer clic en "Agregar al carrito".
* **Resultado Esperado**: El producto es agregado al carrito y se muestra un mensaje de confirmación.

**Caso de Prueba de Sistema 5: Proceso de Pago**

* **ID del Caso de Prueba**: TC005
* **Nombre del Caso de Prueba**: Proceso de Pago
* **Precondiciones**: El usuario tiene productos en el carrito.
* **Pasos del Caso de Prueba**:
  1. Navegar al carrito de compras.
  2. Hacer clic en "Proceder al pago".
  3. Seleccionar método de pago.
  4. Ingresar información de pago.
  5. Confirmar la compra.
* **Resultado Esperado**: La compra es procesada exitosamente y se genera un recibo electrónico.

**Caso de Prueba de Sistema 6: Ver Historial de Pedidos**

* **ID del Caso de Prueba**: TC006
* **Nombre del Caso de Prueba**: Ver Historial de Pedidos
* **Precondiciones**: El usuario está autenticado.
* **Pasos del Caso de Prueba**:
  1. Navegar a la sección de "Mi cuenta".
  2. Seleccionar "Historial de pedidos".
* **Resultado Esperado**: Se muestra el historial de pedidos del usuario con detalles de cada compra.

**Caso de Prueba de Sistema 7: Cancelar Pedido**

* **ID del Caso de Prueba**: TC007
* **Nombre del Caso de Prueba**: Cancelar Pedido
* **Precondiciones**: El usuario tiene un pedido en proceso.
* **Pasos del Caso de Prueba**:
  1. Navegar a la sección de "Mi cuenta".
  2. Seleccionar "Historial de pedidos".
  3. Seleccionar el pedido que desea cancelar.
  4. Hacer clic en "Cancelar pedido".
* **Resultado Esperado**: El pedido es cancelado y se muestra un mensaje de confirmación.

**Caso de Prueba de Sistema 8: Calcular Costos de Envío**

* **ID del Caso de Prueba**: TC008
* **Nombre del Caso de Prueba**: Calcular Costos de Envío
* **Precondiciones**: El usuario tiene productos en el carrito.
* **Pasos del Caso de Prueba**:
  1. Navegar al carrito de compras.
  2. Ingresar la ubicación para calcular el costo de envío.
* **Resultado Esperado**: Se muestra el costo de envío basado en la ubicación del usuario.

**Caso de Prueba de Sistema 9: Recogida en Tienda**

* **ID del Caso de Prueba**: TC009
* **Nombre del Caso de Prueba**: Recogida en Tienda
* **Precondiciones**: El usuario tiene productos en el carrito.
* **Pasos del Caso de Prueba**:
  1. Navegar al carrito de compras.
  2. Seleccionar la opción de "Recogida en tienda".
  3. Seleccionar la tienda de recogida.
  4. Confirmar la compra.
* **Resultado Esperado**: La compra es procesada exitosamente y se genera un recibo electrónico con detalles de la recogida en tienda.

**Caso de Prueba de Sistema 10: Seguridad de la Información**

* **ID del Caso de Prueba**: TC010
* **Nombre del Caso de Prueba**: Seguridad de la Información
* **Precondiciones**: La página está accesible.
* **Pasos del Caso de Prueba**:
  1. Intentar acceder a información sensible sin autenticación.
  2. Realizar una transacción de pago.
* **Resultado Esperado**: La información sensible está protegida y la transacción cumple con los estándares de seguridad.

**Caso de Prueba de Sistema 11: Rendimiento de la Página**

* **ID del Caso de Prueba**: TC011
* **Nombre del Caso de Prueba**: Rendimiento de la Página
* **Precondiciones**: La página está accesible.
* **Pasos del Caso de Prueba**:
  1. Medir el tiempo de carga de la página principal.
  2. Realizar búsquedas y navegar entre diferentes páginas.
* **Resultado Esperado**: La página carga en menos de 3 segundos y maneja picos de tráfico sin problemas.

**Caso de Prueba de Sistema 12: Usabilidad de la Interfaz**

* **ID del Caso de Prueba**: TC012
* **Nombre del Caso de Prueba**: Usabilidad de la Interfaz
* **Precondiciones**: La página está accesible.
* **Pasos del Caso de Prueba**:
  1. Navegar por la página en diferentes dispositivos y navegadores.
  2. Realizar tareas comunes (registro, búsqueda, compra).
* **Resultado Esperado**: La interfaz es intuitiva y fácil de usar en todos los dispositivos y navegadores.

**Caso de Prueba de Sistema 13: Escalabilidad del Sistema**

* **ID del Caso de Prueba**: TC013
* **Nombre del Caso de Prueba**: Escalabilidad del Sistema
* **Precondiciones**: La página está accesible.
* **Pasos del Caso de Prueba**:
  1. Simular un aumento en el número de usuarios y transacciones.
* **Resultado Esperado**: El sistema soporta el aumento sin afectar el rendimiento.

**Caso de Prueba de Sistema 14: Mantenimiento del Sistema**

* **ID del Caso de Prueba**: TC014
* **Nombre del Caso de Prueba**: Mantenimiento del Sistema
* **Precondiciones**: La página está accesible.
* **Pasos del Caso de Prueba**:
  1. Realizar una actualización del sistema.
  2. Verificar que el servicio no se interrumpa.
* **Resultado Esperado**: La actualización se realiza sin interrupciones y la documentación es clara.

**Pruebas de Sistema.**

*1-Realiza alteraciones a la documentación de casos de uso e interfaces de usuario para que produzcan cualquier tipo de error.*

**Alteraciones en Casos de Uso**

**Caso de Uso 1: Registro de Usuario**

* **Nombre del Caso de Uso**: Registro de Usuario
* **Actor Principal**: Usuario
* **Precondiciones**: La página de registro está accesible.
* **Flujo Principal**:
  1. El usuario navega a la página de registro.
  2. El usuario ingresa información válida en los campos requeridos (nombre, correo electrónico, contraseña).
  3. El usuario hace clic en "Registrar".
  4. **Alteración**: El sistema no valida la contraseña y permite registrar contraseñas inseguras.
* **Resultado Esperado**: El usuario es registrado exitosamente y se muestra un mensaje de confirmación.
* **Resultado Alterado**: El usuario es registrado, pero la contraseña no cumple con los requisitos de seguridad.

**Caso de Uso 2: Autenticación de Usuario**

* **Nombre del Caso de Uso**: Autenticación de Usuario
* **Actor Principal**: Usuario
* **Precondiciones**: El usuario está registrado.
* **Flujo Principal**:
  1. El usuario navega a la página de inicio de sesión.
  2. El usuario ingresa correo electrónico y contraseña válidos.
  3. El usuario hace clic en "Iniciar sesión".
  4. **Alteración**: El sistema no verifica correctamente las credenciales y permite el acceso con contraseñas incorrectas.
* **Resultado Esperado**: El usuario es autenticado y redirigido a la página principal.
* **Resultado Alterado**: El usuario es autenticado incluso con credenciales incorrectas.

**Alteraciones en Interfaces de Usuario**

**Página de Registro**

* **Nombre de la Página**: Página de Registro
* **Objetivo**: Permitir a los usuarios crear una cuenta.
* **Componentes Principales**:
  + Formulario de registro con campos para nombre, correo electrónico y contraseña.
  + Botón de "Registrar".
  + Mensaje de confirmación.
  + **Alteración**: El formulario de registro no incluye validación de formato de correo electrónico.
* **Flujo de Usuario**:
  + El usuario ingresa información válida en los campos del formulario.
  + El usuario hace clic en el botón de "Registrar".
  + Se muestra un mensaje de confirmación.
* **Resultado Esperado**: El usuario es registrado exitosamente.
* **Resultado Alterado**: El usuario puede registrar un correo electrónico con formato incorrecto.

**Página de Inicio de Sesión**

* **Nombre de la Página**: Página de Inicio de Sesión
* **Objetivo**: Permitir a los usuarios autenticarse.
* **Componentes Principales**:
  + Formulario de inicio de sesión con campos para correo electrónico y contraseña.
  + Botón de "Iniciar sesión".
  + Enlace para recuperar contraseña.
  + **Alteración**: El botón de "Iniciar sesión" no funciona correctamente y no redirige al usuario.
* **Flujo de Usuario**:
  + El usuario ingresa correo electrónico y contraseña válidos.
  + El usuario hace clic en el botón de "Iniciar sesión".
  + El usuario es redirigido a la página principal.
* **Resultado Esperado**: El usuario es autenticado y redirigido a la página principal.
* **Resultado Alterado**: El usuario no puede iniciar sesión debido a un botón defectuoso.

**Diseño de Arquitectura**

*1-Escribe un prompt para generar el diseño de la arquitectura basándote en los requerimientos funcionales y no funcionales.*

Diseñar la arquitectura de la página de ventas de Home Depot (http://www.homedepot.com.mx) basada en los siguientes requerimientos funcionales y no funcionales, toma en consideración: Diagrama de infraestructura. - Diagrama de componentes. - Diagrama de base de datos.

**Resultado**

**Requerimientos Funcionales:**

1. **Registro de Usuarios**: Crear cuenta, inicio de sesión.
2. **Gestión de Productos**: Visualización y búsqueda de productos.
3. **Carrito de Compras**: Agregar y modificar productos en el carrito.
4. **Proceso de Pago**: Opciones de pago y confirmación de compra.
5. **Gestión de Pedidos**: Historial y seguimiento de pedidos.

**Requerimientos No Funcionales:**

1. **Rendimiento**: Tiempo de respuesta menor a 3 segundos, escalabilidad para manejar picos de tráfico.
2. **Seguridad**: Protección de datos mediante encriptación, autenticación de dos factores.
3. **Usabilidad**: Interfaz intuitiva, compatibilidad con diferentes dispositivos y navegadores.
4. **Mantenimiento**: Actualizaciones sin afectar la disponibilidad, monitoreo para detección y resolución rápida de problemas.

**Elementos del Diseño de Arquitectura:**

1. **Diagrama de Infraestructura**:
   * **Servidores**: AWS, Azure, o Google Cloud.
   * **Balanceo de carga**: Nginx, HAProxy.
   * **Escalabilidad**: Kubernetes, Docker.
   * **Redes**: Configuración de VPC, subredes, y grupos de seguridad.
   * **Almacenamiento**: S3, Blob Storage.
2. **Diagrama de Componentes**:
   * **Frontend**:
     + Frameworks y tecnologías: React, Angular, o Vue.js.
     + Diseño responsivo: CSS, Bootstrap.
     + Autenticación: OAuth, JWT.
   * **Backend**:
     + Lenguajes y frameworks: Node.js, Django, o Spring Boot.
     + API REST: Diseño y documentación de endpoints.
     + Seguridad: Encriptación SSL/TLS, autenticación de dos factores.
   * **Servicios**:
     + Microservicios para gestión de usuarios, productos, carrito de compras, pagos y pedidos.
     + Integración con servicios externos (pasarelas de pago, servicios de envío).
3. **Diagrama de Base de Datos**:
   * **Modelo de datos**:
     + **Usuarios**: Tabla con información de usuarios (ID, nombre, correo electrónico, contraseña encriptada).
     + **Productos**: Tabla con detalles de productos (ID, nombre, descripción, precio, stock).
     + **Carrito de Compras**: Tabla con productos en el carrito (ID de usuario, ID de producto, cantidad).
     + **Pedidos**: Tabla con información de pedidos (ID de pedido, ID de usuario, detalles de productos, estado del pedido).
   * **Base de datos**: MySQL, PostgreSQL, o MongoDB.
   * **Relaciones**: Definición de relaciones entre tablas (one-to-many, many-to-many).

**Consideraciones Adicionales:**

* **Integración continua y despliegue continuo (CI/CD)**: Jenkins, GitLab CI.
* **Pruebas automatizadas**: Selenium, JUnit.
* **Monitoreo y Mantenimiento**:
  + Herramientas de monitoreo: Prometheus, Grafana.
  + Gestión de logs: ELK Stack (Elasticsearch, Logstash, Kibana).

**Prompt**

Genera todos los casos de prueba de integración sobre el documento los diagramas de arquitectura

**Resultado**

**Casos de Prueba de Integración**

| **Nombre del Caso de Prueba** | **Descripción** | **Precondiciones** | **Pasos** | **Resultado Esperado** |
| --- | --- | --- | --- | --- |
| Integración de Registro de Usuario | Verificar la integración del registro de usuario con la base de datos y el sistema de autenticación. | El sistema debe estar configurado y la base de datos debe estar accesible. | 1. Acceder a la página de registro. 2. Ingresar información de usuario. 3. Enviar el formulario de registro. 4. Verificar que los datos se almacenan correctamente en la base de datos. 5. Verificar que el usuario puede iniciar sesión con las credenciales registradas. | El usuario se registra correctamente y puede iniciar sesión. |
| Integración de Búsqueda de Productos | Verificar la integración de la búsqueda de productos con la base de datos y la interfaz de usuario. | La base de datos debe contener productos y el sistema debe estar configurado. | 1. Acceder a la barra de búsqueda. 2. Ingresar criterios de búsqueda. 3. Enviar la búsqueda. 4. Verificar que los resultados de la búsqueda se muestran correctamente en la interfaz de usuario. 5. Verificar que los datos de los productos coinciden con los almacenados en la base de datos. | Los productos se muestran correctamente según los criterios de búsqueda. |
| Integración de Carrito de Compras | Verificar la integración del carrito de compras con la base de datos y la interfaz de usuario. | El usuario debe estar registrado y haber iniciado sesión. | 1. Acceder a la página de un producto. 2. Agregar el producto al carrito. 3. Modificar la cantidad de productos en el carrito. 4. Verificar que los cambios se reflejan correctamente en la base de datos. 5. Verificar que el carrito se muestra correctamente en la interfaz de usuario. | Los productos se agregan y modifican correctamente en el carrito. |
| Integración de Proceso de Pago | Verificar la integración del proceso de pago con la pasarela de pago y la base de datos. | El usuario debe tener productos en su carrito de compras. | 1. Acceder al carrito de compras. 2. Seleccionar el método de pago. 3. Confirmar la compra. 4. Verificar que la transacción se procesa correctamente a través de la pasarela de pago. 5. Verificar que los detalles del pedido se almacenan correctamente en la base de datos. | La compra se procesa correctamente y los detalles del pedido se almacenan en la base de datos. |
| Integración de Seguimiento de Pedidos | Verificar la integración del seguimiento de pedidos con la base de datos y la interfaz de usuario. | El usuario debe haber realizado una compra. | 1. Acceder al historial de pedidos. 2. Seleccionar un pedido para rastrear. 3. Verificar que el estado del pedido se muestra correctamente en la interfaz de usuario. 4. Verificar que los detalles del pedido coinciden con los almacenados en la base de datos. | El estado y los detalles del pedido se muestran correctamente. |

**PRUEBAS DE INTEGRACIÓN**

*Realiza alteraciones a la documentación de arquitectura para que produzcan cualquier tipo de error.*

**Prompt**

Simula realizar alteraciones en la documentación de arquitectura de la página de ventas de Home Depot para que produzca cualquier tipo de error

**Resultado**

**Simulación de Errores en la Documentación de Arquitectura**

**Diagrama de Infraestructura:**

1. **Servidores**:
   * **Alteración**: Configurar servidores en diferentes regiones sin sincronización adecuada.
   * **Consecuencia**: Problemas de latencia y consistencia de datos, lo que puede resultar en una mala experiencia de usuario.
2. **Balanceo de carga**:
   * **Alteración**: Configurar el balanceador de carga sin redundancia.
   * **Consecuencia**: Si el balanceador de carga falla, todo el sistema se cae, causando una interrupción del servicio.
3. **Escalabilidad**:
   * **Alteración**: No configurar adecuadamente los recursos de Kubernetes.
   * **Consecuencia**: Problemas de escalabilidad durante picos de tráfico, lo que puede llevar a tiempos de respuesta lentos o fallos del sistema.

**Diagrama de Componentes:**

1. **Frontend**:
   * **Alteración**: No implementar autenticación segura.
   * **Consecuencia**: Vulnerabilidades de seguridad que permiten accesos no autorizados, comprometiendo la seguridad de los datos de los usuarios.
2. **Backend**:
   * **Alteración**: No documentar correctamente los endpoints de la API REST.
   * **Consecuencia**: Dificultades para los desarrolladores al integrar y mantener la API, lo que puede llevar a errores en la implementación de nuevas funcionalidades.
3. **Servicios**:
   * **Alteración**: No implementar correctamente la integración con servicios externos.
   * **Consecuencia**: Fallos en el procesamiento de pagos y en el seguimiento de envíos, afectando la confiabilidad del sistema.

**Diagrama de Base de Datos:**

1. **Modelo de datos**:
   * **Alteración**: No definir correctamente las relaciones entre tablas.
   * **Consecuencia**: Inconsistencias y errores en la gestión de datos, lo que puede llevar a problemas en la visualización y manejo de la información.
2. **Base de datos**:
   * **Alteración**: No configurar adecuadamente la replicación de la base de datos.
   * **Consecuencia**: Pérdida de datos y problemas de disponibilidad, afectando la integridad y accesibilidad de la información.

**Consideraciones Adicionales:**

1. **Integración continua y despliegue continuo (CI/CD)**:
   * **Alteración**: No configurar correctamente las pipelines de CI/CD.
   * **Consecuencia**: Fallos en el despliegue y en la integración de nuevas funcionalidades, lo que puede llevar a interrupciones del servicio.
2. **Pruebas automatizadas**:
   * **Alteración**: No cubrir adecuadamente los casos de prueba.
   * **Consecuencia**: Bugs y errores no detectados en el sistema, lo que puede afectar la funcionalidad y la experiencia del usuario.
3. **Monitoreo y Mantenimiento**:
   * **Alteración**: No implementar herramientas de monitoreo adecuadas.
   * **Consecuencia**: Dificultades para detectar y resolver problemas rápidamente, lo que puede llevar a tiempos de inactividad prolongados.

*Escribe un prompt para que aplique todos los casos de prueba de integración que hiciste contra los diagramas de arquitectura.*

**Prompt**

Aplicar los casos de prueba de integración contra los diagramas de arquitectura de la página de ventas de Home Depot (http://www.homedepot.com.mx) para verificar la correcta interacción entre los componentes del sistema.

**Resultado**

**Diagramas de Arquitectura:**

1. **Diagrama de Infraestructura**:
   * Servidores: AWS, Azure, o Google Cloud.
   * Balanceo de carga: Nginx, HAProxy.
   * Escalabilidad: Kubernetes, Docker.
   * Redes: Configuración de VPC, subredes, y grupos de seguridad.
   * Almacenamiento: S3, Blob Storage.
2. **Diagrama de Componentes**:
   * Frontend:
     + Frameworks y tecnologías: React, Angular, o Vue.js.
     + Diseño responsivo: CSS, Bootstrap.
     + Autenticación: OAuth, JWT.
   * Backend:
     + Lenguajes y frameworks: Node.js, Django, o Spring Boot.
     + API REST: Diseño y documentación de endpoints.
     + Seguridad: Encriptación SSL/TLS, autenticación de dos factores.
   * Servicios:
     + Microservicios para gestión de usuarios, productos, carrito de compras, pagos y pedidos.
     + Integración con servicios externos (pasarelas de pago, servicios de envío).
3. **Diagrama de Base de Datos**:
   * Modelo de datos:
     + Usuarios: Tabla con información de usuarios (ID, nombre, correo electrónico, contraseña encriptada).
     + Productos: Tabla con detalles de productos (ID, nombre, descripción, precio, stock).
     + Carrito de Compras: Tabla con productos en el carrito (ID de usuario, ID de producto, cantidad).
     + Pedidos: Tabla con información de pedidos (ID de pedido, ID de usuario, detalles de productos, estado del pedido).
   * Base de datos: MySQL, PostgreSQL, o MongoDB.
   * Relaciones: Definición de relaciones entre tablas (one-to-many, many-to-many).

**Casos de Prueba de Integración:**

1. **Integración de Registro de Usuario**:
   * **Descripción**: Verificar la integración del registro de usuario con la base de datos y el sistema de autenticación.
   * **Precondiciones**: El sistema debe estar configurado y la base de datos debe estar accesible.
   * **Pasos**:
     1. Acceder a la página de registro.
     2. Ingresar información de usuario.
     3. Enviar el formulario de registro.
     4. Verificar que los datos se almacenan correctamente en la base de datos.
     5. Verificar que el usuario puede iniciar sesión con las credenciales registradas.
   * **Resultado Esperado**: El usuario se registra correctamente y puede iniciar sesión.
2. **Integración de Búsqueda de Productos**:
   * **Descripción**: Verificar la integración de la búsqueda de productos con la base de datos y la interfaz de usuario.
   * **Precondiciones**: La base de datos debe contener productos y el sistema debe estar configurado.
   * **Pasos**:
     1. Acceder a la barra de búsqueda.
     2. Ingresar criterios de búsqueda.
     3. Enviar la búsqueda.
     4. Verificar que los resultados de la búsqueda se muestran correctamente en la interfaz de usuario.
     5. Verificar que los datos de los productos coinciden con los almacenados en la base de datos.
   * **Resultado Esperado**: Los productos se muestran correctamente según los criterios de búsqueda.
3. **Integración de Carrito de Compras**:
   * **Descripción**: Verificar la integración del carrito de compras con la base de datos y la interfaz de usuario.
   * **Precondiciones**: El usuario debe estar registrado y haber iniciado sesión.
   * **Pasos**:
     1. Acceder a la página de un producto.
     2. Agregar el producto al carrito.
     3. Modificar la cantidad de productos en el carrito.
     4. Verificar que los cambios se reflejan correctamente en la base de datos.
     5. Verificar que el carrito se muestra correctamente en la interfaz de usuario.
   * **Resultado Esperado**: Los productos se agregan y modifican correctamente en el carrito.
4. **Integración de Proceso de Pago**:
   * **Descripción**: Verificar la integración del proceso de pago con la pasarela de pago y la base de datos.
   * **Precondiciones**: El usuario debe tener productos en su carrito de compras.
   * **Pasos**:
     1. Acceder al carrito de compras.
     2. Seleccionar el método de pago.
     3. Confirmar la compra.
     4. Verificar que la transacción se procesa correctamente a través de la pasarela de pago.
     5. Verificar que los detalles del pedido se almacenan correctamente en la base de datos.
   * **Resultado Esperado**: La compra se procesa correctamente y los detalles del pedido se almacenan en la base de datos.
5. **Integración de Seguimiento de Pedidos**:
   * **Descripción**: Verificar la integración del seguimiento de pedidos con la base de datos y la interfaz de usuario.
   * **Precondiciones**: El usuario debe haber realizado una compra.
   * **Pasos**:
     1. Acceder al historial de pedidos.
     2. Seleccionar un pedido para rastrear.
     3. Verificar que el estado del pedido se muestra correctamente en la interfaz de usuario.
     4. Verificar que los detalles del pedido coinciden con los almacenados en la base de datos.
   * **Resultado Esperado**: El estado y los detalles del pedido se muestran correctamente.

**DISEÑO MODULAR**

*Escribe un prompt para generar los flujos y entidades principales basándote en los requerimientos funcionales y no funcionales. Toma en consideración: - Diagramas de clases. - Diagramas de flujo.*

**Prompt**

Generar los flujos y entidades principales para la página de ventas de Home Depot (http://www.homedepot.com.mx) basados en los siguientes requerimientos funcionales y no funcionales.

**Resultado**

Requerimientos Funcionales:

1. Registro de Usuarios: Crear cuenta, inicio de sesión.
2. Gestión de Productos: Visualización y búsqueda de productos.
3. Carrito de Compras: Agregar y modificar productos en el carrito.
4. Proceso de Pago: Opciones de pago y confirmación de compra.
5. Gestión de Pedidos: Historial y seguimiento de pedidos.

Requerimientos No Funcionales:

1. Rendimiento: Tiempo de respuesta menor a 3 segundos, escalabilidad para manejar picos de tráfico.
2. Seguridad: Protección de datos mediante encriptación, autenticación de dos factores.
3. Usabilidad: Interfaz intuitiva, compatibilidad con diferentes dispositivos y navegadores.
4. Mantenimiento: Actualizaciones sin afectar la disponibilidad, monitoreo para detección y resolución rápida de problemas.

Elementos del Diseño de Flujos y Entidades:

Diagramas de Clases:

1. Clase Usuario:
   * Atributos: ID, nombre, correo electrónico, contraseña encriptada.
   * Métodos: registrar(), iniciarSesion(), actualizarPerfil().
2. Clase Producto:
   * Atributos: ID, nombre, descripción, precio, stock.
   * Métodos: buscarProducto(), actualizarStock(), obtenerDetalles().
3. Clase Carrito:
   * Atributos: ID de usuario, lista de productos, cantidad.
   * Métodos: agregarProducto(), modificarCantidad(), eliminarProducto().
4. Clase Pedido:
   * Atributos: ID de pedido, ID de usuario, lista de productos, estado del pedido.
   * Métodos: confirmarPedido(), rastrearPedido(), actualizarEstado().

Diagramas de Flujo:

1. Flujo de Registro de Usuario:
   * Inicio: El usuario accede a la página de registro.
   * Pasos:
     1. El usuario ingresa su información personal.
     2. El usuario envía el formulario de registro.
     3. El sistema valida la información y crea una cuenta.
   * Fin: El usuario recibe una confirmación de registro y puede iniciar sesión.
2. Flujo de Búsqueda de Productos:
   * Inicio: El usuario accede a la barra de búsqueda.
   * Pasos:
     1. El usuario ingresa criterios de búsqueda.
     2. El sistema busca productos en la base de datos.
     3. El sistema muestra los resultados de la búsqueda.
   * Fin: El usuario ve una lista de productos que coinciden con los criterios de búsqueda.
3. Flujo de Carrito de Compras:
   * Inicio: El usuario accede a la página de un producto.
   * Pasos:
     1. El usuario agrega el producto al carrito.
     2. El usuario modifica la cantidad de productos en el carrito.
     3. El usuario elimina productos del carrito si es necesario.
   * Fin: El usuario revisa y confirma los productos en el carrito.
4. Flujo de Proceso de Pago:
   * Inicio: El usuario accede al carrito de compras.
   * Pasos:
     1. El usuario selecciona el método de pago.
     2. El usuario confirma la compra.
     3. El sistema procesa la transacción a través de la pasarela de pago.
   * Fin: El usuario recibe una confirmación de compra y detalles del pedido.
5. Flujo de Seguimiento de Pedidos:
   * Inicio: El usuario accede al historial de pedidos.
   * Pasos:
     1. El usuario selecciona un pedido para rastrear.
     2. El sistema muestra el estado del pedido.
     3. El sistema actualiza el estado del pedido si hay cambios.
   * Fin: El usuario ve el estado actual y los detalles del pedido.

***Escribe un prompt para generar todos los casos de pruebas sobre los diagramas de clases y flujo.***

**Prompt**

Generar todos los casos de prueba basados en los diagramas de clases y flujo para la página de ventas de Home Depot (http://www.homedepot.com.mx), asegurando la correcta funcionalidad y interacción de los componentes del sistema.

**Resultado**

**Diagramas de Clases:**

1. **Clase Usuario**:
   * **Atributos**: ID, nombre, correo electrónico, contraseña encriptada.
   * **Métodos**: registrar(), iniciarSesion(), actualizarPerfil().
   * **Casos de Prueba**:
     + **Prueba de Registro**: Verificar que un usuario puede registrarse correctamente.
     + **Prueba de Inicio de Sesión**: Verificar que un usuario puede iniciar sesión con credenciales válidas.
     + **Prueba de Actualización de Perfil**: Verificar que un usuario puede actualizar su perfil.
2. **Clase Producto**:
   * **Atributos**: ID, nombre, descripción, precio, stock.
   * **Métodos**: buscarProducto(), actualizarStock(), obtenerDetalles().
   * **Casos de Prueba**:
     + **Prueba de Búsqueda de Producto**: Verificar que los productos se pueden buscar correctamente.
     + **Prueba de Actualización de Stock**: Verificar que el stock de un producto se puede actualizar.
     + **Prueba de Obtención de Detalles**: Verificar que los detalles de un producto se pueden obtener correctamente.
3. **Clase Carrito**:
   * **Atributos**: ID de usuario, lista de productos, cantidad.
   * **Métodos**: agregarProducto(), modificarCantidad(), eliminarProducto().
   * **Casos de Prueba**:
     + **Prueba de Agregar Producto al Carrito**: Verificar que un producto se puede agregar al carrito.
     + **Prueba de Modificar Cantidad en el Carrito**: Verificar que la cantidad de productos en el carrito se puede modificar.
     + **Prueba de Eliminar Producto del Carrito**: Verificar que un producto se puede eliminar del carrito.
4. **Clase Pedido**:
   * **Atributos**: ID de pedido, ID de usuario, lista de productos, estado del pedido.
   * **Métodos**: confirmarPedido(), rastrearPedido(), actualizarEstado().
   * **Casos de Prueba**:
     + **Prueba de Confirmación de Pedido**: Verificar que un pedido se puede confirmar correctamente.
     + **Prueba de Rastreo de Pedido**: Verificar que el estado de un pedido se puede rastrear.
     + **Prueba de Actualización de Estado de Pedido**: Verificar que el estado de un pedido se puede actualizar.

**Diagramas de Flujo:**

1. **Flujo de Registro de Usuario**:
   * **Inicio**: El usuario accede a la página de registro.
   * **Pasos**:
     1. El usuario ingresa su información personal.
     2. El usuario envía el formulario de registro.
     3. El sistema valida la información y crea una cuenta.
   * **Fin**: El usuario recibe una confirmación de registro y puede iniciar sesión.
   * **Casos de Prueba**:
     1. **Prueba de Validación de Información**: Verificar que la información ingresada por el usuario se valida correctamente.
     2. **Prueba de Creación de Cuenta**: Verificar que la cuenta se crea correctamente y se almacena en la base de datos.
2. **Flujo de Búsqueda de Productos**:
   * **Inicio**: El usuario accede a la barra de búsqueda.
   * **Pasos**:
     1. El usuario ingresa criterios de búsqueda.
     2. El sistema busca productos en la base de datos.
     3. El sistema muestra los resultados de la búsqueda.
   * **Fin**: El usuario ve una lista de productos que coinciden con los criterios de búsqueda.
   * **Casos de Prueba**:
     1. **Prueba de Búsqueda por Nombre**: Verificar que los productos se pueden buscar por nombre.
     2. **Prueba de Búsqueda por Categoría**: Verificar que los productos se pueden buscar por categoría.
3. **Flujo de Carrito de Compras**:
   * **Inicio**: El usuario accede a la página de un producto.
   * **Pasos**:
     1. El usuario agrega el producto al carrito.
     2. El usuario modifica la cantidad de productos en el carrito.
     3. El usuario elimina productos del carrito si es necesario.
   * **Fin**: El usuario revisa y confirma los productos en el carrito.
   * **Casos de Prueba**:
     1. **Prueba de Agregar Producto**: Verificar que un producto se puede agregar al carrito.
     2. **Prueba de Modificar Cantidad**: Verificar que la cantidad de productos en el carrito se puede modificar.
     3. **Prueba de Eliminar Producto**: Verificar que un producto se puede eliminar del carrito.
4. **Flujo de Proceso de Pago**:
   * **Inicio**: El usuario accede al carrito de compras.
   * **Pasos**:
     1. El usuario selecciona el método de pago.
     2. El usuario confirma la compra.
     3. El sistema procesa la transacción a través de la pasarela de pago.
   * **Fin**: El usuario recibe una confirmación de compra y detalles del pedido.
   * **Casos de Prueba**:
     1. **Prueba de Selección de Método de Pago**: Verificar que el usuario puede seleccionar diferentes métodos de pago.
     2. **Prueba de Confirmación de Compra**: Verificar que la compra se confirma correctamente y se almacena en la base de datos.
5. **Flujo de Seguimiento de Pedidos**:
   * **Inicio**: El usuario accede al historial de pedidos.
   * **Pasos**:
     1. El usuario selecciona un pedido para rastrear.
     2. El sistema muestra el estado del pedido.
     3. El sistema actualiza el estado del pedido si hay cambios.
   * **Fin**: El usuario ve el estado actual y los detalles del pedido.
   * **Casos de Prueba**:
     1. **Prueba de Visualización de Estado**: Verificar que el estado del pedido se muestra correctamente.
     2. **Prueba de Actualización de Estado**: Verificar que el estado del pedido se actualiza correctamente.

**PRUEBAS UNITARIAS**

*Escribe un prompt para generar el código funcional de algunas clases en el lenguaje de programación que prefieras.*

**Prompt**

Generar el código funcional de las clases principales para la página de ventas de Home Depot (http://www.homedepot.com.mx) en el lenguaje de programación Python.

Resultado

**Requerimientos Funcionales:**

1. **Registro de Usuarios**: Crear cuenta, inicio de sesión.
2. **Gestión de Productos**: Visualización y búsqueda de productos.
3. **Carrito de Compras**: Agregar y modificar productos en el carrito.
4. **Proceso de Pago**: Opciones de pago y confirmación de compra.
5. **Gestión de Pedidos**: Historial y seguimiento de pedidos.

**Código Funcional en Python**

**Clase Usuario:**

class Usuario:

def \_\_init\_\_(self, id, nombre, correo\_electronico, contrasena):

self.id = id

self.nombre = nombre

self.correo\_electronico = correo\_electronico

self.contrasena = contrasena

def registrar(self):

# Código para registrar al usuario

pass

def iniciar\_sesion(self, correo\_electronico, contrasena):

# Código para iniciar sesión

pass

def actualizar\_perfil(self, nombre, correo\_electronico, contrasena):

# Código para actualizar el perfil del usuario

pass

**Clase Producto:**

class Producto:

def \_\_init\_\_(self, id, nombre, descripcion, precio, stock):

self.id = id

self.nombre = nombre

self.descripcion = descripcion

self.precio = precio

self.stock = stock

def buscar\_producto(self, criterio):

# Código para buscar productos

pass

def actualizar\_stock(self, cantidad):

# Código para actualizar el stock del producto

pass

def obtener\_detalles(self):

# Código para obtener los detalles del producto

pass

**Clase Carrito:**

class Carrito:

def \_\_init\_\_(self, usuario\_id):

self.usuario\_id = usuario\_id

self.productos = []

def agregar\_producto(self, producto, cantidad):

# Código para agregar producto al carrito

pass

def modificar\_cantidad(self, producto\_id, cantidad):

# Código para modificar la cantidad de productos en el carrito

pass

def eliminar\_producto(self, producto\_id):

# Código para eliminar producto del carrito

pass

**Clase Pedido:**

class Pedido:

def \_\_init\_\_(self, id, usuario\_id, productos, estado):

self.id = id

self.usuario\_id = usuario\_id

self.productos = productos

self.estado = estado

def confirmar\_pedido(self):

# Código para confirmar el pedido

pass

def rastrear\_pedido(self):

# Código para rastrear el pedido

pass

def actualizar\_estado(self, nuevo\_estado):

# Código para actualizar el estado del pedido

pass

*Escribe un prompt para generar el código de pruebas unitarias en el frameworks que prefieras (Nuni, Junit).*

**Prompt**

Generar el código de pruebas unitarias para las clases principales de la página de ventas de Home Depot (http://www.homedepot.com.mx) utilizando el framework JUnit.

**Clases a Probar:**

1. **Clase Usuario**
2. **Clase Producto**
3. **Clase Carrito**
4. **Clase Pedido**

Resultado

**Código de Pruebas Unitarias con JUnit**

**Pruebas para la Clase Usuario:**

import static org.junit.Assert.\*;

import org.junit.Before;

import org.junit.Test;

public class UsuarioTest {

private Usuario usuario;

@Before

public void setUp() {

usuario = new Usuario(1, "Juan", "juan@example.com", "password123");

}

@Test

public void testRegistrar() {

// Código para probar el registro de usuario

assertTrue(usuario.registrar());

}

@Test

public void testIniciarSesion() {

// Código para probar el inicio de sesión

assertTrue(usuario.iniciarSesion("juan@example.com", "password123"));

}

@Test

public void testActualizarPerfil() {

// Código para probar la actualización de perfil

usuario.actualizarPerfil("Juan Perez", "juan.perez@example.com", "newpassword123");

assertEquals("Juan Perez", usuario.getNombre());

assertEquals("juan.perez@example.com", usuario.getCorreoElectronico());

}

}

**Pruebas para la Clase Producto:**

import static org.junit.Assert.\*;

import org.junit.Before;

import org.junit.Test;

public class ProductoTest {

private Producto producto;

@Before

public void setUp() {

producto = new Producto(1, "Martillo", "Martillo de acero", 100.0, 50);

}

@Test

public void testBuscarProducto() {

// Código para probar la búsqueda de productos

Producto resultado = producto.buscarProducto("Martillo");

assertNotNull(resultado);

assertEquals("Martillo", resultado.getNombre());

}

@Test

public void testActualizarStock() {

// Código para probar la actualización de stock

producto.actualizarStock(30);

assertEquals(30, producto.getStock());

}

@Test

public void testObtenerDetalles() {

// Código para probar la obtención de detalles del producto

String detalles = producto.obtenerDetalles();

assertEquals("Martillo de acero", detalles);

}

}

**Pruebas para la Clase Carrito:**

import static org.junit.Assert.\*;

import org.junit.Before;

import org.junit.Test;

public class CarritoTest {

private Carrito carrito;

private Producto producto;

@Before

public void setUp() {

carrito = new Carrito(1);

producto = new Producto(1, "Martillo", "Martillo de acero", 100.0, 50);

}

@Test

public void testAgregarProducto() {

// Código para probar agregar producto al carrito

carrito.agregarProducto(producto, 2);

assertEquals(1, carrito.getProductos().size());

}

@Test

public void testModificarCantidad() {

// Código para probar modificar la cantidad de productos en el carrito

carrito.agregarProducto(producto, 2);

carrito.modificarCantidad(producto.getId(), 5);

assertEquals(5, carrito.getCantidad(producto.getId()));

}

@Test

public void testEliminarProducto() {

// Código para probar eliminar producto del carrito

carrito.agregarProducto(producto, 2);

carrito.eliminarProducto(producto.getId());

assertEquals(0, carrito.getProductos().size());

}

}

**Pruebas para la Clase Pedido:**

import static org.junit.Assert.\*;

import org.junit.Before;

import org.junit.Test;

public class PedidoTest {

private Pedido pedido;

private Producto producto;

@Before

public void setUp() {

producto = new Producto(1, "Martillo", "Martillo de acero", 100.0, 50);

List<Producto> productos = new ArrayList<>();

productos.add(producto);

pedido = new Pedido(1, 1, productos, "Pendiente");

}

@Test

public void testConfirmarPedido() {

// Código para probar la confirmación de pedido

assertTrue(pedido.confirmarPedido());

}

@Test

public void testRastrearPedido() {

// Código para probar el rastreo de pedido

String estado = pedido.rastrearPedido();

assertEquals("Pendiente", estado);

}

@Test

public void testActualizarEstado() {

// Código para probar la actualización de estado del pedido

pedido.actualizarEstado("Enviado");

assertEquals("Enviado", pedido.getEstado());

}

}

*Realiza alteraciones al código funcional para que produzcan cualquier tipo de error*

**Simulación de Errores en el Código Funcional**

**Clase Usuario:**

class Usuario:

def \_\_init\_\_(self, id, nombre, correo\_electronico, contrasena):

self.id = id

self.nombre = nombre

self.correo\_electronico = correo\_electronico

self.contrasena = contrasena

def registrar(self):

# Error: No se valida la información del usuario antes de registrarlo

# Código incorrecto para registrar al usuario

return True

def iniciar\_sesion(self, correo\_electronico, contrasena):

# Error: Comparación incorrecta de contraseñas

if self.correo\_electronico == correo\_electronico and self.contrasena == contrasena:

return True

return False

def actualizar\_perfil(self, nombre, correo\_electronico, contrasena):

# Error: No se actualizan todos los atributos correctamente

self.nombre = nombre

self.correo\_electronico = correo\_electronico

# Falta actualizar la contraseña

**Clase Producto:**

class Producto:

def \_\_init\_\_(self, id, nombre, descripcion, precio, stock):

self.id = id

self.nombre = nombre

self.descripcion = descripcion

self.precio = precio

self.stock = stock

def buscar\_producto(self, criterio):

# Error: No se realiza la búsqueda correctamente

# Código incorrecto para buscar productos

return None

def actualizar\_stock(self, cantidad):

# Error: No se valida la cantidad antes de actualizar el stock

self.stock = cantidad

def obtener\_detalles(self):

# Error: Retorno incorrecto de detalles del producto

return self.nombre

**Clase Carrito:**

class Carrito:

def \_\_init\_\_(self, usuario\_id):

self.usuario\_id = usuario\_id

self.productos = []

def agregar\_producto(self, producto, cantidad):

# Error: No se verifica si el producto ya está en el carrito

self.productos.append((producto, cantidad))

def modificar\_cantidad(self, producto\_id, cantidad):

# Error: No se verifica si el producto está en el carrito antes de modificar la cantidad

for prod in self.productos:

if prod[0].id == producto\_id:

prod[1] = cantidad

def eliminar\_producto(self, producto\_id):

# Error: Eliminación incorrecta de productos

self.productos = [prod for prod in self.productos if prod[0].id != producto\_id]

**Clase Pedido:**

class Pedido:

def \_\_init\_\_(self, id, usuario\_id, productos, estado):

self.id = id

self.usuario\_id = usuario\_id

self.productos = productos

self.estado = estado

def confirmar\_pedido(self):

# Error: No se verifica el estado del pedido antes de confirmarlo

self.estado = "Confirmado"

return True

def rastrear\_pedido(self):

# Error: Retorno incorrecto del estado del pedido

return "Estado desconocido"

def actualizar\_estado(self, nuevo\_estado):

# Error: No se valida el nuevo estado antes de actualizarlo

self.estado = nuevo\_estado

**Consecuencias de los Errores Introducidos:**

1. **Clase Usuario**:
   * **Registrar**: Usuarios pueden registrarse sin validación, lo que puede llevar a datos incorrectos o duplicados.
   * **Iniciar Sesión**: Comparación incorrecta de contraseñas puede permitir accesos no autorizados.
   * **Actualizar Perfil**: Falta de actualización de contraseña puede dejar la cuenta vulnerable.
2. **Clase Producto**:
   * **Buscar Producto**: Búsqueda incorrecta puede resultar en productos no encontrados.
   * **Actualizar Stock**: Actualización sin validación puede llevar a cantidades negativas o incorrectas.
   * **Obtener Detalles**: Retorno incorrecto de detalles puede confundir a los usuarios.
3. **Clase Carrito**:
   * **Agregar Producto**: No verificar si el producto ya está en el carrito puede resultar en duplicados.
   * **Modificar Cantidad**: No verificar si el producto está en el carrito puede llevar a errores de cantidad.
   * **Eliminar Producto**: Eliminación incorrecta puede dejar productos no deseados en el carrito.
4. **Clase Pedido**:
   * **Confirmar Pedido**: Confirmación sin verificar el estado puede llevar a pedidos incorrectos.
   * **Rastrear Pedido**: Retorno incorrecto del estado puede confundir a los usuarios.
   * **Actualizar Estado**: Actualización sin validación puede llevar a estados incorrectos.

**CLASIFICACIÓN**

*Escribe un prompt que te ayude a asignar el grado de severidad y tipo de error a todos los errores que has registrado hasta el momento (utiliza tu archivo de Excel).*

**Prompt**

Asignar el grado de severidad y tipo de error a los errores registrados en las clases funcionales de la página de ventas de Home Depot (http://www.homedepot.com.mx).

**Resultado**

**Errores Registrados:**

1. **Clase Usuario**:
   * **Registrar**: Usuarios pueden registrarse sin validación.
   * **Iniciar Sesión**: Comparación incorrecta de contraseñas.
   * **Actualizar Perfil**: Falta de actualización de contraseña.
2. **Clase Producto**:
   * **Buscar Producto**: Búsqueda incorrecta.
   * **Actualizar Stock**: Actualización sin validación.
   * **Obtener Detalles**: Retorno incorrecto de detalles.
3. **Clase Carrito**:
   * **Agregar Producto**: No verificar si el producto ya está en el carrito.
   * **Modificar Cantidad**: No verificar si el producto está en el carrito.
   * **Eliminar Producto**: Eliminación incorrecta.
4. **Clase Pedido**:
   * **Confirmar Pedido**: Confirmación sin verificar el estado.
   * **Rastrear Pedido**: Retorno incorrecto del estado.
   * **Actualizar Estado**: Actualización sin validación.

**Asignación de Grado de Severidad y Tipo de Error**

**Grados de Severidad:**

1. **Crítico**: El error afecta gravemente la funcionalidad del sistema y puede causar una interrupción completa del servicio.
2. **Alto**: El error afecta significativamente la funcionalidad del sistema y puede causar problemas importantes para los usuarios.
3. **Medio**: El error afecta la funcionalidad del sistema de manera moderada y puede causar inconvenientes para los usuarios.
4. **Bajo**: El error afecta la funcionalidad del sistema de manera mínima y puede causar pequeños inconvenientes para los usuarios.

**Tipos de Error:**

1. **Funcional**: El error afecta la funcionalidad del sistema.
2. **Seguridad**: El error afecta la seguridad del sistema.
3. **Usabilidad**: El error afecta la facilidad de uso del sistema.
4. **Rendimiento**: El error afecta el rendimiento del sistema.

**Asignación de Errores:**

| **Error** | **Grado de Severidad** | **Tipo de Error** |
| --- | --- | --- |
| **Clase Usuario**: Registrar sin validación | Alto | Funcional |
| **Clase Usuario**: Comparación incorrecta de contraseñas | Crítico | Seguridad |
| **Clase Usuario**: Falta de actualización de contraseña | Alto | Seguridad |
| **Clase Producto**: Búsqueda incorrecta | Medio | Funcional |
| **Clase Producto**: Actualización sin validación | Alto | Funcional |
| **Clase Producto**: Retorno incorrecto de detalles | Bajo | Usabilidad |
| **Clase Carrito**: No verificar si el producto ya está en el carrito | Medio | Funcional |
| **Clase Carrito**: No verificar si el producto está en el carrito | Medio | Funcional |
| **Clase Carrito**: Eliminación incorrecta | Medio | Funcional |
| **Clase Pedido**: Confirmación sin verificar el estado | Alto | Funcional |
| **Clase Pedido**: Retorno incorrecto del estado | Bajo | Usabilidad |
| **Clase Pedido**: Actualización sin validación | Alto | Funcional |