

Intro

En el contexto de **Clean Architecture**, los casos de uso son piezas clave, ya que definen las acciones que el sistema debe realizar desde una perspectiva independiente de los detalles técnicos. Un documento de caso de uso o una historia de usuario debe estar orientado a describir lo que hace el sistema, los actores involucrados y las reglas de negocio, manteniéndose desacoplado de cómo será implementado.

Documento de Caso de Uso

1. **Nombre del Caso de Uso:** Registrar Usuario
2. **Actor Primario:** Usuario no registrado
3. **Descripción:** El sistema permitirá que un usuario se registre proporcionando su nombre, correo electrónico, y contraseña. Si los datos son válidos y el correo no está ya registrado, el sistema creará una cuenta nueva y enviará un correo de confirmación.
4. **Precondiciones:**
 - El usuario no debe estar ya registrado en el sistema.
 - El sistema debe estar en funcionamiento y conectado a la base de datos.
5. **Flujo Principal (Escenario Principal):**
 1. El usuario accede a la página de registro.
 2. El sistema muestra un formulario solicitando nombre, correo electrónico y contraseña.
 3. El usuario completa el formulario y lo envía.
 4. El sistema valida los datos ingresados:
 - El correo electrónico debe tener un formato válido.
 - La contraseña debe cumplir con los criterios de seguridad establecidos (mínimo de 8 caracteres, una letra mayúscula, un número, etc.).
 5. Si los datos son válidos, el sistema verifica si el correo ya está registrado.
 6. Si el correo no está registrado, el sistema crea la cuenta de usuario.
 7. El sistema envía un correo electrónico de confirmación al usuario.
 8. El sistema informa al usuario que el registro fue exitoso y debe confirmar su correo electrónico.
6. **Flujo Alternativo (Errores):**
 - **Correo electrónico ya registrado:**
 1. Si el correo está registrado, el sistema informa al usuario que no puede usarlo.
 - **Datos inválidos:**
 1. Si algún campo no cumple con los criterios, el sistema muestra un mensaje de error específico.
7. **Postcondiciones:**
 - El usuario ha sido registrado exitosamente y ha recibido un correo de confirmación.
8. **Reglas de Negocio:**
 - El correo electrónico debe ser único en el sistema.
 - La contraseña debe cumplir con las políticas de seguridad.
9. **Excepciones:**
 - Fallo en el envío del correo de confirmación: El sistema debe reintentar o alertar al administrador.

Historia de Usuario (Formato ágil)

1. **Título:** Registro de usuario
2. **Como:** Usuario no registrado
3. **Quiero:** Poder registrarme en el sistema proporcionando mis datos básicos (nombre, correo, contraseña)
4. **Para:** Poder tener acceso a las funcionalidades que requieren autenticación
5. **Criterios de aceptación:**
 - El sistema debe permitir ingresar nombre, correo electrónico y contraseña.
 - El correo electrónico debe ser único y tener un formato válido.
 - La contraseña debe cumplir con los requisitos de seguridad.
 - El sistema debe notificar al usuario en caso de error o éxito del registro.
 - En caso de éxito, debe enviarse un correo de confirmación.
6. **Notas adicionales:**
 - El sistema debe evitar múltiples intentos de registro con el mismo correo.
 - El registro debe ser confirmado a través de un enlace enviado por correo electrónico.

Relación con Clean Architecture

En **Clean Architecture**, el caso de uso o la historia de usuario describe la **intención** del sistema desde la perspectiva del negocio, sin entrar en detalles de cómo se va a implementar. Esto te permite diseñar interacciones en tu capa de **Aplicación** y luego implementar los detalles en las capas inferiores (como infraestructura o UI).

- **Caso de Uso:** En la arquitectura, esto suele estar representado en el nivel de los **Use Cases** (casos de uso) o **Interactors**, donde las reglas de negocio son aplicadas.
- **Historia de Usuario:** En un contexto ágil, puede inspirar la creación de **Use Cases** más detallados dentro de la arquitectura.

Caso de Uso 2: Realizar Pedido

1. **Nombre del Caso de Uso:** Realizar Pedido
 2. **Actor Primario:** Cliente registrado
 3. **Descripción:** Un cliente puede realizar un pedido seleccionando productos de su carrito de compras. El sistema calculará el precio total, incluyendo impuestos y envío, y generará una orden que luego puede ser pagada.
 4. **Precondiciones:**
 - El cliente debe estar autenticado.
 - El carrito de compras debe contener al menos un producto.
 - El sistema debe estar conectado al inventario para verificar la disponibilidad de productos.
 5. **Flujo Principal (Escenario Principal):**
 1. El cliente accede a su carrito de compras.
 2. El sistema muestra los productos seleccionados, el subtotal, impuestos y costo de envío.
 3. El cliente confirma los productos y procede a realizar el pedido.
 4. El sistema verifica la disponibilidad de los productos en el inventario.
 5. El sistema calcula el costo total del pedido.
 6. El cliente selecciona un método de pago y proporciona los datos correspondientes.
 7. El sistema valida el pago.
 8. El sistema genera el pedido y lo almacena en la base de datos.
 9. El sistema envía una confirmación del pedido al cliente vía correo electrónico.
 6. **Flujo Alternativo (Errores):**
 - **Productos fuera de stock:** Si uno o más productos no están disponibles, el sistema notifica al cliente y ofrece la opción de eliminar los productos fuera de stock o esperar por su disponibilidad.
 - **Pago fallido:** Si el pago no es validado, el sistema notifica al cliente y permite reintentar el pago o seleccionar otro método.
 7. **Postcondiciones:**
 - El pedido ha sido creado y está listo para ser procesado.
 - El cliente ha sido notificado de la creación del pedido.
 8. **Reglas de Negocio:**
 - No se debe permitir que un cliente realice un pedido si no hay productos disponibles.
 - Los impuestos deben calcularse de acuerdo con las leyes locales.
 9. **Excepciones:**
 - Problemas con el inventario: Si el inventario no responde, el sistema debe almacenar el pedido como "pendiente" y notificar al administrador.
-

Historia de Usuario 2: Realizar Pedido

1. **Título:** Realización de pedido
 2. **Como:** Cliente registrado
 3. **Quiero:** Poder realizar un pedido con los productos que seleccioné en mi carrito de compras
 4. **Para:** Completar mi compra y recibir mis productos
 5. **Criterios de aceptación:**
 - El sistema debe mostrar todos los productos en el carrito, con sus precios, impuestos y costo de envío.
 - El sistema debe verificar la disponibilidad de los productos antes de proceder con el pedido.
 - El pedido solo se debe procesar si el pago es validado.
 - Debe generarse una confirmación del pedido, que será enviada al correo del cliente.
 6. **Notas adicionales:**
 - Si algún producto no está disponible, el sistema debe ofrecer al cliente la opción de modificar el pedido.
 - El sistema debe asegurar que los impuestos y costos de envío se calculen correctamente antes de la confirmación final.
-

Caso de Uso 3: Cancelar Pedido

1. **Nombre del Caso de Uso:** Cancelar Pedido
2. **Actor Primario:** Cliente registrado
3. **Descripción:** El sistema permitirá que un cliente cancele un pedido que no ha sido procesado aún por el equipo de logística.
4. **Precondiciones:**
 - El cliente debe estar autenticado.
 - El pedido debe estar en estado "pendiente".
5. **Flujo Principal (Escenario Principal):**
 1. El cliente accede a su historial de pedidos.
 2. El cliente selecciona un pedido en estado "pendiente" y elige la opción de cancelación.

3. El sistema verifica el estado del pedido.
4. Si el pedido es cancelable, el sistema cambia su estado a "cancelado".
5. El sistema notifica al cliente y al equipo de logística de la cancelación.
6. El sistema emite un reembolso (si aplica) y lo notifica al cliente.

6. **Flujo Alternativo (Errores):**

- **Pedido en estado no cancelable:** Si el pedido ya ha sido procesado o enviado, el sistema notifica al cliente que no puede ser cancelado.

7. **Postcondiciones:**

- El pedido ha sido cancelado y el cliente ha sido notificado.

8. **Reglas de Negocio:**

- Solo los pedidos en estado "pendiente" pueden ser cancelados.
- Los reembolsos deben ser emitidos en función de las políticas de devolución.

9. **Excepciones:**

- Fallo en la emisión del reembolso: El sistema debe alertar al administrador si el reembolso no se procesa correctamente.

Historia de Usuario 3: Cancelar Pedido

1. **Título:** Cancelación de pedido
2. **Como:** Cliente registrado
3. **Quiero:** Poder cancelar un pedido que aún no ha sido procesado
4. **Para:** Evitar recibir productos que ya no deseo comprar
5. **Criterios de aceptación:**
 - Solo los pedidos que no han sido procesados pueden ser cancelados.
 - El sistema debe confirmar al cliente que la cancelación fue exitosa.
 - El sistema debe emitir un reembolso si el cliente ya pagó por el pedido.
6. **Notas adicionales:**
 - El cliente debe ser notificado en caso de que el pedido ya no sea cancelable.
 - El reembolso debe procesarse según las políticas del sistema.

Reglas de Negocio

Las reglas de negocio, validaciones y flujos alternativos son cruciales, ya que el objetivo es que estas descripciones guíen la creación de los **interactors** o **casos de uso** dentro de la arquitectura.

Mapeo de Casos de Uso a código

En **Clean Architecture**, los casos de uso se mapean al código como **interactors** o **servicios de aplicación**, que se encargan de ejecutar la lógica de negocio sin depender de detalles técnicos (como frameworks o bases de datos).

Componentes Clave

1. **Casos de Uso (Use Case o Interactor):** Representan la lógica de negocio para cumplir un caso de uso específico.
2. **Interfaces o Puertos:** Interfaces que los interactores usan para comunicarse con otras capas, como repositorios (para acceder a la base de datos) o servicios externos (como una API de pago).
3. **Entidades:** Modelos que contienen las reglas de negocio más esenciales.
4. **Adaptadores (Entradas y Salidas):** Estos adaptadores conectan el caso de uso con las interfaces del mundo exterior (como APIs, UI, o bases de datos).
5. **Repositorios:** Interfaces que representan el acceso a la persistencia (bases de datos).