

UT6 DIAGRAMAS DE COMPORTAMIENTO

RESULTADOS DE APRENDIZAJE ASOCIADOS
6. Genera diagramas de comportamiento valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno
CRITERIOS DE EVALUACIÓN
a) Se han identificado los distintos tipos de diagramas de comportamiento.
b) Se ha reconocido el significado de los diagramas de casos de uso.
c) Se han interpretado diagramas de interacción.
d) Se han elaborado diagramas de interacción sencillos.
e) Se ha interpretado el significado de diagramas de secuencia.
f) Se han elaborado diagramas de secuencia sencillos.
g) Se ha interpretado el significado de diagramas de colaboración
h) Se han elaborado diagramas de colaboración sencillos.
i) Se ha interpretado el significado de diagramas de actividades
j) Se han elaborado diagramas de actividades sencillos.
k) Se han interpretado diagramas de estados.
l) Se han planteado diagramas de estados sencillos.

UT6 DIAGRAMAS DE CASOS DE USO - EXTRA

Índice de contenido

1.- Diferencia entre extend y generalización.....	3
2.- Diferencia entre extend e include	4
2.1.- Ejemplo de include	4
2.2.- Ejemplo de extend	4

1.- Diferencia entre extend y generalización

Un ejemplo para aclarar la diferencia entre <<extend>> y <<generalization>> en los diagramas de casos de uso:

Ejemplo Práctico:

Imagina que tienes un sistema de biblioteca en línea. Aquí hay dos casos de uso principales: Buscar Libro y Reservar Libro.

- <<generalization>> (Generalización):

- Supón que hay un caso de uso general llamado **Buscar** y dos casos de uso especializados: **Buscar Libro** y **Buscar Revista**.
- **Buscar Libro** y **Buscar Revista** son especializaciones del caso de uso **Buscar**. Ambos comparten características comunes de **Buscar** (como ingresar términos de búsqueda, recibir resultados), pero también añaden sus propias reglas o comportamientos específicos.

En este caso, usaríamos <<generalization>> porque tanto **Buscar Libro** como **Buscar Revista** son siempre versiones más específicas de **Buscar**.

- <<extend>> (Extender):

- Considera el caso de uso **Buscar Libro**. Ahora imagina que existe un caso de uso opcional llamado **Ver Reseñas de Libros** que sólo ocurre si el usuario decide clicar en un botón de "Ver Reseñas" después de buscar un libro.
- **Ver Reseñas de Libros** no es una versión especializada de Buscar Libro, sino más bien una funcionalidad adicional que se puede realizar o no, dependiendo de la acción del usuario. No siempre se activa cada vez que se busca un libro.

Aquí utilizaríamos **<<extend>>** porque Ver Reseñas de Libros extiende el caso de uso de Buscar Libro de manera condicional, es decir, solo se ejecuta bajo ciertas condiciones (el usuario decide ver las reseñas).

2.- Diferencia entre extend e include

Vamos a utilizar un sistema de **compras en línea** para ilustrar cómo funcionan las relaciones **<<include>>** y **<<extend>>** en los diagramas de casos de uso, proporcionando un ejemplo práctico de cada uno.

2.1.- Ejemplo de include

Imaginemos que tenemos un caso de uso general llamado **Realizar Compra**. En este proceso de compra, hay varios pasos fundamentales que siempre deben ocurrir:

1. **Seleccionar Productos:** El usuario elige productos para comprar.
2. **Ingresar Información de Pago:** El usuario proporciona detalles de su método de pago.
3. **Confirmar Compra:** El usuario revisa y confirma la compra.

Sin embargo, para que la compra sea válida, siempre es necesario **Verificar Stock** de los productos seleccionados para asegurarse de que están disponibles. Dado que esta verificación es un paso esencial y siempre necesario antes de finalizar la compra, se relacionaría con **Realizar Compra** mediante **<<include>>**.

Es decir, el caso de uso **Verificar Stock** es incluido en Realizar Compra porque es un componente necesario y se incluye siempre en el proceso de compra.

2.2.- Ejemplo de extend

Ahora, supongamos que hay una opción adicional en el sistema para **Aplicar Cupones de Descuento**.

Este no es un paso necesario para completar una compra, pero proporciona una funcionalidad adicional que los usuarios pueden elegir utilizar:

1. **Aplicar Cupones de Descuento:** Si el usuario tiene un cupón, puede optar por aplicarlo para obtener un descuento en su compra.

Dado que no todos los usuarios tienen cupones o desean usarlos en cada compra, este caso de uso no se ejecuta siempre. Se activaría solo si el usuario decide introducir un código de cupón durante el proceso de compra.

Por lo tanto, **Aplicar Cupones de Descuento** se relacionaría con **Realizar Compra** mediante <<extend>>.

Esto significa que **Aplicar Cupones de Descuento** extiende **Realizar Compra** de manera condicional, activándose solo bajo circunstancias específicas.

Resumen:

- <<include>>: *Verificar Stock* es incluido en *Realizar Compra* porque es esencial y siempre necesario para el proceso.
- <<extend>>: *Aplicar Cupones de Descuento* extiende *Realizar Compra*, pero solo se ejecuta bajo condiciones específicas, es decir, si el usuario decide aplicar un cupón.