


# DISEÑO DE SISTEMAS

Juan Carlos Molina Lozano  
Docente

# CONTENIDO

- Introducción
  - Objetivos de la Clase
  - Propósito del Diagrama de Paquetes
  - Características del Diagrama de Paquetes
  - Ventajas y Desventajas
  - Elementos del Diagrama de Paquetes
  - Ejemplos Prácticos
- 

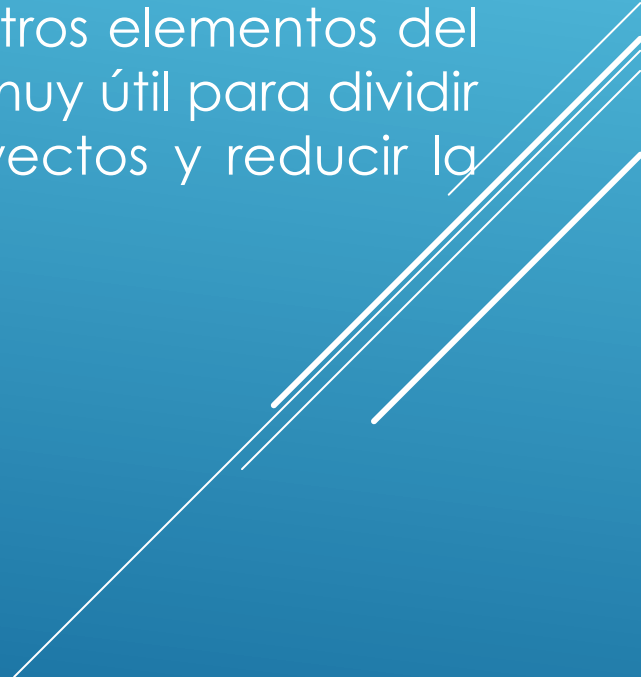
# OBJETIVOS DE LA CLASE

Organizar y representar grandes modelos de software de forma jerárquica, mediante el uso de paquetes que agrupen clases y elementos afines.



# INTRODUCCIÓN

A medida que los sistemas de software crecen en complejidad, es fundamental organizarlos en estructuras lógicas jerárquicas para facilitar su entendimiento. El diagrama de paquetes UML permite agrupar clases, interfaces u otros elementos del modelo en paquetes o módulos lógicos. Este tipo de diagrama es muy útil para dividir el trabajo entre equipos, facilitar la navegación en grandes proyectos y reducir la complejidad visual de otros diagramas como los de clases.



# QUE ES UN DIAGRAMA DE PAQUETES?

Un diagrama de paquetes es un tipo de diagrama estructural en el Lenguaje de Modelado Unificado (UML) que se utiliza para representar la organización y disposición de varios elementos dentro de un sistema. Estos elementos pueden incluir clases, interfaces, componentes y otros paquetes. Esencialmente, es una manera de agrupar elementos relacionados en "paquetes" para simplificar sistemas complejos, de manera similar a como las carpetas ayudan a organizar archivos en un ordenador.

Imaginemos que tenemos un sistema muy grande, como un software bancario. No podemos ver todo de golpe, así que lo dividimos en partes como "Autenticación", "Cuentas", "Pagos", "Reportes", etc. Cada una de estas partes sería un paquete. El diagrama nos muestra cómo están conectadas entre sí.

# PROPÓSITO DEL DIAGRAMA DE PAQUETES

El propósito principal de un diagrama de paquetes es proporcionar una vista de alto nivel de un sistema, mostrando cómo diferentes partes del sistema están agrupadas juntas y cómo interactúan entre sí. Esto hace que los diagramas de paquetes sean particularmente útiles en grandes proyectos de software donde comprender la estructura general es crucial tanto para el desarrollo como para el mantenimiento.

Por ejemplo:

Vista de alto nivel: Miras un mapa de un país con ciudades conectadas por rutas. Eso sería un diagrama de paquetes.

Vista de bajo nivel: Miras una calle específica dentro de una ciudad, con detalles de casas y edificios. Eso sería un diagrama de clases o de secuencia, por ejemplo.

# CARACTERÍSTICAS DEL DIAGRAMA DE PAQUETES

1. Representación de paquetes: En UML, un paquete se dibuja como una carpeta con pestaña y actúa como contenedor de otros elementos. Los paquetes pueden contener otros paquetes, creando una estructura jerárquica.
2. Indicadores de dependencia: Las flechas entre paquetes muestran dependencias, indicando cómo un paquete puede verse afectado por cambios en otro. Esto ayuda a entender la interacción entre partes del sistema.
3. Modularidad y encapsulación: Agrupar elementos relacionados en paquetes promueve la modularidad y permite encapsular detalles internos, facilitando el trabajo independiente dentro de cada paquete.
4. Escalabilidad y mantenibilidad: Organizar el sistema en paquetes mejora su estructura, lo que facilita añadir nuevas funcionalidades y mantener el software a lo largo del tiempo.
5. Herramienta de comunicación: Los diagramas de paquetes ayudan a comunicar la estructura del sistema de forma clara, incluso a personas no técnicas, sirviendo como apoyo en diseño, documentación y discusión del proyecto.
6. Reduce la complejidad visual.

# VENTAJAS Y DESVENTAJAS

## Ventajas

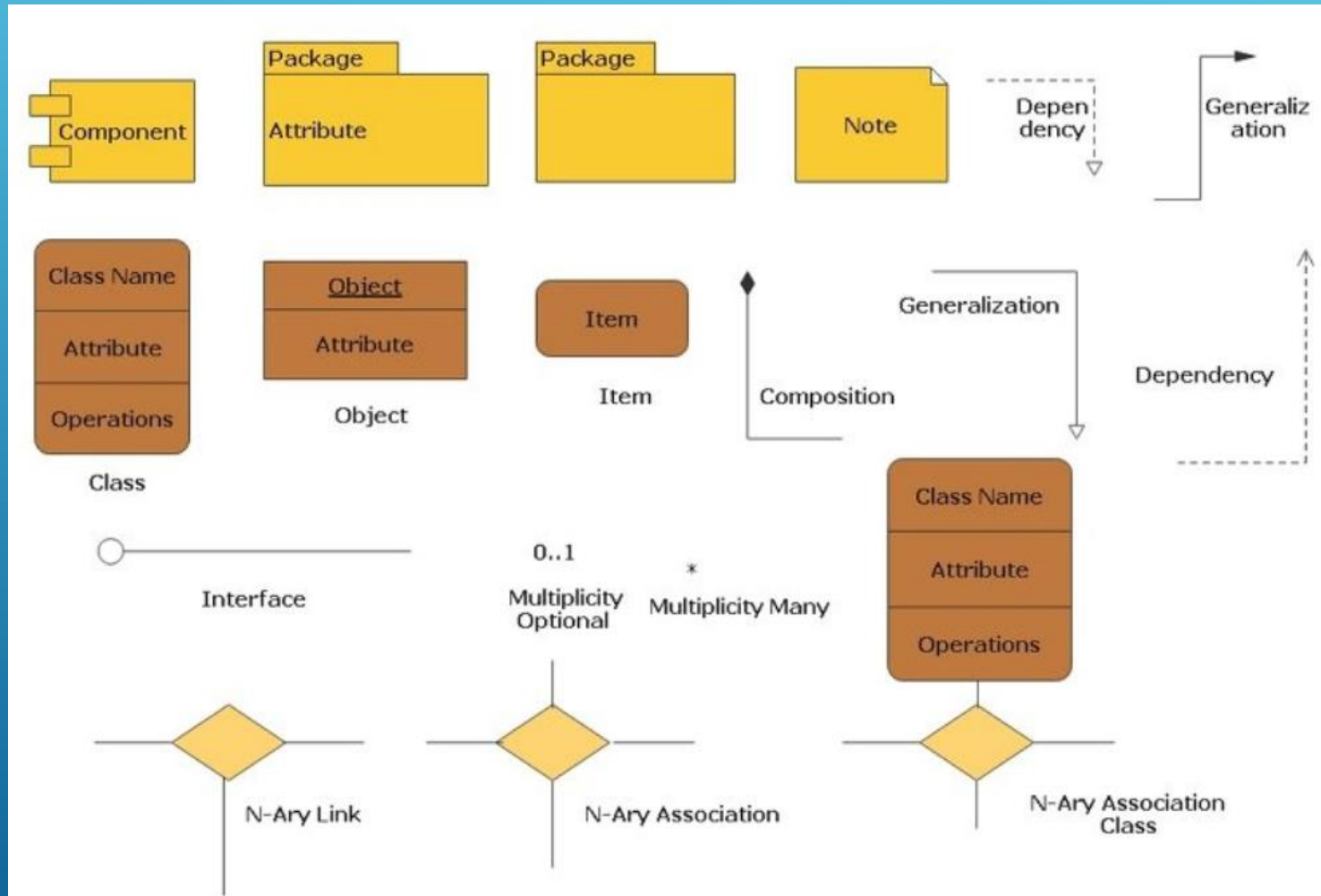
- ✓ Facilita la organización del proyecto.
- ✓ Mejora la lectura y análisis del sistema.
- ✓ Apoya la modularización.

## Desventajas

- ✗ No muestra detalles internos.
- ✗ Puede ocultar relaciones entre clases individuales.



# ELEMENTOS DE UN DIAGRAMA DE COMPONENTES

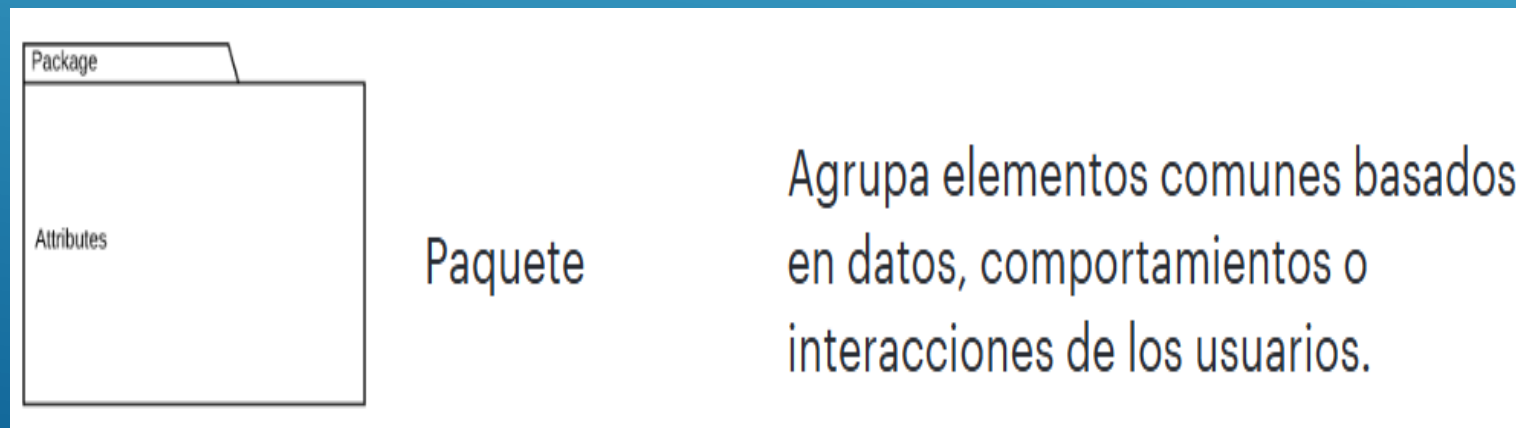


# ELEMENTOS DE UN DIAGRAMA DE COMPONENTES

## Paquete

Estos son los bloques de construcción de un diagrama de paquetes. Piensa en un paquete como un contenedor que contiene varios elementos UML como clases, interfaces o incluso otros paquetes. Es similar a una carpeta en un sistema de archivos.

Cada elemento contenido dentro del paquete debe tener un nombre único.



# ELEMENTOS DE UN DIAGRAMA DE COMPONENTES

## Dependencia

Representación gráfica de cómo un elemento (o un conjunto de elementos) depende de otro o influye a otro. Las dependencias se dividen en dos grupos: dependencias de acceso y de importación (<<import>> y <<access>>).

Estas son las líneas que conectan paquetes, ilustrando cómo los cambios en un paquete podrían afectar a otro. Son vitales para entender la naturaleza interconectada de los componentes del sistema.



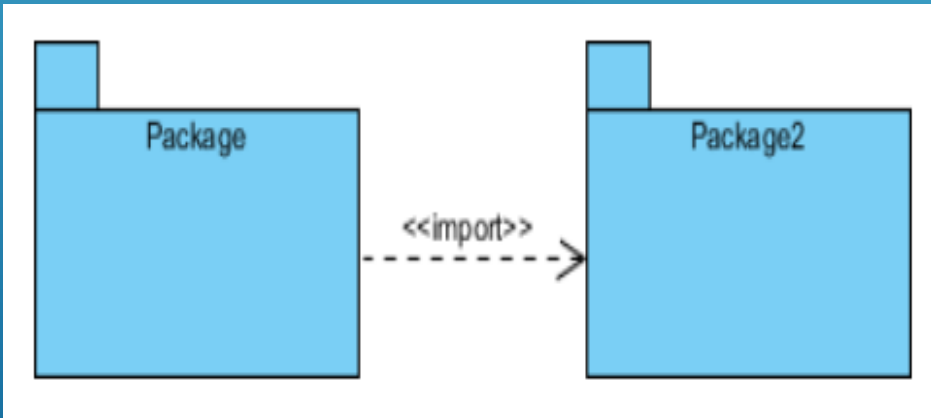
Dependencia

Muestra la relación entre un elemento (paquete, elemento nombrado, etc.) y otro.

# ELEMENTOS DE UN DIAGRAMA DE COMPONENTES

Ejemplo de diagrama de paquete - Importación

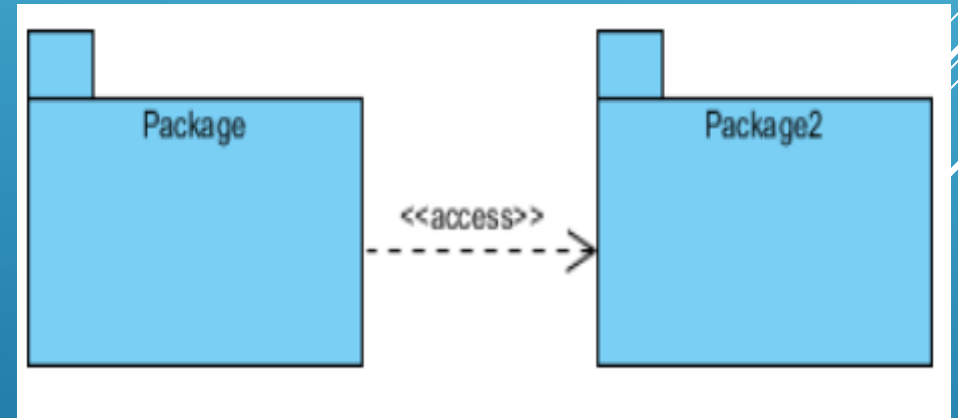
`<<import>>` - un paquete importa la funcionalidad de otro paquete



## Dependencia

Ejemplo de diagrama de paquete - Access

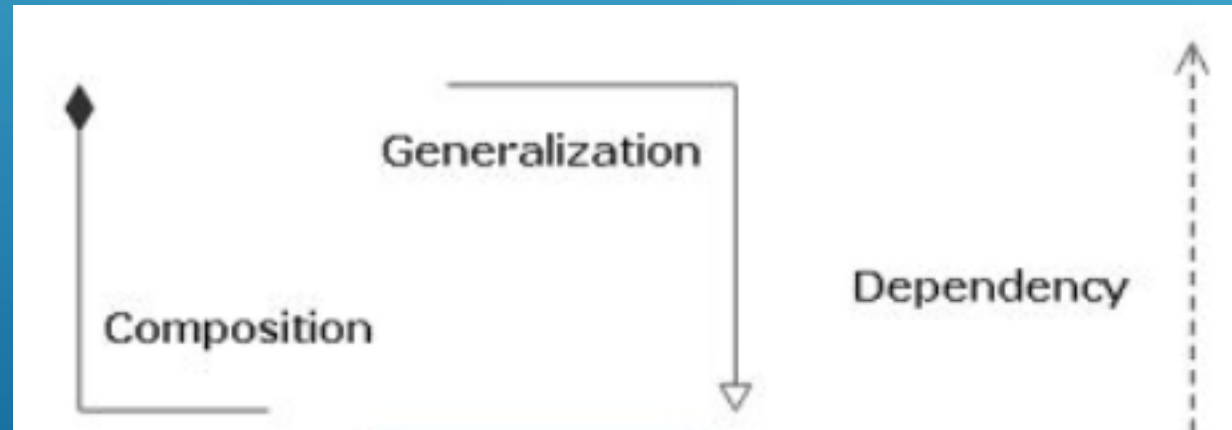
`<<acceso>>` - un paquete requiere ayuda de las funciones de otro paquete.



# ELEMENTOS DE UN DIAGRAMA DE COMPONENTES

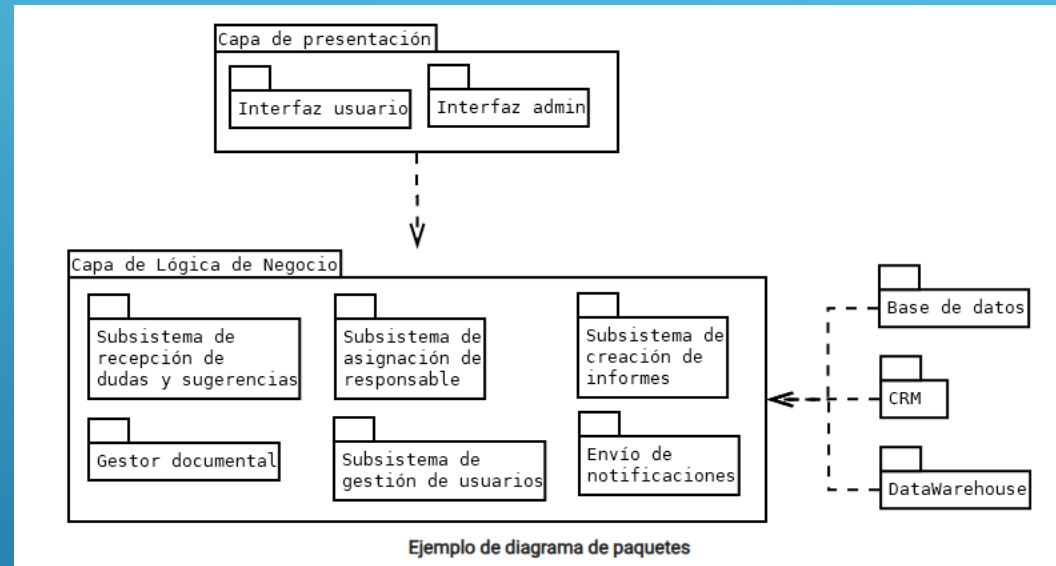
## Relaciones

Relaciones: Más allá de las dependencias, las relaciones en un diagrama de paquetes muestran asociaciones, generalizaciones y realizaciones entre paquetes, ofreciendo perspectivas sobre la jerarquía y estructura del sistema.



# EJEMPLO

## Diagrama de paquetes



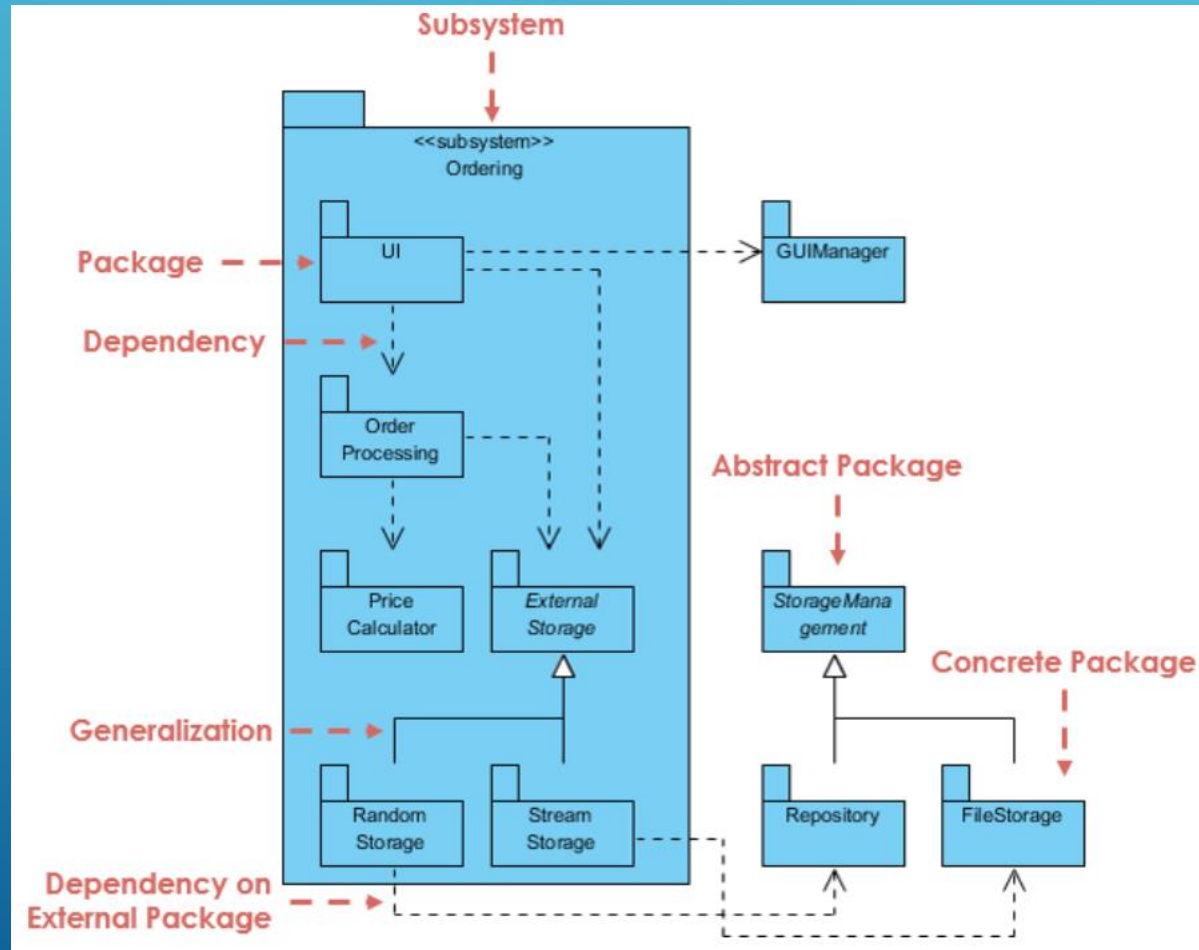
# EJEMPLO

## Diagrama de paquetes



# EJEMPLO

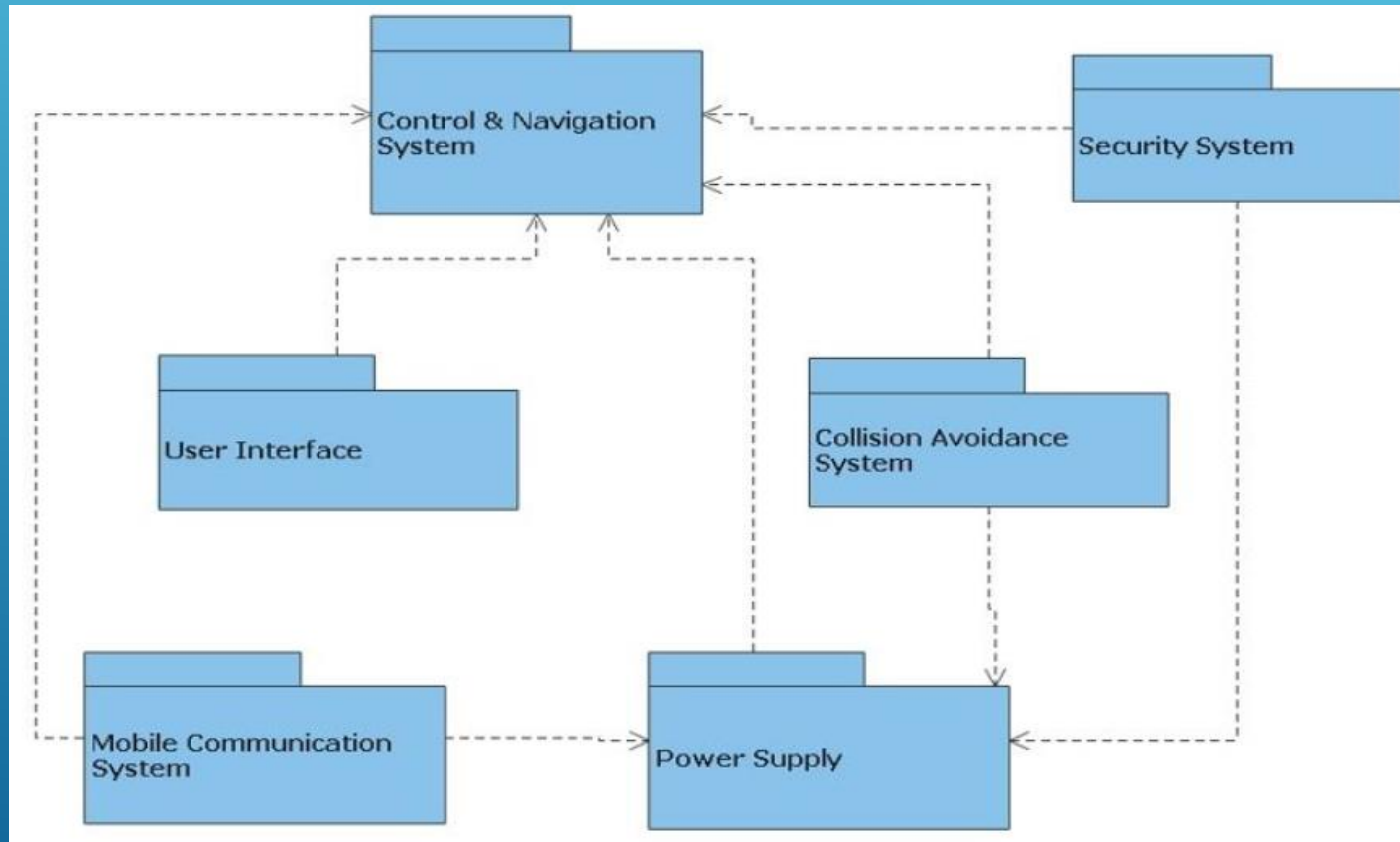
## Diagrama de paquetes: subsistema de pedidos





# EJEMPLO

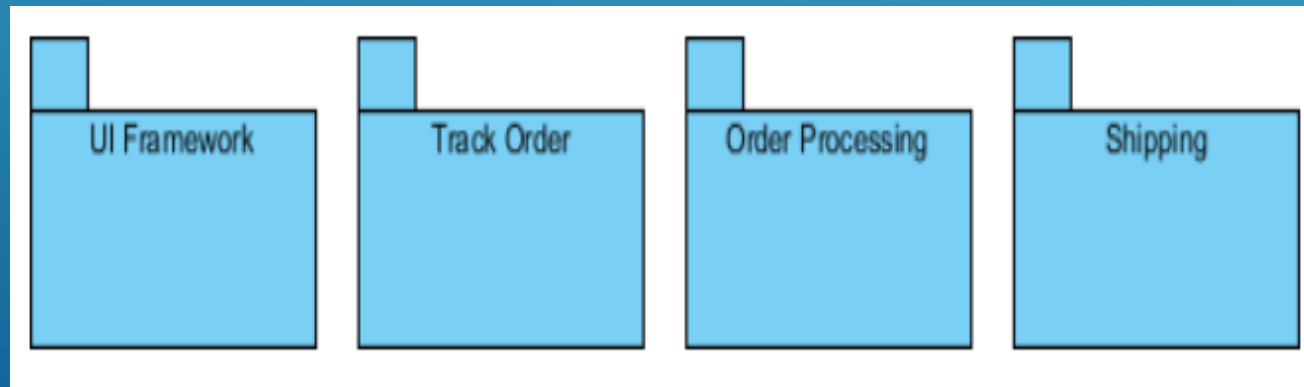
## Diagrama de paquetes: Sistema de control y navegación



# PRACTICA

Sistema de procesamiento de pedidos: descripción del problema

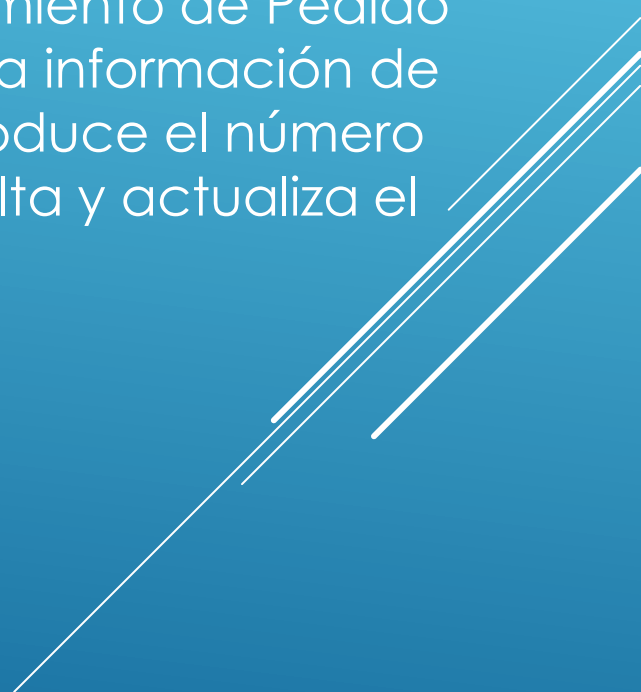
Diseñaremos un diagrama de paquete para el escenario de "Seguimiento de Pedido" de una tienda online. El módulo Seguimiento de Pedido proporciona información de seguimiento de los productos pedidos por los clientes. El cliente introduce el número de serie de seguimiento. El módulo Seguimiento de Pedido lo consulta y actualiza el estado del envío.



# PRACTICA

Sistema de procesamiento de pedidos: descripción del problema

Diseñaremos un diagrama de paquete para el escenario de "Seguimiento de Pedido" de una tienda online. El módulo Seguimiento de Pedido proporciona información de seguimiento de los productos pedidos por los clientes. El cliente introduce el número de serie de seguimiento. El módulo Seguimiento de Pedido lo consulta y actualiza el estado del envío.

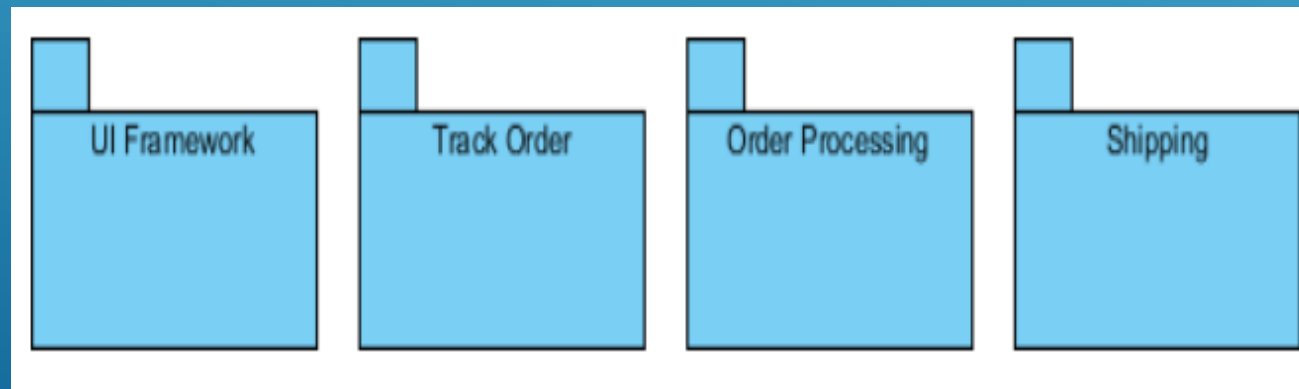


# PRACTICA

## 1. Identificar los paquetes del sistema

Hay un módulo de seguimiento de pedidos, que debe comunicarse con otro módulo para conocer los detalles del pedido. Lo llamaremos "Procesamiento de pedidos".

Luego de obtener los detalles del pedido, debe conocer los detalles de envío, lo llamaremos "Envío".

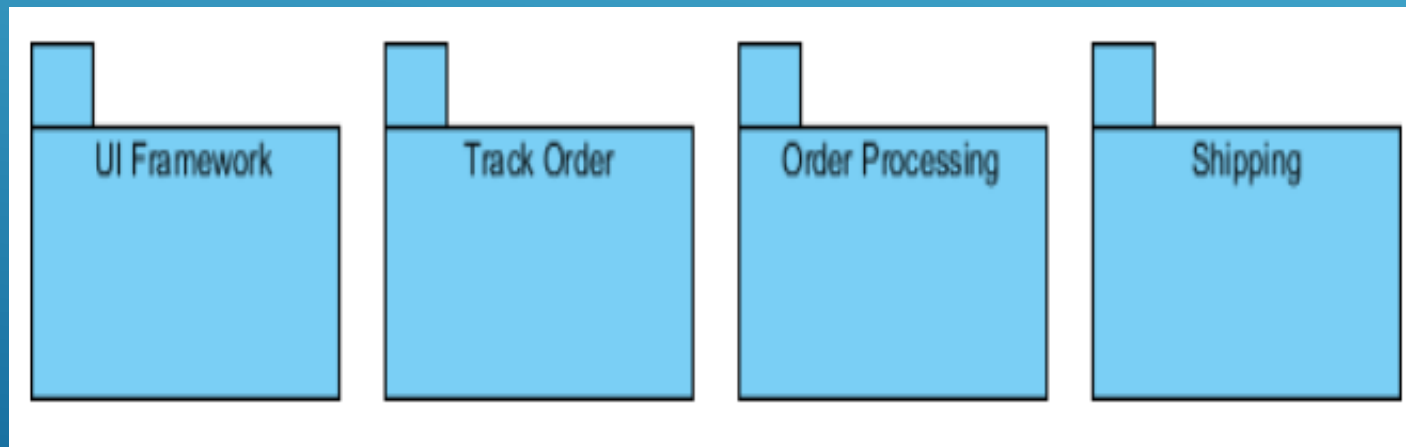


# PRACTICA

## 1. Identificar los paquetes del sistema

Hay un módulo de seguimiento de pedidos, que debe comunicarse con otro módulo para conocer los detalles del pedido. Lo llamaremos "Procesamiento de pedidos".

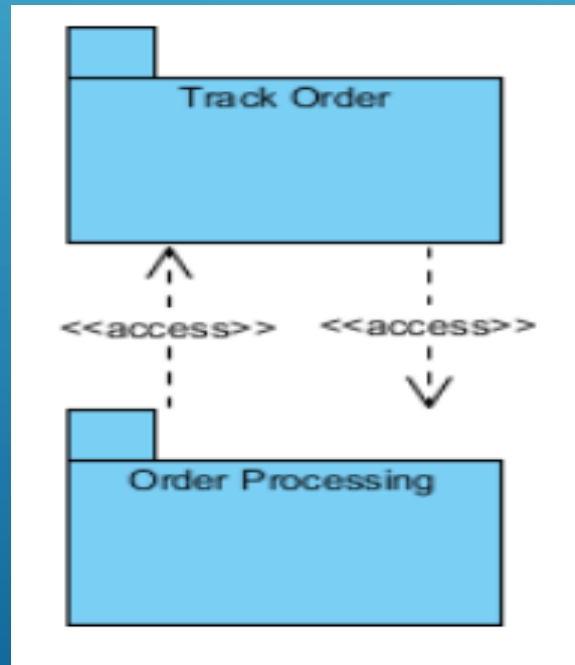
Luego de obtener los detalles del pedido, debe conocer los detalles de envío, lo llamaremos "Envío".



# PRACTICA

## 2. Identificar las dependencias en el Sistema

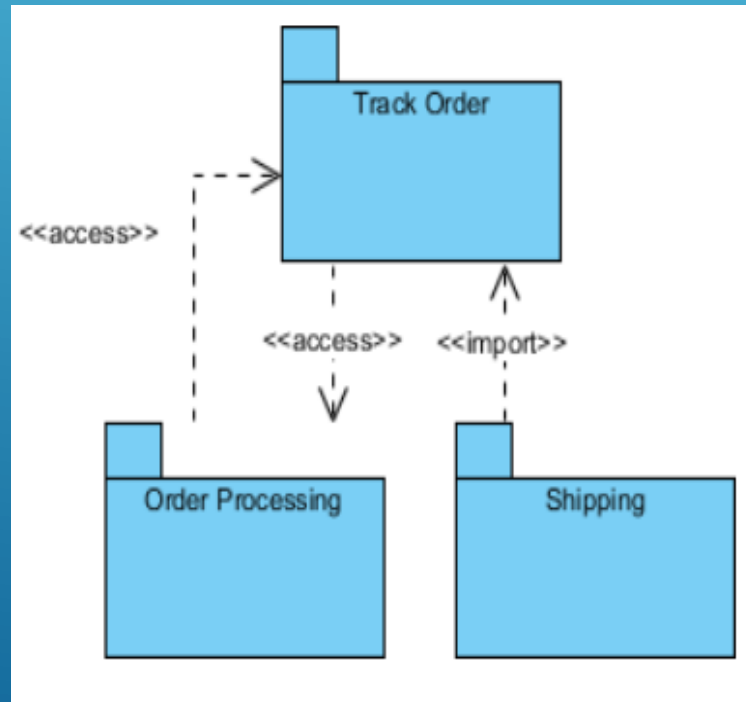
El seguimiento del pedido debe obtener los detalles del pedido desde "Procesamiento de pedidos", y "Procesamiento de pedidos" debe conocer la información de seguimiento proporcionada por el cliente. Los dos módulos se acceden entre sí, lo que permite la doble dependencia de "acceso".



# PRACTICA

## 2. Identificar las dependencias en el Sistema

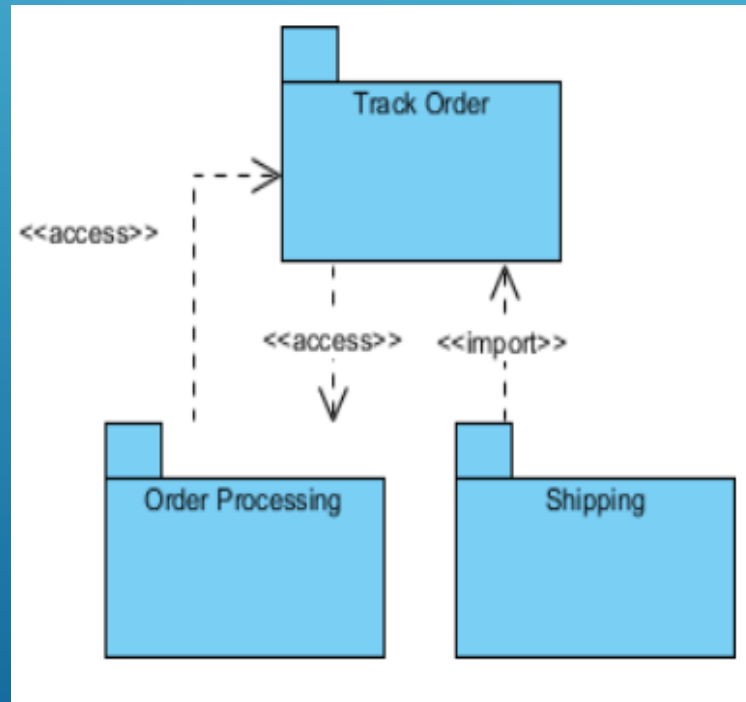
Para conocer la información de envío, "Envío" puede importar "Seguimiento de pedido" para facilitar la navegación.



# PRACTICA

## 2. Identificar las dependencias en el Sistema

Para conocer la información de envío, "Envío" puede importar "Seguimiento de pedido" para facilitar la navegación.

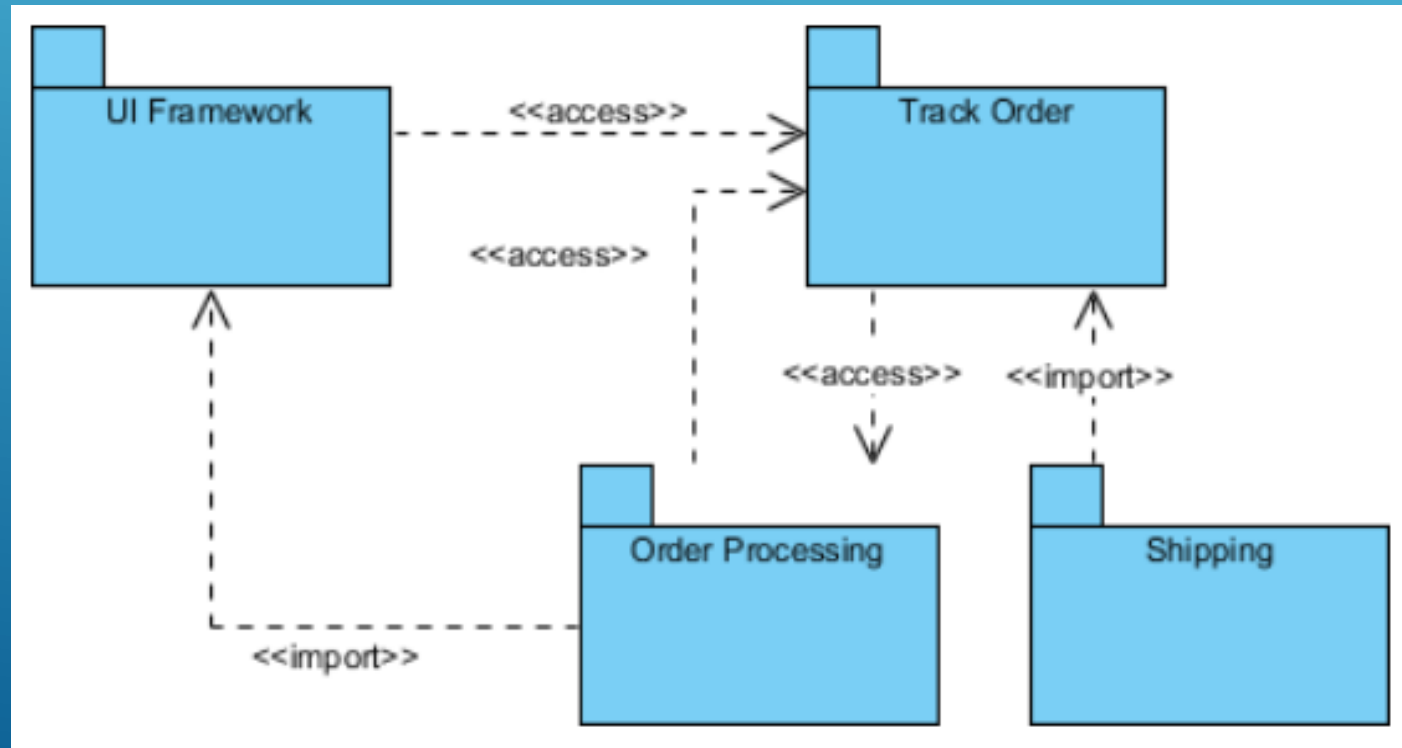




# PRACTICA

## 2. Identificar las dependencias en el Sistema

Finalmente, también se asigna la dependencia de seguimiento de pedidos al marco de interfaz de usuario, lo que completa nuestro diagrama de paquetes para el subsistema de procesamiento de pedidos.



# PRACTICA

Ejemplo 1: Sistema de Videollamadas

Componentes principales:

InterfazUsuario

ControlLlamadas

ServidorStreaming

BaseDeDatos

Relaciones:

InterfazUsuario depende de ControlLlamadas

ControlLlamadas usa servicios de ServidorStreaming

ControlLlamadas accede a la BaseDeDatos

# PRACTICA

## Ejemplo 2: Sistema Académico

### Componentes principales:

GestiónEstudiantes

GestiónCursos

GestiónCalificaciones

BaseDeDatosAcadémica

### Relaciones:

GestiónCalificaciones accede a GestiónEstudiantes y GestiónCursos

Todos acceden a BaseDeDatosAcadémica para persistencia

# PREGUNTAS

