UML estáticos y de estructura

Ing. Luis Guillermo Molero Suárez

CC BY

Diagrama de clases

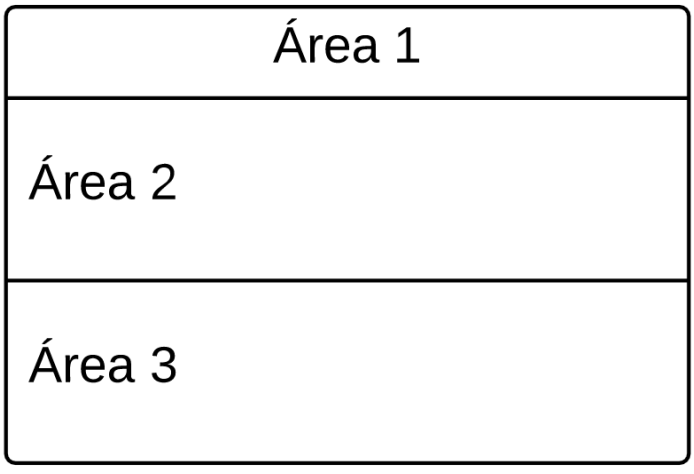
El diagrama de clases compone la organización inicial de un modelo orientado a objetos, bien sea de un sistema completo, de un subsistema o un componente. Asimismo, se muestran las clases con un número reducido de métodos y atributos, es decir, características, restricciones y relaciones, entre ellas, asociación, generalización, dependencia. (Pantaleo & Rinaudo, 2015)

Este diagrama, se orienta únicamente hacia el modelo de desarrollo orientado a objetos, ya que establece las clases que serán utilizadas en la etapa de construcción del software y la forma en cómo se relaciona cada clase. (¿Qué es UML? ¿Qué diagramas componen UML?, s.f.)

Elementos de un diagrama de clase

- **Clases**
  - **Nombre de la clase**: Representa una clase dentro del paradigma orientado a objetos. Este elemento define una entidad dentro del sistema o un grupo de objetos que tienen características similares en cuanto a condiciones y/o significado. Una forma práctica de identificar las clases en un enunciado es buscar los sustantivos que se muestran en el mismo. Algunos ejemplos pueden ser: Persona, animal, mensaje
  - **Atributos**: Enuncia las características de la clase, cuáles son los atributos que representan a ese objeto, ejemplo: tamaño, forma, edad, peso.

- **Métodos**: Identifica cuál será el comportamiento que se van a crear a partir de esa clase, y son de tres tipos: Constructores, accesores y modificadores y operacionales.



Donde, en el ***área uno*** se coloca el nombre de la clase.
Donde, en el ***área dos***, se coloca línea por línea los atributos de la clase de la forma siguiente:

visibilidad nombre_atributo: tipo = valor-inicial { propiedades }

Donde la visibilidad puede ser:

- **(+) Pública**: representa que se puede tener acceso al atributo desde cualquier parte de la aplicación
- **(-) Privada**: solo se puede acceder al atributo desde la misma clase y protegida
- **(#) Protegida**: se puede tener acceso al atributo a partir de la propia clase o de las clases que hereden de ella.

Sabías que…
El atributo también se puede colocar de la forma:

nombre del atributo y tipo

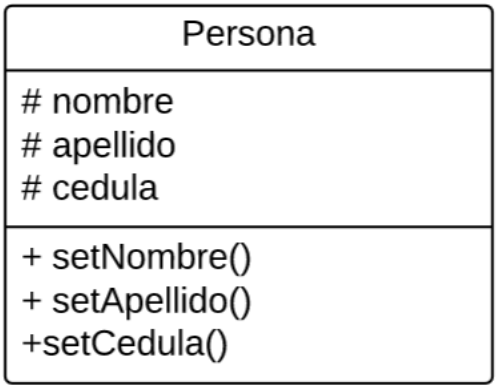
o

nombre del atributo.

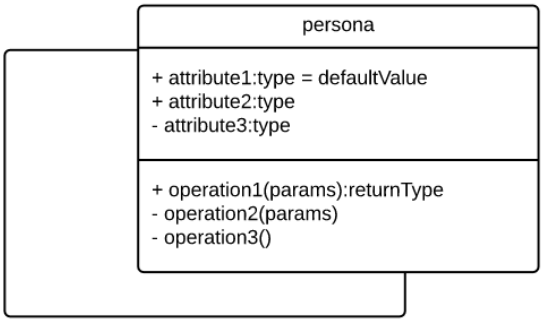
Donde, en el ***área tres***, se coloca el nombre de las funciones (métodos) que cuentan dentro de la clase

visibilidad nombre_funcion (parametros) : tipo-devuelto { propiedades }

Sabías que…
Al igual que el atributo, en la función también se puede colocar el *nombre de la función* y el *tipo*.

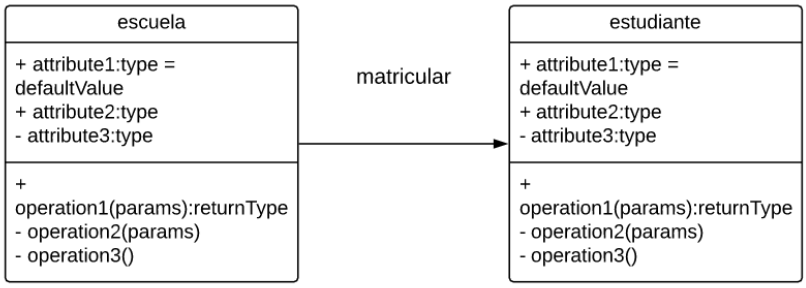


- **Relaciones**
  - **Dependencia**: Puede ser entre dos o más clases o una clase hacia sí misma (dependencia reflexiva)
    Se representa con una línea que une a las clases que varía dependiendo del tipo de relación.

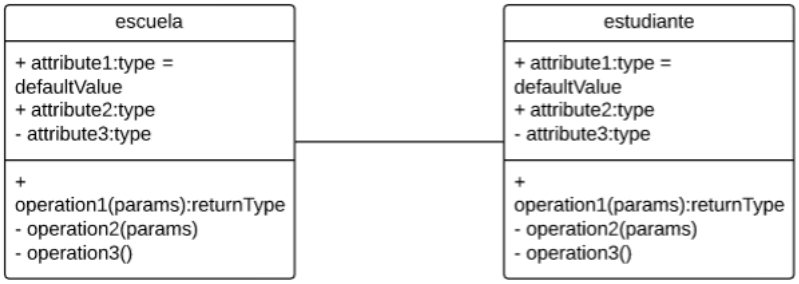


  - **Multiplicidad**: identifica el número de elementos dentro de una relación. Para ello, se utiliza: “1”, “0”, “*”, “n”.

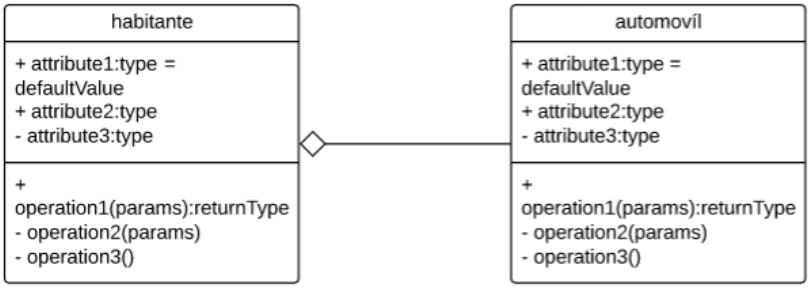
- **Nombre de la asociación**: Suele utilizarse un verbo entre las dos clases que define la acción.



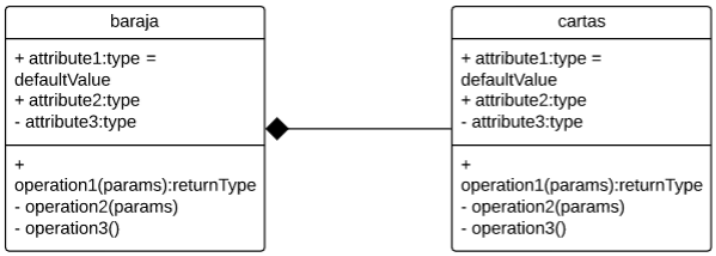
- **Tipos de relaciones**
  - **Asociación**: Es una de las más comunes relaciones y representa una dependencia semántica.



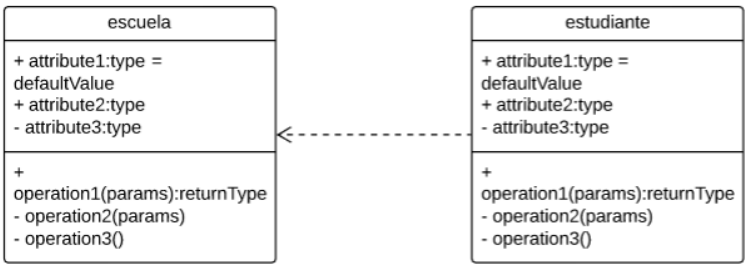
  - **Agregación**: Relación en la cual, la existencia del objeto de una clase no está limitada a la existencia del objeto de la clase a la que se asocia.



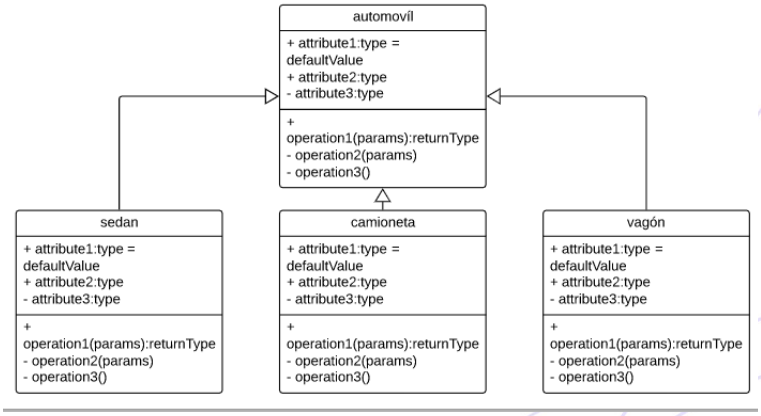
  - **Composición**: Relación en la cual, la existencia del objeto de una clase está limitada a la existencia del objeto de la clase a la que se asocia, es decir, es relación altamente acoplada.



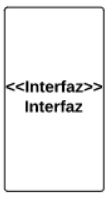
- **Dependencia**: Una relación donde una clase depende de otra para ofrecer sus funcionalidades.



- **Herencia**: Una clase recibe los atributos y métodos de otra clase, por ende, estos se adhieren a los atributos y métodos que la clase por sí misma ya tiene.



- **Interfaces**: Es un conjunto de métodos abstractos que definen un comportamiento que puede ser compartido por varias clases.

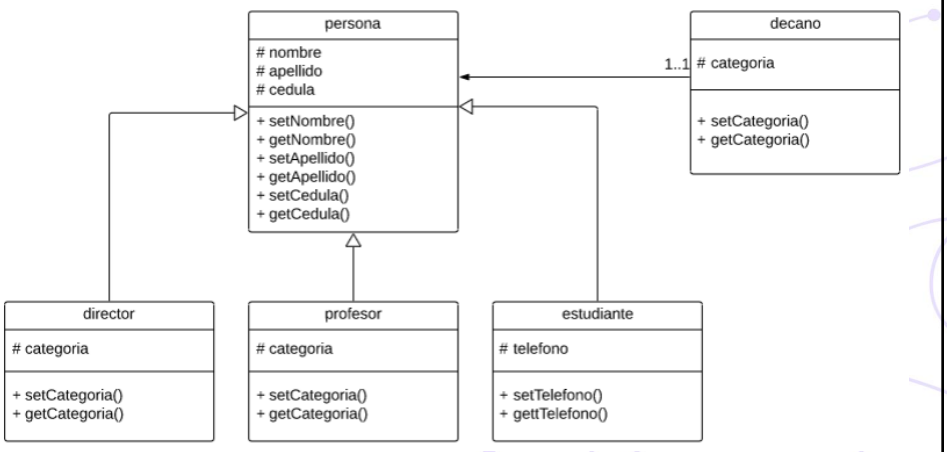


Ejercicios diagrama de clases

Ejercicio 1

Una universidad necesita almacenar información sobre las decanaturas, empleados y estudiantes, los cuales se identifican por su nombre, apellido y cédula.
Asimismo, las decanaturas poseen un nombre y funcionarios tales como decano, directores y profesores donde cada uno de los cuales tiene una categoría y un grupo de empleados subordinados.
Por parte de los estudiantes, se necesita sus datos acerca como nombre, apellido, y teléfono.
Finalmente, el sistema debe mostrar los datos de empleados y estudiantes.

Represente mediante un diagrama de clases la siguiente especificaciones.



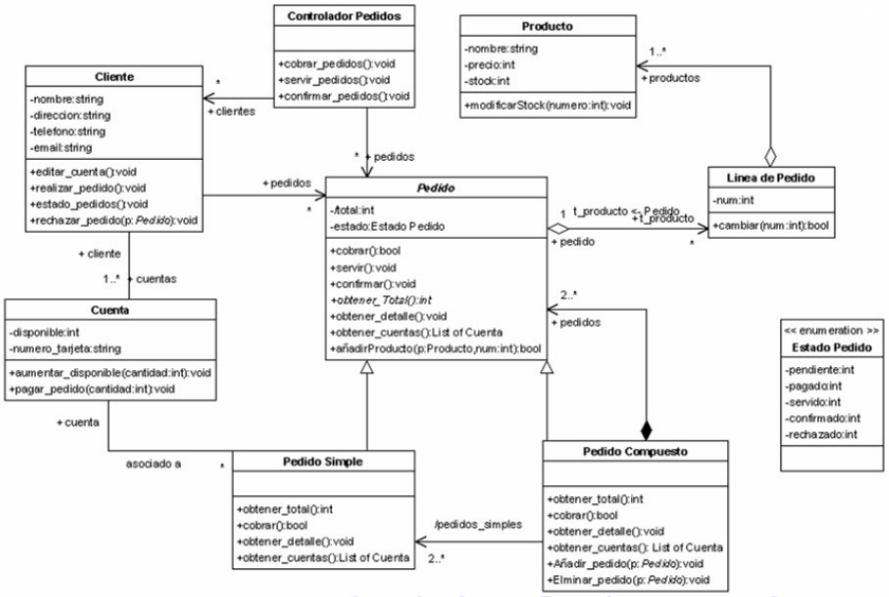
Ejercicio 2

Se quiere diseñar un sistema que permita la gestión de pedidos, por lo cual, la aplicación debe manejar las siguientes funcionalidades:

- Para el registro del cliente se deben tener en cuenta los atributos nombre, dirección, teléfono y correo electrónico.
- Un cliente puede registrar una o varias cuentas para cancelar los pedidos.
- Un cliente puede iniciar un pedido si tiene disponibilidad de dinero en alguna cuenta.
- Un cliente puede agrupar los pedidos como simples o compuestos, donde los pedidos simples están asociados a una cuenta de pago y pueden contienen hasta un máximo de 20 unidades de productos.

Existe una clase (de la cual debe haber una única instancia en la aplicación) responsable del cobro, orden de distribución y confirmación de los pedidos.

El cobro de los pedidos se hace una vez al día, y el proceso consiste en comprobar todos los pedidos pendientes de cobro, y cobrarlos de la cuenta de pago correspondiente.



Enlaces recomendados:

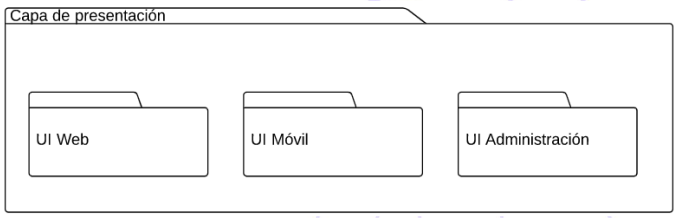
	Los siguientes recursos, sirven para consolidar conocimientos en el diagrama de clases de UML. • https://www.youtube.com/watch?v=kNV-NvBuH7M • https://www.youtube.com/watch?v=NEmZZX8Xgns • https://www.youtube.com/watch?v=wkTuCIScFQI • https://www.youtube.com/watch?v=sKqMDsDoeQ8
	Para más información acerca de los diagramas de clases de UML, puedes consultar: • Introducción al UML: Lenguaje para modelar objetos • Sistemas organizacionales. Teoría y práctica Disponibles en: https://books.google.es/

Diagrama de paquetes

Una de las situaciones más cotidianas en el desarrollo de sistemas orientados a objetos consiste en dividir un sistema de gran volumen en sistemas más pequeños, en virtud, de que los sistemas de gran volumen se hacen más complejos en su manejo, así como también, más complejos de entender sus cambios. (Forwel & Scott, 1997).

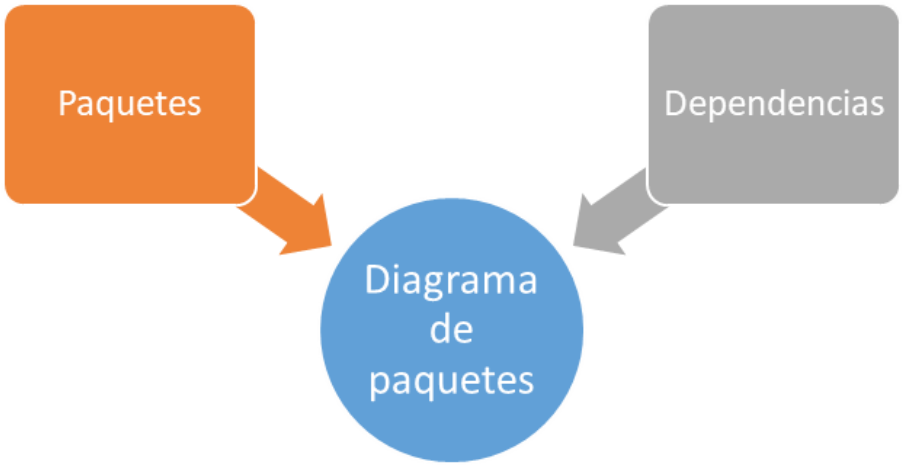
Para ello, los procedimientos estructurados se apalancaron en la descomposición funcional, esto hace caso en el sentido que un sistemas holísticamente hablando se define como un conjunto de subfunciones, que se va desintegrando en más subfunciones y así sucesivamente, en paquetes.

De lo anterior se desprende, que el diagrama de paquetes permite separar en unidades funcionales (paquetes/componentes) un gran sistema de información que está siendo modelado, lo cual no es más que la definición lógica de los distintos paquetes que forman la aplicación y cuál es su dependencia entre ellos. (Diagrama de paquetes, s.f.).

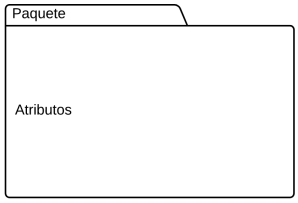


Elementos de un diagrama de paquetes

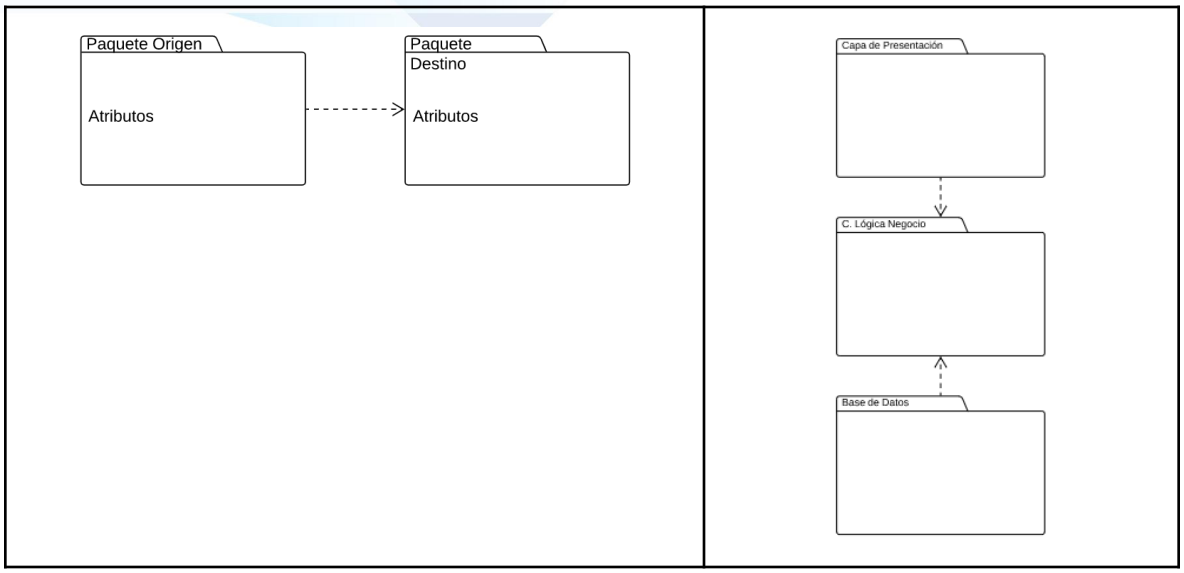
El diagrama de paquetes está compuesto por dos elementos: los paquetes y las dependencias entre paquetes así.



- **Paquete**: es un conjunto de elementos (funcionalidades: clases, casos de uso, componentes, entre otros) que maximizan la cohesión y otorgará la máxima claridad al diagrama, en conclusión, al sistema.



- **Dependencia**: muestra las relaciones entre los distintos paquetes.



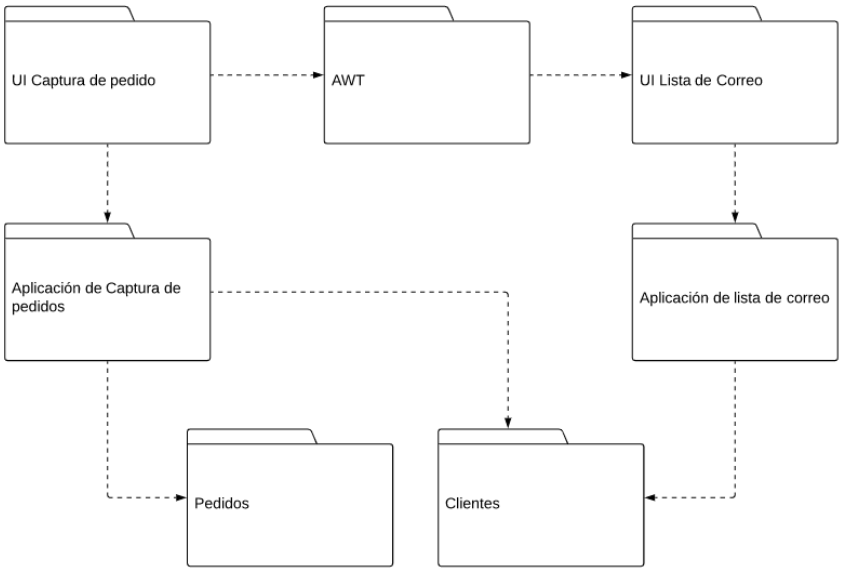
Ejercicios diagrama de paquetes

Ejercicio 1

Se desea realizar un sistema para clientes y pedidos, que cuenta con dos clases del mismo nombre, asimismo, ambos paquetes constituyen parte de un paquete que engloba todo el dominio.

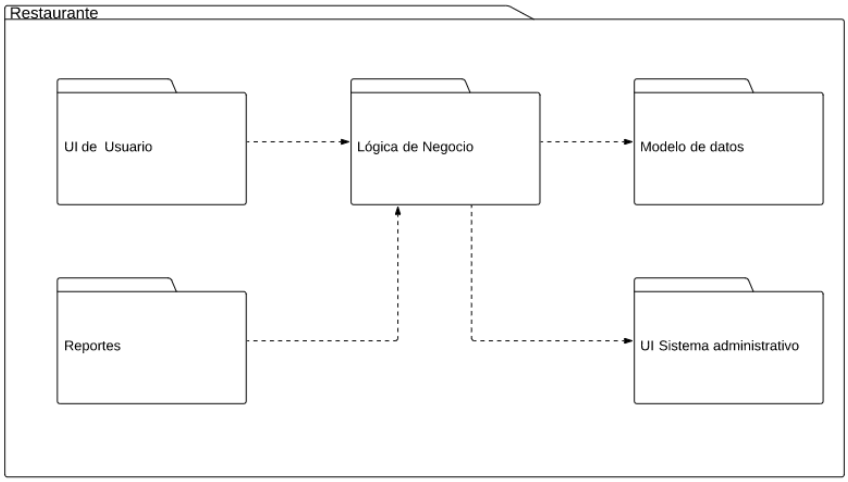
Por otra parte, la aplicación de captura de datos tiene dependencias con los otros dos paquetes del dominio.

La IU (Interfaz de Usuario) para capturar pedidos tiene subordinaciones con la aplicación “captura de pedidos” y con AWT (Juego de herramientas GUI de Java). (Forwel & Scott, 1997)



Ejercicio 2

Se desea diseñar un sistemas para un restaurante para lo cual se hace indispensable diseñar el modelo de paquetes inicial.



Enlaces recomendados

	Los siguientes recursos, sirven para consolidar conocimientos en el diagrama de paquetes de UML. • https://www.youtube.com/watch?v=ataioNckj-E • https://www.youtube.com/watch?v=MetjzGM4d5w
	Para más información acerca de los diagramas de clases de UML, puedes consultar: • Introducción al UML: Lenguaje para modelar objetos • Sistemas organizacionales. Teoría y práctica Disponibles en: https://books.google.es/

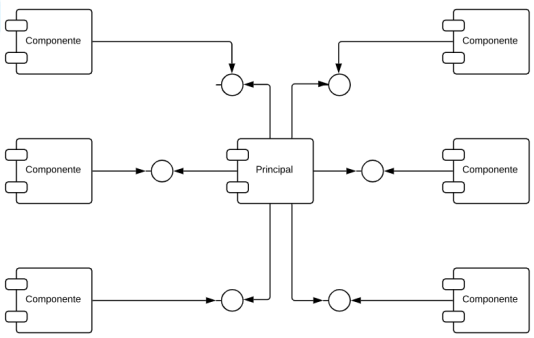
Diagrama de componentes

Los diagramas de componentes forman parte de los diagramas estáticos o de estructura y permiten documentar los componentes de los diferentes subsistemas.

Esta vista por su parte permite observar la distribución completa del sistema, en ese sentido, permite visualizar las dependencias entre los diferentes componentes de este. (Pantaleo & Rinaudo, 2015).

Este diagrama generalmente se utiliza luego del diagrama de clases, en vista de que necesita la información que arroja este diagrama, y proporciona una vista de alto nivel (estático) de todos los componentes dentro del sistema: programas ejecutables, base de datos, archivos, librerías, componentes de hardware, unidad de negocio, entre otros.

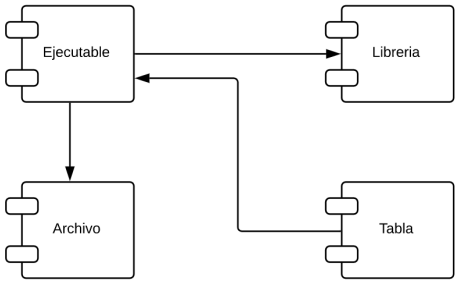
Es necesario inicialmente para su construcción, identificar los componentes a utilizar por el sistema de información, así como también, las diferentes interfaces.



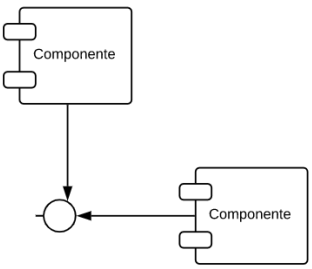
Elementos de un diagrama de componentes

- **Componente**: es una abstracción lógicas del sistema que puede representar a dos tipos de elementos: el componente lógico (componente del negocio, procesos, entre otros) y el componente físico (Python, .NET, C#).

Otra notación permite utilizar el diagrama de paquetes para diagramar la unión de varios componentes.



- **Interfaz**: este elemento siempre está ligada a un componente y su utilidad se basa en representar la zona del módulo que se utiliza para la comunicación con otro componente.

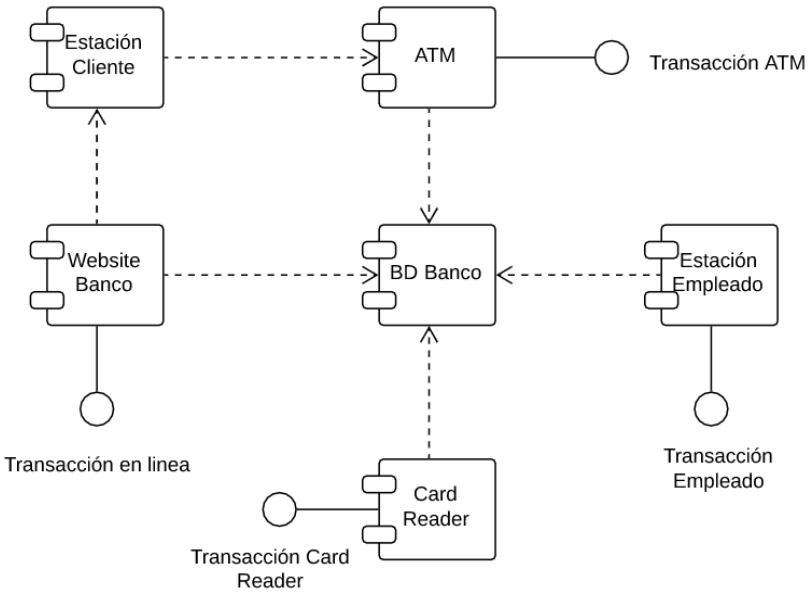


- **Relación de dependencia**: requiere que un componente requiera de otro para ejecutar su actividad, se representa por una línea punteada que conecta a un componente con otro.

Ejercicios diagrama de componentes

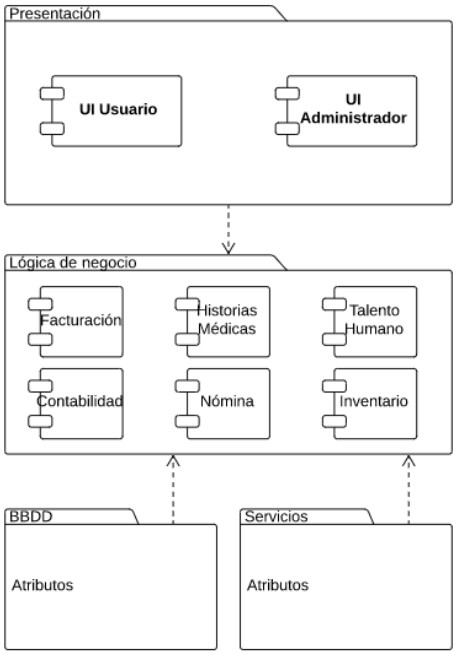
Ejercicio 1

Realice el diagrama de componentes de un sistema de gestión de transacciones bancarias incluyendo el cajero electronico (ATM).



Ejercicio 2

Realice un diagrama de componentes de una clínica.



Enlaces recomendados:

Los siguientes recursos, sirven para consolidar conocimientos en el diagrama de componentes de UML.

- https://www.youtube.com/watch?v=oOycG_n1ARs
- https://www.youtube.com/watch?v=YN1n1bQmG1k

Para más información acerca de los diagramas de clases de UML, puedes consultar:

- Introducción al UML: Lenguaje para modelar objetos
- Sistemas organizacionales. Teoría y práctica

Disponibles en: https://books.google.es/