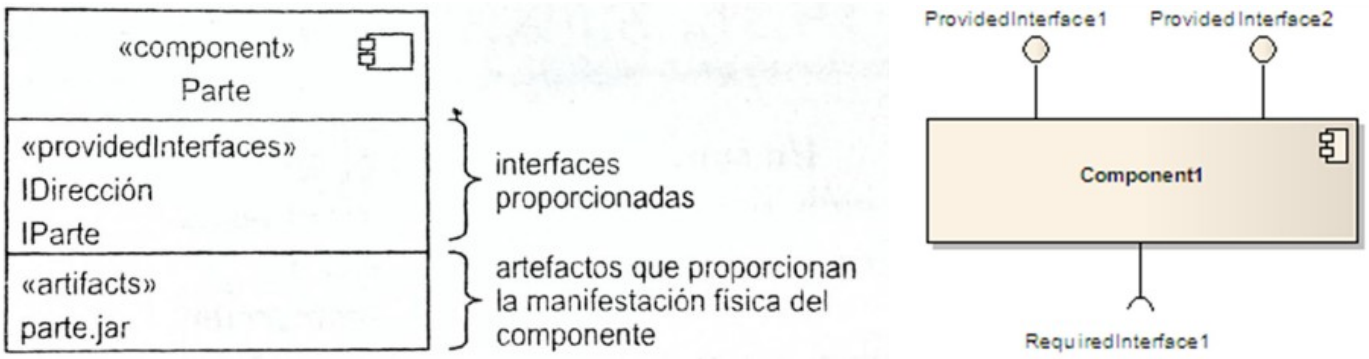
Interfaces y componentes

- ***Componente***

Representa una parte modular de un sistema que encapsula sus contenidos y cuya manifestación se reemplaza dentro de su entorno. Actúa como una caja negra cuyo comportamiento externo está completamente definido por sus interfaces proporcionadas y requeridas.
Los componentes pueden mostrarse como caja blanca.



- ***Diseño de Componente-Interfaz***

Primero se definen los componentes “las grandes piezas del sistema” y después las interfaces correspondientes.
Una vez que los componentes y las interfaces se han definido es posible dividir la implementación del sistema entre los participantes organizándolos en varios grupos los desarrolladores son libres de implementar los componentes.

- ***Diseño a partir de clases***

Utilizando las clases dominio y el método a partir de las clases se tiende más a la resolución del problema, en caso de que la complejidad de la solución se incremente o se identifique un grupo de clases que pueda depurarse y reutilizarse con más facilidad si se les encapsula en componentes.

<<Estereotipo>> - Semántica

<<buildComponent>>
Un componente que define un conjunto de elementos para fines organizativos o de desarrollo a nivel de sistema.

<<entity>>
Un componente de información persistente que representa un concepto de negocio.

<<implementation>>
Una definición de componentes que no tiene especificación, es una implementación para una <<specification>> aparte con la que tiene una dependencia.

<<specification>>
Un clasificador que especifica un dominio de objetos sin definir la implementación física de esos objetos, por ejemplo, un componente estereotipado por <<specification>> solamente tiene interfaces proporcionadas y requeridas y no clasificadores de realización.

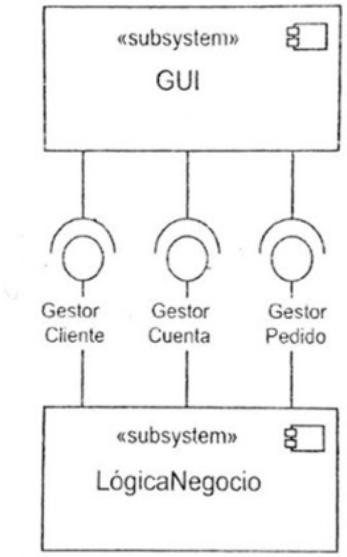
<<process>>
un componente basado en transacción.

<<service>>
Un componente funcional sin estado que computa un valor.

<<subsystem>>
Una unidad de descomposición jerárquica para grandes sistemas.

- ***Subsistema***

Un subsistema es un componente que actúa como una unidad de descomposición para un sistema mayor
Un subsistema es una construcción lógica que se utiliza para descomponer un sistema grande en bloques manejables.
Desglosar un sistema en subsistemas resuelve un problema de desarrollo difícil en muchos subproblemas más pequeños y manejables.
Un objetivo común de diseño es minimizar la comunicación y dependencias entre subsistemas.



Patrones de diseño

- ***Fachada/Facade***

Conoce cuáles clases del subsistema son responsables de una petición y delega las peticiones de los clientes en los objetos del subsistema.

Clases del subsistema: implementan la funcionalidad del subsistema, manejan el trabajo asignado por el objeto Facade y además de esto no tienen ningún conocimiento del Facade.

Clasificación del Patrón: Estructural

Intención: Proporcionar una interfaz simplificada para un grupo de subsistemas o un sistema complejo.

Motivación: Simplificar el acceso a un conjunto de clases proporcionando una única clase que todos utilizan para comunicarse con dicho conjunto de clases Reducir la complejidad y minimizar dependencias.

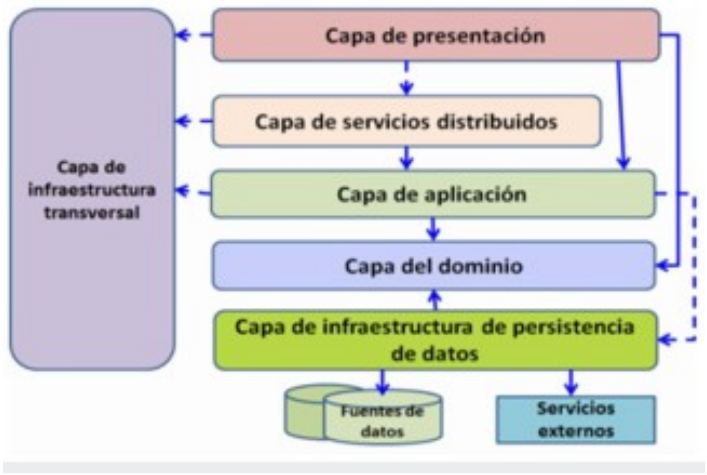
- ***Disposicion en capas***

Es un patrón de arquitectura que nos permite facilitar el entendimiento y el mantenimiento de la misma, para esto necesitamos organizar la colección de subsistemas e interfaces de forma coherente.

1. Se comienza en el nivel mas de abstraccion del sistema.
2. Se trabaja la abstraccion poniendo la capa **J** en la parte superior de la capa **J-1** hasta llegar al nivel superior de la funcionalidad, llamandolo capa **N**.

La principal característica estructural del patrón de capas es que los servicios de la capa **J** sólo son utilizados por dependencias. La responsabilidad de la capa **J** es proveer servicios usados por la capa **J+1** y delegar sub tareas a la capa **J-1**.

Ejemplo:



Presentación: responsable de mostrar información e interpretar acciones

Servicios distribuidos: Cuando una aplicación actúa como proveedor de servicios para otras aplicaciones remotas, o incluso también localizaciones pública negocio servicios. (habitualmente Servicios Web) proporciona un medio de acceso remoto basado en canales de comunicación y mensajes de datos.

Aplicación: Define los trabajos que la aplicación como tal debe de realizar y redirige a los objetos del dominio y de infraestructura (persistencia, etc.) que son los que internamente deben resolver los problemas. Sirve principalmente para coordinar la lógica del flujo del caso de uso.

Dominio: Esta capa es responsable de representar conceptos de negocio e implementación de las reglas de dominio. Estos componentes implementan la funcionalidad principal del sistema y encapsulan toda la lógica de negocio relevante.

Persistencia de datos: proporciona la capacidad de persistir datos asi como logicamente acceder a ellos. Pueden ser datos propios o datos expuestos por otros (web services)**.**