



**Universidad Tecnológica
del Norte de Guanajuato**
Organismo Público Descentralizado del Gobierno del Estado

"Educación y progreso para la vida"

Tecnologías de la información y la comunicación

Programación de aplicaciones

Unidad III Patrones de diseño

RODRÍGUEZ GARCÍA Anastacio

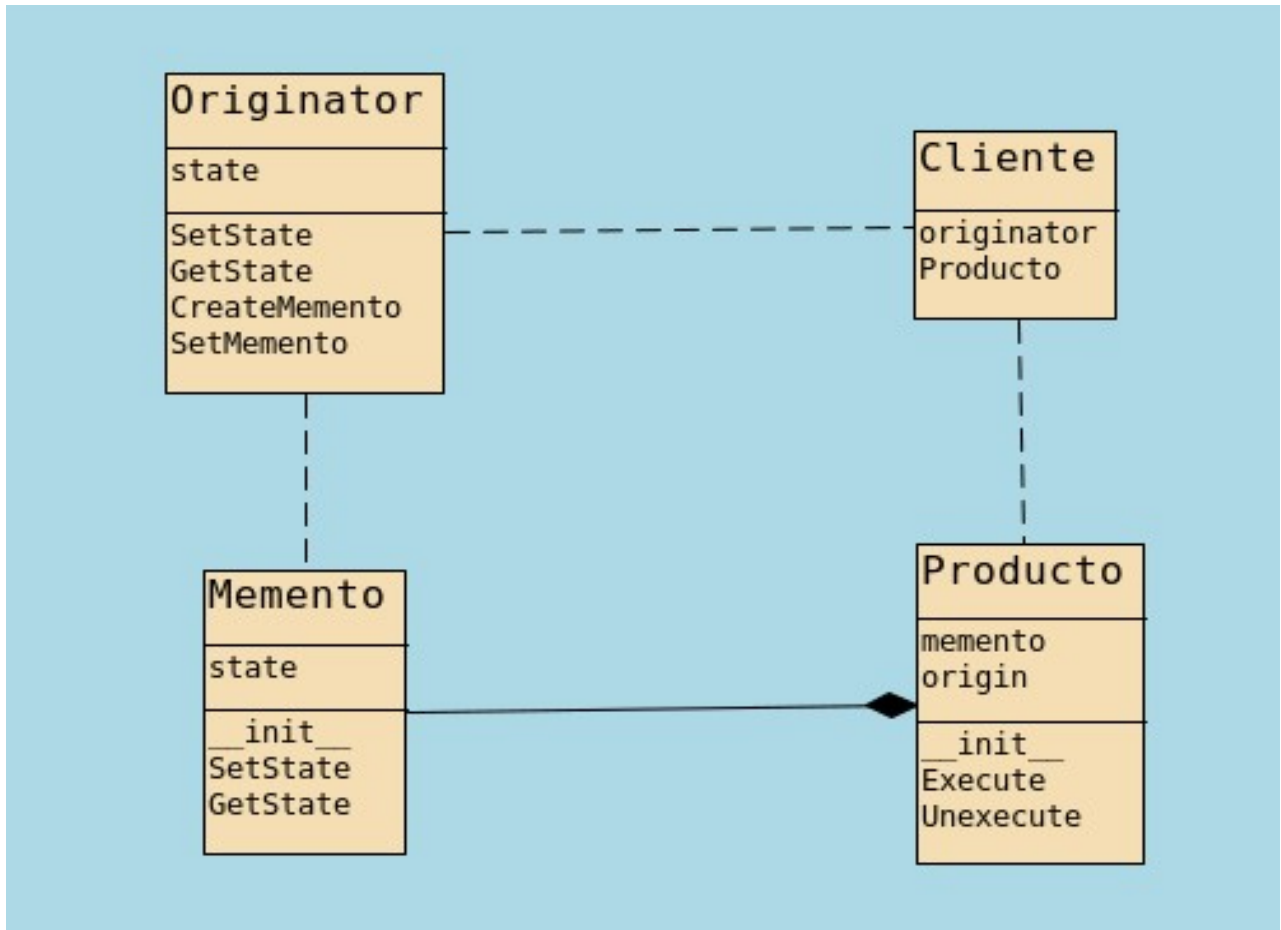
RANGEL ALVARADO Martha Guadalupe

No.Control: 1216100812

Dolores Hidalgo C.I.N a 25 de Noviembre del 2019

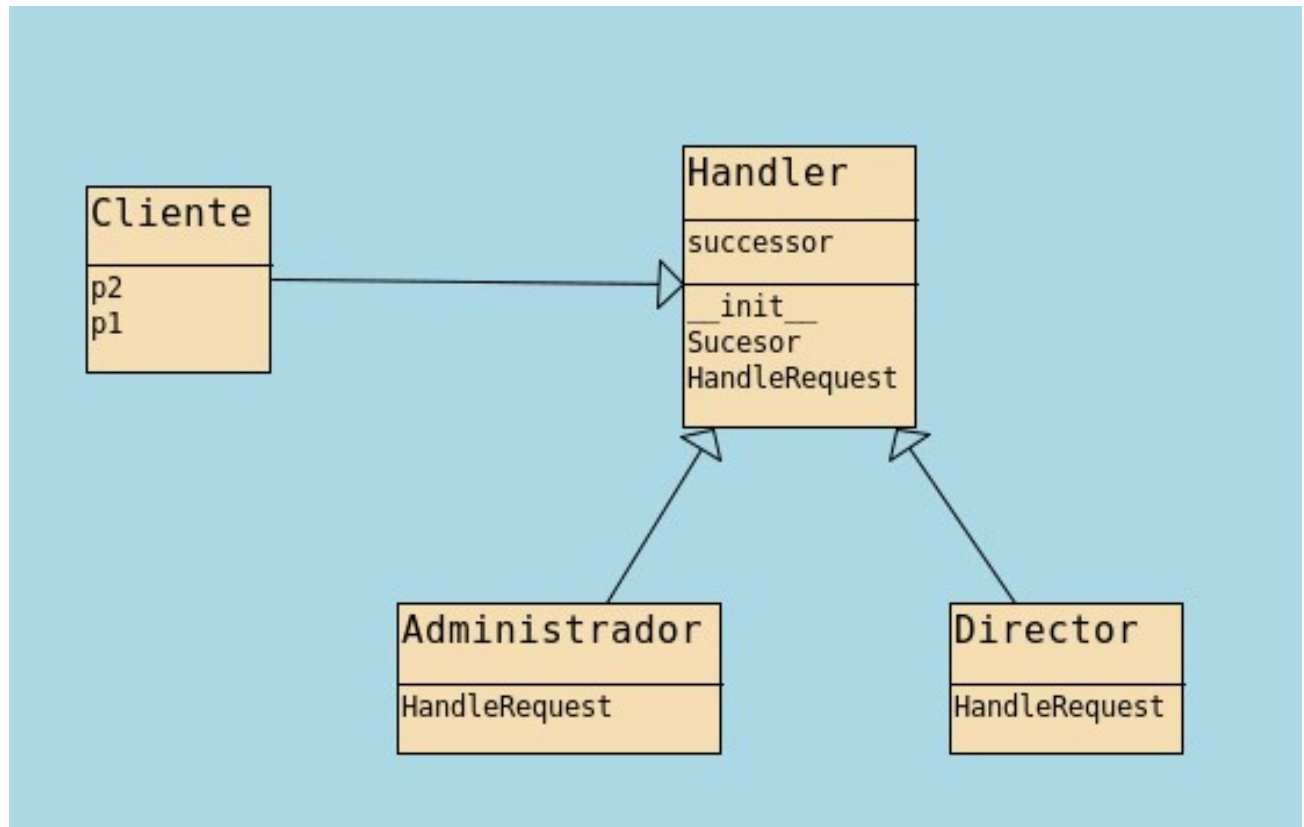
Memento

Como su nombre lo dice, este patrón de diseño permite capturar el estado interno de un objeto para poder tomarlo en algún futuro, mismo si este ya no es el mismo. Para ejemplificarlo, imaginemos un almacén que mantiene un control de las entradas y salidas de productos, y para el cual es importante conocer la cantidad en existencia de cierto producto en determinado tiempo, para poder conocerlo haremos uso del patrón memento, que nos permitirá conocer esa cantidad.



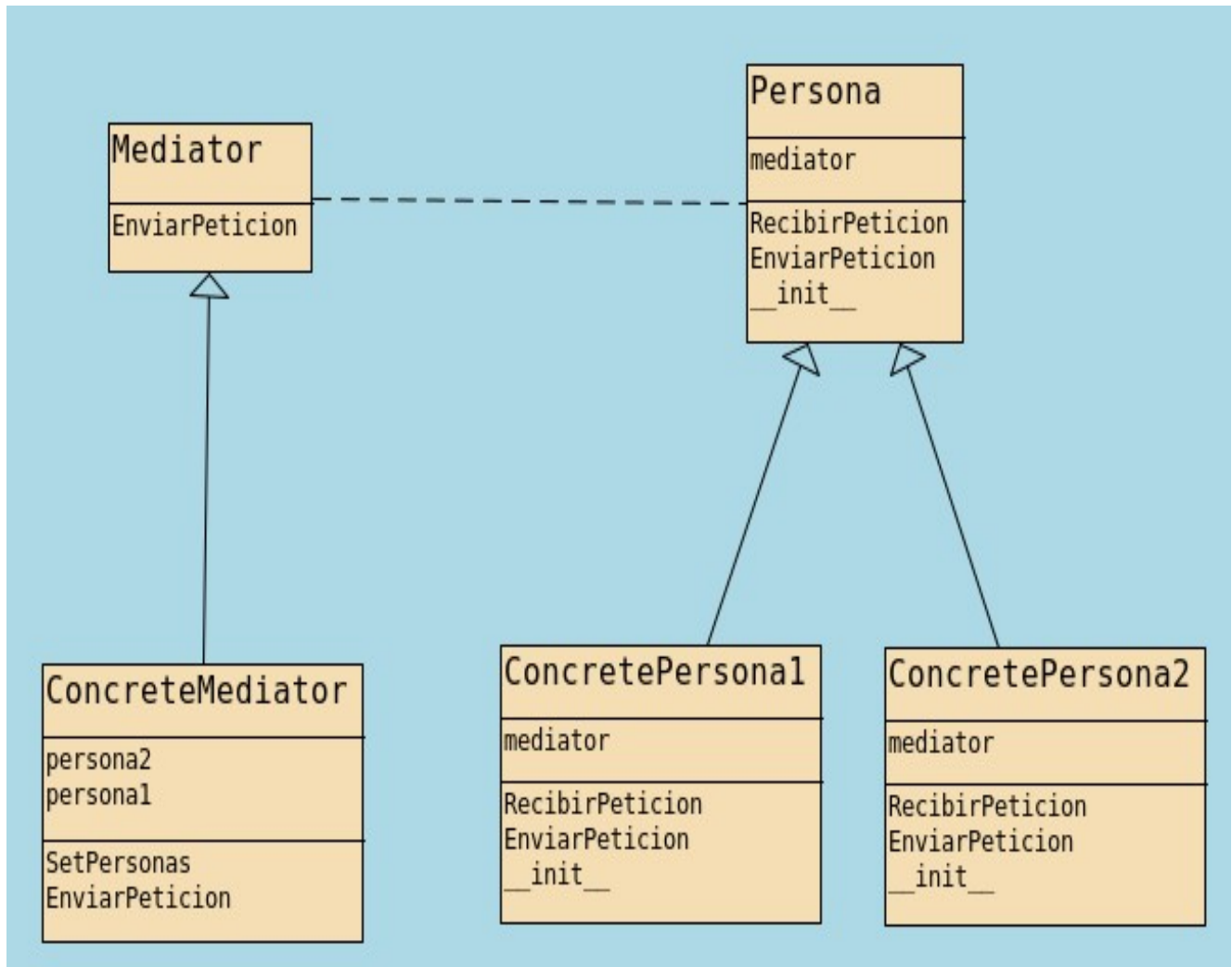
Chain of responsibility

Permite establecer una cadena de objetos receptores a través de los cuales se pasa una petición formulada por un objeto emisor. En el siguiente objeto se tienen los objetos coordinadora, administrador y director, imaginemos un hospital en el cual la coordinadora puede firmar permisos laborales pero no pases de salida sin embargo el administrador si puede autorizar pases de salida por 2 horas pero no puede autorizar vacaciones pero el director si. En este ejemplo, dependiendo la situación dependerá la persona que de respuesta a ella.



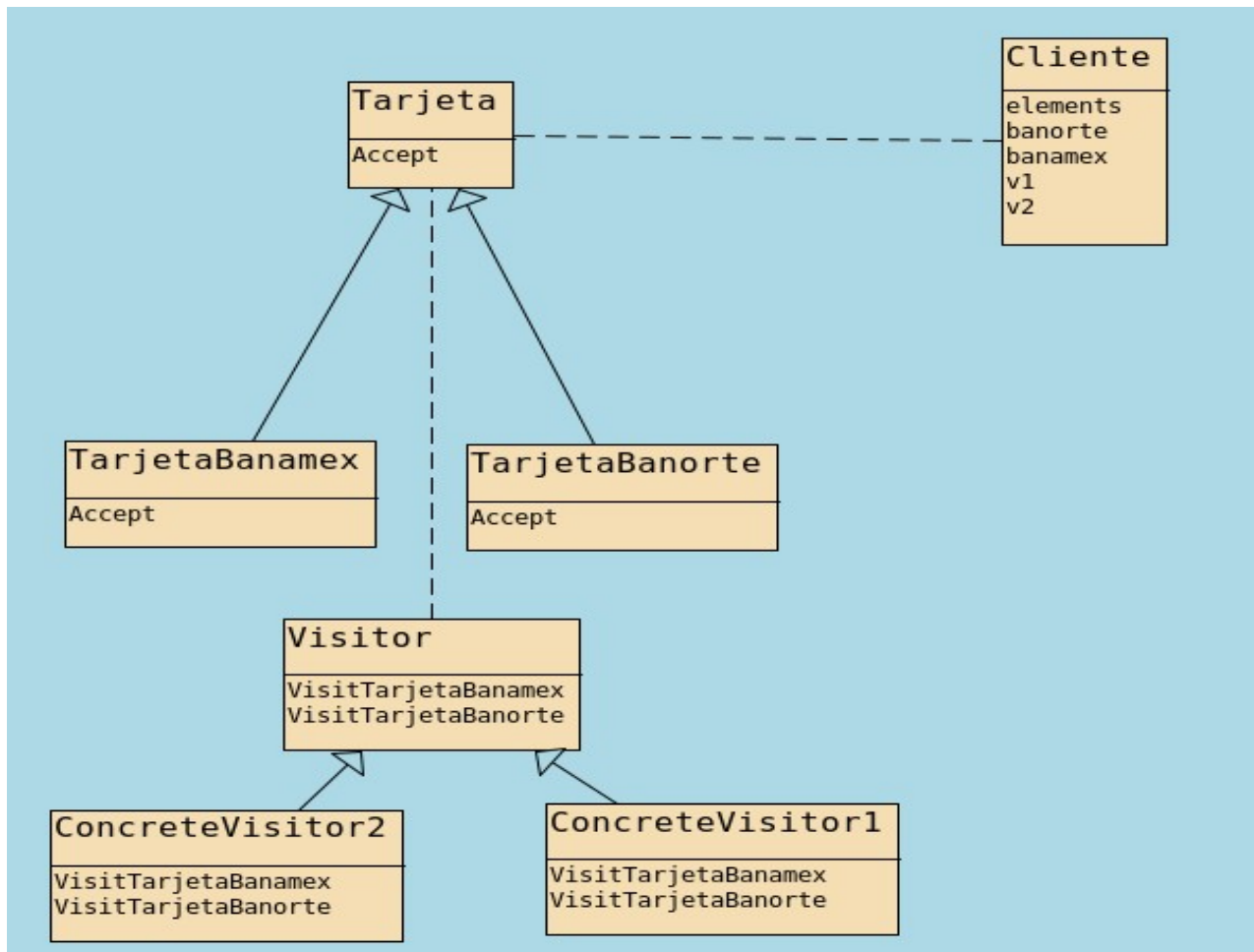
Mediador

Un Mediator es un patrón de diseño que define un objeto que hace de procesador central, coordinando las relaciones entre sus asociados o participantes. Para ejemplificar este patrón, volvamos al ejemplo de un hospital, en el cual existe un administrador encargado de satisfacer los insumos de este, entonces, una solicitud de compra de un ordenador es enviada al administrador por parte de uno de los trabajadores del hospital, entonces, para poder dar seguimiento, el administrador debe hablar con los proveedores, actuando como intermediario entre estos dos tipos de personas.



Visitor

Como ya se sabe, este patrón permite agregar diferentes funcionalidades o clases que trabajan con una misma interfaz si que esta sea modificada, es por eso que el ejemplo que se usará es el de un cajero automático que recibe diferentes tipos de tarjetas y dependiendo de ellas dependerá el costo de retiro de efectivo.



Iterator

Este patrón de diseño permite iterar sobre una lista o colección si poner en riesgo la integridad de los objetos, para ejemplificarlo se pensó en un control de correspondencia donde esta es recibida por una secretaria y va siendo almacenada de acuerdo a su número de folio, si bien la secretaria podrá recorrer toda la lista de correspondencia, no tendrá el acceso a lo que esta contiene.

