

# Decisiones de Diseño

## DocumentoComercial

Debe ser una clase con tipo y nro por estar como requerimiento en el enunciado.

valor total por el cual se realiza la operación y medio de pago, los datos del documento comercial (tipo y número de documento comercial) y los datos propios, según lo definido en “sobre las organizaciones”.

## OperacionDeEgreso

Como un egreso es un gasto realizado por la organización, sólo tiene como atributos el medio de pago, los documentos comerciales, la fecha en la que se realiza la operación y el tipo de egreso. Una operación de egreso tiene una lista de ingresos asociados que se irá llenando de acuerdo al criterio para asociarlos que elija la organización.

## OperaciónDelIngreso

La operación de ingreso tiene un monto total, una descripción y una lista de operaciones de egreso, la cual se irá llenando de acuerdo al criterio para asociarlos que elija la organización.

## TipoEgreso

Creo esta clase abstracta en caso de tener distintos egresos (como pueden ser una compra, un alquiler, un pago de sueldo, entre otros).

Tiene como método abstracto `valorTotal()` ya que debe redefinirlo cada subclase de `TipoEgreso` y lo necesito para saber cuánto dinero se va a gastar.

Tiene un método abstracto `validar()` que será redefinido en cada subclase.

## Compra

Es una de las subclases de `TipoEgreso` en la cual especifico que items se van a comprar, la suma de sus montos, el proveedor del cual se adquieren, etc.

Si bien el método de validación de la compra se encuentra en el Validador de compra, creemos apropiado que la compra pueda validarse a sí misma, por eso agregamos el método también en la compra, el cual llama al del validador.

## BandejaDeMensajes

Cada compra tiene una Bandeja de mensajes propia ya que pueden variar los mensajes guardados en cada una de ellas. Creemos que es mejor que sea una clase antes que por ej una lista de strings directamente en la compra porque tiene métodos propios como agregarMensaje o imprimirMensajes.

## TipoEntidad

Esta clase abstracta es composición (y no herencia) de la organización debido a que la Entidad puede cambiar entre Base y Jurídica.

## Actividad

Es una clase abstracta que tiene como subclases a cada una de las actividades con arrays de la cantidad máxima de empleados y el promedio máximo de ventas anuales. Las mismas tienen valores distintos en ellos, por eso las definimos como clases. Esos valores son los que se usan para definir la Estructura de la Empresa.

## ValidadorDeContraseña

Es un Singleton ya que solo va a existir uno en todo el sistema, el cual es llamado desde el usuario para comprobar si su contraseña es correcta. Hacemos esta clase separada del ValidadorDeCompra ya que validan cosas distintas, se aplican en diferentes clases y no comparten comportamiento.

## ValidadorDeCompra

Es un Singleton ya que solo va a existir uno en todo el sistema y es una clase aparte para no romper el encapsulamiento.

El método validarCompra administra los mensajes de la Bandeja de mensajes, mientras que esta última es la encargada de imprimirlos cuando un usuario revisor lo pide.

## CriterioCompra

Es una clase abstracta que la Compra tiene como atributo. Si la compra requiere presupuestos, este criterio será el que el ValidadorDeCompra use para verificar si la compra se realizó correctamente. Utilizamos composición porque creemos que puede modificarse el criterio. Por ahora sólo tenemos el criterio de MenorValor, pero pueden agregarse otros más adelante si se quiere. Cada subclase redefine el método seleccionarPresupuesto según cómo lo haga.

## Presupuesto

Tienen una lista de los ítems con sus valores (los cuales utilizo para calcular el valor total del presupuesto), un Proveedor (para saber quién lo propone ya que cada uno puede ofrecer un valor distinto de los ítems), documentos comerciales asociados y el detalle (simplemente un string).

## Ítem

Además, de cada operación de egreso de fondos se deben conocer los ítems (el detalle) incluyendo los productos o servicios incluidos (su descripción) y valor total del ítem (la restante información no es necesaria para el objetivo del sistema)

Del ítem necesito saber su descripción y el valor total ya que es requerimiento.

También cada ítem puede tener categorías, por lo que tiene como atributo una lista de ellas.

El método insertarCategoria(Categoria) se encarga de agregar una categoría a la lista verificando que no haya otra con el mismo Criterio que la que se quiere agregar.

## Criterio

Solo a modo informativo, esta clase tiene un atributo CriterioPadre, para saber la jerarquía entre criterios. También tiene un atributo nombre que sirve como identificador a fines prácticos.

## Categoría

Tiene como atributo el Criterio al cual se asocia, y un atributo nombre que sirve como identificador a fines prácticos.

## Proveedor

Es una clase abstracta de la cual heredan PersonaProveedora y EntidadJurídicaProveedora. Éstas sólo comparten la dirección postal por lo que es el único atributo que tengo en esta clase. Luego, cada subclase agrega sus propios atributos.

Creo esta clase abstracta como “clase intermedia” ya que ambas subclases deben poder ponerse como proveedor en Presupuesto y Compra.

## Empresa y OSC

Empresa y OSC heredan de organización porque la misma solamente puede ser una de las dos. Además, solo la empresa tiene una estructura (Micro, Pequeña, Mediana Tramo 1, Mediana Tramo 2).