

## Gof 2 Ej 1

Se pide:

- 1) es posible que se añadan distintos complementos en el futuro
- 2) se desea evitar una explosión de clases para cada una de las posibles combinaciones a la hora de calcular una nómina
- 3) el código que genera las órdenes de pago al banco no debe verse afectado porque una nómina tenga más o menos complementos (simplemente quiere poder calcular el total de cada empleado).
- 4) Implementar el sistema en Java que tome como entrada un fichero como el del ejemplo y saque órdenes de pago

Se propone el siguiente diseño (de izquierda a derecha):

La clase LectorFichero tendrá la lógica para extraer los datos de un fichero usando una lista de objetos Empleado para almacenarlos. Como el enunciado especifica que habrá cambios en la forma de guardar o leer datos dejamos la funcionalidad como una única clase.(4)

En la clase Empleado guardaremos la información de cada empleado incluyendo la nómina, inicialmente a cero será modificada por la cadena de responsabilidades como corresponda.

En la clase Main llamaremos a la clase CalculadoraNomina que se encargará de configurar la cadena de responsabilidad dejando a la clase Main las responsabilidades mostrar datos a través de consola (podrían delegarse en otra clase).

CalculadoraNomina desacopla la lógica de calcular la nómina del resto del programa. En caso de querer añadir un complemento (punto de expansión) se añadirá la lógica del complemento como una clase que extiende a AbstractNominaHandler y se configurará en CalculadoraNomina mediante el método setSiguiente(). (1,2 y 3)

