



## Programación (DAM-SP)

José Ramón Jiménez Reyes

### Taller mecánico

Seguimos trabajando con nuestro cliente del taller mecánico y ahora nos comenta unos nuevos requisitos a aplicar a la versión anterior que ya le habíamos mostrado.

Nos comenta que para cada trabajo los mecánicos no solo pueden añadir horas a dicho trabajo cuando éste está abierto, si no que también pueden añadir precio del material utilizado.

También nos dice que en el taller realiza tres tipos de trabajos y que para cada uno sigue unas reglas de negocio diferentes a la hora de calcular su precio final. Para todos los trabajos se cobra un precio fijo que se calcula como:  $30 * n.^{\circ} \text{ horas} + 10 * n.^{\circ} \text{ días}$ .

Además para cada trabajo se cobra un precio específico que se suma al fijo anterior y que se calcula de la siguiente forma:

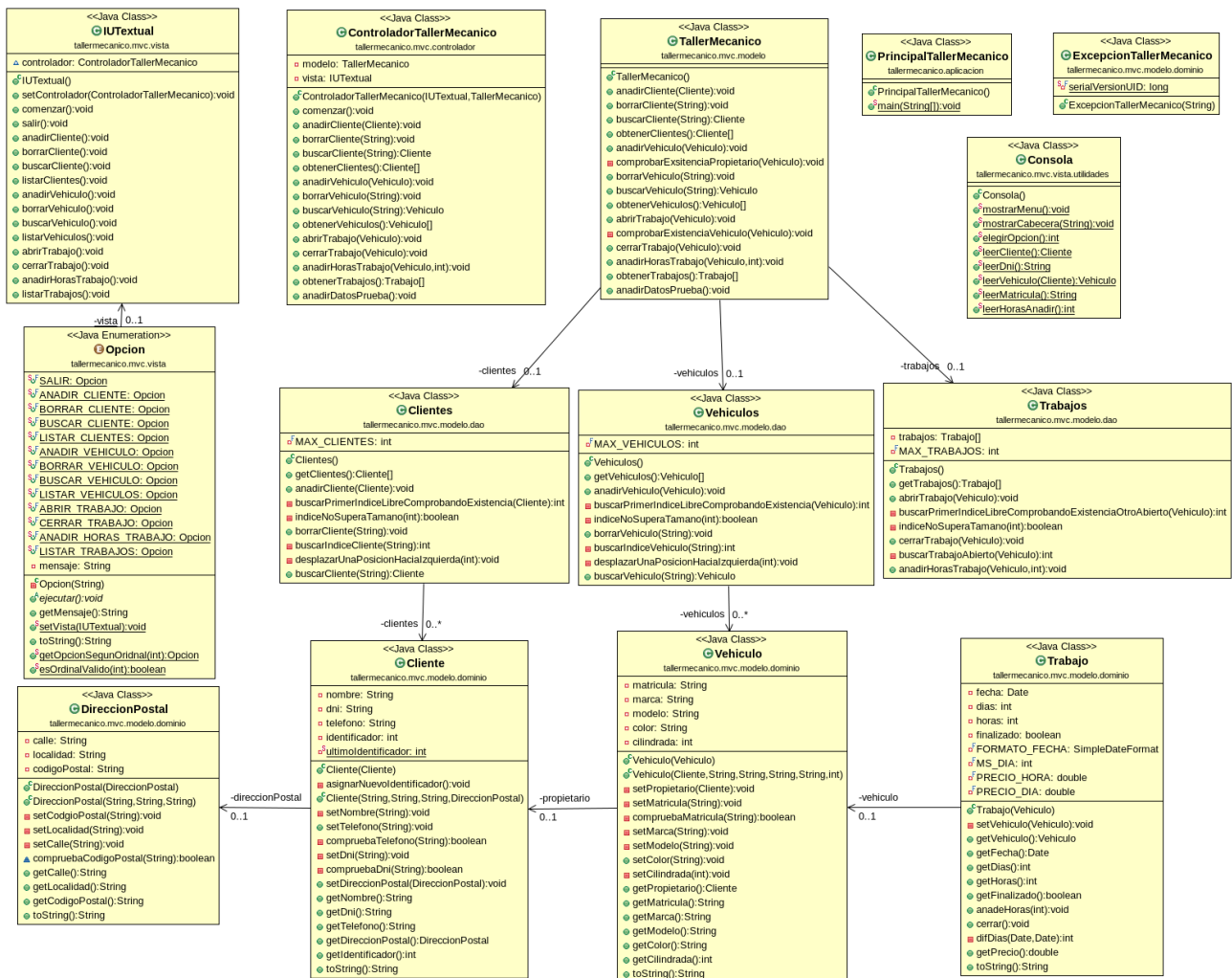
- Revisión: precio del material \* 1.0
- Chapa y pintura: precio del material \* 1.5
- Mecánico: precio del material \* 2.0

## Refactorización

Para aplicar los nuevos requisitos que nos pide el cliente, lo primero que vamos a hacer con nuestro código es refactorizarlo, para limpiar dicho código, autodocumentarlo y dejarlo preparado para nuevas ampliaciones.

Recuerda que después de cada paso en la refactorización es muy importante que lo pruebes y compruebes que todo sigue funcionando igual que al principio.

Para ello, aplicaremos todo lo visto en clase (patrón MVC, patrón DAO, implementación de un menú con un enumerado, extraer el código para hacer los métodos más pequeños y autodocumentados en cada una de sus acciones, etc.) para llegar a un diagrama de clases como el que sigue:



## Implementación de nuevos requisitos utilizando herencia y polimorfismo

Una vez que hemos refactorizado, hemos dejado el código limpio y preparado para nuevas ampliaciones y/o modificaciones y que hemos comprobado que nuestra aplicación se comporta exactamente igual que en la versión anterior, es el momento de aplicar los nuevos requisitos que nos pide el cliente.



# Programación (DAM-SP)

José Ramón Jiménez Reyes

Para aplicar los nuevos requisitos utilizaremos todos los conocimientos adquiridos sobre herencia, polimorfismo, ligadura dinámica, interfaces, clases abstractas, etc. La idea es llegar a un diagrama de clases como el que sigue y que la aplicación cumpla con los nuevos requisitos impuestos por el cliente.

