

Diseño de Sistemas - Recuperatorio Segundo Parcial

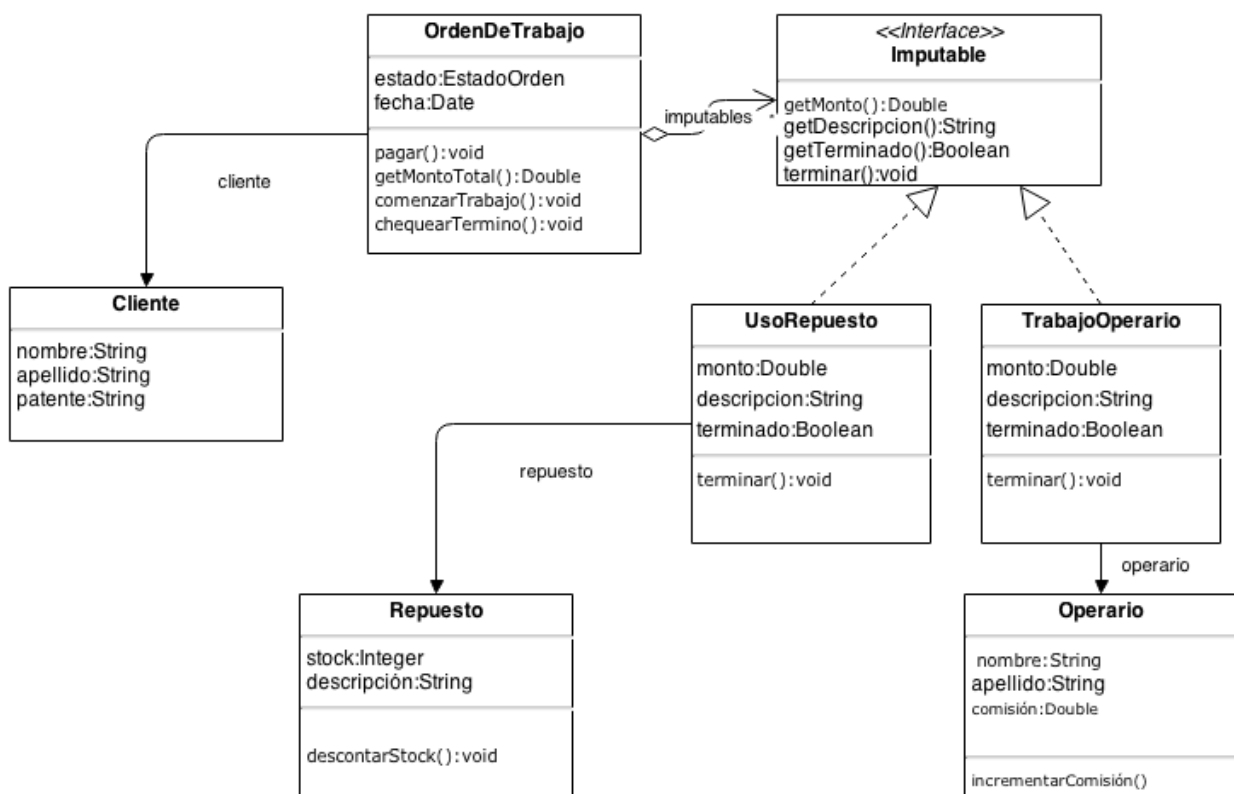
Recuerden que lo que nos interesa es la **justificación de las elecciones**, así que siempre justifiquen sus decisiones.

Siempre pueden crear nuevos objetos de considerarse necesario o modificar el diseño presentado (claro siempre justificando).

Punto 1:

Se cuenta con el siguiente diseño:

- La persistencia se accede a través de las Home que sean necesarias.
- Este sistema permite la gestión de ordenes de trabajo de un taller mecánico, su presupuesto y su cobro al cliente.
- Las ordenes de trabajo pueden tener los siguientes estados, representados por el Enum *EstadoOrden*, Creada, EnCurso, Terminada, Cobrada
- Se modeló explícitamente siguiendo este diagrama de clases:



- Las ordenes de trabajo estan terminadas cuando todas sus tareas imputables estan terminadas.
- Cuando un imputable se termina puede tomar dos acciones, dependiendo de su tipo:
 - Si es un uso de un repuesto se descuenta el stock del mismo
 - Si es una intervención del operario se incrementa la comisión del mismo.

- Solo se puede terminar una tarea seleccionada y si no esta terminada, en función de eso bloquear el botón.
- Solo se puede cobrar una orden de trabajo terminada.
- Se desea poder mostrar la información de la siguiente forma:

Crear Orden de Trabajo

Cliente:

Ramirez, Juan

Total:

3350

Fecha:

26/11/2013

Tarea a agregar:

Cambiar Aceite

Agregar

Descripción	Monto	Terminado
Cambiar Amortiguadores	1500	☐
Amortiguadores (x2)	1200	☐
Cambiar Aceite	200	☐
Aceite "Aceitin" 5L	450	☐

Comenzar Trabajo

Cancelar

Gestionar Orden de Trabajo

Cliente:

Ramirez, Juan

Total:

3350

Fecha:

26/11/2013

Estado:

En Curso

Descripción	Monto	Terminado
Cambiar Amortiguadores	1500	☒
Amortiguadores (x2)	1200	☒
Cambiar Aceite	200	☐
Aceite "Aceitin" 5L	450	☐

Cobrar

Terminar Tarea

Cliente:

Ramirez, Juan

Fecha:

26/11/2013

Descripción	Monto	Terminado
Cambiar Amortiguadores	1500	☒
Amortiguadores (x2)	1200	☐
Cambiar Aceite	200	☐
Aceite "Aceitin" 5L	450	☐

Terminar Tarea

Se pide

Comunicar (**justificando adecuadamente**) el diseño de objetos que permita modelar las pantallas presentadas contemplando las reglas de negocio antes enunciadas.

Debe de comunicar, de que forma:

1. Modelaría la vista
2. Comunicaría la vista con el modelo antes presentado
3. Realizaría las validaciones requeridas

Tener en cuenta para la justificación de las elecciones:

- Duplicación de código
- Simpleza del modelo
- Facilidad de testeo

Punto 2: Persistencia relacional y mapeo objetos/relacional

Dado el modelo de dominio presentado en el punto anterior, se pide:

- El modelo de datos necesario para persistirlo
- Información de mapeo entre ambos modelos.
- Modificaciones que son necesarias realizar al modelo de dominio para poder persistirlo.
- Justificación de las decisiones tomadas.
- Información adicional que es necesaria sobre el dominio, en particular para tomar las decisiones anteriores.

También pueden solicitar información necesaria para poder tomar decisiones de performance.