



## Ampliación

Nos piden ampliar nuestra solución de forma que ahora se gestionen los cursos a los que asisten los mecánicos y que sirven para certificarlos como acreditados en los distintos tipos de vehículos que hay.

Al registrar los cursos que se ofertan se deberá indicar el código, nombre, fecha de comienzo y finalización, número de horas totales del curso y % de esas horas que se dedica a cada uno de los diferentes tipos de vehículo.

El sistema también debe registrar la participación de los mecánicos en los cursos, debiendo capturar el % de asistencia a clase y si ha superado el curso o no (para ello tiene que haber tenido como mínimo una asistencia del 85%).

Existirá un proceso en el sistema que, a partir de la información de los cursos a los que ha asistido y aprobado cada mecánico y de las horas que se necesitan para ser considerado formado en un tipo de vehículo, expida certificados de mecánico acreditado en un tipo de vehículo.

La operación de asignación de una orden de trabajo a un mecánico sufre una ligera modificación de tal manera que, ahora, no se puede asignar una orden de trabajo a un mecánico si no está acreditado para ese tipo de vehículo.

Además, se quieren una serie de listados que se indicarán más adelante.

## Generación de certificados

A petición del Administrador se lanzará un proceso de generación de certificados de formación. El proceso se regula con las siguientes reglas:

- Cada tipo de vehículo necesita un número determinado de horas de formación.
- Un curso puede dedicarse a uno o varios tipos de vehículo. Por lo tanto, además del número total de horas, se necesita registrar el desglose de horas (su % sobre el total del curso) que dedica a los tipos de vehículo para los que forme.
- Los mecánicos asisten a cursos y se registra su % de asistencia a las clases y si lo ha aprobado o no.
- El número total de horas aprobadas por un mecánico para un tipo de vehículo será el sumatorio de: la duración de cada curso al que asistió multiplicado por el % de dedicación del curso al tipo de vehículo y multiplicado por el % de asistencia al curso (si ha aprobado).
- Si un mecánico ha aprobado suficientes horas de formación específicas para un determinado tipo de vehículo (quizá en varios cursos) pasa a obtener el certificado de acreditado en ese tipo de vehículo a partir de esa fecha (que se quiere conocer).

## Actores y casos de uso

Administrador	<ul style="list-style-type: none"><li>– Gestión de cursos</li><li>– Gestión de asistencia a cursos</li><li>– Generación de certificados</li><li>– Listados</li></ul>
---------------	--



Jefe de Taller

– Gestión de órdenes de trabajo

### **Jefe de Taller. Gestión de órdenes de trabajo (modificado)**

Se implementará un CRUD de órdenes de trabajo con las siguientes características:

- Registrar una orden de trabajo. Podemos asumir que el vehículo ya existe en la aplicación. Se pedirá además del vehículo al que se refiere, una descripción del trabajo a hacer. Se asignará la fecha del sistema en la que se registra la orden. Inicialmente la orden de trabajo estará en estado ABIERTA.
- Modificar datos de una orden de trabajo. Solo la descripción puede ser cambiada y la orden de trabajo debe estar en estado ABIERTA o ASIGNADA.
- Eliminar una orden de trabajo. Solo si no tiene intervenciones.
- Asignar una orden de trabajo a un mecánico. Al asignar una orden de trabajo a un mecánico se mostrará un listado de todos los mecánicos que están certificados para ese tipo de vehículo y el usuario elegirá de entre ellos. No se puede asignar una orden de trabajo a un mecánico que no esté certificado para ese tipo de vehículo.

### **Administrador. Gestión de cursos**

Se implementará un CRUD de cursos con las siguientes características:

- Se darán de alta cursos, registrando el código, nombre, descripción, la fecha de comienzo (DD/MM/YYYY), la fecha de finalización (DD/MM/YYYY), el número de horas totales del curso y el % de las horas totales que son específicas para cada tipo de vehículo.
- Eliminar un curso. Solo si no tiene mecánicos registrados.
- Modificar datos de un curso. Mientras no haya sido, o esté siendo, impartido.
- Listar todos los cursos.

### **Administrador. Gestión de asistencia a cursos**

Se implementará un CRUD para las asistencias de los mecánicos a cursos con las siguientes características:

- Se darán de alta asistencias, registrando el mecánico que asiste, el curso al que asistió, el % de asistencia a clases y si ha aprobado el curso. El sistema debe controlar que para poder aprobar es necesario tener un mínimo de un 85% de asistencia.
- Eliminar una asistencia.
- Listar todas las asistencias a un curso. El usuario seleccionará el curso y se mostrará la información de cada mecánico que haya asistido ordenado por apellidos y nombre. La información a mostrar será: apellidos y nombre del mecánico, % de horas asistidas y si ha superado el curso o no.



## Administrador. Generación de certificados

Periódicamente se ejecutará el proceso que se encarga de generar los nuevos certificados para los mecánicos que cumplan los requisitos especificados en Generación de certificados. No requiere interacción del usuario.

En un despliegue real este proceso sería lanzado como una tarea programada del sistema (cron, Windows task scheduler, etc.). Por simplicidad se deja como una opción del menú del administrador.

## Administrador. Listados

Se implementarán los siguientes listados (además de los ya mencionados anteriormente en otros casos de uso):

- Listado de horas de formación de un mecánico. El sistema pedirá la identificación del mecánico y mostrará un listado con el total de horas de los cursos en los que se registró, el total de horas a las que asistió de esos cursos y un desglose de las que asistió por cada tipo de vehículo.

*Total de horas de los cursos: 999*

*Total de horas asistidas: 999*

*Horas asistidas para tipo vehículo1: 999*

*Horas asistidas para tipo vehículo3: 999*

- Listado de mecánicos que han asistido a formación por tipo de vehículo. Para cada tipo de vehículo se mostrará la información de cada mecánico que han participado en cursos de ese tipo junto al número de horas a las que asistió.

*Tipo de vehículo 1*

*Mecánico2, 999 horas*

*Mecánico5, 999 horas*

*Tipo de vehículo 3*

*Mecánico2, 999 horas*

*Mecánico7, 999 horas*

- Listado de mecánicos certificados para cada tipo de vehículo. Similar al anterior, pero mostrará los mecánicos certificados y la fecha de obtención del certificado.

*Tipo de vehículo 1*

*Mecánico2, dd/mm/aaaa*

*Mecánico5, dd/mm/aaaa*

*Tipo de vehículo 3*

*Mecánico2, dd/mm/aaaa*

*Mecánico7, dd/mm/aaaa*



## ¿Qué casos de uso se deben implementar?

- Si tu UO módulo 3 es 0 ( $uo \% 3 == 0$ ):
  - \* Gestión de cursos
  - \* Generación de certificados
  - \* Listado de mecánicos certificados para cada tipo de vehículo
- Si tu UO módulo 3 es 1 ( $uo \% 3 == 1$ ):
  - \* Gestión de asistencia a cursos
  - \* Generación de certificados
  - \* Listado de horas de formación de un mecánico
  - \* Gestión de órdenes de trabajo. Únicamente registro, actualización y eliminación.
- Si tu UO módulo 3 es 2 ( $uo \% 3 == 2$ ):
  - \* Generación de certificados
  - \* Listado de mecánicos que han asistido a formación por tipo de vehículo
  - \* Gestión de órdenes de trabajo excepto “Ver detalle de orden de trabajo”.

Independientemente de los casos de uso que te toquen es necesario que el modelo de dominio se implemente completo y correcto para todos ellos (como si tuvieras que implementarlos todos). Para suplir la falta del caso de uso que mantenga los datos que correspondan se insertarán en la base de datos los registros necesarios.

## Criterios de Diseño

El desarrollo debe hacerse a partir de lo obtenido en las clases de laboratorio correspondientes y de las partes que se dan implementadas: capa de presentación e interfaces de la capa de servicio.

A continuación, se mencionan una serie de criterios de diseño **obligatorios**. El no utilizarlos tendrá una penalización en la calificación.

- El diseño debe ajustarse a la **estructura de paquetes** vista en clase.
- Todo el código entregado pasará por un **detector de plagio**.
- Es necesario realizar correctamente las **validaciones** de datos necesarias y el **tratamiento de errores** (excepciones) y hacerlo en el lugar correcto (no se puede interactuar con el usuario en ningún otro sitio que la capa de presentación). No es válido que la aplicación termine de forma abrupta con una traza de excepción en pantalla.
- Todas las operaciones deben mantener la **integridad referencial** de los datos (cuidado con borrados o modificaciones).
- El código debe ajustarse a las **Java Code Conventions**<sup>1</sup>. No debe tener warnings (serán tratados como errores de compilación).
- El **código** debe estar “limpio”, bien sangrado, legible y comentado correctamente (javadoc donde proceda). No debe haber código “muerto”, ni código para depuración (println...).

---

<sup>1</sup> <http://www.oracle.com/technetwork/java/codeconventions-150003.pdf>



- El sistema tiene que funcionar **al menos con el motor HSQLDB**.
- Todas las sentencias SQL/JPQL deben estar **externalizadas**.
- Se debe entregar un único fichero comprimido con el siguiente contenido:
  - \* Un fichero de texto de nombre README.txt con tu **información personal** (UO, nombre y apellidos), qué **casos de uso** te ha tocado implementar y comentarios que quieras hacer. Si decides hacer alguna ampliación indícala también.
  - \* **Proyectos Eclipse** completos (sin el directorio */bin* para aligerar).
  - \* **Diagrama UML de clases** representando el nuevo modelo de dominio de la ampliación. Debe ser un diagrama de análisis (esto es, conceptual) y no debe incluir detalles técnicos propios de diseño o implementación. **No debe ser hecho con ingeniería inversa**. Y tampoco se admiten esquemas dibujados a mano y luego escaneados o fotografiados.
  - \* **Diagrama de tablas** (modelo relacional) como en el enunciado original (no es un diagrama entidad-relación) con las nuevas tablas añadidas y aquellas que se hayan modificado o se relacionen con las nuevas. Este diagrama *debe* ser hecho con una **herramienta de ingeniería inversa**<sup>2</sup>.
  - \* La **base de datos** debe estar rellena con datos suficientes para poder pasar pruebas (esto incluye las nuevas tablas creadas). **Si no se entregan datos para poder realizar las pruebas la entrega será nula**.
  - \* No se debe modificar la biblioteca AlbUtil. Si consideras que algún método de esta librería se puede mejorar, o echas de menos alguno, hazlo en un nuevo paquete dentro del proyecto que se entrega. Si te parece que merece la pena añadirlo a la librería original puedes comentarlo en el fichero README.txt.

## Recuerda

Además del profesor en las horas presenciales y las de tutorías, tienes a tu disposición en el campus virtual el foro para consultas sobre cualquier duda de la asignatura, del enunciado, posibles soluciones, etc. Úsalo tanto como necesites. El foro es una herramienta muy útil para que los profesores nos podamos dar cuenta de posibles problemas que os surjan y así poder ayudarlos.

Esta práctica, como muchas, te va a llevar un tiempo. Los profesores han estimado, y contrastado después de varios cursos, que el tiempo a emplear está ajustado al programa y a la guía docente. Pero ten en cuenta que esas estimaciones se hacen contando con que la vas haciendo al ritmo que te indica el profesor, en paralelo con las clases presenciales de teoría y prácticas, y que le dedicas cada semana el tiempo de trabajo autónomo que se espera (1,5 horas por cada hora presencial).

Dejar todo el trabajo para la última semana invalida por completo esa planificación y la metodología docente que los profesores están tratando de aplicar. El resultado probable, si ese es el caso, es que pierdas mucho tiempo tratando recuperar el hilo de las cosas, te encuentres con infinidad de problemas menores que no sabes ubicar (de los que

---

<sup>2</sup> En las clases de laboratorio usamos DBVisualizer.



casi seguro te advirtió el profesor en clase, pero no te acuerdas ya...), te hagas un lío leyendo por foros, la ansiedad por entregar te lleve a copiar fragmentos de código que encuentras por ahí, pero que no entiendes del todo, y que, al final, se te atragante y no llegues a captar la esencia conceptual del tema de la práctica.

De lo que entregues tendrás que hacer una demostración de autoría y además tienes el examen de teoría. La demostración tiene como fin dejar claro que has hecho tú la práctica y que dominas lo que has entregado. El examen de teoría suele consistir en preguntas cortas sobre conceptos con los que te tuviste que enfrentar en la práctica. Por lo tanto, aunque hayas conseguido entregar, de nada te servirá el atragantón si al final no has entendido los conceptos básicos.