

## Proyecto programado #2

### Objetivo:

Utilizar los conceptos de programación de objetos en JAVA y diagramas UML vistos en clase y aplicarlos en posibles escenarios, para la solución de problemas.

### Notas:

El proyecto debe ser entregado con limite el día y hora de entrega. No se aceptan proyectos por correo. Día y hora límite de entrega: martes 2 de diciembre, antes de las 22:00, en el grupo Teams del curso.

Debe entregar un archivo para toda la solución del proyecto en formato ZIP. Asegúrese de incluir todos los archivos Netbeans, Diagramas, Datos, Imágenes, etc.

La estructura del proyecto Netbeans debe respetar la debida separación funcional: Conceptos, Util, Aplicación, Presentación.

Data, Export y Diagramas serán las carpetas donde deberán almacenarse los archivos serializados binarios .DAT, los XML y los DRAWIO respectivamente.

Imágenes será la carpeta donde guarde las posibles imágenes que utilice en el proyecto.

No se permite cambiar la estructura de los archivos XML, debe ajustarse fielmente a la especificación.

### Contexto:

El proyecto trata de la gestión e información en el ambiente de un taller de mecánica automotriz; y es una modificación del proyecto 1 de este curso.

### Evaluaciones:

1) (10 pts.) Los objetos serializados.

Para los conceptos usados en la aplicación, ahora se deberá implementar la “Interfaz Serializable” en las clases de conceptos, que permitirá salvar el estado en de los objetos en archivos binarios, todos con extensión .DAT; a saber, se Leerán y/o Escribirán los objetos vía serialización en los archivos correspondientes:

- CLIENTES.DAT: objetos de clientes
- SERVICIOS.DAT: objetos de servicios
- MECANICOS.DAT: objetos de mecánicos
- SOLICITUDES.DAT: objetos de solicitudes
- ESTADOS.DAT: objetos de estados de la solicitud

Estos archivos tendrán su carpeta dentro del proyecto, llamada “Data”.

2) (15 pts.) Las clases (conceptos), formatos y diagramas

Para esta fase, se requiere de la implementación de 1 concepto principal: Solicitudes de Servicios. Y en donde este concepto tendrá 3 diferentes acciones principales, en donde algunas completaran información. Estas acciones son Solicitar, Atender y Consultar.

El concepto de Solicitud deberá incluir la siguiente información: Identificación (consecutivo), Placa, Tipo de Servicio, Cliente, Observaciones, Mecánico, Estado y Otros Servicios. Este último será una lista opcional que podrá agregarse uno a uno.

La información: Cliente, Mecanico, Estado y Otros Servicios es tomada de otros conceptos creados y existentes (definidos de previo).

Debe construir el diagrama de clases en el formato DRAWIO y las relaciones necesarias para representar correctamente el modelo.

Para las acciones Solicitar, Atender y Consultar, deberá construir los diagramas de caso de uso UML usando DRAWIO, de acuerdo a la información que solicita en las interacciones con la aplicación, definidas por la implementación visual adelante en evaluación (4).

Se esperan al menos 3 diagramas de caso de uso, uno por cada acción.

Las implementaciones en Java de estos modelos y diagramas deben coincidir fielmente con el con la especificación de datos XML, que se indica a continuación:

a. XML Solicitudes (etiquetas y estructura)

```
<solicitudes>
  <solicitud id="000000000">
    <placa> </placa>
    <servicio> </servicio>
    <cliente> </cliente>
    <mecanico> </mecanico>
    <estado> </estado>
    <observaciones> </observaciones>
    <otros_servicios>
      <id>000</id>
      <id>001</id>
    </otros_servicios>
  </solicitud>
  ...
</solicitudes>
```

b. XML Estados (etiquetas, estructura y datos)

```
<estados>
  <estado id="000">
    <nombre> Nuevo</nombre>
  </estado>
  <estado id="001">
    <nombre> En Revision</nombre>
  </estado>
  <estado id="003">
    <nombre> Pendiente de Repuestos</nombre>
  </estado>
  <estado id="004">
    <nombre> En Pruebas</nombre>
  </estado>
  <estado id="005">
    <nombre> Completado</nombre>
  </estado>
  ...
</estados>
```

3) (25 pts.) La implementación del manejo de archivos y las operaciones de “Importar”, “Exportar” y “Limpiar”.

Para el manejo de los archivos, es necesario la serialización de los objetos, según se indica en la “evaluación 1”. Quiere decir que los objetos se cargaran/salvaran de los archivos binarios “.DAT”.

Los archivos XML descritos en la evaluación (2) previos XML’s del proyecto 1, se deben implementar para el manejo exclusivo de las operaciones “IMPORTAR” e “EXPORTAR”, utilizando analizadores de XML, que permitirán vía archivo, la carga (lectura) y modificación (escritura) en archivos XML.

La carpeta “Export” tendrá los archivos XML.

La operación “IMPORTAR”: elimina todos los objetos actuales y carga la información en los XML’s, para cada uno de los conceptos, a saber: clientes, servicios, mecánicos, solicitudes y estados. Además de “rescribir” los archivos .DAT, de todos ellos.

La operación “EXPORTAR”: elimina todos los archivos XML actuales y escribe la información para cada uno de los conceptos, a saber: clientes, servicios, mecánicos, solicitudes y estados; generando/reescribiendo los archivos .XML, de todos ellos.

La operación “LIMPIAR”, elimina todos los objetos actuales y elimina la información en los DAT’s, para cada uno de los conceptos, a saber: clientes, servicios, mecánicos, solicitudes y estados. Los archivos .DAT, deberán quedar vacíos, sin objetos guardados.

Los archivos “.DAT” deben ser la entrada para construir los objetos de los conceptos indicados en la evaluación (2) y los conceptos previos del proyecto 1. Es decir que los XML son únicamente un mecanismo de carga y descarga de las operaciones de “Importar” y “Exportar”. No debe utilizarse el XML para otra cosa que no sea esta función. Este programa maneja la información en objetos.

Debe incluir la implementación del analizador XML en el paquete Util dentro de Netbeans.

4) (50 pts.) La implementación visual y funcional de la aplicación.

La aplicación debe ser implementado usando objetos visuales de Java-Swing.

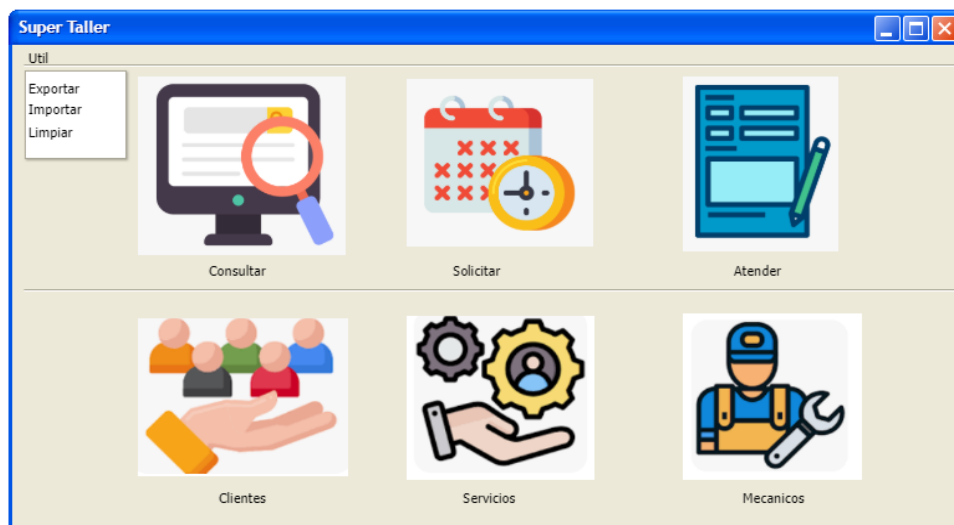
La ventana principal se encargará de dar acceso a todas las funciones a implementar y las ya implementadas en la fase 1 (proyecto 1). Para cada concepto y acción indicados en evaluación (2), es decir: Solicitar, Atender, o Consultar de las solicitudes de servicio.

El acceso a estas acciones será por la implementación de un icono/imagen, que mostrará cada una de las respectivas ventanas, que tendrán programadas dichas funciones.

Adema se requiere que existe un menú que permitirá las acciones de “Exportar”, “Importar” y “Limpiar”, cuya función de detalla en la evaluación (3).

A continuación, se detallan las ventanas:

a. Principal

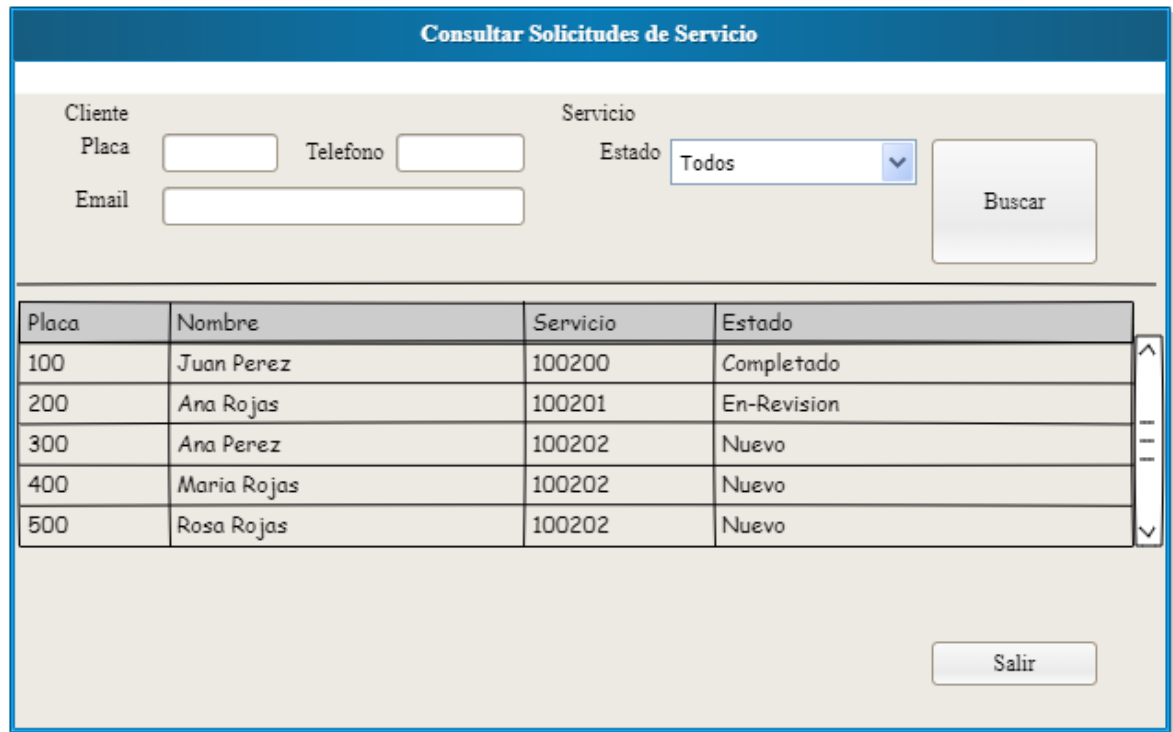


Suerte ¡!

En esta ventana se implementan las imágenes/iconos que dará acceso a las operaciones vía eventos.

El menú “Util” es nuevo en la implementación de la segunda fase.

b. Consultar



Placa	Nombre	Servicio	Estado
100	Juan Perez	100200	Completado
200	Ana Rojas	100201	En-Revision
300	Ana Perez	100202	Nuevo
400	Maria Rojas	100202	Nuevo
500	Rosa Rojas	100202	Nuevo

La ventana presenta dos secciones: superior, encargada de hacer las búsquedas y la inferior que se encarga de mostrar los objetos cargados desde el .DAT y del archivo correspondiente a Solicitudes.

Las operaciones aplican sobre la presentación visual, la representación de los objetos y el archivo .DAT.

La parte superior es la operación de buscar y/o filtrar, y debe mostrar la información correspondiente en la tabla, debe mostrar los datos programando un evento que le permita aplicar en conjunto (AND) sobre los filtros. Pej, el estado “Nuevo” de la placa “100” deberá interpretarse como que ambas condiciones deben darse para mostrar los datos en la tabla, cuando se presione el botón “buscar”. Si no hay información en los filtros de búsqueda, y se presiona el botón “buscar”, la tabla traerá todos los servicios. Debe prever que los datos sean más de lo que visualmente se pueda mostrar, así que la tabla debe tener la función de “scroll” para ver los demás datos según la búsqueda.

c. Crear Solicitud (Solicitar)

The image shows two windows from a software application. The main window is titled "Crear Solicitud de Servicio" and contains the following elements:

- Placa:** A text input field.
- Tipo de Servicio:** A dropdown menu with "Cambio de Aceite" selected.
- Cliente:** A dropdown menu with "Nuevo Cliente" selected, and a "+" button next to it.
- Observaciones:** A text area containing the placeholder text: "Secrevit fontes liquidum locoque pronaque? Illas semine campoque declivia oppida corpora nam inter fuit discordia tellus solidumque iunctarum erat: quae terrenae ubi rerum recessit iudicis aestu fixo".
- Buttons:** "Salvar" and "Cancelar" buttons at the bottom right.

A blue arrow points from the "Salvar" button to a smaller, secondary dialog box titled "# Servicio". This secondary dialog box contains:

- An information icon (i) in a circle.
- The text: "Su solicitud ha sido ingresada exitosamente, Numero 100201".
- An "OK" button.

La ventana presenta la captura de información para crear una nueva solicitud de servicio. Las operaciones aplican sobre la presentación visual, la representación de los objetos y el archivo .DAT.

Los controles de listas (combos), cargan la información de los conceptos asociados, así para el "Tipo de Servicio" y para el "Cliente".

El botón "Salvar" permite guardar la información en el archivo .DAT correspondiente a "Solicitudes". Debe además mostrar un pequeño dialogo indicando que la solicitud fue salvada en el archivo, y mostrar cual fue el consecutivo generado.

El botón "Cancelar" solo cierra el dialogo sin realizar ninguna operación.

d. Atender

**Atender Solicitud de Servicio**

#Servicio: 100201 Placa: 200

Tipo de Servicio: Cambio de Aceite

Mecanico: Rodrigo Perez Estado: En Proceso

Observaciones:

Secrevit fontes liquidum locoque pronaque?  
Illas semine campoque declivia oppida corpora nam inter fuit  
discordia tellus solidumque iunctarum erat: quae terrenae ubi  
rerum recessit iudicis aestu fixo

Otros Servicios (adicionales)

	Tipo	Servicio
<input type="checkbox"/>	002	Mecanica Menor
<input checked="" type="checkbox"/>	003	Revision de Frenos

Salvar Cancelar

La ventana presenta la operación de atención a las solicitudes de servicio.

Es lo primero que debe de seleccionarse para iniciar o continuar con la atención de una solicitud.

Si es la primera vez, la solicitud no tendrá un mecanico asignado, y este campo aparecerá en blanco, de forma que el “combo” de selección, permita seleccionar a alguno de los mecánicos. Este combo debe cargar con los mecánicos definidos en el concepto mecánicos. Los estados se deben mostrar en el “combo” de selección, y la primera vez se mostrará el estado “NUEVO” (código 000).

El campo placa y tipo de servicio aparecerán deshabilitados para no cambiarlo en esta pantalla.

Las observaciones si son editables y estará permitido hacerle cambios.

Se habilita agregar “Otros Servicios Adicionales” que serían tomados del concepto “Servicios”, y estos deben agregarse en la sección “otros\_servicios” del objeto, y guardado en el objeto de esta solicitud.

El botón “Salvar” permite guardar la información en el archivo .DAT correspondiente a “Solicitudes”. Debe además mostrar un pequeño dialogo indicando que la solicitud fue salvada en el archivo, y mostrar cual fue el consecutivo generado.

El botón “Cancelar” solo cierra el dialogo sin realizar ninguna operación.

Las operaciones aplican sobre la presentación visual, la representación de los objetos y el archivo .DAT.

Notas adicionales:

1. Puede hacerse en grupos de 2 personas, máximo. Si hace el proyecto en grupos, solo uno de los integrantes sube la solución a Teams. OJO: en el “main” y mediante un comentario del código, se debe indicar quienes son los integrantes y sus respectivos carnes.
2. Se vale que las imágenes y la forma de acomodarse en la ventana pueda ser sugerida por alguno de sus compañeros de diseño o un tercero (opcional).
3. El programa principal debe mostrar un objeto ventana que hereda de JFrame.
4. El programa termina cuando se cierra la venta.
5. Utilice las versiones de Java y NetBeans que autorizamos en este curso, y cuyos instaladores están disponibles en el Teams del curso.
6. NO utilice otras librerías que no hayamos visto en clase. En caso de duda pregunte al profesor.