

# **Trabajo Practico N° 4:** **Laboratorio de Programación**



*Ignacio Martín Canay*

# Introducción:

El presente documento detalla y explica las funcionalidad del Trabajo Practico N°4. Dicho Trabajo Practico abarca los temas vistos entre las clases 19 y 25, a saber:

- Excepciones
- Test Unitarios
- Tipos Genéricos
- Interfaces
- Archivos
- SQL
- Bases de Datos
- Eventos y Delegados
- Métodos de extensión

Cada uno de estos temas fue utilizado a lo largo del trabajo y la implementación de cada uno de ellos sera explicada a continuación.

La consigna era generar un programa que simulara una fabrica. Para realizarlo se eligió una Fabrica de muebles. Dicha fabrica produce 2 tipos distintos de muebles: **Roperos** y **Sillones**. Cada uno de estos muebles esta hecho de un material base y aparte tienen un segundo material del cual están hechos sus estantes o el tapizado, respectivamente. El programa de la fabrica tiene la posibilidad de hacer estos muebles de distintos **Materiales**, elegidos de una lista. La primera vez que se inicie el programa no tendrá solo 3 materiales básicos cargados. Pudiendo Agregar nuevos materiales a la lista a través de la interfaz de la fabrica(**frmFabrica**), utilizando el botón **Agregar Material**.

Para Fabricar un mueble se utiliza el botón **Fabricar mueble**, esto llama a un formulario nuevo de tipo **frmAgregarMueble** con las opciones para Crear el nuevo mueble. Un nuevo mueble tiene que tener un nombre, un material del cual esta hecha la base y un material del cual están hechos los estantes, en el caso de los Roperos, y el tapizado en el caso de los Sillones. Los Roperos nuevos que están hechos de madera inicialmente no tienen barniz, pero luego pueden ser barnizados en la sección de diseño de la fabrica. Los sillones, roperos (que no sean de madera) y los estantes de los roperos (sin importar el material) no pueden ser barnizados, pero si pueden ser pintados mas adelante.

Ya que este formulario representa la función principal de la fabrica, es aquí donde se han utilizado eventos, delegados e hilos. Al clickearse en el botón **btnAgregarMueble** se lanza un nuevo hilo con el método **AgregarProducto**. La vida de este hilo depende del mueble que se este queriendo fabricar. Según el tipo de mueble demora 15 segundos en fabricarse un ropero y 10 segundos un sillón. Al cabo de este lapso el mueble es creado y agregado a la lista de muebles, esto se informa por medio de un rich text box con la información del mueble que se ha creado, que se actualiza de forma asincronica desde el hilo secundario invocando el rich text box para cargarle la información del ultimo mueble que se creo.

Clickeando en el botón **Diseñar mueble** se instancia un nuevo formulario de tipo **frmDiseño** que contiene una list box con los nombres de todos los **Muebles** (Roperos y Sillones) que se fabricaron. Seleccionando un elemento de la lista y eligiendo los colores para la base y los estantes o el tapizado estos se pueden pintar. Nótese que en el caso de que el elemento seleccionado de la lista sea un ropero y su base este hecha de madera, se podrá elegir el color con el cual se barnizara la estructura, esto es posible gracias a la Interfaz **IBarnizable** empleada en la clase Roperos y el método de extensión **EsMaterial**, que permite comprobar que material es.

El botón **Mostrar Lista de muebles** abre un nuevo formulario de tipo **frmListaDeMueble** mostrando la listas de los muebles que se fabricaron. Aquí se detalla toda la información de

cada uno de ellos. Este formulario posee un botón [Generar log.txt](#) para generar un archivo de texto con la información mostrada en el rich text box.

El botón de [Guardar](#), guarda la lista de muebles en la base de datos.

El botón [Cargar](#), carga la lista de muebles desde la base de datos.

Dentro del formulario [frmListaDeMateriales](#) hay un botón [Generar log.txt](#) que genera un archivo de texto “Lista de materiales Log.txt” en la carpeta ubicada en:

“..\TP4 Recuperatorio\UI\bin\Debug”.

Aquí también está el botón [Guardar](#) que genera un archivo xml “Materiales.xml” en la carpeta ubicada en: “..\TP4 Recuperatorio\UI\bin\Debug” con la información de todos los materiales en la lista de la fábrica. Y también el botón [Cargar Archivo](#) que lee el archivo ubicado en: “..\TP4 Recuperatorio\UI\bin\Debug” y carga los materiales a la lista de la fábrica. Nótese que el método empleado para realizar esto también se emplea en la carga del formulario principal para que automáticamente cargue la lista con los tres materiales básicos para que la fábrica pueda funcionar, estos son: Madera, Hierro y Algodón.

En cuanto a los métodos de extensión se han utilizado en dos oportunidades. El primero es un método de extensión para Materiales, [EsMaterial](#) para comprobar el tipo de material. Este método, como se explicó anteriormente, es usado para el control de los combobox del formulario de diseño de muebles.

El segundo método es una extensión de `List<Mueble>` [ContieneMueble](#) y retorna un booleano dependiendo si el Mueble que se pasa como parámetro está contenido dentro de la lista. Se utilizó para hacer los test de las clases las instancias de `Ropero` y `Mueble` para probar si eran correctamente ingresadas en la lista.

## Excepciones:

Se utilizaron excepciones para evitar que el programa detenga su funcionamiento ante un error no controlado. Dentro del archivo “ManejoDeExcepciones.cs” podemos encontrar todas las excepciones creadas por mi. Estas son:

**MaterialNullException:** Se utiliza cuando el material de un objeto mueble es null.

**NombreNullException:** Se utiliza cuando el nombre de un objeto mueble es null.

**FaltaSeleccionException:** Se utiliza cuando no se ha seleccionado un objeto en un combo box.

**BarnizarSillaException:** Se utiliza cuando se intenta barnizar un sillón.

**BarnizarEstantesException:** Se utiliza cuando se intenta barnizar un Estante.

## Test Unitarios:

Se realizaron dos test unitarios.

El primero test llamado “roperoBarnizar”. El Arrengue (simulación de la datos) emula un ropero con base de madera y estantes de hierro. Se prueba la función del método “Barnizar”. Dicho método devuelve un booleano que es verdadero si se pudo barnizar la base del Ropero. En este caso el Asser (la respuesta esperada) es “true”. La prueba de este test arroja un resultado positivo.

El segundo test llamado “sillonColoreado”. El Arrengue emula un sillón con base de madera y con tapizado de algodón. Se prueba en este test la funcionalidad del método “PintarBase” que cambia el color de la base del sillón por un color del tipo ConsoleColor y devuelve un booleano indicando si pudo o no realizar el cambio. En este caso el Asser también es “true”. La prueba arroja un resultado positivo.

## Tipos Genéricos:

Principalmente los tipos genéricos fueron utilizados para las funciones de Guardado y Cargado del programa. Como se puede observar en el archivo “Xml.cs”

Esto fue hecho para poder utilizar las mismas funciones de guardado y cargado tanto para los distintos muebles como para los materiales.

## Interfaces:

Se utilizaron dos interfaces. “IBarnizable” e “Icoloreable”. Estas interfaces se utilizaron para delimitar el comportamiento de las Clases Ropero y Sillón. Principalmente para poder permitir que los roperos con base de madera puedan ser barnizados. Y para poder colorear al resto de los muebles que no cumplen con esa condición.

## Archivos:

Para el manejo de archivos se utilizaron dos clases. La primera es lo que concierne al manejo de archivos de Texto, utilizados para guardar un registro de las operaciones en la clase “ConsoleTest”. Todo lo que allí ocurra queda registrado en el archivo ubicado en el directorio “...\Trabajo Practico 3\frmFabrica\bin\Debug” con el nombre de archivo “Console log.txt”.

La segunda clase de archivos utilizados es Xml. Esta clase se utiliza para registrar dos archivos llamados “Lista de Muebles.xml” y “Lista de Materiales.xml” donde se registran en formato xml tanto los Muebles en la lista de muebles de la fabrica con sus atributos, como los materiales.