

TP 4 - Fabrica de autos

La aplicacion trata de simular como es el proceso de creacion de un vehiculo, en estos casos de Autos o Motocicletas, donde podemos crearlos a nuestro gusto con las opciones que nos ofrecen, pudiendo ponerle un nombre a la creacion y guardar nuestra configuracion en un txt. (El tiempo que tarda en crear el archivo es la simulacion de creacion del auto en el taller)

Ademas podremos guardarnos nuestras creaciones en la base de datos del taller para que otras personas puedan reutilizar modelos ya creados.

-Importante: Un vehiculo se considera igual siempre y cuando ya exista un vehiculo con el mismo nombre y hecho por el mismo creador.

Clase 15: Excepciones - se usa en todo el codigo con el objetivo de contener cualquier error en tiempo de ejecucion, mayormente se encuentra en Entidades.Taller ya que es la clase estática que maneja el taller y en la aplicacion de consola donde se utiliza mayormente para mostrar las excepciones por cajas de texto.

Clase 16: Test Unitarios - se hicieron 5 test donde se prueba las principales funcionalidades de la aplicacion, especificamente Entidades.Taller.

Clase 17: Tipos Genéricos -

- Se implemento en los casos de las funciones Entidades.Taller.MostrarUnoDeLaLista para que pueda ser utilizada siempre y cuando la clase que se quiera usar como parametro tenga una herencia de "Vehiculo" y la interfaz "IDetalles"

- Se implementó en Entidades.Vehiculos.Existe con el proposito que el parametro que se use lo de la clase hija que herede de "Vehiculo"

Clase 18: Interfaces - Se creó una interfaz que ayuda a que la clase que la implemente tenga una funcion que muestre los datos. Esta funcion no se hizo dentro de una clase abstracta ya que no todas las clases las heredaría, así que se optó por la interfaz

Clase 19: Archivos y serializacio - Entidades.Archivos

- Archivos : Se implemento en el archivo .txt que se crea al "fabricar" un vehiculo, dependiendo si es un auto o una motocicleta el mensaje es personalizado y el tiempo que tarda en "crearse" sera segun que partes le hayamos puesto al vehiculo

-Serializacion: Se implemento por la necesidad de guardar el nombre del usuario que crear los vehiculos, se decidio esto para que el usuario se pueda loguear una sola vez y sin necesidad de acceder a una base de datos.

Clase 21/22: SQL - DBConexion

-Se implementó con la idea de poder guardar los vehiculos en un lugar por fuera del codigo que en cualquier momento se puedan volver a fabricar. Se encuentra en

App.traeAutosToolStripMenuItem_Click

App.insertarTodosLosAutosToolStripMenuItem_Click

App.traeMotocicletasToolStripMenuItem_Click

App.insertarTodosLasMotocicletasToolStripMenuItem_Click

Clase 23: Hilos

-Se implemento con la idea de poder simular el tiempo de fabricacion de un auto, el tiempo que tarda en fabricarse cada auto es independiente a que partes se le haya puesto al vehiculo y con la ayuda de los hilos se pueden fabricar vehiculos en simultaneo. Se encuentra en

App.FrmPrincipal.ActivarFabricacion

Clase 24: Eventos y delegados

-Se utilizaron para poder agrupar las funciones que dan comienzo a la fabricacion. Se encuentra en Entidades.Auto, Entidades.Motocicleta y App.ActivarFabricacion.

Clase 25: Metodos de Extencion

-Se implemento en la clase Entidades.ExtencionDeFormato en esta clase se extiende los distintos formatos de texto que se muestran en la aplicación