

Alumno: Gabriel Celano Division: 2do C

Diagrama de clase Archivos:

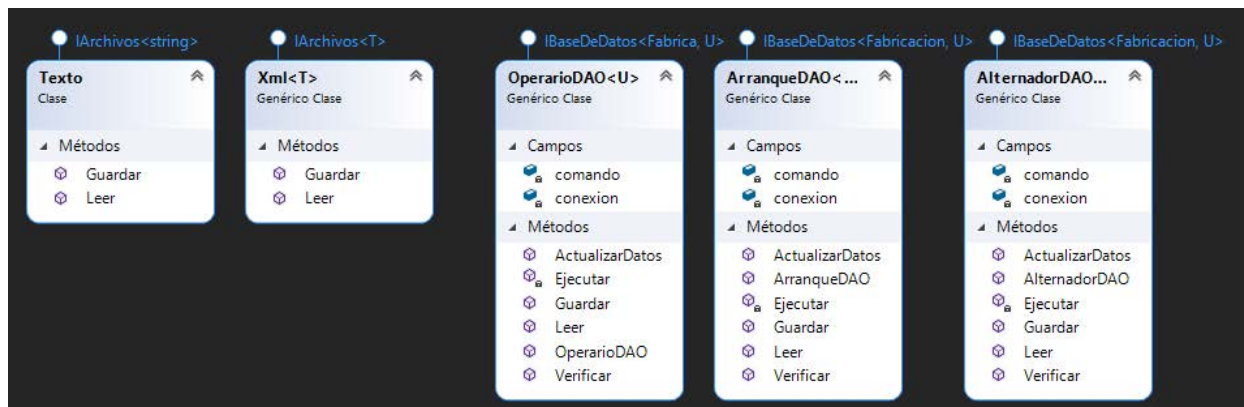


Diagrama de clases Abstractas:

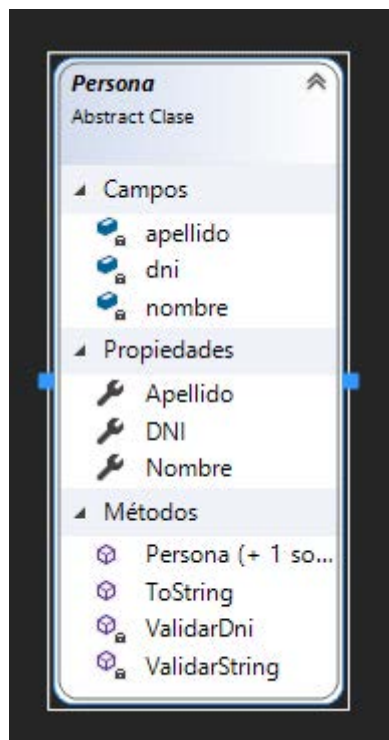


Diagrama de clase Enumerados:



Diagrama de clase Instanciable:

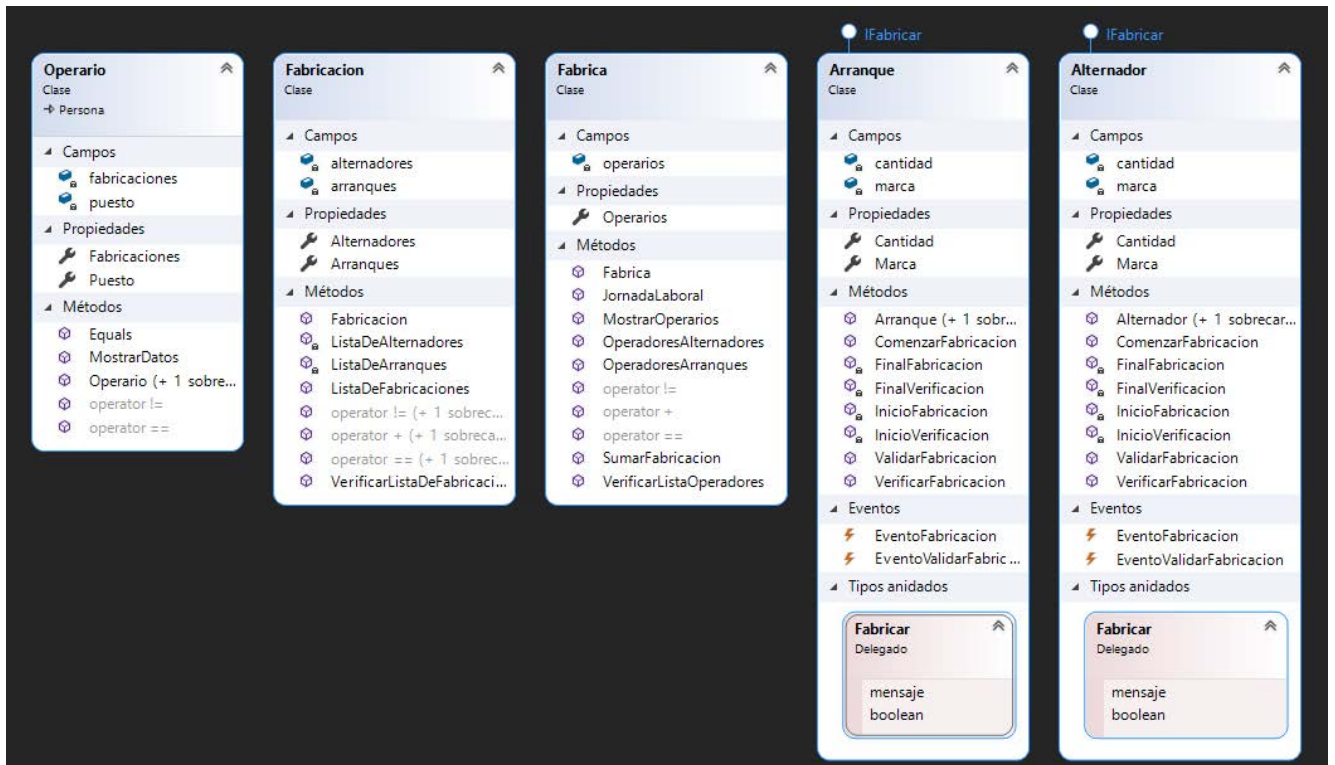
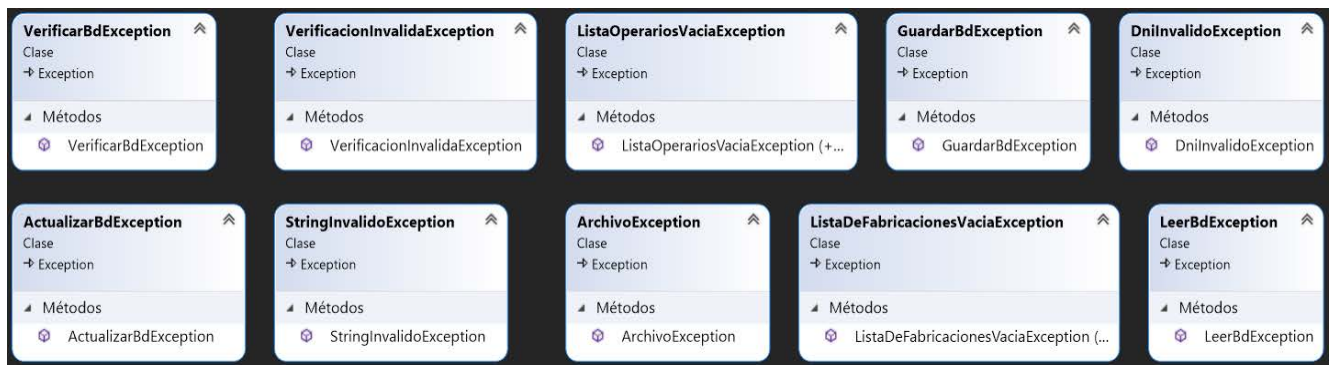


Diagrama de clase Excepciones:



Proyecto Archivos:

ArranqueDAO, AlternadorDAO, OperarioDAO: Clases genericas, que implementan la interfaz IBaseDeDatos, se encargan del manejo de sus respectivas tablas en la base de datos.

Xml: Clase generica que implementa la interfaz IArchivo, se encarga de serializar y deserializar listas.

Texto: Clase que implementa la interfaz IArchivo, se encarga de guardar y leer un archivo del tipo texto.

Proyecto Clases Abstractas:

Persona:

- *Las propiedades que contiene, se setean previa validación.
- *Tiene un constructor sin parámetros para la serializacion de la clase.
- *Metodos ValidarDni y ValidarString que validan los valores ingresados.
- *Sobreescritura del metodo ToString(), devuelve los datos de la Persona.

Proyecto Clases Instanciables:

Fabrica:

- *Atributo de lista Operarios.
- *Constructor por defecto que instancia la lista de Operarios.
- *Un operario, se repite si esto ya se encuentra en las lista.
- *Se agrega un operario, a la lista si este aun no se encuentran en ella.
- *Método SumarFabricacion suma una fabricación al operador
- *Método VerificarListaOperadores verifica si la lista de operadores contiene algo, sino lanza una excepción.
- *Método MostrarOperarios que llama a los metodos OperadoresArranques y OperadoresAlternadores y retornan la lista completa de Operarios.
- *Metodo Jornada Laboral que retorna la lista de opararios que trabajaron el dia de la fecha.

Operario:

- *Constructor por defecto sin parámetros para poder serializar la clase.
- *Un operador es igual a otro si estos comparten le mismo DNI.
- *Método MostrarDatos retorna los datos completos del operario.

Fabricacion:

- *Atributos de listas de tipo Alternador y Arranques.
- *Constructor por defecto que instancia las listas.
- *Un alternador o un arranque es igual a otro si este comparte la misma marca.
- *Se puede agregar un alternador o arranque a sus respectivas listas si estos no se encuentran en ellas.
- *Método ListaDeFabricaciones que llama a los métodos ListaDeAlternadores y ListaDeArranques que devuelven la lista de fabricaciones.
- *Metodo VerificarListaDeFabricaciones retorna true si alguna de las dos listas contienen alguna fabricacion, de caso contrario lanza una excepcion.

Arranque y Alternador:

- *Contienen un delegado Fabricar que retorna void y con los parametros string y bool.
- *Dos eventos del tipo Fabricar: EventoFabricado y EventoValidarFabricacion.
- *Constructor por defecto sin parametros.
- *Metodo ValidarFabricacion, valida la marca del objeto.
- *Metodos InicioFabricacion y FinalFabricacion invocan al EventoFabricacion que pasa por parametro un mensaje y un valor booleano.
- *Metodos InicioVerificacion y FinalVerificacion que invocan al EventoValidarFabricacion que pasa por parametro un mensaje y un valor booleano(previa validacion por el metodo ValidarFabricacion).

Enumerados:

- *EPuestoDeTrabajo: contiene FabricanteDeArranques y FabricanteDeAlternadores.
- *EMarca: contiene Valeo, Marelli, Bosch y Denso.

MiExtension:

- *Método de extensión para la clase string que retorna un valor booleano y pasa por parámetro el string y devuelve un atributo del tipo EMarca, este convierte el valor string pasado por parámetro y lo convierte en EMarca.

Proyecto de clases Excepciones:

Exepciones:

- *ArchivoException
- *DniInvalidoException
- *ListaDeFabricacionesVaciaException
- *ListaOperariosVaciaException
- *OperarioInvalidoException
- *StringInvalidoException
- *VerificacionInvalidaException
- *VerificarBdException
- *LeerBdException
- *GuardarBdException
- *ActualizarBdException

Proyecto de Interfaces:

IArchivos:

- *Implementa el método Guardar que retorna bool y pasa por parámetro un string y un objeto del tipo genérico T.
- *Implementa el método Leer que retorna un dato del tipo T genérico y recibe por parámetro un string.

IBaseDeDatos:

- *Implementa los métodos Guardar(retorna bool) y ActualizarDatos(retorna void), ambos reciben por parámetro un dato del tipo T genérico y otro de tipo U.
- *El método Leer retorna T de tipo genérico y recibe por parámetro T.
- *Verificar retorna bool y recibe por parámetro un tipo U genérico.

IFabricar:

- *Contiene dos métodos que retornan void, ComenzarFabricacion y VerificarFabricacion

Proyecto Formularios:

FrmFabrica:

- *El boton Alta llama al formulario FrmAltaOperario.
- *Los botones Alternadores y Arranques, llaman a sus respectivos formularios FrmAlternadores, y FrmArranque.
- *Leer BD devuelve la base de datos y guarda los datos en listas.
- *Fabricaciones y Operarios muestran en el RichTextBox sus listas.
- *Guardar XML Serializa las listas y Leer las Deserealiza.
- *Guardar TXT guarda la jornada en un archivo txt, y Leer muestra el contenido en el RichTextBox.

FrmAltaOperario:

- *Verificar instancia el operario y verifica si hay algún error en los campos o si el operario ingresado ya se encuentra registrado.
- *Alta termina de guardar al operario en la lista y la base de datos.

FrmAlternadores y FrmArranques:

- *Asignar Operador asigna un operador del DataGridView.
- * Marca asigna la marca a fabricar.
- *Fabricar Arranque comienza la fabricacion en un hilo secundario.
- *Verificar Fabricacion verifica la fabricacion en un hilo secundario.