

Alumno: Gabriel Celano Division: 2do C

Diagrama de clase Archivos:



Diagrama de clases Abstractas:

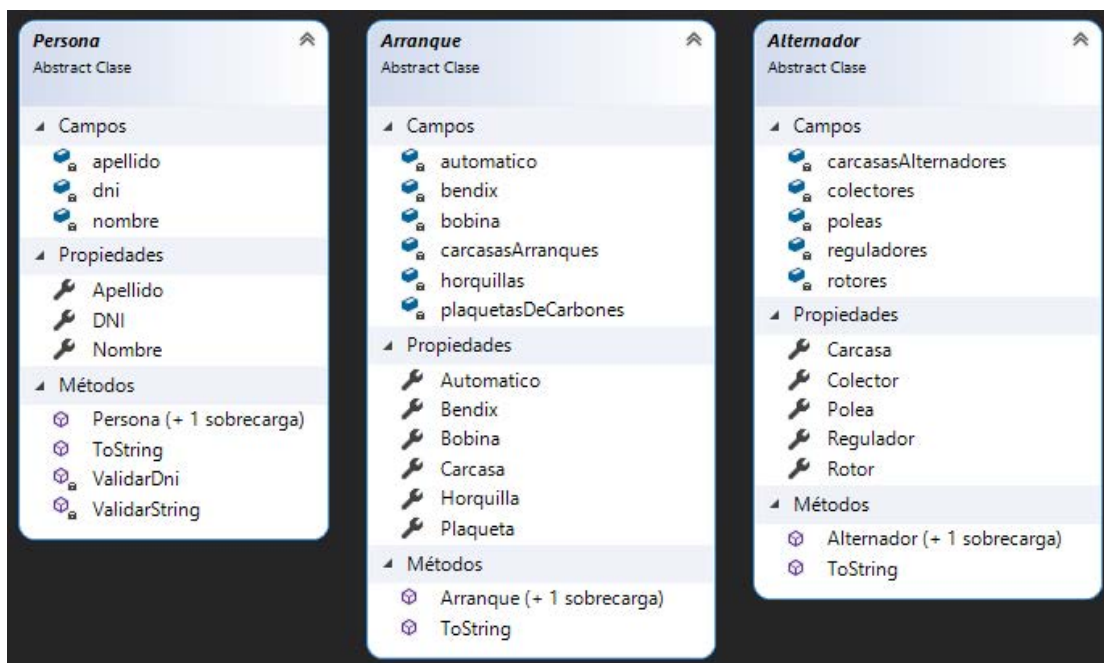


Diagrama de clase Enumerados:

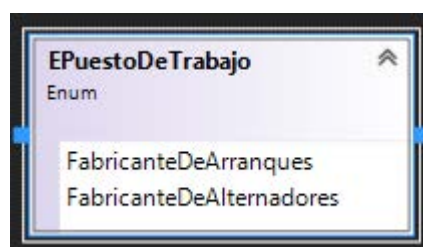


Diagrama de clase Instanciable:

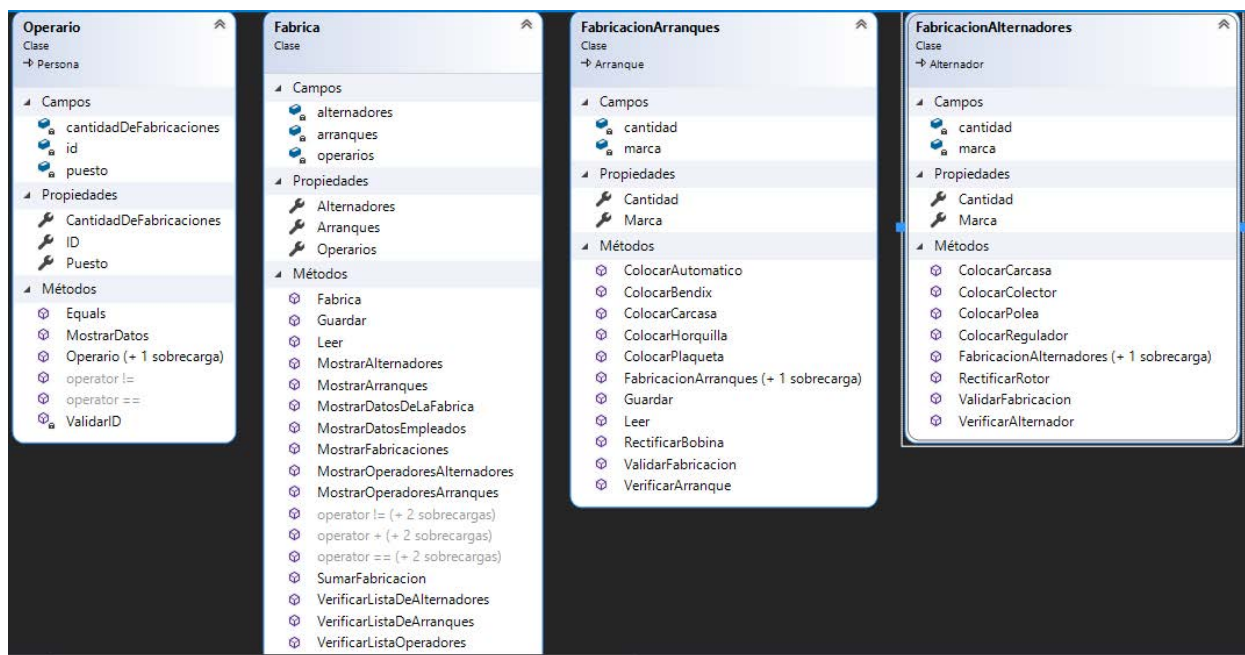
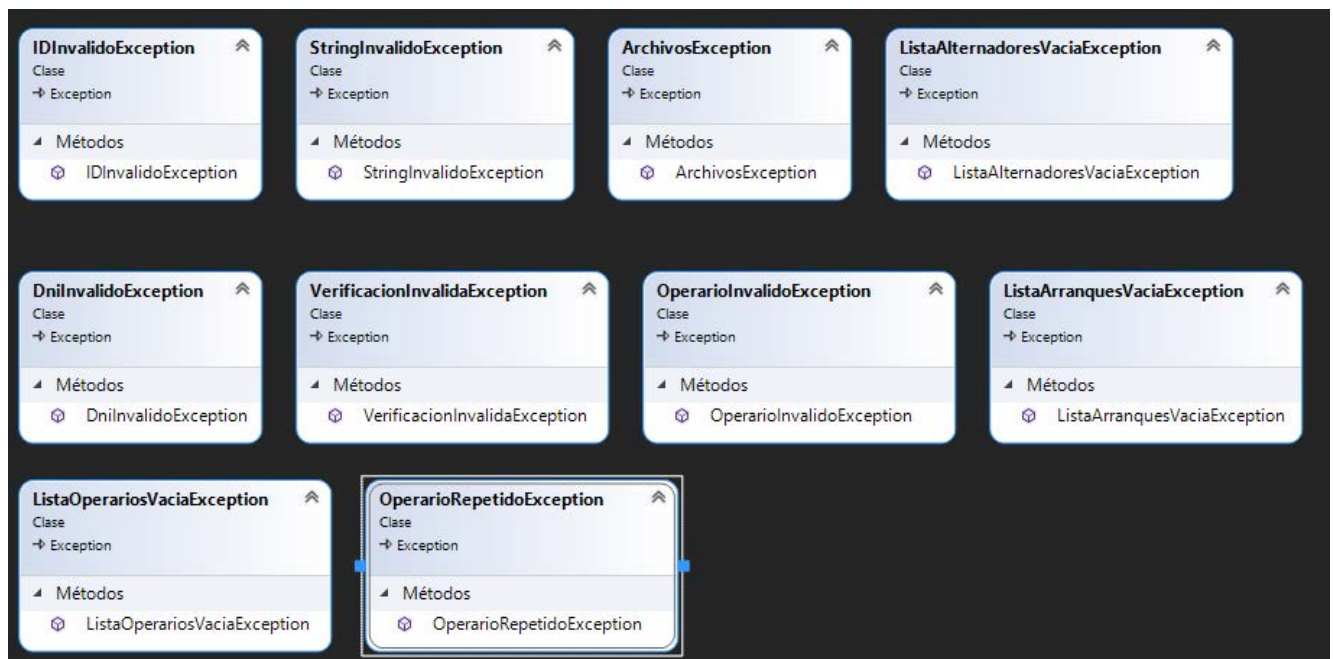


Diagrama de clase Excepciones:



Proyecto Archivos:

IArchivo: interfaz implementada en las clases Xml y Texto, para guardar y leer archivos.

Texto y Xml: implementan los métodos de la interfaz y para guardar y leer los archivos(tanto xml y txt).

Proyecto Clases Abstractas:

Persona:

- *Atributos Nombre, Apellido y Dni
- *Método ValidarDni retorna el dni valido o lanza la excepcion DniInvalidoException
- *Método ValidarString retorna el string valido o lanza la excepcion StringInvalidoException
- *Sobreescritura del método ToString que retorna los datos de la Persona

Arranque:

- *Atributos booleanos Carcasa, Horquilla, Bendix, Bobina, Automático, Plaqueta
- *Sobreescritura del método ToString que retorna los datos del arranque

Alternador:

- *Atributos booleanos Carcasa, Polea, Rotor, Colector, Regulador
- *Sobreescritura del metodo ToString que retorna los datos del alternador

Proyecto Clases Instanciables:

Fabrica:

- *Atributos de lista Alternadores, Arranques, Operarios
- *Constructor de clase que instancia las listas
- *Un operario, alternador o arranque se repite si estos ya se encuentra en las listas
- *Se agrega un operario, alternador o arranque a la lista si estos aun no se encuentran en ellas
- *Método SumarFabricacion suma una fabricación al operador
- *Método VerificarListaOperadores verifica si la lista de operadores contiene algo, sino lanza una excepción
- *Método VerificarListaAlternadores verifica si la lista de alternadores contiene algo, sino lanza una excepción
- *Método VerificarListaArranques verifica si la lista de arranques contiene algo, sino lanza una excepción
- *Método DatosDeLaFabrica muestra todos los empleados y fabricaciones

JornadaLaboral:

- *Atributos de lista Operarios
- *Constructor de clase que instancia la lista
- *Un operario se repite si este se encuentra en la lista
- *Un operario se agrega si este no se encuentra en la lista
- *Metodo MostrarJornada muestra a todos los operarios con mas de una fabricacion realizada y todas las fabricaciones de la jornada laboral

FabricacionAlternadores:

- *Atributos Marca y Cantidad
- *Métodos con los procesos de fabricación
- *Método VerificarAlternador muestra como va la fabricación del alternador
- *Método ValidarFabricacion valida si el alternador esta bien fabricado

FabricacionArranques:

- *Atributos Marca y Cantidad
- *Métodos con los procesos de fabricación
- *Método VerificarArranque muestra como va la fabricación del Arranque
- *Método ValidarFabricacion valida si el arranque esta bien fabricado

Operario:

- *Atributos Puesto(de tipo Enumerado.EPuesto), Cantidad de fabricaciones e id
- *Sobrecarga del método Equals un Operador es igual a otro si son del mismo tipo
- *Método ValidarId valida si el id ingresado es correcto, caso contrario lanza IDInvalidoException
- *Método MostrarDatos retorna los datos del Operario

Proyecto clases Enumerados:

EPuestoDeTrabajo:

- *Contiene los puestos de tipo enumerado FabricanteDeArranques y FabricanteDeAlternadores

Proyecto de clases Excepciones:

Exepciones:

- *ArchivosException
- *DniInvalidoException
- *IDInvalidoException
- *ListaAlternadoresVaciaException
- *ListaArranquesVaciaException
- *ListaOperariosVaciaException
- *OperarioInvalidoException
- *OperarioRepetidoException
- *StringInvalidoException
- *VerificacionInvalidaException