

Este proyecto permite la carga, comparacion , analisis e importacion de datos de tipo Placas de video.

-Carga:se cargan los datos al sistema, y tmb se le puede agregar datos nuevos con nuevos valores que se veran reflejados en la parte de comparacion. Ademas se podra guardar la placa creada en formato XML.

Carga de Datos

CARGA DE NUEVA PLACA AL SISTEMA

Nombre	<input type="text"/>	Dato nuevo:	<input type="text"/>
Marca	Msi <input type="button" value="v"/>	Valor:	<input type="text"/>
Tipo de Memoria	SinAsignar <input type="button" value="v"/>	<input type="button" value="Carga dato nuevo"/>	
Capacidad de ram	<input type="text" value="0"/> GB	Detalles de la placa creada:	
Frecuencia	<input type="text" value="0"/> GB/s		
Consumo	<input type="text" value="0"/> W		
Longitud	<input type="text" value="0"/> CM		
Interfaz	<input type="text" value="0.0"/> PCI		
Rendimiento Minería Bitcoin	<input type="text" value="0.000"/> Bit/24hs	<input type="button" value="Crear"/>	
Rendimiento Minería Ethereum	<input type="text" value="0.000"/> Eth/24hs	<input type="button" value="Guardar en Xml"/>	

-Comparación: Una vez realizada la carga de datos, ya sea manual o importandola, dichas placas se mostraran en unos combobox ubicados en cada extremo.En el combobox comparaciones estan los datos a comparar, en este combobox tambien se veran reflejados los nuevos datos que tengan las placas ya sean creadas o traidas desde la importacion.Se elige sobre qué se desea comparar y en el Richtextbox mostrara la informacion resultante.

The screenshot shows a Windows application window titled "Comparar". The interface has a dark blue background with a digital rain effect. At the top, there are three dropdown menus: "GTX 100", "Comparar Memoria", and "GTX 600". To the right of these is a "Comparar" button. Below the dropdowns, there are three main sections. On the left, a section titled "Comparaciones:" contains a RichText control with the following text: "\*\*\*\*Comparacion en Memoria entre GTX 100 y GTX 600\*\*\*\*", "GTX 100 tiene el tipo de memoria es mas actual", "GTX 600 tiene mas capacidad de ram", and "GTX 100 tiene mas frecuencia de memoria". In the center, there is a section titled "GTX 100" with a RichText control showing: "Tipo de memoria: GDDR5", "Capacidad de Ram: 1 GB", and "Frecuencia de memoria: 14". On the right, there is a section titled "GTX 600" with a RichText control showing: "Tipo de memoria: SinAsignar", "Capacidad de Ram: 3 GB", and "Frecuencia de memoria: 11". At the bottom, there are two buttons: "Guardar Informacion en formato .txt" and "Guardar Informacion en formato .json".

Modelo	Tipo de memoria	Capacidad de Ram	Frecuencia de memoria
GTX 100	GDDR5	1 GB	14
GTX 600	SinAsignar	3 GB	11

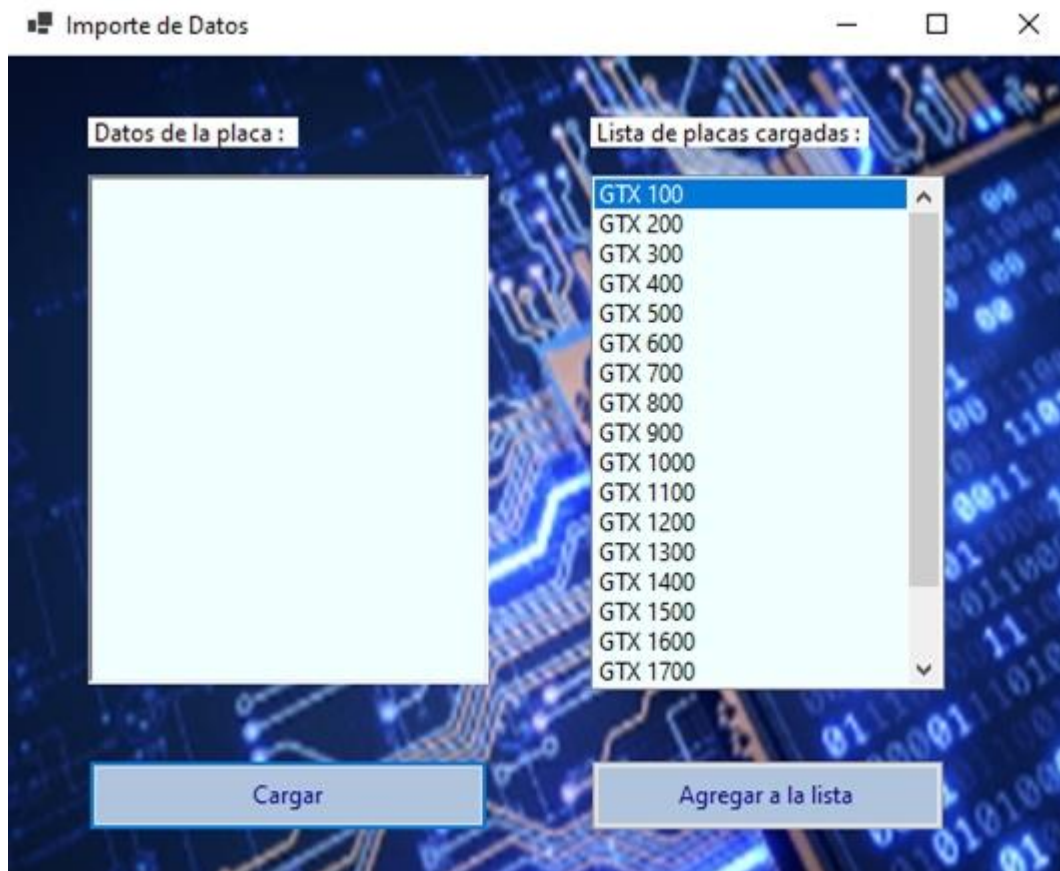
-Análisis de datos: Se podrá analizar las placas cargadas en el sistema según marca, tipo de memoria, capacidad de ram y consumo.

Al seleccionar el botón analizar en cualquier opción, se mostrará en el richtextbox una descripción detallando el porcentaje de placas que cumplen la condición y se actualizará el listbox con esas placas.

The screenshot shows a software application titled "Analisis". On the left, a listbox titled "Placas segun marca Msi" contains two items: "GTX 800" and "GTX 1800". The main area contains several analysis filters, each with a dropdown menu and an "Analizar" button:

- Analizar segun marca:** Dropdown set to "Msi". Result: "En el sistema hay 10.00 % de placas registradas con la marca Msi".
- Analizar segun tipo de memoria:** Dropdown set to "GDDR1". Result: "En el sistema hay 10.00 % de placas registradas con tipo de memoria GDDR1".
- Analizar segun capacidad de Ram:** Dropdown set to "entre 8 y 16 gb inclus". Result: "En el sistema hay 50.00 % de placas registradas con ram entre 8 y 16 gb inclusive".
- Analizar segun Consumo:** Dropdown set to "entre 25 y 50 W inclu:". Result: "En el sistema hay 25.00 % de placas registradas con consumo entre 25 y 50 W inclusive".
- Analizar Placa segun ram:** Dropdown set to "Gigabyte". Result: "Gigabyte tiene un promedio de ram de 8.8gb, esto significa : 151.43 % mas de ram que Intel, el cual tiene un promedio de 3.5gb 25.71 % mas de ram que Gigabyte, el cual tiene un promedio de 7gb 7.95 % menos de ram que Msi, el cual tiene un promedio de 9.5gb 35.38 % mas de ram que Asus, el cual tiene un promedio de 6.5gb".
- Analizar Consumo segun el tipo de memoria:** Dropdown set to "GDDR3". Result: "Consumo promedio tipo memoria GDDR3 es de: 47.50 W. El tipo de memoria GDDR3 no tiene valores para analizar en las placas tipo Msi. El tipo de memoria GDDR3 tiene 53.23 % menos de consumo en las placas tipo Asus. El tipo de memoria GDDR3 tiene 184.21 % mas de consumo en las placas tipo Gigabyte. El tipo de memoria GDDR3 no tiene valores para analizar en las placas tipo Intel. El tipo de memoria GDDR3 tiene 150.53 % mas de consumo en las placas tipo Amd".

**Importacion:** Permite cargar placas, ya creadas, al sistema .Cargandola con todos sus datos generales y tambien datos nuevos que se agregaran a las comparaciones , sin que estas se repitan. Dentro de la solucion existe una carpeta "DatosParaCargar" en donde se podran importar archivos xml para agregarlos al sistema.



The screenshot shows a software window titled "Importe de Datos". It contains two main sections: "Datos de la placa :" on the left, which is a large empty text area, and "Lista de placas cargadas :" on the right, which is a list box containing 17 items from "GTX 100" to "GTX 1700". At the bottom, there are two buttons: "Cargar" and "Agregar a la lista".

Lista de placas cargadas :
GTX 100
GTX 200
GTX 300
GTX 400
GTX 500
GTX 600
GTX 700
GTX 800
GTX 900
GTX 1000
GTX 1100
GTX 1200
GTX 1300
GTX 1400
GTX 1500
GTX 1600
GTX 1700

Buttons: Cargar, Agregar a la lista



**Base de datos:** Permite crear una placa, insertarla en la database; leer las placas que contiene la database , poder visualizar y eliminar una placa de la database; poder guardar en el sistema placas traídas desde la database.

Base de Datos

Placas dentro de la DataBase:

info (doble click en el listbox)

Nombre	<input type="text"/>
Marca	Msi
Tipo de Memoria	SinAsignar
Capacidad de ram	0 GB
Frecuencia	0 GB/s
Consumo	0 W
Longitud	0 CM
Interfaz	0.0 PCI
Rendimiento Minería Bitcoin	0.000 Bit/24hs
Rendimiento Minería Ethereum	0.000 Eth/24hs

Guardar en la base de datos

Cargar Placas desde BD

Eliminar de DataBase

Guardar en Sistema

Implementacion de Temas:

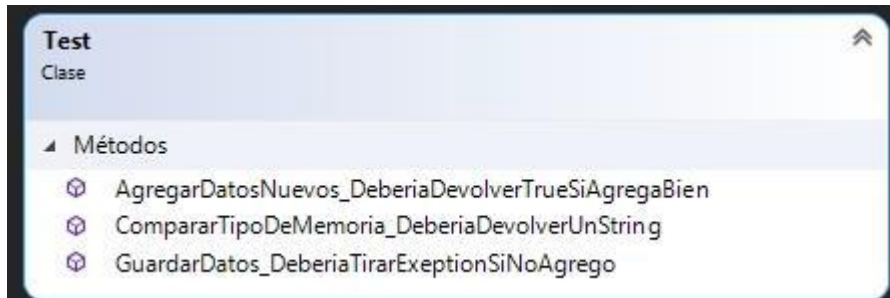
## Excepciones



*ArchivoException: se lanza cuando no se pueda leer o guardar un archivo . Se recomienda desactivar el antivirus cuando se quiera guardar los informes en formato txt.*

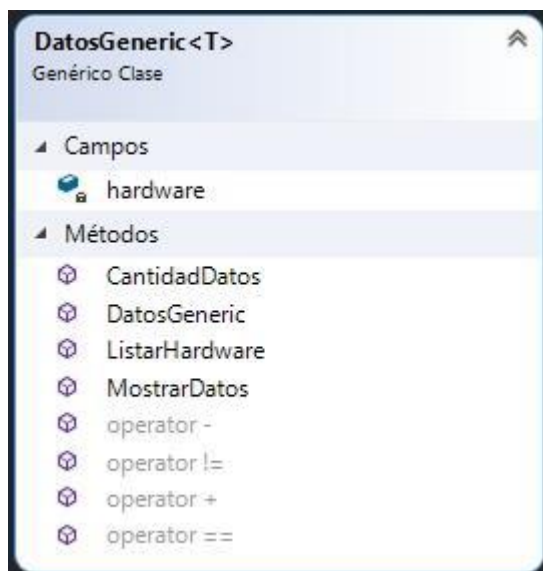
*NoAgregaDatosExeption: se lanza cuando no se puede agregar datos al objeto PlacaDeVideo ya sea por estar repetido o tenga formato incorrecto.*

# Pruebas Unitarias



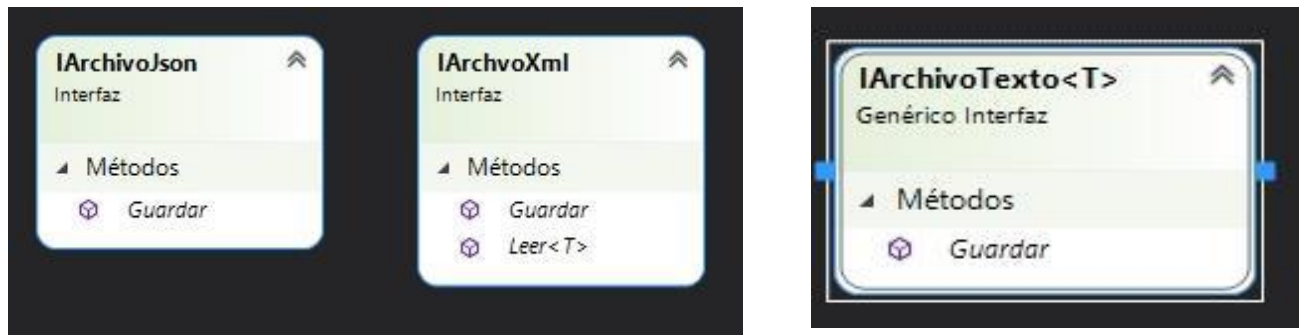
*Realiza los test correspondientes a las funciones de agregar y comparar .Ademas tambien realiza un test para verificar el funcionamiento de la exepcion "NoAgregaDatosExeption".*

## Tipos Genéricos



*Clase genmerica, la cual deriva de una clase Hardware que tenga un constructor sin parametros.Usada para almacenar los datos de una lista de PlacaVideo.*

# Interfaces



*Usadas para la serializacion y desearizacion de archivos tanto .json como .xml y para guardar datos txt*

## Archivos y serialización

**Archivos:**



*Dentro de la clase se encuentran los codigos para guardar los datos en formato .txt*

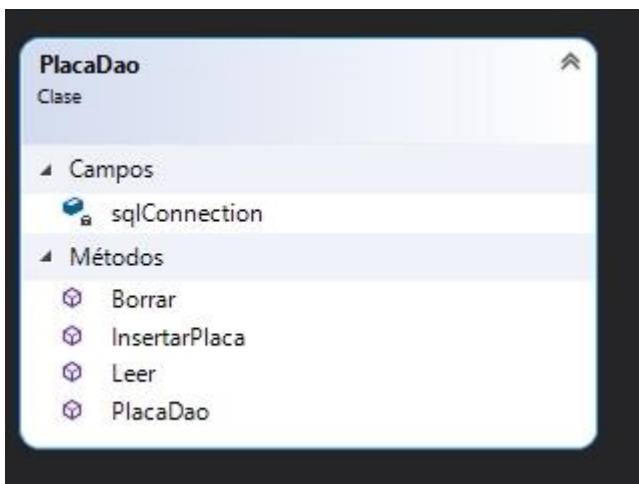


### ***Serializacion:***



*Las clases Sistema y PlacaVideo implementaras los metodos que reciben de las interfases, Dichos metodos contienen el codigo para serializar y desearizar el objeto PlacaVideo a xml .En sistema se guardara un archivo .json donde contendra la informacion de comparaciones realizadas por el usuario.*

## Conexión a bases de datos



*La clase PlacaDao es la encargada de manejar los metodos necesarios para poder conectarse a la base de datos.*

# Delegados Eventosy expresiones lambda

```
39 referencias
public class Sistema : IArchivoJson
{
    public delegate void Cargar();
    public event Cargar CargarLista;

    private DatosGeneric<PlacaVideo> datosOriginales;

    private List<PlacaVideo> listaDePlacasACargarLado1;
    private List<PlacaVideo> listaDePlacasACargarLado2;
    private List<Comparar> comparaciones;
    private int capacidadDatosAAlmacenar;
    /// <summary>
    /// Constructor, inicializa la lista
    /// </summary>
    1 referencia
    public Sistema()
    {
        this.datosOriginales = new DatosGeneric<PlacaVideo>();
        this.listaDePlacasACargarLado1 = new List<PlacaVideo>();
        this.comparaciones = new List<Comparar>();

        this.CargarLista += CargarPlacas;
        this.CargarLista += CargarDatos;
        this.CargarLista += CargarComparaciones;

        CargarLista.Invoke();

        this.listaDePlacasACargarLado2 = new List<PlacaVideo>(listaDePlacasACargarLado1);
    }
}
```

*Delegados y Eventos usados en la clase Sistema para la carga de datos de las listas.*

## Expresiones Lambda:



```
/// analizamos segun tipo de memoria
/// </summary>
/// <param name="tipo"></param>
/// <returns></returns>
2 referencias
public string AnalizarTipoMemoria(PlacaVideo.ETipoMemoria tipo)
{
    double cantLista = this.listaDePlacasACargarLado1.Count;
    List<PlacaVideo> listaAnalisis = new List<PlacaVideo>();
    listaAnalisis = this.listaDePlacasACargarLado1.FindAll((l) => l.TipoDeMemoria == tipo);
    double cantAnalisis = listaAnalisis.Count;
    double analisisFinal;

    analisisFinal = (cantAnalisis * 100) / cantLista;

    return $"En el sistema hay {string.Format("{0:0.00}", analisisFinal)} % de placas registradas con tipo de memoria {tipo}";
}

/// <summary>
/// analizamos segun ram
/// </summary>
/// <param name="datos"></param>
/// <returns></returns>
2 referencias
public string AnalizarSegunRam(string datos)
{
    string retorno = "";
    double cantLista = this.listaDePlacasACargarLado1.Count;
    List<PlacaVideo> listaAnalisis = new List<PlacaVideo>();
    double cantAnalisis;
    double analisisFinal;

    switch (datos)
    {
        case "entre 1 y 4 gb inclusive":
            listaAnalisis = this.listaDePlacasACargarLado1.FindAll((l) => l.CapacidadDeRam >= 1 && l.CapacidadDeRam <= 4);
            cantAnalisis = listaAnalisis.Count;
            analisisFinal = (cantAnalisis * 100) / cantLista;
```



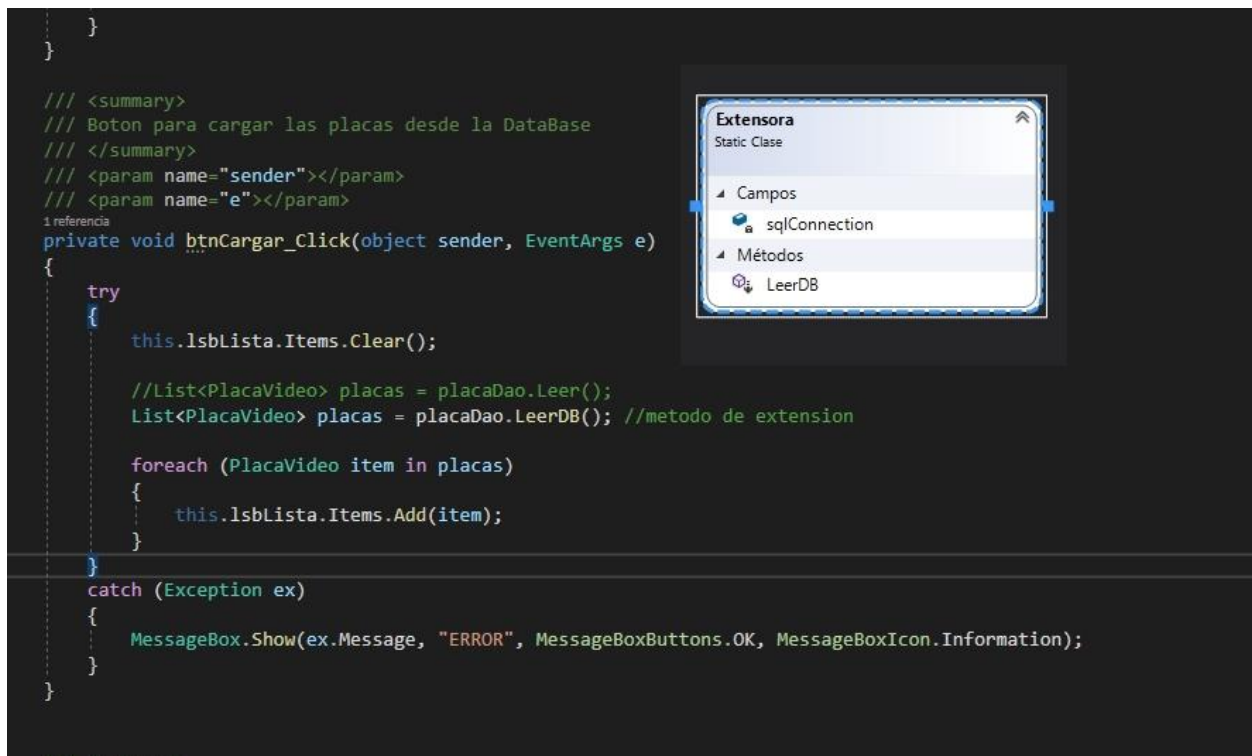
*Usadas para los metodos de Analizar datos en la clase Sistema.*

# Hilos

*En el form FormImporteDeDatos, se implementa un hilo , que permite iniciar un progressbar al momento de apretar el boton “Agregar a la lista”*

*el cual tarda unos 10 segundos en completarse, permitiendo en ese tiempo realizar otras tareas como cargar la base de datos, realizar comparaciones o cargar otra placa al sistema.*

## Métodos de extensión



*Se crea la clase Extensora, para extender la clase PlacaDao y asi tener un metodo de extension que permita leer la base de datos.*