**¿Cómo lanzo una excepción propia?**

Utilizando la instrucción throw.

**¿Cómo capturo una excepción?**

Mediante un bloque try {} catch {}.

**¿Qué función cumple el finally?**

Contiene un conjunto de instrucciones que se van a ejecutar sea cual sea el resultado del flujo de control (si se produce la excepción o no).

**¿Puedo tener varios catch? ¿Por qué?**

Sí, se puede porque un bloque try puede producir diversas clases de excepciones.

**¿Qué etiqueta marca una clase de test?**

[TestClass]

**¿Cómo corro un test unitario?**

Prueba -> Ejecutar -> Todas las pruebas

Ir al menú Pruebas / Ventanas / Explorador de Pruebas. Click derecho sobre la prueba -> Ejecutar

**¿Para qué sirven los test unitarios?**

Se busca escribir casos de prueba para cada función no trivial o método en el modulo para comprobar que el resultado sea el esperado.

**¿Cuántos tipos de testing hay?**

Hay 3 tipos:

* Test unitarios
* Test funcionales
* Test integrales

**¿Qué es una interface?**

Es la descripción de uno o más métodos que una clase debe posteriormente implementar.

**¿Una Interface puede tener propiedades? ¿Y atributos?**

Se pueden especificar propiedades, pero sin la implementación. C# no permite especificar atributos en las interfaces.

**¿Cómo implemento una interface?**

class NombreClase : INombreInterface

**Si heredo de una clase que tiene interface, ¿La estoy implementando?**

No la estoy implementando en la clase hija (¿?)

**Si mi interface tiene la siguiente firma Interface Iarchivos<T>, cuando la implemento, ¿Que se le pasa a T?**

A T se le puede pasar cualquier tipo de dato ya que es un tipo genérico.

**¿Cuándo se reemplace el T por una clase o tipo y para qué sirve?**

Se reemplaza al momento la declaración del objeto o método. Sirve para programar clases o métodos genéricos, capaces de implementar un tipado fuerte, sin necesidad de conocer a priori los tipos que se utilizarán.

**¿Qué restricciones puede tener?**

Ver Tabla Generics.

**¿Qué librería necesito para utilizar archivos?**

System.IO

**¿Cómo agrego datos a un archivo de texto?**

Utilizando el StreamWriter:

using System.IO;

// ...

// Abro el archivo ubicado en una dirección de la máquina

StreamWriter sw = new StreamWriter("C:\\prueba.txt");

// Agrego una línea de texto

sw.WriteLine("Hola mundo!!");

// Cierro el archivo

sw.Close();

**¿Qué es la serializacion?**

Es el proceso de convertir un objeto en memoria en una secuencia lineal de bytes.

**¿Qué diferencia hay entre la serializacion binaria y xml?**

La serialización XML solo serializa los atributos y propiedades (lectura-escritura) públicos. La serialización binaria puede serializar atributos y propiedades públicos y privados y se debe agregar el marcador [Serializable].

**¿Con que método serializo en XML?**

Serializo con el método Serialize(System.IO.Stream stream, Object o) que poseen los objetos de la clase XMLSerializer.

**¿Cómo deserealizo un archivo binario?**

using System.IO;

using System.Runtime.Serialization.Formatters.Binary;

//Objeto que alojará los datos contenidos en el archivo binario.

Dato aux = new Dato();

FileStream fs; //Objeto que leerá en binario.

BinaryFormatter ser; //Objeto que Deserializará.

//Se indica ubicación del archivo binario y el modo.

fs = new FileStream(pathBinario, FileMode.Open);

//Se crea el objeto deserializador.

ser = new BinaryFormatter();

//Deserializa el archivo contenido en fs, lo guarda en aux.

aux = (Dato)ser.Deserialize(fs);

//Se cierra el objeto fs.

fs.Close();

**¿Cómo verifico que exista un archivo?**

Por medio del método File.Exists(string path).

**¿Qué excepción tengo cuando no encuentra el path solicitado?**

FileNotFoundException.

**¿Qué es una tupla?**

Es un objeto único de datos relacionados en una tabla (sería la fila de la tabla). Es una lista ordenada de elementos.

**¿Con que elemento me conecto a una base de datos?**

Me conecto utilizando una cadena de conexión (Connection String) y creando un objeto de la clase SQLConnection al cual le vas a pasar dicha cadena al momento de instanciarlo.

**¿Qué elemento ejecuta una consulta?**

Un objeto de la clase SQLCommand

**¿Cómo inserto un dato en una tabla?**

String consulta;

consulta = “INSERT INTO Personas (nombre) VALUES(‘Pedro’)”;

comando.CommandText = consulta;

conexión.Open();

comando.ExecuteNonQuery();

**¿Cómo hago una consulta a una tabla provincia?**

consulta = “UPDATE provincial SET nombre = ‘Buenos Aires’ WHERE id = 1”;

**¿Cómo borro una localidad?**

consulta = “DELETE FROM provincia WHERE id = 1”;

**¿Con que clase leo en c# una consulta y como la recorro?**

Leo la consulta con un objeto de la clase SQLDataReader y la recorro con un while;

**¿Qué funcionalidad nos brinda un hilo?**

Nos permiten ejecutar varias tareas al mismo tiempo.

**¿Cómo lo lanzo?**

Thread t = new Thread();

t.Start();

**¿Cómo lo freno?**

Thread.Sleep(milisegundos);

**¿Cómo lo bajo?**

if(t.isAlive())

{

t.Abort();

}

**¿Qué es un evento?**

Es el modo que tiene una clase de proporcionar notificaciones a sus clientes cuando ocurre algo en particular dentro del objeto.

**¿Cuántos manejadores puede haber por evento?**

Todos los que se suscriban al evento (¿?).

**¿Cómo asigno un manejador a un evento?**

Con el operador +=.

objEmisor.evento += MetodoManejador;