Parcial 2

1. **Enumere y explique los 4 eventos que maneja el FileSystemWatcher**

Created Evento de cuando se crea un archivo en una carpeta.

Changed Evento que detecta el cambio de tamaño, o atributos del archivo.

Deleted Evento que detecta cuando se borra un archivo.

Renamed Evento que detecta cuando un archivo cambia de nombre.

1. **¿A cuáles de los siguientes espacios de nombres pertenece el objeto Process?**

* System.Diagnostics
* System.Process
* System.Threading
* Todas son correctas
* Ninguna es correcta

1. **¿El objeto EventLog pertenece al espacio de nombres System.Diagnostics?**

* Verdadero
* Falso

1. **¿Cuál es el objeto dentro del Framework.Net que utiliza para manejar las expresiones regulares?**

Regex (pertenece al espacio de nombres System.Text.RegularExpressions)

1. **¿Cuáles de los siguientes métodos pertenecen al objeto DataSet para interactuar con un archivo XML?**

* ReadXMLSchema
* LoadXml
* LoadXmlSchema
* Todas las anteriores son correctas
* Ninguna es correcta

1. **¿Cuál/es de los siguientes métodos de la clase Process me permiten interrumpir la ejecución de un proceso?**

* Abort
* Interrupt
* Stop
* Todas son correctas
* Ninguna es correcta (La respuesta es proceso.KILL() )

1. **¿Cuál/es de los siguientes elementos pertenecen al objeto que maneja las expresiones regulares?**

* isMatch (devuelve true si existe coincidencia)
* Matches (busca todas las ocurrencias)
* Match (busca la ocurrencia según el patrón)
* Todas son correctas
* Ninguna es correcta

1. **¿Cuál/es de las siguientes afirmaciones de XML es/son válidas?**

* Solo puede tener 1 solo nodo raíz
* Las etiquetas no son sensibles a las mayúsculas
* Todas las etiquetas deben cerrarse
* Todas son correctas
* Ninguna es correcta

1. **¿Cuál/es de los siguientes métodos de XML es/son permiten ir al siguiente nodo del mismo nivel?**

* NextNode
* NextSibling
* NextElement
* Todas las etiquetas deben cerrarse
* Ninguna es correcta

1. **¿Cual/es de los siguientes métodos de XML es/son permiten ir al nodo anterior del mismo nivel?**

* PreviousNode
* Previous
* PreviousElement
* Todas las etiquetas deben cerrarse
* Todas son correctas
* Ninguna es correcta (La correcta es PreviousSibling)

1. **¿Cuál es el objeto que me permite leer un archivo xml?**

* XMLFile
* XML
* XMLDocument
* Todas las etiquetas deben cerrarse
* Todas son correctas
* Ninguna es Correcta

1. **¿Cuál es el método que me permite leer un archivo XML?**

* Read
* LoadXml
* OpenXml
* Todas son correctas
* Ninguna es correcta

1. **¿Cuál es el método que me permite ejecutar un proceso dentro de la clase Process?**

* Init
* Begin
* Start (Process.Start(ruta))
* Todas son correctas
* Ninguna es correcta

1. **Enumere 3 tipos de prioridad de un registro del objeto EventLog**

Information, Warning, Error

**¿Cuál es el método que me permite ejecutar un escribir un registro en el visor de sucesos?**

* WriteEntry
* Putentry
* SaveEntry
* Todas son correctas
* Ninguna es correcta

PING PONG parcial 2

1. **¿Para terminar un sub Proceso que tengo que hacer?**

Utilizar el método Abort.

1. **Método sleep de un subproceso ¿qué unidad de tiempo manejo?**

Milisegundos.

1. **¿Para qué sirve un subproceso?**

Para dejar corriendo una tarea en segundo plano, que permite seguir interactuando.

1. **¿Cuáles son los 5 niveles de prioridad de un subproceso?**

ThreadPriority.AboveNormal

ThreadPriority.BelowNormal

ThreadPriority.Highest

ThreadPriority.Lowest

ThreadPriority.Normal

1. **¿Cómo matar un proceso?**

Kill() en el proceso

1. **¿A qué espacio de nombres pertenece Process?**

System.Diagnostics

1. **¿A qué espacio de nombres pertenece Thread?**

System.Threading

1. **¿Para qué sirve el Event Log?**

Para registrar diferentes sucesos que ocurren a las aplicaciones.

1. **¿Cuáles son los niveles de prioridad del Event Log?**

Error, Warning, Information, Seguridad, Auditoria.

1. **Cosas a tener en cuenta para XML bien formados.**

Debe tener 1 solo nodo Raiz, las etiquetas deben abrir y cerrarse (es sensible a las mayúsculas y deben corresponderse la de abrir con cerrar).

Si respeta las reglas es un documento bien formado.

1. **Objeto por excelencia para navegar XML que puede explorar todas sus ramas.**

XmlDocument

1. **Métodos para cargarlo**

Load(ruta) para un archivo, y loadXML(string) para un string en memoria.

1. **¿Qué es un archivo XML esquema?**

Es el archivo que contiene la estructura del XML

1. **¿Si quiero ir al primer NODO que método uso?**

FirstChild

1. **¿Para ir al siguiente NODO que método uso?**

NextSibling

1. **¿Para ir al anterior NODO que método uso?**

PreviousSibling

1. **¿Para ir al NODO PADRE que método uso?**

ParentNode

1. **¿Para ir a la raíz del xml que método uso?**

DocumentElement

1. **Con que puedo usar LINQ**

Colecciones, Bases de Datos, XML

1. **¿Si hago <NODO/> es válido?**

Sí, es válido, es lo mismo que una etiqueta vacía <NODO></NODO>

1. **¿Para utilizar LINKQ con XML, que espacios de nombres utilizo?**

System.Xml.Linq

**Varios**

FileSystemWatcher

Escucha las notificaciones de cambio del sistema de archivos y genera eventos cuando cambia un directorio o un archivo de un directorio.

Created Evento de cuando se crea un archivo en una carpeta.

Changed Evento que detecta el cambio de tamaño, o atributos del archivo.

Deleted Evento que detecta cuando se borra un archivo.

Renamed Evento que detecta cuando un archivo cambia de nombre.

**Procesos**

Un proceso es una instancia de un programa.

Espacio de nombre System.Diagnostics

Process.Start("iexplore", "D:\gente.xml") “Inicia un proceso externo”

Application.DoEvents “Llamar a este método hace que el proceso actual quede suspendido mientras se procesan todos los mensajes en espera de la ventana”

Process.GetProcesses() “Obtener los procesos”

Espacio de Nombre para Hilos

System.Threading

Thread.Sleep(100) “Suspende el subproceso actual durante la cantidad de tiempo especificada.”

**Event LOG**

EventLog.WriteEntry("Application", Mensaje, EventLogEntryType.Error)

EventLog.GetEventLogs() “Obtener Eventos del LOG”

**Expresiones Regulares**

Estándares de REGEX

* Perl
* PCRE
* POSIX

Espacio de Nombres

System.Text.RegularExpressions

Métodos

IsMatch devuelve true si existe coincidencia.

MATCH busca la ocurrencia según el patrón

MATCHES busca todas las ocurrencias.

**XML**

Ds = objeto DataSet

¿Cómo escribir el Esquema al disco?

ds.WriteXmlSchema("D:\GENTE\_estructura.xml")

¿Cómo escribir el XML?

ds.WriteXml("D:\GENTE.xml")

¿Cómo leer el Esquema desde el disco?

ds.ReadXmlSchema("D:\GENTE\_estructura.xml")

¿Cómo leer el XML desde el disco?

ds.ReadXml("D:\GENTE.xml")

Desde XmlDocument

Dim doc As New XmlDocument

doc.Load("D:\PERSONAS.xml")

Preguntas Primer Parcial

1. **Que es la sigla ACID**

Atomicidad

Consistencia

Aislamiento

Durabilidad

1. **¿Qué es una DB consistente?**

Que no tenga registros huérfanos, que la información es confiable

1. **¿Para qué sirve una transacción?**

Para que se ejecute todo o nada, pasa de un estado consistente a otro.

1. **Método encargado de confirmar una transacción**

Commit

1. **¿Método en cargado de volver atrás?**

Rollback

1. **¿Quién instancia una transacción?**

El objeto SqlConnection sobre una conexión ya abierta

1. **¿Cuál es el método encargado de instanciar una transacción?**

conexion.BeginTransaction

1. **¿Diferencia entre relación y restricción?**

En la relación de tablas: la clave foránea es la clave primaria en otro.  
La RIR impide borrar claves primarias sin borrar los hijos.

1. **3 Tipos de Elementos Command**

CommandType.StoredProcedure

CommandType.TableDirect

CommandType.Text

1. **¿Cuál es la diferencia entre ADO conectado Vs ADO desconectado?**

El conectado devuelve el resultado en la memoria del servidor (ocupa recursos del servidor) y el desconectado devuelve el resultado en “mi memoria”. (Ocupa recursos del cliente).

1. **ADO Conectado, el objeto por excelencia**

SqlDataReader

1. **¿Para qué sirve ExecuteScalar?**

Devuelve la primera columna del primer registro. (Sirve para máximo, mínimo, count)

1. **¿Qué devuelve ExecuteNonQuery?**

La cantidad de filas afectadas.

1. **Que método instancia el DataReader**

El ExecuteReader del Sqlcommand

1. **Interface que implementan los Parameter**

IDBDataParameter

1. **Elementos necesarios para acceder a una DB de Access**

Son necesarios 4 elementos:  
\* DataSource  
\* Security Integrated  
\* Initial Catalog  
\* Provider

1. **¿Cuándo no lleva Provider?**

Con SQLSERVER porque se considera la DB nativa de .net

1. **¿A que estoy obligado si seteo, PERSIST SECURITY INFO en true?**

Necesito pasar los atributos UserID y Password de la Base de Datos, es poco recomendable ya que esto viaja en texto plano y es poco seguro.

1. **Cuál es el objeto por excelencia para ADO DESCONECTADO**

SqlDataAdapter

1. **¿Porque dividir en capas?**

Por el bajo acoplamiento, cohesión, para escalar sistemas

Independencia Funcional, lo único que tiene que haber es colaboración.

1. **Cuando hablo de capas puedo decir que forzosamente una capa es un proyecto o es más de un proyecto?**Una capa puede tener más de un proyecto.
2. **Que ventajas da usar parámetros contra usar SQL**

Hace comprobación binaria, y todo es parte de los valores, evitar el injectSQL.

1. **Operaciones básicas de la DB**

Leer, Escribir

1. **¿Dentro de la escritura?**

Insert, update, delete.

1. **¿Dentro de la Lectura?**

select

1. **¿Cuál es la regla de declaración de parámetros en SQLSERVER?**

Tienen que comenzar con ARROBA @

1. **Como se llama el Objeto que captura el error de SQL**

SqlException

1. **Para que utilizo procedimientos almacenados?**Los utilizo porque son más performantes, ya que se encuentran precompilados en la base de datos.
2. **Es válido que el DELETE devuelva 0 registros afectados?**Es válido que el delete devuelva 0 registro afectados, ya que puede ser que en mi filtro no entren registros con esas características.
3. **Es válido que el INSERT devuelva 0 registros afectados?**   
   No es válido, ya que al realizar un insert debería si o si devolver 1 registro afectado al menos.
4. **A que se conoce con el nombre de entidad?**Una entidad es cuando le quito el comportamiento a una clase, quedando asi solo su identidad.
5. **Que es una clase de concurrencia?**Una clase de concurrencia es cuando la clase no posee estado pero si acción.

**CLASE 1**

Elementos que se necesitan para conectar .Net a SqlServer:

DataSource: Origen de datos.

El origen de datos puede ser:

* IP y un puerto por default de escucha.
* IP : Puerto \ Instancia
* .\

InitialCatalog: Catalogo inicial (la base de datos a la cual yo quiero impactar)

IntegrateSecurity: Seguridad Integrada

Provider: El motor que voy a utilizar, si utilizo .net contra sql no se lo utiliza sino explota la app.

xxConnection: Elemento que se encarga de abrir y cerrar conexiones.

ConnectionString: Contiene los parámetros para conectarme a la base de datos.

xxCommand: Elemento que envía los comandos que quiero impactar contra la base de datos. (El objeto comando tiene contención física del conexión)

Parameters: Objetos SQL parameters.

xxDataReader: Elemento que se encarga de transformar información de la base de datos a información legible por uno.

SQL: Lenguaje estructurado de consulta

Archivo secuencial: No tienen una longitud fija.

“El motor de base de datos almacena en tablas, en las tablas se almacenan registros.”

Seleccionar (Consulta SQL):  
**SELECT** CAMPO1, CAMPO2, CAMPO3   
**FROM** TABLA   
**WHERE** Genero = F **AND** Edad > 18  
**ORDER BY** Edad **DESC,** EstadoCivil **ASC**

CREATE PROCEDURE ListarTestigos

as

begin

SELECT \*

FROM Testigo

ORDER BY Nombre

end

Alta (Consulta SQL):  
**INSERT INTO** TABLA  
(CAMPO1, CAMPO2, CAMPO3)  
**VALUES(** 5 , ’Texto’ , 3.14 **)**

CREATE PROCEDURE InsertarTestigo

@ID int, @Nombre varchar(50), @Apellido varchar(50), @Edad int, @Comentario varchar(90)

AS

BEGIN

INSERT INTO Testigo(ID,Nombre,Apellido,Edad,Comentario) values(@ID,@Nombre,@Apellido,@Edad,@Comentario)

END

Editar (Consulta SQL):  
**UPDATE** TABLA   
**SET**CAMPO1 = 5 ,  
CAMPO2 = ‘Mi auto’ ,  
CAMPO3 = 2.5  
**WHERE** DNI = ’28.555.000’

Eliminar (Consulta SQL):  
**DELETE FROM** TABLA **WHERE** ID = 5

Se necesitan Espacios de Nombre:

Imports System.Data  
Imports System.Data.SqlCliente   
(Tiene todos los objetos necesarios para conectarme y manipular una base de datos de SQL)

**Cliente** ----------------------------- **Servidor**  
DataReader------------------------------→ BOF de la tabla  
DataReader.True------------------------→ Va pasando de registro en registro cada vez q trae True

ADO NET en MODO CONECTADO: consumo recursos del lado del servidor.

**El SqlCommand es el que se encarga de inicializar el SqlReader**

El executeNonQuery es recomendable trabajarlo dentro del trycatch.

CLAVE PRIMARIA: Columna que me permite hacer único a un registro.

ExecuteScalar = retorna una sola celda de toda la consulta, devuelve la primera columna del primer registro.

**Clase 2**

Relación: Clave primaria relacionada con clave foránea. (Es el juego del clave)

Restricción de referencial: Impedimento que te da el motor para que no queden registros huérfanos.

Base de Datos Consistente: Base de datos sin tabla con registros huérfanos.

Huérfano: Que apunta algo inexistente (Base de datos inconsistente)

**Clase 4**

Consistencia con respecto a los registros huérfanos.

Cada dependencia apunta a su contenedor.

Si los datos no son fiables, la base de datos es inconsistente.  
Pasar de un estado consistente a otro.

**Transaccion**: Para llevar a cabo una serie de operación y después al final si está todo bien hago commit (confirmo)y si algo salió mal hago un rollback.   
No todos los motores de base de datos soportan transacciones.

**ACID: (En ingles):**  
Atomicidad  
Consistencia  
Aislamiento  
Durabilidad

Objeto para la transacciones -> SQL**Transaction**

El objeto **SQLTransaction no se puede iniciar**, lo tienen que **iniciar por mi**. (Lo hace el **objeto** **conexión** con el método **BEGIN TRANSACTION**)  
Las transacciones siempre van contra una conexión, para tener una transacción tengo que tener una conexión abierta.

El SQLConnection es el que se encarga de iniciar el SQLTransaction

Estoy obligado a agregar la transacción al SqlCommand

**Procedimientos almacenados**: Subrutinas que están precompiladas en la base de datos.

PROCEDIMIENTO ALMACENADO

**CREAT**: Para crearla  
**ALTER**: Para modificarla

CREATE PROCEDURE <nombre>  
  
AS  
--Parametros que el SP espera recibir

@ID INT, @NOM VARCHAR (50)  
BEGIN

--Instrucciones SQL que quiero que mi SP contenga y haga.

END

CON **DECLARE** DECLARO UNA VARIABLE DENTRO DE UN SP

CON

**DATATABLE**: es una **tabla** que va a tener en su interior una **colección** de **datacolumns** y una colección de **datarows**, como es genérica no depende de un motor de base de datos especifico.  
Solamente voy a necesitar un **objeto** que **lea** la **consulta** **y** **me** **cargue** esta tabla **sqldataadapter.**

El **SQLDATAADAPTER** tiene 4 objetos de tipo Command:  
SELECT  
INSERT  
DELETE  
UPDATE

El comando SELECTCOMMAND del SQLDATAADAPTER se encarga de seleccionar el comando es un SQLCOMMAND.  
El **Adapter** tiene un comando **FILL** que me **llena** la **tabla** con el resultado de lo que ejecuta el **comando** que le **pase** en **selectcommand**.

**CLASE 5**

Ejecutable

DLL de Negocio

DLL de acceso a datos

La **arquitectura en capa busca** que cada capa sean lo mas **cohesivas** posibles (sean lo mas **independiente** posible una de otra)

**Capa:** es la agrupación conceptual de determinados elementos que justamente coinciden en determinada funcionalidad.

Entre proyecto cliente y proyecto servidor, necesito:

Que el cliente conozca al servidor, debo realizar una referencia de uno a otro.

Siempre **se** **arranca** a programar por la **capa** de **NEGOCIO**.

**Espacio de nombre**: agrupación lógica de funcionalidad. (Cada proyecto tiene un espacio de nombre raíz). Sirve para quitar ambigüedad (si tengo 2 clases con el mismo nombre, pero ahora van a tener diferentes espacio de nombre Ej: Sergio.Usuario <> Nicolas.Usuario).

Todos los **parámetros** **implementan** la **interfaz** **IDbDataParameters,** esto sirve para desacoplar.

**HASHTABLE**= **Tabla** que se maneja en **tupla** (una **clave** y un **elemento**)  
La diferencia con el sorter list, es que la clave permanece en el orden que yo la agregue.

**REFERENCIAS 3 capas:**

APLICACIÓN: Negocio

NEGOCIO: Datos

**En 4 capas:**

Se divide en 2 la capa de Negocio.

**Estado** por un lado **y** el **comportamiento** por el otro, se obtiene entidad.

**ENTIDAD**: Posee estado, pero no comportamiento.

Se genera una capa transveral que posee entidades, BUSINESS ENTITY (BE)

Al negocio se le pone BLL (Capa de enlace de negocio – Business Library)

La de acceso a datos = DAL (DATA ACCESS LEVEL)

En la DAL voy a tener puertas de entradas, **MAPPER** (Se encargan de transformar registros en entidades de negocio y viceversa)

**CLASE 6**

**GDI**: Graphic Device Interface. (Permite trabajar tanto con gráficos vectoriales como imágenes)

**Grafico vectorial**: Aquel que trazando puntos se van marcando líneas, cuadrados, etc. (Se va trazando a medida que se va dibujando).

Mapa de bits:

En el mundo informático la **coordenada 0,0** va a estar dada en el **punto izquierdo superior de la pantalla**.

**Point** = contiene una tupla X e Y, define un punto dentro de mi pantalla o contenedor, etc.

**Superficie**= determinada por el ancho y por el alto.

**Size**= contiene los 2 valores (ancho y alto)

Objeto **Rectangle**, tiene la X y la Y y un size.

El **objeto graphic** se desprende del espacio de nombre **System.Draw**

* + No es un objeto que yo lo pueda instanciar, es un objeto que lo va a construir de manera estatica y de forma compartida la misma clase Grafic.

Si quiero dibujar sobre el formulario:

G = Me.CreateGraphics

Tengo 3 formas de poder dibujar contra 3 distintos elementos:

1. Contra un formulario:
   1. **FROM HWND** (HWND manejador de la ventana, con este el sistema operativo se da cuenta que ventana esta activa. Este handle es una dirección de memoria)
   2. **FROM HDC** (Control de dispositivo, objetos que tienen que estar forzosamente contenidos dentro de una ventana)
   3. **FROM IMAGE** (Apartir de una imagen que tenga cargada en memoria, todos los dibujos que vaya haciendo lo puedo ir impactando en una imagen)

El Objeto Grapich necesita a la hora de dibujar:

* **Lapiz = Objeto PEN (que contiene en su interior otro objeto llamado COLOR, que nos resuelve toda la temática del color) [RGBA-rojo,verde,azul,nivel de transparencia]**
  + Ancho del lápiz
  + Estilo del lápiz (línea solida, punteada,etc)
* El lápiz sirve para dibujar líneas o bordes nada mas.

**Los formularios y controles tienen una propiedad LOCATION que es donde esta en arranque.**

**Para rellenar los objetos utilizo el FILLRECTANGLE, FILLELLIPCE.**

**BRUSH= brocha, la cual tiene una iteración de color. [Primero tengo que rellenar, es decir, pintar y luego dibujar el contorno]**

**Codigo de captcha = Texto que no es string, termina siendo un dibujo.**

**Que es un archivo bmp ‘mapa de bits’?   
dividio en 2 partes**

1. **Encabezado: Va a tener un determinado sector donde arranca el dibujo y donde termina.**
2. **Cuerpo: Con bits**

El JPG en su interior también tiene:

1. **Encabezado: es más grande que el de bmp, tiene un sector de comentario. Indica el tipo de compresión que va a tener el cuerpo.**
2. **Cuerpo: esta comprimido.**

Recuperatorio

1. **Enumere las siglas acid**

Atomicidad (se ejecuta todo o nada)

Consistencia (pasa de un estado consistente a otro)

Aislamiento (una operación no afecta a otra)

Durabilidad (persiste y no se puede deshacer)

1. **Cuáles de los siguientes objetos se usa en ado desconectado**

* sqldatareader
* Sqldatareader
* Datareader
* Todas son correctas
* Ninguna es correcta

1. **Ado desconectado, gestiona una conexión abriéndola para recuperar datos y cerrándola después para volver abrirla para escribir y volver a cerrarla**

**Verdadero** Falso

1. **Cuál es el objeto que se encarga de abrir una conexión a la base de datos**

SqlConnection

1. **Acerca de la siguiente afirmacion el objeto sqltransaction posee un constructor friend (hasta aca correcta) y no permite instanciarlo Verdadera (el contructor friend no permite instanciar) y debo usar el metodo begintransaction del sqlconection**

**Verdadero** Falso

1. **Cuál de los siguientes métodos pertenecen al objeto sqlcommand**

* Execute
* Executenonquery
* Executereader
* Executescalar
* Todas son correctas
* Ninguna es correcta

1. **Cuál de los siguientes métodos pertenece al objeto SqlDataAdapter**

* Update
* Begintransaction
* Fill
* Todas son correctas
* Ninguna es correcta

1. **Cuál es la propiedad que permite establecer los elementos para conectar a una base de datos en el objeto conection**

ConnectionString

1. **Cuál de los siguientes métodos permite vincular un objeto graphics al formulario**

* Graphics.FromHwnd
* Graphics.fromForm
* Graphics.form
* Graphics.formparent
* Todas son Correctas
* Ninguna es Correcta

1. **Desarrolle las instrucciones necesarias para actualizar un registro de la tabla alumno que posee un id nombre apellido, se debe incluir, Transaction y commit**