Aplicada 1

Tarea: A Capitulo 1-4

Leer capitulos 1-4 hacer Auto Evaluación y los siglos ejercicios: Cap. 1 ejercicios 1 y 5, Cap. 2. 1, 3, 5, cap. 3: 1, 4, 5 cap. 4 1, 7, 5

Capitulo: 1 C# .NET

1- Cual es la historia del desarrollo de los aplicaciones para windows?

Hace algunos años la única forma como se podía programar para windows era hacer uso de un compilador de C o C++ y de una API de windows. El API es una gran colección de funciones que se relacionan, las que nos permiten comunicarnos con el sistema operativo. Por medio del API de win 32 se programaban las ventanas, botones y demás elementos.

El Problema de este tipo de programación es que el API de win 32 es realmente complejo y enorme. Con miles de funciones en su interior, por lo que pocos programadores podían conocerlo en su totalidad, Pero la complejidad no solamente estaba en la cantidad de funciones, tambien en la sintaxis y la forma como se programa.

Para facilitar la programación de aplicaciones para windows surgen diferentes opciones; la finalidad de estos intentos era poder hacer las aplicaciones sin tener que pasar por la complejidad de win 32. Uno de estos intentos fue conocido como OWL, sin embargo, obtuvo mas exito MFC, creado por Microsoft.

-MFC es un conjunto de clases que envuelve a win 32 y facilita la programación. Con MFC los procesos más comunes se agrupan en funciones de tal forma que con una simple llamada a la función MFC se puede una determinada tarea, para la que antes necesitabamos por lo menos 10 llamadas en win 32 y muchos parametros.

Sin embargo win 32 esta debajo de MFC, la programación MFC simplifica mucho las cosas, pero muchos programadores venian de paradigmas de programación estructurada no se sentian a gusto en él. Otras opciones surgieron en Visual Basic este lenguaje logró gran popularidad, especialmente en Latinoamérica. Visual Basic tambien trabajaba por arriba de win 32, pero basa su sintaxis en el antiguo lenguaje

Basic. Es muy sencillo de aprender y una caracteristica que le dio gran popularidad fue la facilidad con la que se podian crear interfaces de usuarios y conectividad a base de datos. Pero hasta antes de la version .NET, este lenguaje tenia ciertos limites ya que no se podia llevar a cabo programacion orientada a objetos con el.

Otro lenguaje que surge, pero con su propio Framework, es Java, su principal ventaja es ser multiplataforma. Una de sus caracteristicas es el uso del runtime, la app en lugar de correr directamente en el microprocesador se ejecuta en un programa llamado runtime y este se encarga de ejecutar el codigo en el microprocesador correspondien

Cuando deseabamos tener un programa que se pudiera ejecutar, era necesario compilarlo. Cada uno de los lenguajes tenia su propio compilador pero no era sencillo poder compartir el codigo de C++ con el de Visual Basic ya que el traducir entre lenguales era dificil. Para poder compartir codigo entre lenguajes surge un modelo conocido como COM, este permite crear componentes binarios, esto quiere decir que

posible programar un componente en visual Basic y un programador de c++ puede tomarlo y hacer uso de él. Esto se debe a que el componente ya es código compilado y no código fuente en el lenguaje de origen. La programación de COM también tenía sus complejidades y surge ATL para ayudar en su desarrollo.

Con esto, llega el momento en el cual es necesario ordenar ; facilitar y ordenar el desarrollo de las aplicaciones para windows con es filosofía surge .NET

7-¿Qué problemas ayuda a resolver .NET?

.NET Framework soluciona los problemas mediante CLR que es independiente del lenguaje, de la plataforma y del estandar y con el uso del estandar del mercado extra la independencia del lenguaje. .NET permite a los desarrolladores generar una aplicación en cualquier lenguaje basado en .NET y saber que la aplicación web servirá en cualquier cliente que lo soporte.

3-¿Que es un assembly?
es una coleccion de tipos y recursos compilados para funcionar en conjunto y formar una unidad logica de funcionalidad.

4-¿cuál es la definición de CIL y CLR?
CIL es un lenguaje ensamblador orientado a objetos, y está basado en pilas. Es ejecutado por una maquina virtual.

CLR es un programa de ejecución comun a todos los lenguajes. Este programa se encarga de leer el codigo generado por el compilador y empieza su ejecución.

5-¿Cuál es la diferencia entre Write() y Writeline()?
Mientras write() y writeline() son metodos de clase consola. La diferencia entre write() y writeline() es que console.write se usa para imprimir datos, sin imprimir la nueva linea, mientras que console.writeline se usa para imprimir datos junto con la impresión de la nueva linea.

6-¿cuál es el significado de jitter?
Al compilador jit también se conoce como jitter. forma parte del runtime y es muy eficiente, si el programa necesita volver a ejecutar un codigo que ya se ha compilado, el jitter en lugar de volver a compilar ejecuta lo ya compilado, mejorando de esta forma el desempeño y los tiempos de respuestas de cara al usuario.

7-Mencione algunos compiladore de c# que podemos utilizar.
visual studio, BaseAddress, PathMap code pag, entre otros

8-¿Qué es el CTS?
El CTS define los tipos de datos de .NET y las construcciones de programacion de los lenguajes que el CLR puede utilizar de forma adecuada y correcta.

9-¿ Porqué .NET Puede ser multiplataforma?
El framework de .NET se puede ejecutar en muchas plataformas, no solo en Windows. Esto significa que podemos programar en una plataforma y si otra plataforma tiene runtime nuestro programa se ejecutará sin ningún problema.

10-¿ Qué otro lenguaje que use .NET existe?
C++, Visual Basic, C#, J#

11-¿ Cuál es la ultima versión de .NET
.NET Framework 4.6.2 es el ultimo

17-¿ Qué otro lenguaje que use .NET existe?
Visual basic, C++, Python, Fortan

Capitulo 2

Los elementos basicos de un programa.

1-¿Qué es un algoritmo?

El programa de la computadora no es otra cosa que la lista de pasos ordenados que la computadora tiene que seguir para hacer una actividad. A esta lista de pasos le llamamos algoritmos. Un algoritmo son los pasos necesarios para llevar a cabo una accion

2-¿Qué caracteristicas tienen los algoritmos?

Un algoritmo debe de cumplir ciertas caracteristicas de acuerdo con [3] en las caracteristicas fundamentales estan: un algoritmo debe ser preciso e indicar el orden de la realizacion de cada paso. Un algoritmo debe estar definido. si se sigue un algoritmo dos veces, se debe obtener el mismo resultado cada vez. Una caracteristica importante del algoritmo es que tiene un punto de inicio y punto final lo que indica que los pasos se llevan a cabo de forma secuencial uno tras otro.

3-¿Que es una sentencia?
Es una linea de codigo al mando de una tarea, cada programa consiste en una secuencia de sentencias

4-¿como se finalizan las sentencias?
la sentencias se finalizan con punto y coma.

5-¿cual es la diferencia entre write() y writeline()?
la diferencia entre los metodos write() y writeline() despues de escribir el mensaje writeline() inserta un salto de linea, por lo que lo proximo que se escriba aparecerá en el renglón sigte. por su parte el metodo write() no lleva a cabo ningun salto de linea y lo sigte que se escriba será en la misma linea

6-¿Qué es una cadena?
Es una coleccion de caracteres, es decir, letras, números y signos. las cadenas se delimita con comillas dobles.

7-¿Qué es una variable?
una variable en programacion es un elemento de datos cuyo valor puede cambiar durante el curso de la ejecución de un programa.

8-¿ como mostramo el valor de una variable?
se le asigna un valor determinado a una variable. y este valor se requiere mostrarlo en una etiqueta <p7 identificada id='valor'

9-¿ como le solicitamos un dato al usuario?
C# nos provee un método que pertenece a la clase consola. El método se llama

10-¿ Cómo se convierte una cadena a un valor numérico?
El método convert.i() toma como parámetro una variable de cadena y devuelve el valor numérico equivalente.

11-¿ Qué son los operadores Aritmeticos?
los operadores aritméticos realizan operaciones matemáticas, como suma, resters, multiplicac y división siempre que usemos una metodol logia correcta de resolución de problemas

17-¿ De que formas podemos agrupar operaciones Aritmeticas?
por medio de paréntesis

Capitulo 3

El Programa toma decisiones

1-¿Qué son las estructuras selectivas?
son aquellas que nos permiten hacer una selección entre dos o más rutas de ejecución posibles, esta se llevara a cabo según el valor de una expresión

2-¿cuándo se ejecuta la sentencia if?
la sentencia condicional if se usa para tomar decisiones, este evalua basicamente una operación logica, es decir, una expresión que de como resulta True o False y ejecuta la pieza de código sigte siempre y cuándo el resultado sea verdadero.

3-¿Què colocamos entre parentesis en if? una expresión

4-¿Qué es una expresión relacional?
se usan para expresar la relación que exste entre dos valores. los valores pueden estar contenidos adentro de variables o se coloca explicitamente.

5-¿Cuales son los operadores de una expresión relacional?

signo	operador
==	Igualdad
!=	NO Igual
>	Mayor que
<	Menor que
>=	Mayor que igual
<=	Menor que igual

6-¿A qué valores posibles puede evaluar una expresión relacional o lógica?
la expresión sera evaluada, pero el resultado de la evaluacion tendra unicamente dos valores posible True y false

7-¿Qué es una expresión lógica?
son operaciones que dan como resultado dos posibles valores falso o verdaderos.

8-¿cuáles son los operadores de las expresiones lógicas?

operador	significado	
&&	y	and
||	o	or
!	no	not

9-¿Qué operador solamente necesita un operando?

El de la negación

10-¿ como funciona el switch?
En el programa de las operaciones cada una de ellas es un caso de uso. El valor de la variable se compara con un valor para cada caso, si el valor coincide entonces se empieza a ejecución el codigo a partir de esa linea

11-¿qué es la variable de comparación y como se coloca?
la variable de comparacion sera la variable opcion, ya que el valor de esta será comparado para cada operación.

12-¿ como definimos los casos y como usamos break?
para indicar un caso, usamos case seguido del valor de comparación y dos puntos, usamos break al final de cada codigo de caso

Capitulo 4

Creación de ciclos

1- ¿Qué es un ciclo?
es una secuencia de instrucciones de código que se ejecuta repetidas veces, hasta que la condición asignada a dicho bucle deja de cumplirse.

2- ¿Cuáles son las partes del ciclo for?
El ciclo for tiene cuatro partes principales: Inicialización, condición, incremento y codigo.

3- ¿Como colocamos el valor inicial de conteo en un ciclo for?
por medio de una asignación

4.- ¿Cómo colocamos el valor final de conteo en ciclo for?
Por medio de un incremento

5.- ¿Cómo se lleva a cabo el incremento en un ciclo for?
El valor del incremento puede ser un valor colocado explícitamente o el valor que se encuentra adentro de una variable.

6.- ¿Cómo funciona el ciclo do while?
funciona para que el programa se repeta el número dos veces necesarios aun sin saber cuantas veces son.

7.- ¿Por qué el ciclo while se lleva a cabo al menos una vez?
Porque si las condiciones no se cumple desde el inicio entonces el ciclo nunca se lleva a cabo

8.- ¿se necesita punto y coma al finalizar el ciclo do while?
no lleva punto y coma al final del do while

9.- ¿cómo funciona el ciclo while?
un ciclo while es una estructura que se utiliza para ejecutar un bloque de código repetidamente hasta que se cumple una condición determinada. Al ejecutarse, se ejecuta el código dentro del ciclo while y luego se evalua la condición de la terminal

10.- ¿cuántas veces se puede repetir el ciclo while?
El bucle se ejecuta siempre una vez, y al final se evalua la condición para decir si se ejecuta otra vez el bucle o se termina su ejecución.

11-¿Qué tipo de condicion podemos colocar en el ciclo while?
la condicion ha de ser una sentencia que devuelva un valor booleano, y esta puede ser el valor boleano.

12-¿se coloca un bloque de código en el ciclo while? Si