|  |  |
| --- | --- |
| Fecha : | 19/04/2023 |
| Nombre: | JOSE DE JESUS SANCHEZ XIHUITL |

**C#**

**Instrucciones: Lee detenidamente y conteste verdadero (V) o falso (F) en el recuadro final.**

1. En una aplicación de consola para leer los parámetros utilizaremos un **array** de variables de tipo **string** que llamaremos “**args**“. **(V)**
2. Para cambiar el color de fondo de una aplicación de consola utilizaremos **Console. ColorBackground = System.Color(“Green”); (F)**
3. La siguiente instrucción **Console.WriteLine(“\nChar: {0} y {1}”, Char.MinValue, Char.MaxValue);**nos devolverá una “a” minúscula y una “Z” mayúscula. **(F)**
4. El tipo “**long**” es el tipo de número entero más alto que podemos representar en C#. **(F)**
5. El tipo “**decimal**” ocupa 128 bits (el doble que un tipo **long**) y es el tipo de número con decimales más grande que se puede representar en C#. **(V)**
6. El tipo “**double**” ocupa lo mismo que el tipo “**ulong**“. **(V)**
7. El tipo “**char**” ocupa 2 bytes que es lo mismo que el tipo entero “**short**” que a su vez también ocupa lo mismo que el tipo “**ushort**“, es decir 16 bits. **(V)**
8. Las variables en C# a diferencia de Java, no distinguen entre mayúsculas y minúsculas. **(F)**
9. No se pueden utilizar palabras reservadas como nombres de variables. **(V)**
10. Las variables siempre empiezan por una letra o por el símbolo “**@**“. **(V)**
11. Para escribir una comilla simple en una variable alfanumérica podemos utilizar la secuencia de escape \’. **(V)**
12. Este es un ejemplo de declaración y asignación de una variable lógica: **Bool EsCierto = true; (F)**
13. Una variable tipo **Object** podemos almacenar cualquier cosa. **(V)**
14. El tipo **dynamic** representa una dirección de memoria. **(F)**
15. Mediante la instrucción **“$”** podemos asignar el valor **null** a cualquier variable. **(F)**
16. Para saber si una variable tiene asignado algún valor distinto de **null** se puede utilizar la propiedad **HasValue**. Por ejemplo if (CodigoPostal.HasValue), siendo CodigoPostal de tipo “**int**”. **(V)**
17. Para asignar el valor **null** a una variable tipo string podemos asignarla directamente **variable = null;** o bien asignado la cadena vacía **variable = “”**; (F)
18. V1=null; V2=false; V3=(V1&V2)     ; V4=(V1|V2); V5=V3&V4; dará como resultado en V5 “**null**“. **(F)**
19. La conversión explícita y = (long) x; daría un error de compilación si la variable x fuera de tipo double. **(F)**
20. Console.WriteLine (**String.Format**(“{0:p}”,0.2)); dará como resultado **20,00%. (V)**
21. Console.WriteLine (**String.Format** (“Importe = {0:e}”,12.35)); dará como resultado **12,35€. (F)**
22. Console.WriteLine (**String.Format**(“{0:T}”,**DateTime.Now**)); dará como resultado la Hora del Sistema hasta los segundos. **(V)**
23. El método **Parse** se utiliza para convertir una cadena en algún tipo numérico, de igual manera se utilizará el método **ToString** para pasar un tipo numérico a cadena. **(V)**
24. El método **TryParse** es un método sobrecargado que devuelve un valor booleano true si la conversión se realiza correctamente y false en caso contrario. **(V)**
25. En caso de no incializar una varibale numérica el valor inicial será **cero**; si es una cadena el carácter vacío, a valor **null** si es del tipo Object, y a true si es del tipo boolean. **(F)**
26. Las inicializaciones por defecto de las variables no se produce cuando se declaran dentro de una función. **(V)**
27. El ámbito de las variables puede ser a nivel de bloque, a nivel de función o a nivel de clase, siendo la de mayor ámbito la que se declara a nivel de bloque. **(F)**
28. El nivel de acceso **public**no se pueden utilizar en el interior de las funciones. **(V)**
29. El nivel de acceso **protected** restringe el acceso a la variable únicamente al código de la clase donde está definida, así como a las de todas las clases que hereden de esta. **(V)**
30. El nivel de acceso **internal**no se puede utilizar en el interior de una función. **(V)**
31. El nivel de acceso **private** no se puede utilizar en el interior de una función. **(V)**
32. Sino indicamos nada a una variable; esta se considera como **public**. **(F)**
33. No se puede calcular el valor de una constante a partir de una variable. **(V)**
34. No se puede calcular el valor de una constante a partir de una constante en tiempo de ejecución, sí se puede en cambio en tiempo de compilación. **(V)**
35. Todas las enumeraciones comienzan por el índice 0. **(F)**

**RESPUESTAS**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5 F** | **6 F** | **7** | **8** | **9** | **10** |
| **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** |
| **21** | **22** | **23** | **24** | **25** | **26** | **27** | **28** | **29** | **30** |
| **31** | **32** | **33** | **34** | **35** |  |  |  |  |  |

**MSSQL**

**Instrucciones: Lee detenidamente y Realice las consultas solicitadas en un archivo de SQL enumerando las consultas de acuerdo a su operación.**

**Nota:** La creación de la BD se encuentra en el archivo adjunto (Creación de BD).

***Problema***

**Enunciado**: Sean las siguientes tablas de una base de datos.

**Empleado** (nDIEmp, nomEmp, sexEmp, fecNac, fecIncorporacion, salEmp, comis,

codSuc, jefeDI, cargo, nroDepto)

**Departamento** (codDepto, nombreDpto, ciudad, director)

**Consultas solicitadas.**

1. Obtener los datos completos de los empleados.
2. Obtener los datos completos de los departamentos.
3. Obtener los datos de los empleados con cargo 'Secretaria'.
4. Obtener el nombre y salario de los empleados.
5. Obtener los datos de los empleados vendedores, ordenado por nombre.
6. Listar el nombre de los departamentos.
7. Listar el nombre de los departamentos, ordenado por nombre.
8. Listar el nombre de los departamentos, ordenado por ciudad.
9. Listar el nombre de los departamentos, ordenado por ciudad, en orden inverso.
10. Obtener el nombre y cargo de todos los empleados, ordenado por salario.
11. Obtener el nombre y cargo de todos los empleados, ordenado por cargo y por salario.
12. Obtener el nombre y cargo de todos los empleados, en orden inverso por cargo.
13. Listar los salarios y comisiones de los empleados del departamento 2000.
14. Listar los salarios y comisiones de los empleados del departamento 2000, ordenado por comisión.
15. Listar todas las comisiones.
16. Listar las comisiones que sean diferentes, ordenada por valor.
17. Listar los diferentes salarios.
18. Obtener el valor total a pagar que resulta de sumar a los empleados del departamento 3000 una
19. bonificación de $500.000, en orden alfabético del empleado
20. Obtener la lista de los empleados que ganan una comisión superior a su sueldo.
21. Listar los empleados cuya comisión es menor o igual que el 30% de su sueldo.
22. Elabore un listado donde para cada fila, figure ‘Nombre’ y ‘Cargo’ antes del valor respectivo para
23. cada empleado
24. Hallar el salario y la comisión de aquellos empleados cuyo número de documento de identidad es
25. superior al '19.709.802'
26. Listar los empleados cuyo salario es menor o igual que el 40% de su comisión
27. Divida los empleados, generando un grupo cuyo nombre inicie por la letra J y termine en la letra Z.
28. Liste estos empleados y su cargo por orden alfabético.
29. Listar el salario, la comisión, el salario total (salario + comisión), documento de identidad del
30. empleado y nombre, de aquellos empleados que tienen comisión superior a $1.000.000, ordenar el
31. informe por el número del documento de identidad
32. Obtener un listado similar al anterior, pero de aquellos empleados que NO tienen comisión
33. Hallar el nombre de los empleados que tienen un salario superior a $1.000.000, y tienen como jefe al
34. empleado con documento de identidad '31.840.269'
35. Hallar el conjunto complementario del resultado del ejercicio anterior.
36. Hallar los empleados cuyo nombre no contiene la cadena “MA”
37. Obtener los nombres de los departamentos que no sean “Ventas” ni “Investigación” ni ‘MANTENIMIENTO’, ordenados por ciudad.

CREATE TABLE Empleado(

nDIEmp int IDENTITY(1,1) PRIMARY KEY,

nombEmp varchar(30) not null,

sexEmp varchar(5) not null,

fecNac date not null,

fecIncorporacion date not null,

salEmp int not null,

comis int not null,

codSuc int not null,

jefeDI int not null,

cargo varchar(50) not null,

nroDepto int not null

)

insert into Empleado (nombEmp, sexEmp, fecNac, fecIncorporacion, salEmp, comis, codSuc, jefeDI, cargo, nroDepto)

values('Maria', 'M', '1990-08-04', '2019-02-10', 2500, 1200, 3214, 10, 'Secretaria', 3)

insert into Empleado (nombEmp, sexEmp, fecNac, fecIncorporacion, salEmp, comis, codSuc, jefeDI, cargo, nroDepto)

values('Diana', 'M', '1994-11-14', '2018-04-01', 2500, 1200, 3214, 10, 'Secretaria', 2)

insert into Empleado (nombEmp, sexEmp, fecNac, fecIncorporacion, salEmp, comis, codSuc, jefeDI, cargo, nroDepto)

values('Jose', 'H', '1996-08-25', '2018-11-15', 2200, 800, 3214, 3, 'Vendedor', 5)

insert into Empleado (nombEmp, sexEmp, fecNac, fecIncorporacion, salEmp, comis, codSuc, jefeDI, cargo, nroDepto)

values('Pedro', 'H', '1996-01-02', '2019-02-19', 2600, 1000, 3214, 3, 'Vendedor', 6)

insert into Empleado (nombEmp, sexEmp, fecNac, fecIncorporacion, salEmp, comis, codSuc, jefeDI, cargo, nroDepto)

values('Alexa', 'M', '1997-09-10', '2019-01-11', 1900, 1200, 3214, 3, 'Vendedor', 6)

insert into Empleado (nombEmp, sexEmp, fecNac, fecIncorporacion, salEmp, comis, codSuc, jefeDI, cargo, nroDepto)

values('Felipe', 'H', '1995-06-12', '2019-10-25', 2100, 3900, 3214, 3, 'Vendedor', 6)

insert into Empleado (nombEmp, sexEmp, fecNac, fecIncorporacion, salEmp, comis, codSuc, jefeDI, cargo, nroDepto)

values('Andres', 'H', '1990-01-05', '2019-08-27', 3000, 1900, 3214, 3, 'Gerente Ventas', 2000)

insert into Empleado (nombEmp, sexEmp, fecNac, fecIncorporacion, salEmp, comis, codSuc, jefeDI, cargo, nroDepto)

values('Ricardo', 'H', '2000-12-28', '2019-03-14', 3000, 1900, 3214, 3, 'Gerente Ventas', 2000)

insert into Empleado (nombEmp, sexEmp, fecNac, fecIncorporacion, salEmp, comis, codSuc, jefeDI, cargo, nroDepto)

values('Lupita', 'M', '1997-07-12', '2019-02-19', 1450, 700, 3214, 5, 'Contador', 3000)

insert into Empleado (nombEmp, sexEmp, fecNac, fecIncorporacion, salEmp, comis, codSuc, jefeDI, cargo, nroDepto)

values('Brenda', 'M', '1999-08-03', '2018-12-08', 2900, 1300, 3214, 5, 'Contador', 3000)

/\*TABLA Departamento\*/

CREATE TABLE Departamento(

codDepto int not null,

nombreDpto varchar(50) not null,

ciudad varchar(30) not null,

director varchar(50) not null

)

insert into Departamento (codDepto, nombreDpto, ciudad, director)

values(6, 'Ventas', 'Ciudad de mexico',3214)

insert into Departamento (codDepto, nombreDpto, ciudad, director)

values(3, 'Ventas', 'Ciudad de mexico',3214)

insert into Departamento (codDepto, nombreDpto, ciudad, director)

values(5, 'Administracion', 'Queretaro',3214)