2da Evaluación de Programación II

Total de puntos 54/98

Temas a evaluar

1. Colecciones y Archivos.
2. Estructura, Enum, Propiedades y Serialización.
3. Introducción y Conexiones a Bases de Datos.
4. Lenguaje SQL (DDL y DML).
5. Delegados y Expresiones Lambda.
6. Programación Multi-hilo y Concurrencia.
7. ThereadPool, Task y Clase Parallel.

Correo electrónico *

niripil.mail@gmail.com

0 de 0 puntos

Introduzca la Contraseña: *

CUDI2

Legajo *

117126

Nombre y Apellido *

Mailen del Rosario Edith Niripil

Comienzo de la Evaluación 22 de 34 puntos

Proceso de conversión de un JSON / flujo de bytes (Stream) en un objeto. * 1/1

Seleccione cual es la definición correcta:

- (●) Deserialización
- ( ) Colecciones
- ( ) Serialización
- ( ) Hilos
- ( ) Propiedades

* 0/1

```
using System;
using System.Text;
using System.Collections;
using System.Collections.Generic;
using System.ComponentModel;

public class Test
{
   public static void Main(string[] args)
   {
      Dictionary<string, int> edades = new Dictionary<string, int>(); 1

      Console.WriteLine("Edades: ");

      edades.Add("Fernando", 45);
      edades.Add("Natalia", 15);
      edades["Pedro"] = 34;
      edades["Maria"] = 23;

      foreach (KeyValuePair<string, int> persona in edades)
      {

         Console.WriteLine("Nombre: {0} Edad {1}", persona.Key, persona.Value);
      }

   }
}
```

Para obtener el punto debe indicar correctamente, que colecciones se aplican en el programa:

- [x] Diccionario
- [ ] Lista ordenada
- [ ] Lista de arreglo
- [ ] Par clave-valor
- [ ] Diccionario ordenado

*0/1

```
using System;
using System.Runtime.CompilerServices;

namespace MyApp // Note: actual namespace depends on the project name.
{
   internal class Program
   {
      static void Main(string[] args)
      {



         Console.ReadLine();
      }
      static void EjecutarTarea()
      {
         for (int i = 0; i < 10; i++)
         {
            var miThread = Thread.CurrentThread.ManagedThreadId;
            Thread.Sleep(1000);
            Console.WriteLine("Esta vuelta de bucle corresponde al Thread: " +
miThread);
         }
      }



      {
         for (int i = 0; i < 10; i++)
         {
            var miThread = Thread.CurrentThread.ManagedThreadId;
            Thread.Sleep(1000);
            Console.WriteLine("Esto es otra tarea Tarea Correspondiente al hilo: "
+ miThread);
         }
      }
   }
}
```

En el siguiente código fuente, indique cuales son las respuestas correctas:

- [ ] static void EjecutarOtraTarea()
- [ ] Task tareas2 = tareas.ContinueWith(EjecutarOtraTarea);

- [x] Task tareas = Task.Run(() => EjecutarTarea());
- [ ] Task tareas = tareas.ContinueWith(EjecutarTarea);
- [ ] Thread tareas = Thread.Run(() => EjecutarTarea());
- [ ] static void EjecutarOtraTarea(Task obj)

Representa un nodo en una colección LinkedList. Esta clase no se puede heredar. *1/1

Seleccione cual es la instrucción correcta:

- (●) LinkedListNode
- ( ) ArrayList
- ( ) SortedList
- ( ) LinkedList
- ( ) List

Representa una colección de pares clave-valor clasificados ascendentemente por la clave. *1/1

Seleccione cual es la instrucción correcta:

- ( ) ArrayList
- ( ) List
- (●) SortedList
- ( ) LinkedList
- ( ) LinkedListNode

Seleccione las líneas que son necesarias para crear una propiedad. * 1/1

Tome en cuenta que para obtener el punto, se debe tener las tres líneas del formato escritas de forma correcta:

- [ ] { set { return myVar; } get { myVar = value; }}
- [x] { get { return myVar; } set { myVar = value; }}
- [ ] public int myVar;
- [x] private int myVar;
- [ ] private int MyProperty
- [x] public int MyProperty

Es una versión ordenada de Hashtable. En este tipo de colección, se puede acceder a los elementos por la clave o por el índice. *1/1

Seleccione cual es la instrucción correcta:

- ( ) Dictionary
- ( ) Stack
- ( ) ArrayList
- (•) SortedList
- ( ) Hashtable

1/1

```
using System;
using System.Text;
using System.Collections;
using System.Collections.Generic;
public class Test
{
   public static void Main(string[] args)
   {
      Queue<int> numeros = new Queue<int>();

      Console.WriteLine("¿Numeros: ");

      foreach (int numero in new int[] { 10, 8, 6, 7, 8, 9 })
      {

         numeros.Enqueue(numero);

      }

      Console.WriteLine("Recorriendo la Cola: ");

      foreach (int elemento in numeros)
       {

          Console.WriteLine(elemento);

       }
      Console.WriteLine("Eliminando la cola: ");
      numeros.Dequeue();
      foreach (int elemento in numeros)
      {

         Console.WriteLine(elemento);

      }

   }
}
```

Para obtener el punto debe indicar correctamente, todas las sentencias que se realizan en el programa:

- [x] Se crea una Cola

- [ ] Introduce un elemento al inicio
- [ ] Elimina un elemento al final
- [ ] Se crea una Pila
- [x] Elimina un elemento al inicio
- [x] Agrega un elemento al final

Representa una matriz unidimensional dinámica. * 0/1

Seleccione cual es la instrucción correcta:

- ( ) List
- ( ) LinkedList
- ( ) LinkedListNode
- ( ) SortedList
- (•) ArrayList

Es una lista del tipo FIFO: First In First Out - primero en entrar, primero en salir. *1/1

Seleccione cual es la instrucción correcta:

- ( ) Stack
- ( ) Hashtable
- (•) Queue
- ( ) SortedList
- ( ) ArrayList

Para los tipos Enum que **símbolo** se utiliza para trabajar con valores nulos (null). *1/1

?

Representa una lista doblemente enlazada. * 1/1

Seleccione cual es la instrucción correcta:

- ArrayList
- LinkedListNode
- List
- LinkedList
- SortedList

*

0/1

```
using System;
using System.IO;

namespace archivos
{
   internal class Program
   {
      static void Main(string[] args)
      {

         TextReader Leer_archivo;
         Leer_archivo = new StreamReader("archivo.doc");
         Console.WriteLine(Leer_archivo.ReadToEnd());
         Leer_archivo.Close();

      }
   }
}
```

Para obtener el punto, debe seleccionar todo lo que hace el código fuente correctamente:

- [ ] Escribe un mensaje dentro del archivo
- [ ] Crea un archivo
- [ ] Cierra el archivo
- [x] Lee lo escrito en el archivo
- [ ] Añade el mensaje dentro del archivo sin borralo

Representa una colección de pares clave-valor organizados según la clave. * 0/1

Seleccione cual es la instrucción correcta:

- ( ) LinkedListNode
- ( ) KeyValuePair
- (•) SortedDictionary
- ( ) LinkedList
- ( ) Dictionary

Es una lista del tipo LIFO: Last In First Out - último en entrar, primero en salir. * 1/1

Seleccione cual es la instrucción correcta:

- ( ) SortedList
- (•) Stack
- ( ) ArrayList
- ( ) Queue
- ( ) Hashtable

Es una forma de agrupar objetos, sus elementos no tienen por qué compartir el mismo tipo de datos (por ejemplo, los objetos de cualquier tipo se pueden agrupar en una sola colección del tipo Object), y su número de elementos puede aumentar y reducirse dinámicamente a medida que cambian las necesidades del programa. *1/1

Seleccione cual es el concepto correcto:

- ○ Propiedades
- ○ Serialización
- ○ Hilos
- ◉ Colecciones
- ○ Concurrencias

*0/1

```
using System;
using System.Threading;
using System.Threading.Tasks;  // Necesario para utilizar la clase Task.

namespace MyApp
{
   internal class Program
   {
      static void Main(string[] args)
      {

         tarea.Start();

         Task tarea2 = new Task(() =>
         {
            for (int j = 0; j < 100; j++)
            {

               Thread.Sleep(1000);
               Console.WriteLine("Tarea Correspondiente al hilo: " + miThread + "
Ejecutandose desde el Main");
            }
         });

         Console.ReadLine();
      }

      static void EjecutarTarea()
      {
         for (int i = 0; i < 100; i++)
         {
            var miThread = Thread.CurrentThread.ManagedThreadId;
            Thread.Sleep(1000);
            Console.WriteLine("Esta vuelta de bucle corresponde al Thread: " +
miThread);
         }
      }
   }
}
```

En el siguiente código fuente, indique cuales son las líneas que faltan:

- [ ] var miThread = Task.CurrentTask.ManagedTaskId;

- [x] Task tarea = new Task(EjecutarTarea);
- [ ] Thread tarea = new Thread(EjecutarTarea);
- [ ] var miThread = Thread.CurrentThread.ManagedThreadId;
- [ ] tarea2.Start();

Representa un elemento clave-valor * 1/1

Seleccione cual es la instrucción correcta:

- ( ) Stack
- ( ) Dictionary
- ( ) Ninguno
- (•) KeyValuePair
- ( ) SortedDictionary

Implementa una matriz cuyo tamaño aumenta o disminuye dinámicamente. * 1/1

Seleccione cual es la instrucción correcta:

- ( ) List
- (•) ArrayList
- ( ) LinkedListNode
- ( ) SortedList
- ( ) LinkedList

using System; * 0/1
using System.Threading;
using System.Threading.Tasks;

```
namespace Hilo
{
   internal class Program
   {
      static void Main(string[] args)
      {

         var hilo1 = new Thread(() =>
         {
            for (int i = 0; i < 5; i++)
            {
               Console.WriteLine("Hilo 1");
               Thread.Sleep(1000);
            }

         });
         hilo1.Start();


         Thread hilo2 = new Thread(segundo);
         hilo2.Start();

         Thread hilo3 = new Thread(tercero);
         hilo3.Start();
      }
      static void segundo()
      {
         for (int i = 0; i < 5; i++)
         {
            Console.WriteLine("Hilo 2");
            Thread.Sleep(1000);
         }
      }
      static void tercero()
      {
         for (int i = 0; i < 5; i++)
         {
            Console.WriteLine("Hilo 3");
```

```
            Thread.Sleep(1000);
        }
    }
  }
}
```

En el siguiente código fuente, indique cuales son las líneas que faltan:

- [ ] var tareaTerminada = new ThreadCompletionSource<bool>();
- [ ] var tareaTerminada = new TaskCompletionSource<bool>();
- [ ] var resultado = tareaTerminada.Task.Result;
- [ ] var resultado = tareaTerminada.Thread.Result;
- [ ] tareaTerminada.TrySetFinally(true);
- [x] tareaTerminada.TrySetResult(true);

Para los tipos Enum que instrucción o palabra reservada por C# se utiliza para que se convierta a cadena de caracteres: *1/1

Escriba la sentencia con todos sus detalles:

ToString()

using System; *0/1

```
namespace MyApp // Note: actual namespace depends on the project name.
{
   internal class Program
   {
      static void Main(string[] args)
      {
         for (int i = 0; i < 500; i++)
         {

         }
         Console.ReadLine();
      }



      {
         int nTarea = (int)o;
         Console.WriteLine($"El correo N°:
{Thread.CurrentThread.ManagedThreadId} ha comenzado su tarea " + nTarea);
         Thread.Sleep(1000);
         Console.WriteLine($"El correo N°:
{Thread.CurrentThread.ManagedThreadId} ha terminado su tarea " + nTarea);
      }
   }
}
```

En el siguiente código fuente, indique cuales son las líneas que faltan:

- [ ] ThreadPool.StarkUserWorkItem(EjecutarTarea);
- [ ] static void EjecutarTarea()
- [ ] int nTarea = o;
- [x] ThreadPool.QueueUserWorkItem(EjecutarTarea, i);
- [ ] int nTarea = (int)o;
- [ ] static void EjecutarTarea(Object o)

*0/1

```
using System;
using System.Text;
using System.Collections;
using System.Collections.Generic;
public class Test
{
   public static void Main(string[] args)
   {
      List<int> numeros = new List<int>();
      Console.WriteLine("¿Para Salir precionar 0: ");

      int elem = 1;

      while (elem !=0 )
      {

        elem = int.Parse(Console.ReadLine());
         numeros.Add(elem);

      }

      numeros.RemoveAt(numeros.Count - 1);
      Console.WriteLine("Los elementos que se introducen son los siguientes:
");

      foreach (int elemento in numeros)
      {

         Console.WriteLine(elemento);

      }
   }
}
```

Para obtener el punto debe indicar correctamente, todas las sentencias que se realizan en el programa:

- [ ] Agrega un nodo al inicio de la lista
- [x] Agrega un elemento u objeto a final de la lista
- [x] Remueve un nodo al final de la lista
- [ ] Remueve el elemento a la expecificación del indice de la lista

- [x] Declara una lista de colección

*1/1

```
using System;
using System.Text;
using System.Collections;
using System.Collections.Generic;
public class Test
{
   public static void Main(string[] args)
   {
      LinkedList<int> numeros = new LinkedList<int>();

      Console.WriteLine("¿Numeros ");

      foreach (int numero in new int[] {10,8,6,7,8,9})
      {

          numeros.AddFirst(numero);

      }

      numeros.Remove(8);

      LinkedListNode<int> nodoImportante = new LinkedListNode<int>(15);
      numeros.AddLast(nodoImportante);
      Console.WriteLine("Los elementos que se introducen son los siguientes:
");

foreach (int elemento in numeros)
{

    Console.WriteLine(elemento);

}

for (LinkedListNode<int> nodo=numeros.First; nodo!=null; nodo=nodo.Next)
{
    int numero = nodo.Value;
    Console.WriteLine(numero);

}
}
}
```

Para obtener el punto debe indicar correctamente, todas las sentencias que se realizan en el programa:

- [x] Crea un lista doblemente enlazada
- [x] Trabaja con un nodo en una colección de la lista doblemente enlazada
- [x] Agrega un nodo al final de la lista doblemente enlazada
- [ ] Remueve la ultima ocurrecia del valor de la lista doblemente enlazada
- [x] Agrega un nodo al inicio de la lista doblemente enlazada
- [x] Remueve la primera ocurrecia del valor de la lista doblemente enlazada

Indique que se puede hacer en los siguientes segmentos de memoria: *

Tome en cuenta que para obtener los 2 puntos, no se debe fallar en ninguna de las diferencias:

	Objetos	Los llamados a las funciones	Variables superglobales	Los parámetros de las funciones llamadas	Las variables locales	Otra información de alto uso de mémoria	Puntua
Heap	☑	☐	☑	☐	☐	☑	1/1
Stack	☐	☑	☐	☑	☑	☐	1/1

La clase base abstracta de todos los tipos de enumeración. Proporciona una serie de métodos para obtener información sobre un tipo de enumeración y sus valores. *1/1

Seleccione cual es la clase correcta:

- ○ Hilos
- ○ Class
- ◉ Enum
- ○ Struct
- ○ Delegados

Colección de elementos de tipo clave-valor organizados en función del código hash de la clave. Sólo se puede tener acceso a los elementos de la colección a través de la clave del elemento. *1/1

Seleccione cual es la instrucción correcta:

- ○ Queue
- ○ Dictionary
- ○ SortedList
- ○ ArrayList
- ◉ Hashtable

0/1

```
using System;
using System.IO;

namespace archivos
{
   internal class Program
   {
      static void Main(string[] args)
      {

         StreamWriter archivo = File.AppendText("archivo.doc");
         string mensaje;
         mensaje= Console.ReadLine();
         archivo.WriteLine(mensaje);
         archivo.Close();
         Console.ReadKey();

      }
   }
}
```

Para obtener el punto, debe seleccionar todo lo que hace el código fuente correctamente:

- [x] Crea un archivo
- [x] Cierra el archivo
- [ ] Lee lo escrito en el archivo
- [x] Escribe un mensaje dentro del archivo
- [x] Añade el mensaje dentro del archivo sin borralo

* 0/1

```
using System;
using System.Text;
using System.Collections;
using System.Collections.Generic;
public class Test
{
   public static void Main(string[] args)
   {
      Stack<int> numeros = new Stack<int>();

      Console.WriteLine("¿Numeros: ");

      foreach (int numero in new int[] { 10, 8, 6, 7, 8, 9 })
      {

         numeros.Push(numero);

      }

      Console.WriteLine("Recorriendo la Pila: ");

      foreach (int elemento in numeros)
      {

         Console.WriteLine(elemento);

      }
      Console.WriteLine("Eliminando la Pila: ");
      numeros.Pop();
      foreach (int elemento in numeros)
      {

         Console.WriteLine(elemento);

      }

   }
}
```

Para obtener el punto debe indicar correctamente, todas las sentencias que se realizan en el programa:

- [x] Se crea una Pila

- [x] Agrega un elemento al final
- [ ] Se crea una Cola
- [ ] Introduce un elemento al inicio
- [x] Elimina un elemento al final
- [ ] Elimina un elemento al inicio

Es un diccionario ordenado con operaciones de inserción y eliminación, lo que hace de él una alternativa útil a SortedList. *1/1

Seleccione cual es la instrucción correcta:

- ( ) KeyValuePair
- (•) SortedDictionary
- ( ) Stack.
- ( ) Dictionary
- ( ) Hashtable

using System; *1/1

```
namespace Estructura
{
   class Program
   {
      static void Main(string[] args)
      {
         Trabajador Trabajador1 = new Trabajador(120000, 25000);
         Trabajador1.modificaSalario(Trabajador1, 10000);
         Console.WriteLine(Trabajador1);
      }
   }

   public struct Trabajador
   {
      public double salarioBase, comision;

      public Trabajador (int salarioBase, int comision)
      {
         this. salarioBase = salarioBase;
         this.comision = comision;
      }

      public override string ToString()
      {
         return string.Format("Salario y comisión del Trabajador {0}, {1}, ",
this.salarioBase, this.comision);
      }

      public void modificaSalario(Trabajador traba, double aumentar)
      {
         traba.salarioBase += aumentar;
         traba.comision += aumentar*10;
      }
   }
}
```

Indique cual es la respuesta, si se cambia struct por class en el programa:

- ○ Ninguna de las anteriores
- ◉ Salario y comisión del Trabajador 130000, 125000

- ○ Salario y comisión del Trabajador 120000, 25000
- ○ Salario y comisión del Trabajador 130000, 35000
- ○ Salario y comisión del Trabajador 220000, 125000

*

1/1

```
using System;
using System.IO;

namespace archivos
{
   internal class Program
   {
      static void Main(string[] args)
      {
         TextWriter archivo;
         archivo = new StreamWriter("archivo.doc");
         string mensaje;
         mensaje= Console.ReadLine();
         archivo.WriteLine(mensaje);
         archivo.Close();
         Console.Clear();
         Console.WriteLine("Se ha guardado el archivo ...");
         Console.ReadKey();
      }
   }
}
```

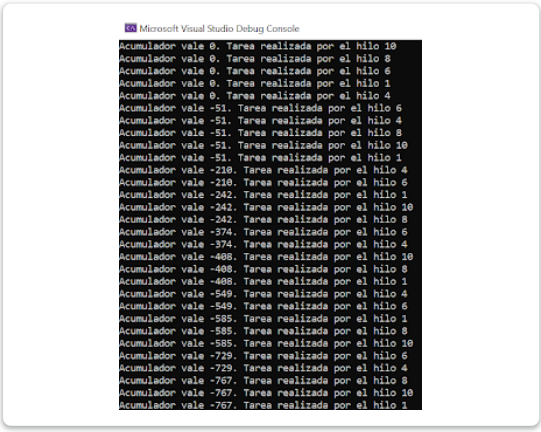
Para obtener el punto, debe seleccionar todo lo que hace el código fuente correctamente:

- ☑ Cierra el archivo
- ☐ Añade el mensaje dentro del archivo sin borralo
- ☑ Crea un archivo
- ☑ Escribe un mensaje dentro del archivo
- ☐ Lee lo escrito en el archivo

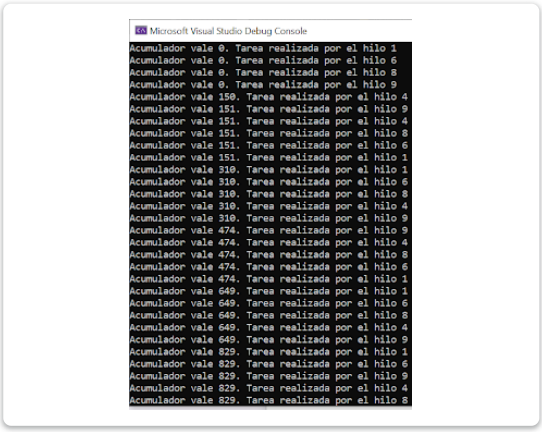
* 0/1

```
using System;
using System.Runtime.CompilerServices;
using static System.Runtime.InteropServices.JavaScript.JSType;

namespace MyApp // Note: actual namespace depends on the project name.
{
   internal class Program
   {
      private static int acumulador = 0;
      static void Main(string[] args)
      {
         Parallel.For(0, 100, dato =>
         {
            Console.WriteLine($"Acumulador vale {acumulador}. Tarea " +
            $"realizada por el hilo {Thread.CurrentThread.ManagedThreadId}");
            if (acumulador == 0)
            {
               acumulador += dato;
               Thread.Sleep(100);
            }
            else
            {
               acumulador += dato;
               Thread.Sleep(100);
            }
         });
      }
   }
}
```

En el siguiente código fuente, indique cual es la respuesta correcta:

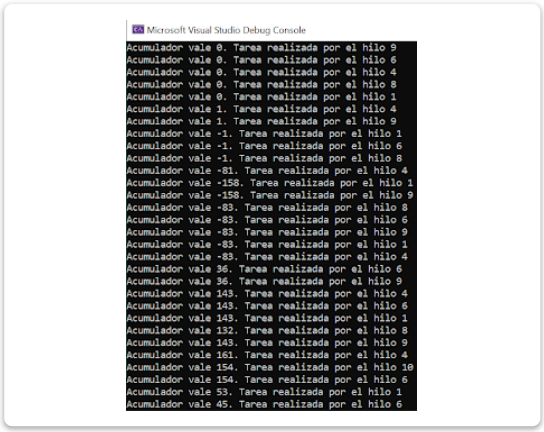


Opción 2



Opción 1

Ninguna de las anteriores



Opción 3

2da Evaluación de Programación II 32 de 64 puntos

Evaluación individual virtual para obtener la puntuación cualitativa de los conocimientos del estudiante adquiridos en clase.

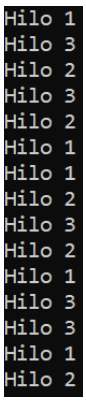
*

1/1

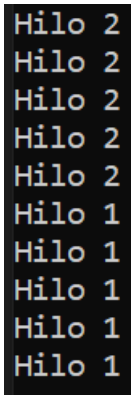
```
using System;

namespace Hilo
{
   internal class Program
   {
      static void Main(string[] args)
      {
         Thread s = new Thread(segundo);
         s.Start();


         Thread t = new Thread(tercero);
         t.Start();

         Console.WriteLine("Hilo 1");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 1");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 1");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 1");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 1");
      }

      static void segundo()
      {
         Console.WriteLine("Hilo 2");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 2");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 2");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 2");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 2");

      }
      static void tercero()
      {
         Console.WriteLine("Hilo 3");
```

```
            Thread.Sleep(1000);
            Console.WriteLine("Hilo 3");
            Thread.Sleep(1000);
            Console.WriteLine("Hilo 3");
            Thread.Sleep(1000);
            Console.WriteLine("Hilo 3");
            Thread.Sleep(1000);
            Console.WriteLine("Hilo 3");

        }
    }
}
```

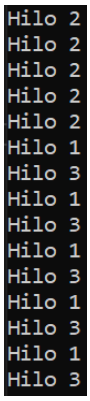
En el siguiente código fuente, indique cual es la respuesta correcta:



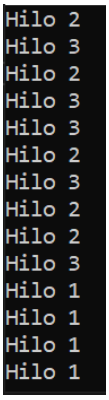
Opción 2



Opción 4



Opción 3



Opción 1

----------Class---------- * 1/1

//Declaración de l a función tabla()

static void tabla(){

for (i n t i = 1; i <= 10; i++) {Console.WriteLine("1 x {0} = {1}", i, 1 * i) ; }

}

----------Main----------

//Bucle que ejecuta 3 veces la función

tabla() f o r (i n t i = 1; i <= 3; i++) {tabla();}

En el siguiente código fuente indique que lineas se debe cambiar para que muestre la tabla de multiplicar del 1 al 10:

- [x] for (int i = 1; i <= 10; i++) { Console.WriteLine("{0} x {1} = {2}", t, i, t * i); } //en Main
- [ ] for (int i = 1; i <= 10; i++) { tabla(); }//en Class()
- [ ] for (int i = 1; i < 10; i++) { Console.WriteLine("{0} x {1} = {2}", i, t, t * i); }//en Main
- [x] for (int i = 1; i <= 10; i++) { tabla(i); }//en Class()
- [ ] for (int i = 1; i < 10; i++) { tabla(i); }//en Class()

Un conjunto de condiciones que deben cumplir los datos para reflejar la realidad deseada. *1/1

Seleccione cual es la definición correcta:

- ( ) Base de datos
- (•) Las restricciones de integridad
- ( ) Bases de datos relacionales
- ( ) Las estructuras de datos de la base
- ( ) Sistema Gestor de Base de Datos

Agregar un nuevo atributo dirección con un varchar(100) en la tabla estudiantes *0/1

Debe escribir la sentencia correcta en SQL:

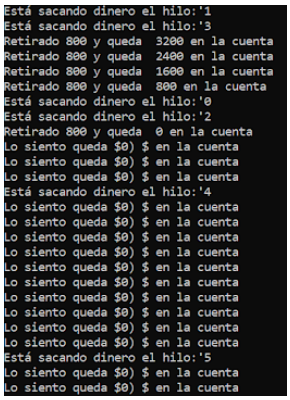
ALTER TABLE estudiantes ADD direccion VARCHAR(100);

*1/1

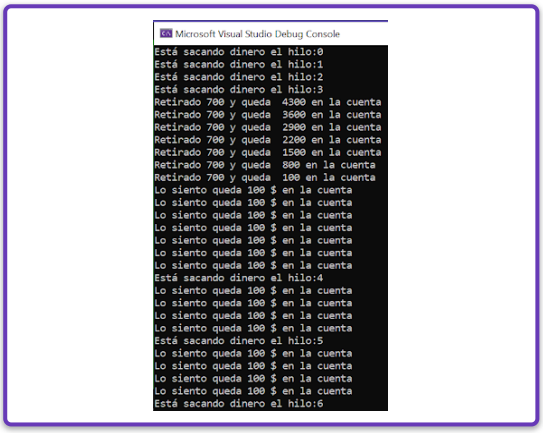
```
namespace Hilo
{
   internal class Program
   {
      static void Main(string[] args)
      {
         CuentaBancaria CuentaFamilia = new CuentaBancaria(5000);
         Thread[] hilosPersonas = new Thread[15];

         for (int i = 0; i < 15;i++)
         {
            Thread t = new Thread(CuentaFamilia.VamosRetirarEfectivo);
            t.Name = i.ToString();
            hilosPersonas[i] = t;
         }
         for (int i = 0;i < 15;i++) hilosPersonas[i].Start();
      }

      class CuentaBancaria
      {
         private Object bloqueaSaldoPositivo = new Object();
         double Saldo { set; get; }
         public CuentaBancaria(double Saldo)
         {
            this.Saldo = Saldo;
         }
      public double RetirarEfectivo(double Cantidad)
      {
         if ((Saldo - Cantidad) < 0)
         {
            Console.WriteLine($"Lo siento queda ${Saldo}) $ en la cuenta");
            return Saldo;
         }
         lock(bloqueaSaldoPositivo)
         {
            if(Saldo >= Cantidad)
            {
               Console.WriteLine("Retirado {0} y queda  {1} en la cuenta",
Cantidad, (Saldo - Cantidad), Thread.CurrentThread.Name);
                  Saldo -= Cantidad;
            }
            return Saldo;
```

```
            }
        }
        public void VamosRetirarEfectivo()
        {
            Console.WriteLine("Está sacando dinero el hilo:´{0}",
Thread.CurrentThread.Name);
            for(int i= 0;i < 4; i++) RetirarEfectivo(700);
        }
    }
  }
}
```

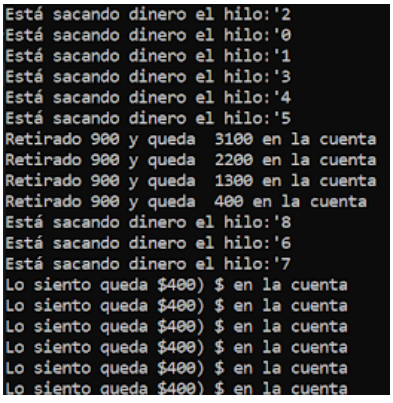
En el siguiente código fuente, indique cual es la respuesta correcta:



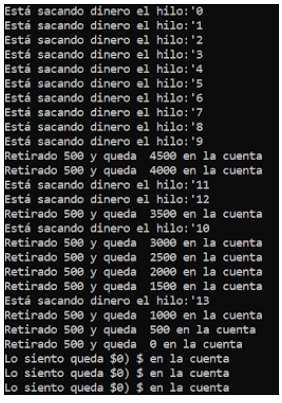
Opción 2



Opción 3



Opción 1



Opción 4

- ◯ Ninguna Opción

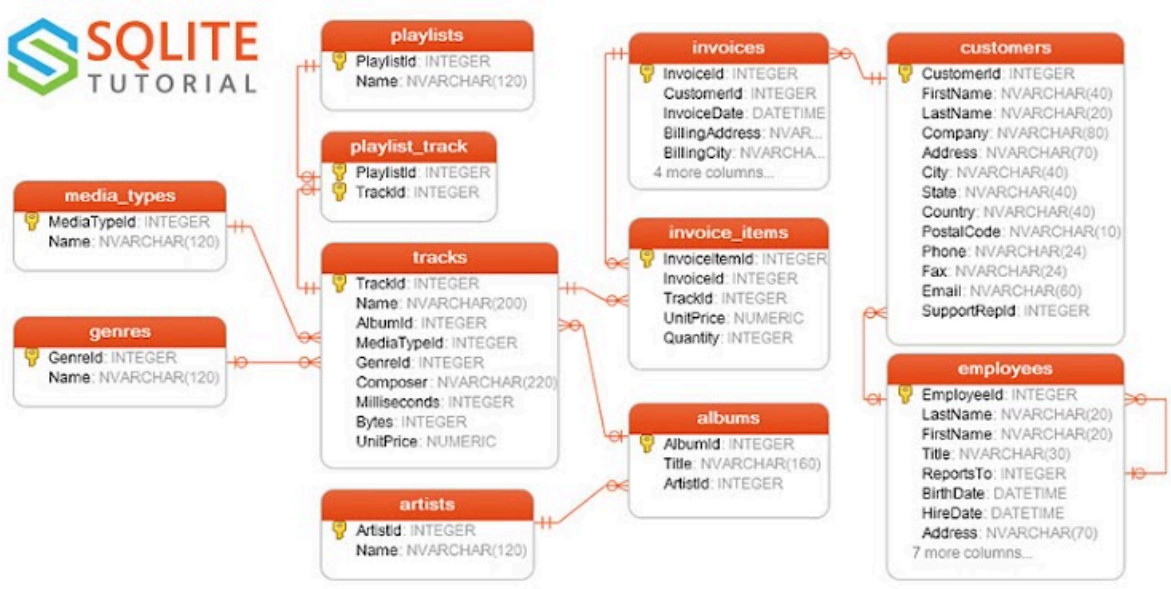
Crear una tablas de personas con DNI de 11 numeros enteros, nombre y apellido de 30 caracteres alfanúmericos y fecha de nacimiento: *0/1

Para obtener el punto, debe seleccionar la lineas de SQL que son correctas:

- ☑);
- ☑ fecha_nac date,
- ☐ dni int(11);
- ☑ dni int(11),
- ☑ apellido varchar(30),
- ☑ nombre varchar(30),
- ☐)
- ☐ fecha_nac date
- ☐ apellido string(30),
- ☐ nombre varchar(30).
- ☑ CREATE TABLE alumnos (

Traer todas las columnas de la tabla Tracks y ordenar por nombre. * 1/1

Coloque el código SQL correcto en la siguiente base de datos que aparece en la figura:



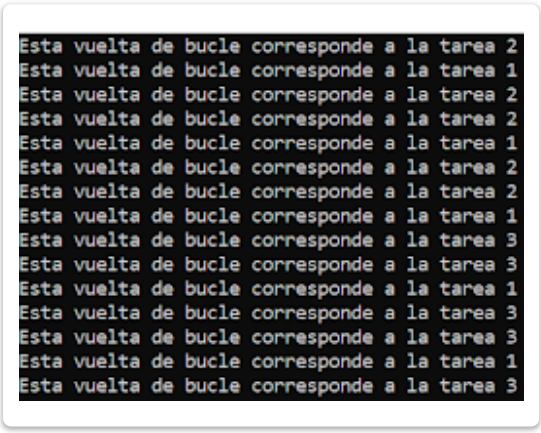
- (●) SELECT * FROM tracks ORDER BY Name;
- ( ) SELECT * FROM tracks ORDER BY Name, AlbumId DESC;
- ( ) SELECT * FROM tracks ORDER BY Name, AlbumId;
- ( ) SELECT * FROM tracks ORDER BY Name DESC;
- ( ) SELECT * FROM tracks ORDER BY AlbumId DESC;

* 0/1

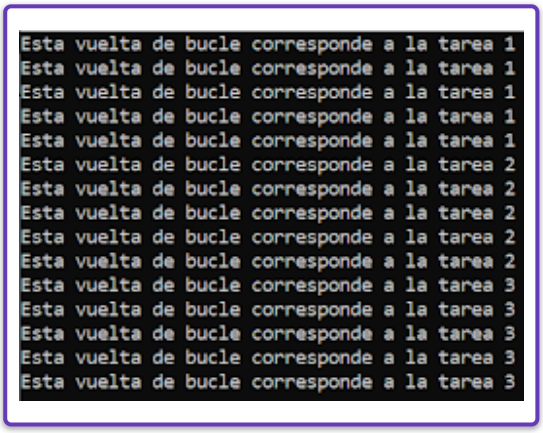
```
using System;
using System.Runtime.CompilerServices;

namespace MyApp // Note: actual namespace depends on the project name.
{
    internal class Program
    {
        static void Main(string[] args)
        {
            RealizarTodasTareas();
            Console.ReadLine();
        }

        static void RealizarTodasTareas()
        {
            var tarea1 = Task.Run(() => { EjecutarTarea(); });

            var tarea2 = Task.Run(() => { EjecutarTarea2(); });
            Task.WaitAny(tarea1, tarea2);
            var tarea3 = Task.Run(() => { EjecutarTarea3(); });



        }

        static void EjecutarTarea()
        {
            for (int i = 0; i < 5; i++)
            {
                var miThread = Thread.CurrentThread.ManagedThreadId;
                Thread.Sleep(1000);
                Console.WriteLine("Esta vuelta de bucle corresponde a la tarea 1");
            }
        }
        static void EjecutarTarea2()
        {
            for (int i = 0; i < 5; i++)
            {
                var miThread = Thread.CurrentThread.ManagedThreadId;
                Thread.Sleep(500);
                Console.WriteLine("Esta vuelta de bucle corresponde a la tarea 2");
            }
        }
```

```
        static void EjecutarTarea3()
        {
            for (int i = 0; i < 5; i++)
            {
                var miThread = Thread.CurrentThread.ManagedThreadId;
                Thread.Sleep(500);
                Console.WriteLine("Esta vuelta de bucle corresponde a la tarea 3");
            }
        }

    }
}
```

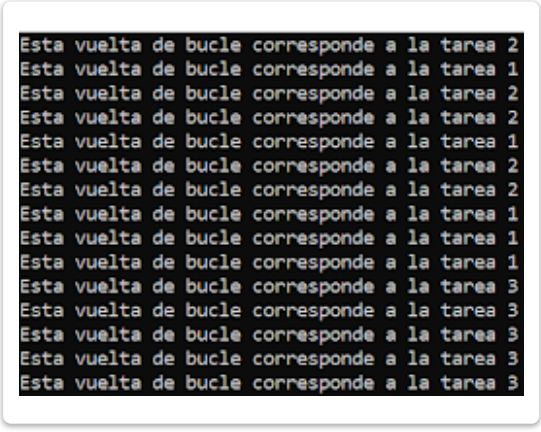
En el siguiente código fuente, indique cual es la respuesta correcta:



Opción 1



Opción 2



Opción 3

Ninguna

Permiten gestionar el acceso a los datos, así como también su almacenamiento, modifificación, eliminación, consulta y el múltiple acceso a ellos, que podrá ser desde distintas aplicaciones y usuarios. Inclusive podrán gestionar permisos para que una parte de los datos estén disponibles para ciertos usuarios y no para otros. Todo esto es resuelto por un motor de base de datos, generando una independencia entre la base de datos y la aplicación que la consulte.. *0/1

Seleccione cual es la definición correcta:

Sistema Gestor de Base de Datos

Para usar funciones que cosas hay que hacer: * 1/1

Para obtener el punto debe seleccionar las respuestas correctas:

- [x] Ejecutar la función.
- [ ] Reducir la función.
- [ ] Encriptar la función.
- [x] Declarar la función.
- [ ] Retornar la función.

De los siguientes SGBD o DBMS, cuales son Relacional y cuales son No Relacional: *

Tome en cuenta que, para obtener los 2 puntos no se debe fallar en ninguna de las diferencias:

	MySQL	MongoDB	MariaDB	Redis	Cassandra	PostgreSQL	Puntuaci
NO RELACIONAL:	☐	☑	☑	☐	☑	☐	0/1
RELACIONAL	☑	☐	☐	☑	☐	☑	0/1

----------Class---------- * 0/1

```
static int cuadrado(int x){

return x*x;

}

----------Main()----------

Console.WriteLine(cuadrado(2));

----------
```

Seleccione la función lambda que sustituye a la función tradicional en C#:

- ○ static cuadrado (int x) => x*x;
- ◉ int cuadrado (int x) => x*x;
- ○ static int cuadrado (int x) => x*x;
- ○ static int (x) => x*x;
- ○ static int cuadrado (x) => x*x;

De los siguientes SGBD o DBMS, cuales son Relacional y cuales son No Relacional: *

Tome en cuenta que, para obtener los 2 puntos no se debe fallar en ninguna de las diferencias:

	Microsoft Access	SQL Server	DynamoDB	Oracle	CouchDB	SQLite	Puntuación
RELACIONAL	☑	☑	☐	☑	☐	☑	1/1
NO RELACIONAL:	☐	☐	☑	☐	☑	☐	1/1

Es un tipo que representa una firma de método particular. Una instancia de este tipo se refiere a un método particular con una firma coincidente. *1/1

Seleccione cual es la definición correcta:

Delegado

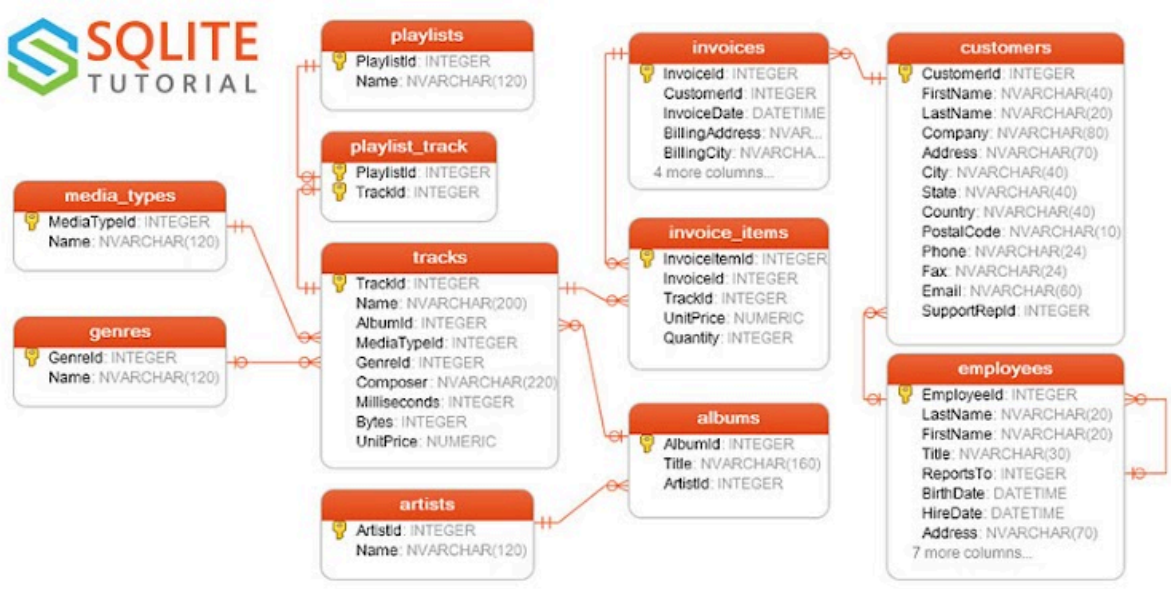
El tipo de los datos que hay en la base y la forma en que se relacionan. * 1/1

Seleccione cual es la definición correcta:

Las estructuras de datos de la base

Mostrar el nombre de las canciones que comiencen con 'All' * 1/1

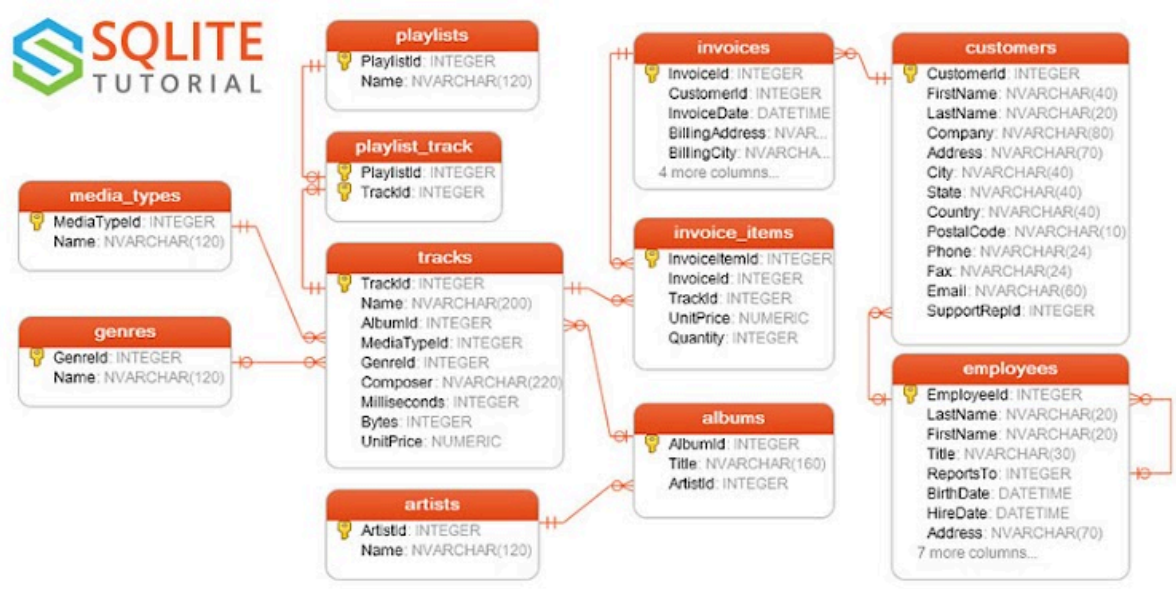
Coloque el código SQL correcto en la siguiente base de datos que aparece en la figura:



- (●) SELECT name FROM tracks WHERE name LIKE 'ALL%';
- ( ) SELECT name FROM tracks WHERE name LIKE '%ALL%';
- ( ) SELECT name FROM tracks WHERE name LIKE '%ALL';
- ( ) SELECT name FROM tracks WHERE name LIKE '%ALL____%';
- ( ) Ninguno

Encontrar los álbumes con 12 o más canciones: * 0/1

Coloque el código SQL correcto en la siguiente base de datos que aparece en la figura:



- ○ SELECT AlbumId, COUNT (TrackId) FROM tracks GROUP BY AlbumId HAVING count (TrackId) >=12;
- ◉ SELECT *, COUNT (TrackId) FROM tracks GROUP BY AlbumId HAVING count (TrackId) >=12;
- ○ SELECT AlbumId, COUNT (TrackId) FROM tracks GROUP BY AlbumId HAVING count (TrackId) LIMIT 12;
- ○ SELECT AlbumId, MAX(TrackId) FROM tracks GROUP BY AlbumId HAVING count (TrackId) >=12;
- ○ NINGUNA

using System; * 1/1

```
namespace Mensaje
{
   internal class Program
   {
      static void Main(string[] args)
      {
         Mensaje Envia = new Mensaje(Bienvenido.Saludo);
         Envia("Llegando");
         Envia = new Mensaje(Despedida.Chao);
         Envia("Saliendo");
      }

      delegate void Mensaje(string mensaje);
      class Bienvenido
      {
         public static void Saludo(string mensaje)
         {
         Console.WriteLine("Mensaje de Bienvenida: {0}", mensaje);
         }
      }
      class Despedida
      {
         public static void Chao(string mensaje)
         {
            Console.WriteLine("Mensaje de Depedida: {0}", mensaje);
         }
      }
   }
}
```

En el siguiente código fuente indique que tipos de sentencias se utiliza:

- [ ] Predicado
- [x] Delegado
- [ ] Listas
- [ ] Hilos
- [x] Funciones

Razones para declarar funciones: * 1/1

Para obtener el punto debe seleccionar las razones correctas:

- [x] Una función me permite modularizar.
- [ ] Para hacer más rápido el programa.
- [ ] Sirve para encriptar el código fuente de tal forma que no pueda ser hackeado.
- [x] Por claridad. Para que programa no tenga muchas líneas y sea difícil seguirlo.
- [x] En el momento de trabajar en equipo dividimos el trabajo en partes.

----------Class---------- * 1/1

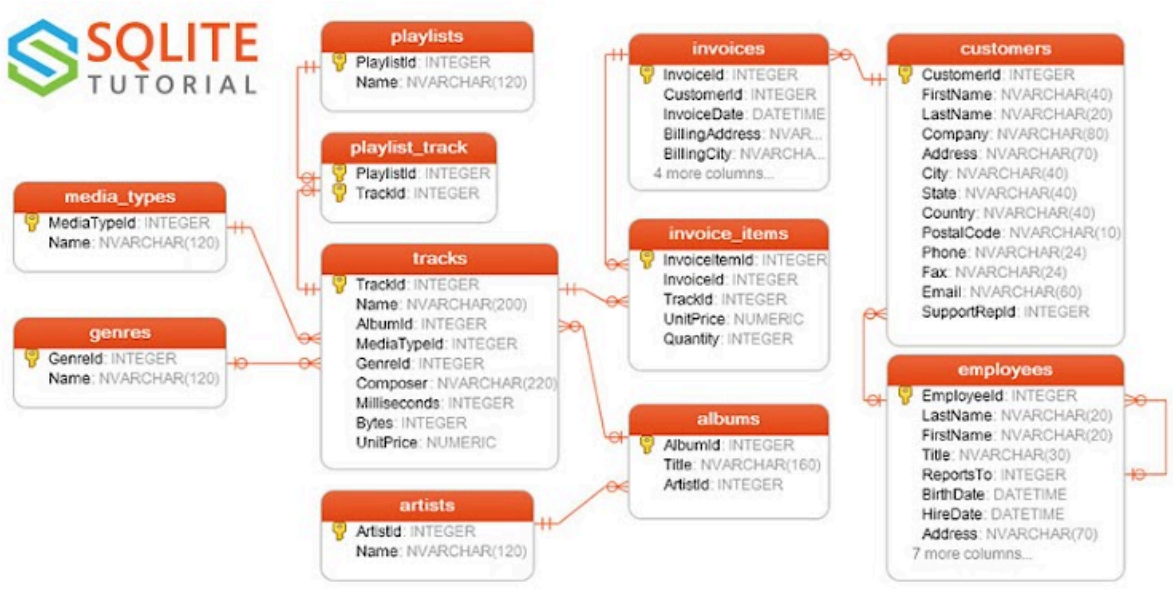
```
static float dividir(float num1, float num2)
    {

        return num1 / num2;

    }

----------Main()----------

Console.WriteLine(dividir(2, 4));

------------------------------------------------------------
```

Seleccione la función lambda que sustituye a la función tradicional en C#:

- ( ) float dividir(float num1, float num2) => num1 \ num2;
- (•) static float dividir(float num1, float num2) => num1 / num2;
- ( ) dividir (num1, num2) => num1\num2;
- ( ) static float dividir(float num1, float num2) => num1 \ num2;
- ( ) static dividir (num1, num2) => num1/num2;

Encontrar todas las facturas de Brasilia, Edmonton, y Vancouver, luego ordenar descendentemente por invoice ID. *1/1

Coloque el código SQL correcto en la siguiente base de datos que aparece en la figura:



- (●) SELECT * FROM invoices WHERE BillingCity IN ('Brasilia', 'Edmonton', 'Vancouver') ORDER BY InvoiceId DESC;
- ( ) SELECT * FROM invoices WHERE BillingCity IN ('Brasilia', 'Edmonton', 'Vancouver') ORDER BY InvoiceId;
- ( ) SELECT invoiceid FROM invoices WHERE BillingCity IN ('Brasilia', 'Edmonton', 'Vancouver') ORDER BY InvoiceId DESC;
- ( ) SELECT invoiceid FROM invoices WHERE BillingCity IN ('Brasilia', 'Edmonton', 'Vancouver') ORDER BY InvoiceId;
- ( ) NINGUNA

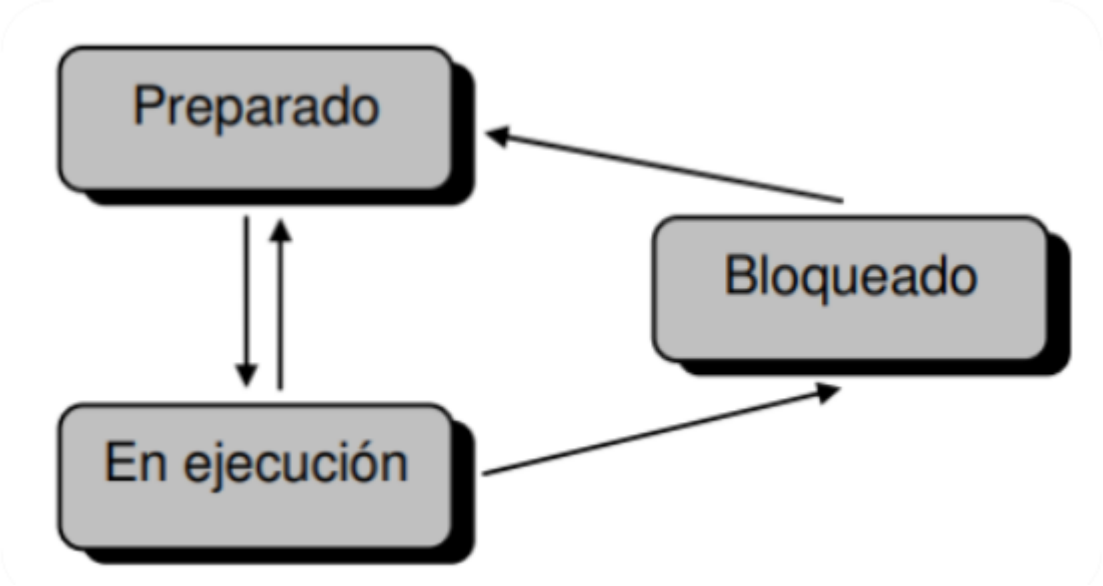
Es un sistema de software de propósito general que facilita la definición, construcción, manipulación y compartición de bases de datos entre usuarios y aplicaciones. *1/1

Seleccione cual es la definición correcta:

Sistema Gestor de Base de Datos ▾

En la siguiente figura puede observar que existen varias regiones de memoria: *

Seleccione las definiciones correctas:



	El hilo ha sido creado pero aún no ha sido activado. Cuando se active pasará al estado preparado.	El hilo está activo y está a la espera de que le sea asignada la UCP.	El hilo está activo y le ha sido asignada la UCP (sólo los hilos activos, preparados, pueden ser realizados).	El hilo espera que otro elimine el inmovilización, puede estar dormido o esperando.	El hilo ha finalizado pero todavía no ha sido recogido por su padre. Los hilos muertos no pueden alcanzar ningún otro estado.	Puntuación
Nuevo	☑	☐	☐	☐	☐	1/1
En ejecución	☐	☐	☑	☐	☐	1/1
Preparado	☐	☑	☐	☐	☐	1/1
Muerto	☐	☐	☐	☐	☑	1/1
Bloqueado	☐	☐	☐	☑	☐	1/1

*0/1

```
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace MOVIMIENTO
{
   class Program
   {
      static void Main(string[] args)
      {
         string NOM;
         byte K;
         Console.Write("COLOQUE EL NÚMERO DE HILO:"); NOM =
Console.ReadLine();
         for (K = 1; K <= 70; K++)
         {
            Console.SetCursorPosition(K, 12);
            Console.Write(" " + NOM);
            //REALIZAMOS UNA PAUSA;
            System.Threading.Thread.Sleep(100);
         }
         Console.WriteLine();
         Console.Write("Pulse una Tecla:"); Console.ReadLine();
      }
   }
}
```

En el siguiente código fuente indique que hace el programa:

- ○ Mueve un carácter de izquierda a derecha, todavía no ha finalizado cuando sale el otro.
- ○ Mueve un carácter de arriba hacia abajo, apenas termina uno comienza a desplazarse el otro.
- ○ Mueve una palabra completa de izquierda a derecha
- ◉ Mueve un carácter de izquierda a derecha, apenas termina uno comienza a desplazarse el otro.
- ○ Mueve un carácter de arriba hacia abajo, todavía no ha finalizado cuando sale el otro.

Es la unidad de ejecución de un proceso y está asociado con una secuencia de instrucciones, un conjunto de registros y una pila. *0/1

Seleccione cual es la definición correcta:

Concurrencia

----------Class---------- * 0/1

```
static float dividir(float num1, float num2)
      {
         float producto = num1 / num2;
         return producto;
      }
```

----------Main()----------

```
Console.WriteLine(dividir(2, 3));
```

Seleccione las opciones necesarias para sustituir esta función tradicional en C# :

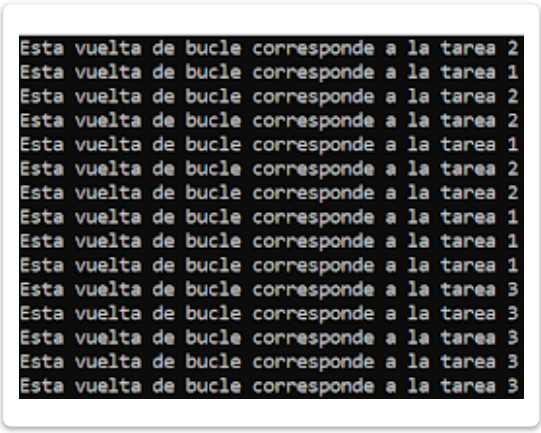
- [ ] static dividir (num1, num2) => num1/num2; // en Class
- [ ] static float dividir(float num1, float num2) => num1 \ num2; // en Class
- [x] float dividir(float num1, float num2) => num1 / num2; //en Class
- [ ] Calculadoras dividir = new Calculadoras((num1, num2)=> {int Total = num1 / num2;return Total;});//en Main
- [ ] public delegate int Calculadoras(int num1, int num2); //en Class

* 0/1

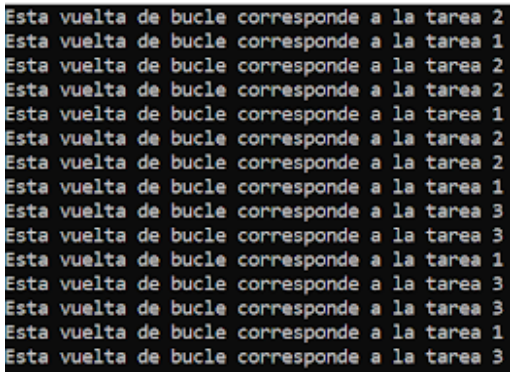
```
using System;
using System.Runtime.CompilerServices;

namespace MyApp // Note: actual namespace depends on the project name.
{
   internal class Program
   {
      static void Main(string[] args)
      {
         RealizarTodasTareas();
         Console.ReadLine();
      }

      static void RealizarTodasTareas()
      {
         var tarea1 = Task.Run(() => { EjecutarTarea(); });

         var tarea2 = Task.Run(() => { EjecutarTarea2(); });
         Task.WaitAll(tarea1, tarea2);
         var tarea3 = Task.Run(() => { EjecutarTarea3(); });



      }

      static void EjecutarTarea()
      {
         for (int i = 0; i < 5; i++)
         {
            var miThread = Thread.CurrentThread.ManagedThreadId;
            Thread.Sleep(1000);
            Console.WriteLine("Esta vuelta de bucle corresponde a la tarea 1");
         }
      }
      static void EjecutarTarea2()
      {
         for (int i = 0; i < 5; i++)
         {
            var miThread = Thread.CurrentThread.ManagedThreadId;
            Thread.Sleep(500);
            Console.WriteLine("Esta vuelta de bucle corresponde a la tarea 2");
         }
      }
```

```
        static void EjecutarTarea3()
        {
            for (int i = 0; i < 5; i++)
            {
                var miThread = Thread.CurrentThread.ManagedThreadId;
                Thread.Sleep(500);
                Console.WriteLine("Esta vuelta de bucle corresponde a la tarea 3");
            }
        }

    }
}
```

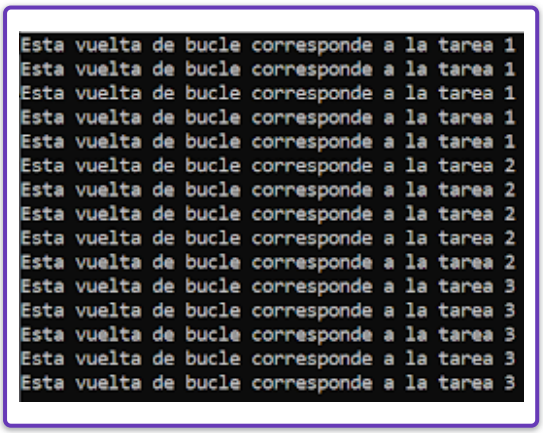
En el siguiente código fuente, indique cual es la respuesta correcta:



○ Opción 3



○ Opción 1



◉ Opción 2

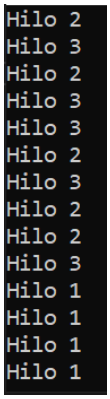
○ Ninguna

using System; * 0/1

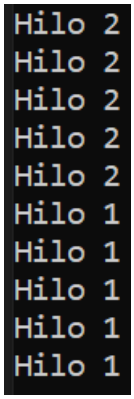
```
namespace Hilo
{
   internal class Program
   {
      static void Main(string[] args)
      {
         Thread s = new Thread(segundo);
         s.Start();

         Thread t = new Thread(tercero);
         t.Start();
         s.Join();

         Console.WriteLine("Hilo 1");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 1");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 1");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 1");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 1");
      }

      static void segundo()
      {
         Console.WriteLine("Hilo 2");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 2");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 2");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 2");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 2");

      }
      static void tercero()
      {
         Console.WriteLine("Hilo 3");
```

```
            Thread.Sleep(1000);
            Console.WriteLine("Hilo 3");
            Thread.Sleep(1000);
            Console.WriteLine("Hilo 3");
            Thread.Sleep(1000);
            Console.WriteLine("Hilo 3");
            Thread.Sleep(1000);
            Console.WriteLine("Hilo 3");

        }
    }
}
```

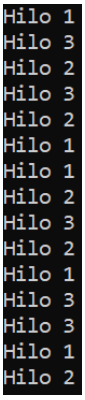
En el siguiente código fuente, indique cual es la respuesta correcta:



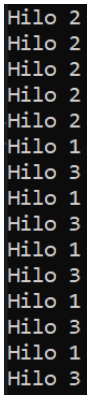
Opción 3



Opción 4



Opción 2



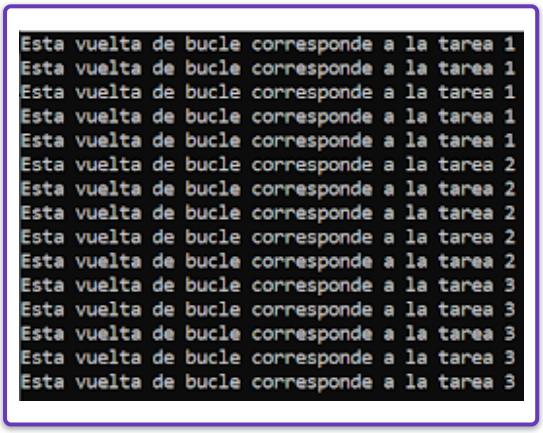
Opción 1

* 1/1

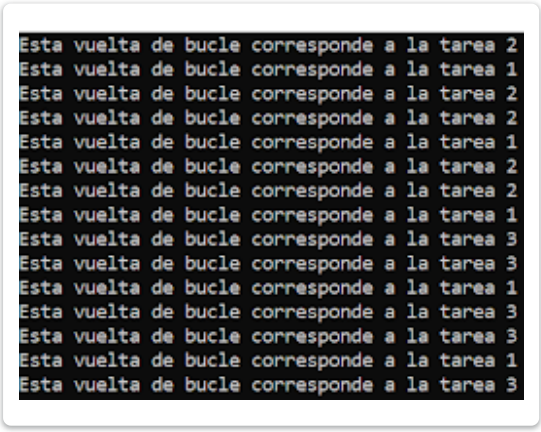
```
using System;
using System.Runtime.CompilerServices;

namespace MyApp // Note: actual namespace depends on the project name.
{
    internal class Program
    {
        static void Main(string[] args)
        {
            RealizarTodasTareas();
            Console.ReadLine();
        }

        static void RealizarTodasTareas()
        {
            var tarea1 = Task.Run(() => { EjecutarTarea(); });
            tarea1.Wait();
            var tarea2 = Task.Run(() => { EjecutarTarea2(); });
            tarea2.Wait();
            var tarea3 = Task.Run(() => { EjecutarTarea3(); });



        }

        static void EjecutarTarea()
        {
            for (int i = 0; i < 5; i++)
            {
                var miThread = Thread.CurrentThread.ManagedThreadId;
                Thread.Sleep(1000);
                Console.WriteLine("Esta vuelta de bucle corresponde a la tarea 1");
            }
        }
        static void EjecutarTarea2()
        {
            for (int i = 0; i < 5; i++)
            {
                var miThread = Thread.CurrentThread.ManagedThreadId;
                Thread.Sleep(500);
                Console.WriteLine("Esta vuelta de bucle corresponde a la tarea 2");
            }
        }
```

```
        static void EjecutarTarea3()
        {
            for (int i = 0; i < 5; i++)
            {
                var miThread = Thread.CurrentThread.ManagedThreadId;
                Thread.Sleep(500);
                Console.WriteLine("Esta vuelta de bucle corresponde a la tarea 3");
            }
        }

    }
}
```

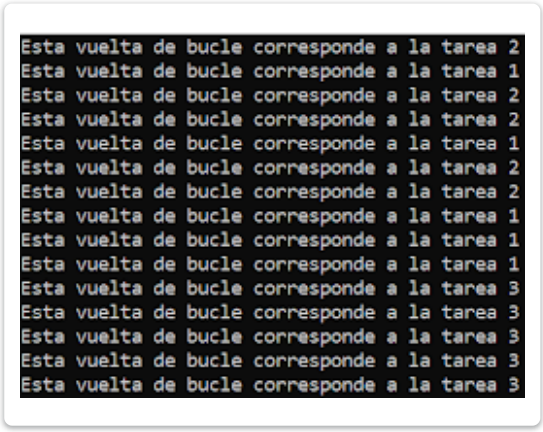
En el siguiente código fuente, indique cual es la respuesta correcta:



- (●) Opción 2
- ( ) Ninguna



- ( ) Opción 1



- ( ) Opción 3

Hacen que sea un poco más fácil, más rápido y más breve la creación de una función. Es una alternativa más compacta para expresar una función. Es más limitada, como para funciones más chicas se pueden crear y que no ocupen tanto espacio en memoria. *1/1

Seleccione cual es la definición correcta:

Función Lambda

----------Class---------- *

0/1

```
// Función tradicional

static int trescientos(int a){

return a + 300;

}

----------Main()----------

Console.WriteLine(trescientos(3));
```

Seleccione las opciones necesarias y cuales puede sustituir esta función tradicional en C# :

- [x] public delegate int Calculadora(int num);//en Class
- [x] Calculadora cien = new Calculadora ((a) => { return a + 300;}); //en Main()
- [ ] Calculadora cien = new Calculadora (a => a + 300);//en Main()
- [ ] public delegate int Calculadoras(int num1, int num2);//en Class
- [ ] Calculadora cien = new Calculadora ((a) => a + 300);//en Main()

----------Class---------- * 0/1

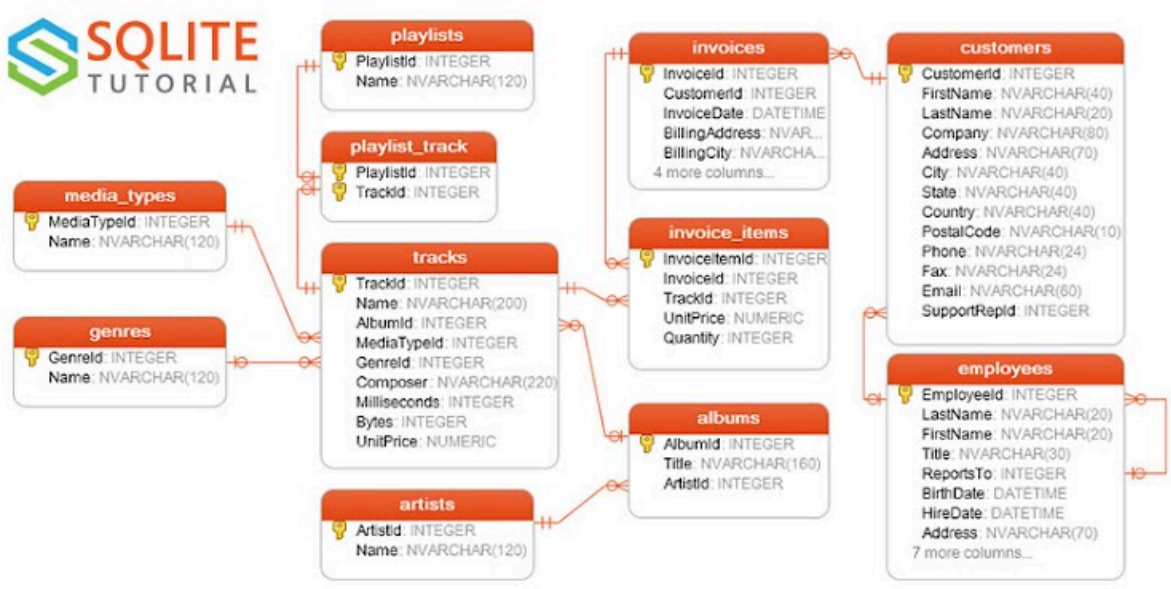
```
// Función tradicional

static int sustraccion(int num1, int num2){

        int resta= num1 - num2;

        return resta;

}

----------Main()----------

Console.WriteLine( sustraccion (3,4));
```

Seleccione las opciones necesarias y cuales puede sustituir esta función tradicional en C# :

- [x] Calculadoras sustraccion = new Calculadoras((num1, num2) => { int resta = num1 - num2; return resta; });//en Main()
- [ ] Calculadora sustraccion = new Calculadora (num1 => num1 - num2);//en Main()
- [x] Calculadoras sustraccion = new Calculadoras((num1, num2) => num1 - num2);//en Main()
- [ ] public delegate int Calculadoras(int num1, int num2);//en Class
- [ ] public delegate int Calculadora(int num);//en Class

Traer todas las canciones que duren 5.000.000 milisegundos o más. * 1/1

Coloque el código SQL correcto en la siguiente base de datos que aparece en la figura:



- (●) SELECT * FROM tracks WHERE Milliseconds >= 5000000;
- ( ) SELECT TrackId FROM tracks WHERE Milliseconds >= 5000000;
- ( ) SELECT TrackId FROM tracks WHERE Milliseconds > 5000000;
- ( ) SELECT * FROM tracks WHERE Milliseconds > 5000000;
- ( ) Ninguna

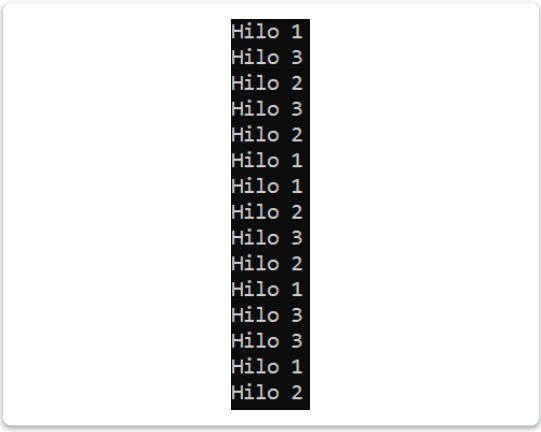
*

1/1

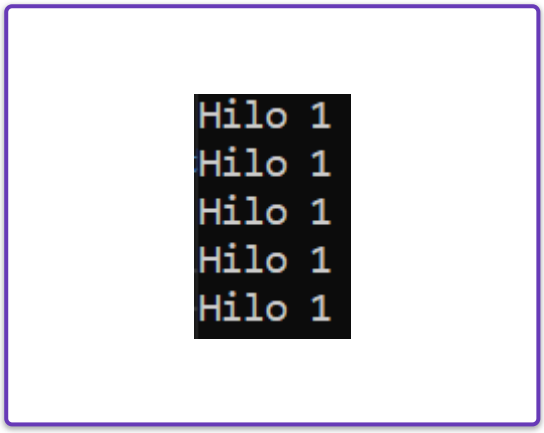
```
using System;

namespace Hilo
{
   internal class Program
   {
      static void Main(string[] args)
      {
         Thread s = new Thread(segundo);

         Thread t = new Thread(tercero);

         Console.WriteLine("Hilo 1");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 1");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 1");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 1");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 1");
      }

      static void segundo()
      {
         Console.WriteLine("Hilo 2");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 2");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 2");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 2");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 2");

      }
      static void tercero()
      {
         Console.WriteLine("Hilo 3");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 3");
         Thread.Sleep(1000);
```

```
            Console.WriteLine("Hilo 3");
            Thread.Sleep(1000);
            Console.WriteLine("Hilo 3");
            Thread.Sleep(1000);
            Console.WriteLine("Hilo 3");

        }
    }
}
```

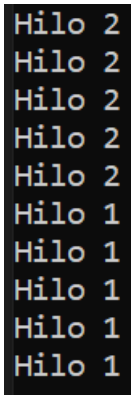
En el siguiente código fuente, indique cual es la respuesta correcta:



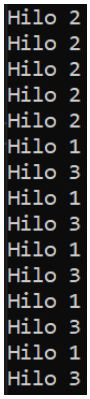
Opción 2



Opción 4



Opción 1



Opción 3

* 0/1

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
   class Program
   {
      public static string Mid(string param, int startIndex, int length)
      {
         string result = param.Substring(startIndex, length);
         return result;
      }
      static void Main(string[] args)
      {
         string NOM;
         string CAR;
         int K = 0;
         int P = 0;
         int CI = 0;
         Console.Write("COLOQUE EL NÚMERO DE HILO: ");
         NOM = Console.ReadLine();
         Console.Clear();
         Console.SetCursorPosition(25, 1);
         Console.Write(NOM);
         CI = 25;
         NOM = NOM.ToUpper();
         for (P = 1; P <= NOM.Length; P++)
         {
            CAR = Mid(NOM, P - 1, 1);
            for (K = 1; K <= 22; K++)
            {
               Console.SetCursorPosition(CI, K);
               Console.Write(CAR);
               // REALIZAMOS UNA PAUSA
               System.Threading.Thread.Sleep(50);
               Console.SetCursorPosition(CI, K);
               Console.Write(" ");
            }
            Console.SetCursorPosition(CI, K);
            Console.Write(CAR);
            CI = CI + 1;
```

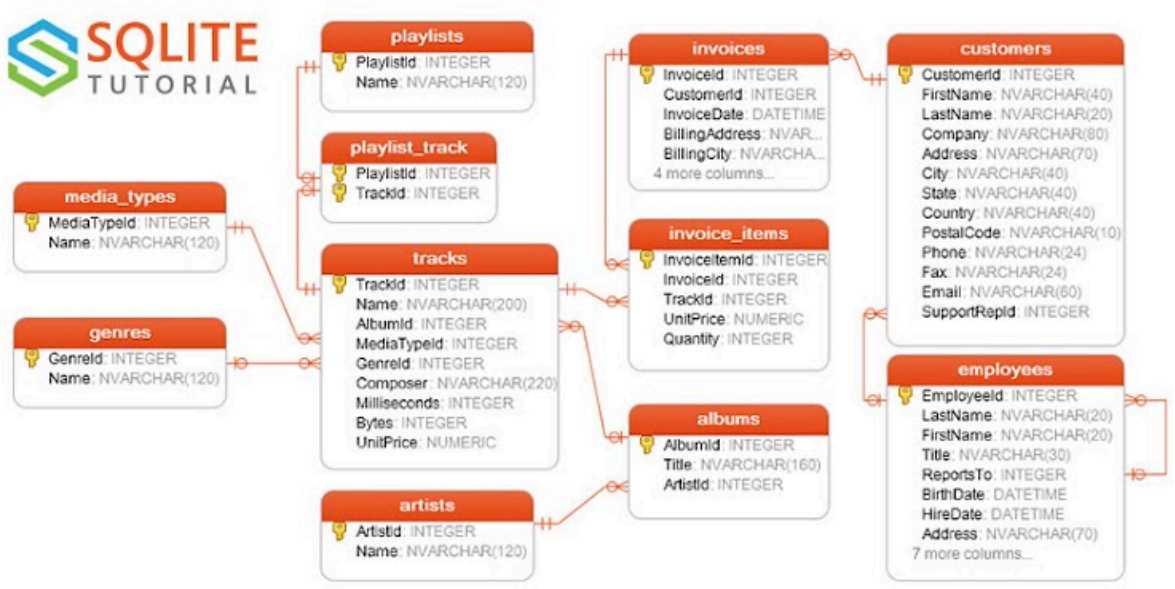
```
            }
            Console.WriteLine();
            Console.Write("Pulse una Tecla:");
            Console.ReadKey();
        }
    }
}
```

En el siguiente código fuente indique que hace el programa:

- (●) Mueve un carácter de izquierda a derecha, apenas termina uno comienza a desplazarse el otro.
- ( ) Mueve un carácter de izquierda a derecha, todavía no ha finalizado cuando sale el otro.
- ( ) Mueve un carácter de arriba hacia abajo, apenas termina uno comienza a desplazarse el otro.
- ( ) Mueve un carácter de arriba hacia abajo, todavía no ha finalizado cuando sale el otro.
- ( ) Mueve una palabra completa de izquierda a derecha

Encontrar todos los mails de los clientes que comiencen con “J” y sean de gmail.com. *1/1

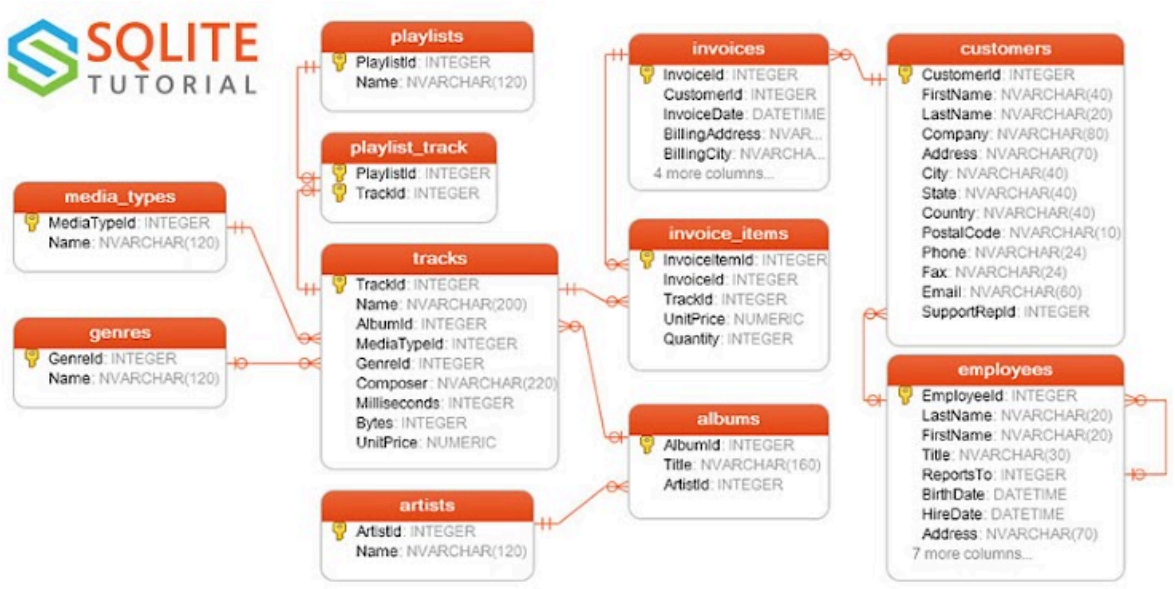
Coloque el código SQL correcto en la siguiente base de datos que aparece en la figura:



- (●) SELECT Email FROM customers WHERE Email LIKE 'j%gmail.com';
- ( ) SELECT Email FROM customers WHERE Email LIKE '%j___@gmail.com%';
- ( ) SELECT Email FROM customers WHERE Email LIKE '%j____gmail.com%';
- ( ) SELECT Email FROM customers WHERE Email LIKE '%j%gmail.com%';
- ( ) NINGUNA

Encontrar todas las facturas para los clientes 56 y 58 donde el total esté entre $1.00 and $5.00. (invoices) *1/1

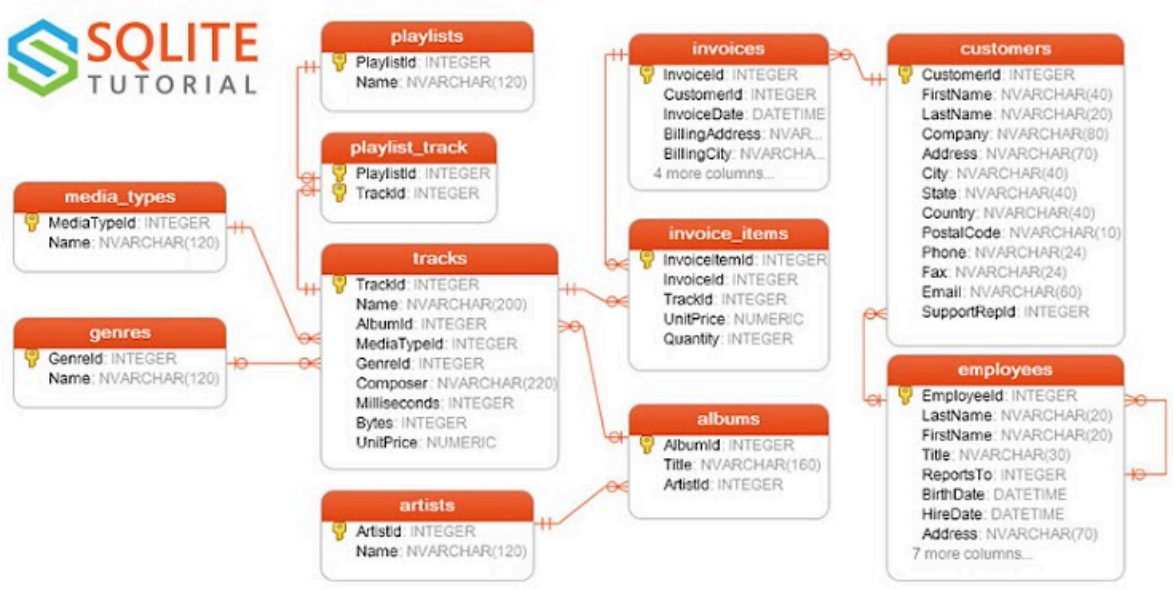
Coloque el código SQL correcto en la siguiente base de datos que aparece en la figura:



- (●) SELECT * FROM invoices WHERE CustomerId IN ('56', '58') AND total BETWEEN 1 AND 5;
- ( ) SELECT Invoiceid FROM invoices WHERE CustomerId IN ('56', '58') AND total BETWEEN 1 AND 5;
- ( ) SELECT * FROM invoices WHERE CustomerId IN ('56', '58') AND total BETWEEN 1 OR 5;
- ( ) SELECT * FROM invoices WHERE CustomerId IN ('56', '58') OR total BETWEEN 1 OR 5;
- ( ) NINGUNA

Traer todas las columnas de la tabla Tracks y ordenar por nombre e identificador del álbum. *1/1

Coloque el código SQL correcto en la siguiente base de datos que aparece en la figura:



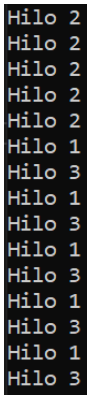
- ○ SELECT * FROM tracks ORDER BY Name;
- ○ SELECT * FROM tracks ORDER BY Name, AlbumId DESC;
- ◉ SELECT * FROM tracks ORDER BY Name, AlbumId;
- ○ SELECT * FROM tracks ORDER BY Name DESC;
- ○ SELECT * FROM tracks ORDER BY AlbumId DESC;

using System; * 0/1

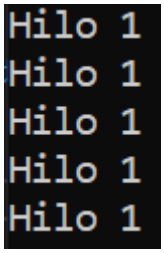
```
namespace Hilo
{
   internal class Program
   {
      static void Main(string[] args)
      {
         Thread s = new Thread(segundo);
         s.Start();
         s.Join();

         Thread t = new Thread(tercero);


         Console.WriteLine("Hilo 1");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 1");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 1");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 1");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 1");
      }

      static void segundo()
      {
         Console.WriteLine("Hilo 2");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 2");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 2");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 2");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 2");

      }
      static void tercero()
      {
         Console.WriteLine("Hilo 3");
```

```
            Thread.Sleep(1000);
            Console.WriteLine("Hilo 3");
            Thread.Sleep(1000);
            Console.WriteLine("Hilo 3");
            Thread.Sleep(1000);
            Console.WriteLine("Hilo 3");
            Thread.Sleep(1000);
            Console.WriteLine("Hilo 3");

        }
    }
}
```

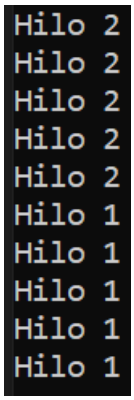
En el siguiente código fuente, indique cual es la respuesta correcta:



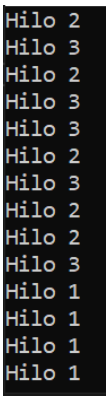
Opción 3



Opción 4



Opción 2



Opción 1

Si el programa permitiera lanzar un hilo secundario por cada petición que hiciera programa, estaríamos en el caso de múltiples hilos ejecutándose al mismo tiempo y compitiendo por quien termina primero. *0/1

Seleccione cual es la definición correcta:

Funciones

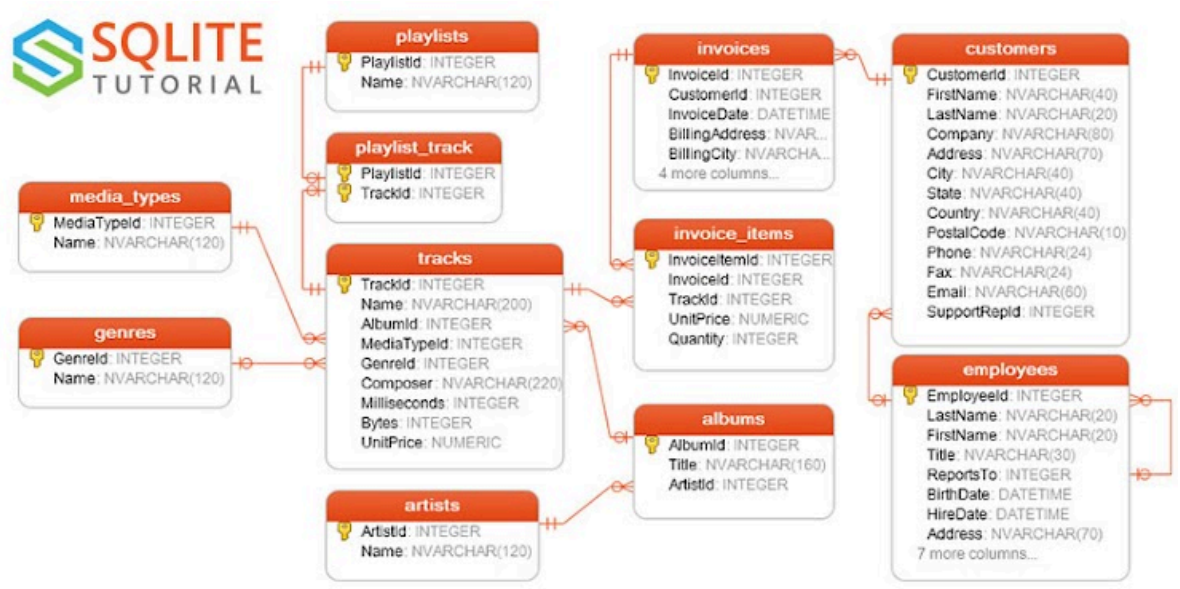
Borrar la base de datos CUDI * 1/1

Debe escribir la sentencia correcta en SQL:

DROP DATABASE CUDI;

Mostrar el número de compras hechas por cada cliente y ordenarlas por la cantidad descendentemente (invoices). *0/1

Coloque el código SQL correcto en la siguiente base de datos que aparece en la figura:



- ○ SELECT CustomerID, COUNT (InvoiceId) FROM invoices GROUP BY CustomerID ORDER BY InvoiceId DESC;
- ◉ SELECT *, COUNT (InvoiceId) FROM invoices GROUP BY CustomerID ORDER BY InvoiceId DESC;
- ○ SELECT CustomerID, COUNT (InvoiceId) FROM invoices GROUP BY CustomerID HAVING InvoiceId DESC;
- ○ SELECT CustomerID, COUNT (InvoiceId) FROM invoices GROUP BY CustomerID ORDER BY InvoiceId;
- ○ NINGUNA

En la tabla siguiente se muestran las acciones que pueden provocar un cambio de estado, junto con el nuevo estado correspondiente. *

A continuación, seleccione las definiciones correctas de algunos de los métodos que se pueden utilizar para controlar hilos independientes:

Acción	Nuevo estado resultante
Otro hilo llama a **Thread.Start**	No cambia
El hilo responde a **Thread.Start** y empieza a ejecutarse	**Running**
El hilo llama a **Thread.Sleep**	**WaitSleepJoin**
El hilo llama a **Monitor.Wait** en otro objeto	**WaitSleepJoin**
El hilo llama a **Thread.Join** en otro hilo	**WaitSleepJoin**
Otro hilo llama a **Thread.Interrupt**	**Running**
Otro hilo llama a **Thread.Abort**	**AbortRequested**
El hilo responde a un **Thread.Abort**	**Aborted**

	Inicia la ejecución de un hilo.	Detiene un hilo durante un tiempo determinado.	Detiene un hilo cuando alcanza un punto seguro.	Deja en espera un hilo hasta que finaliza otro hilo diferente. Si se utiliza con un valor de tiempo de espera, este método devolverá true cuando el hilo finaliza en el tiempo asignado.	Puntuación
Start	☐	☐	☐	☑	0/1
Sleep	☑	☐	☐	☐	0/1
Join	☐	☑	☐	☐	0/1
Abort	☐	☐	☑	☐	1/1

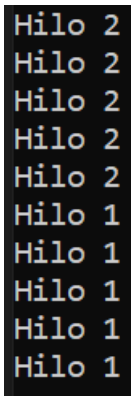
*

1/1

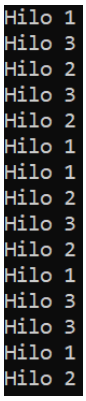
```
using System;

namespace Hilo
{
   internal class Program
   {
      static void Main(string[] args)
      {
         Thread s = new Thread(segundo);
         s.Start();
         s.Join();

         Thread t = new Thread(tercero);
         t.Start();
         s.Join();

         Console.WriteLine("Hilo 1");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 1");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 1");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 1");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 1");
      }

      static void segundo()
      {
         Console.WriteLine("Hilo 2");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 2");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 2");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 2");
         Thread.Sleep(1000);
         Console.WriteLine("Hilo 2");

      }
      static void tercero()
      {
```

```
            Console.WriteLine("Hilo 3");
            Thread.Sleep(1000);
            Console.WriteLine("Hilo 3");
            Thread.Sleep(1000);
            Console.WriteLine("Hilo 3");
            Thread.Sleep(1000);
            Console.WriteLine("Hilo 3");
            Thread.Sleep(1000);
            Console.WriteLine("Hilo 3");

        }
    }
}
```

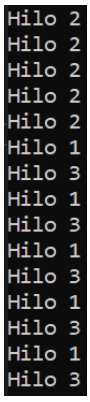
En el siguiente código fuente, indique cual es la respuesta correcta:



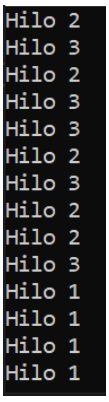
Opción 4



Opción 2



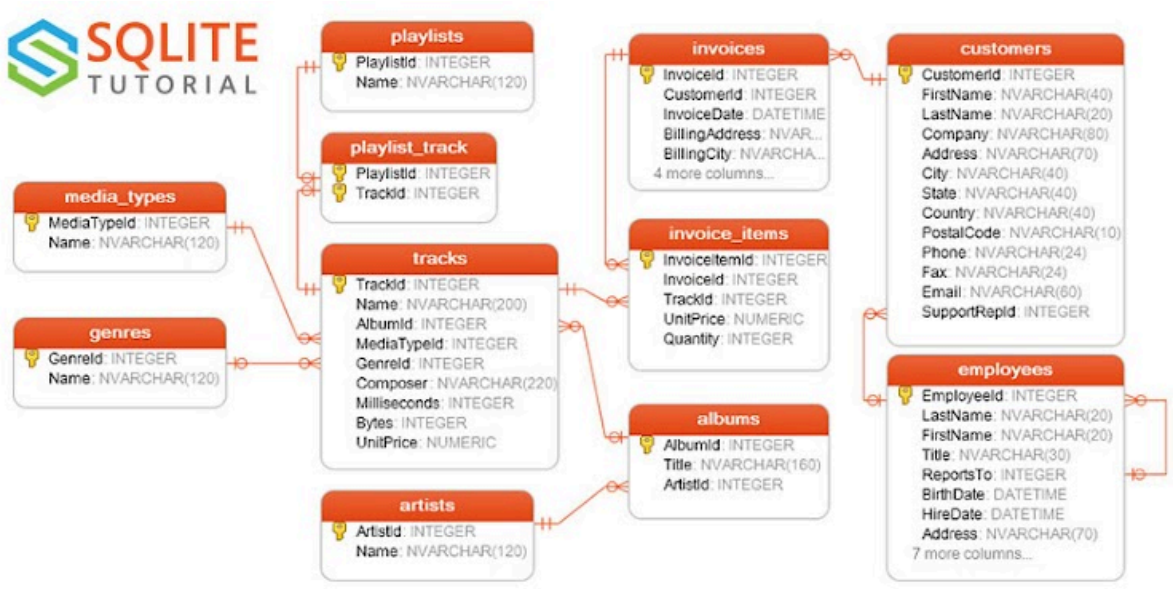
Opción 3



Opción 1

Traer todas las columnas de la tabla Tracks y ordenar por nombre identificado del álbum de forma decreciente. *0/1

Coloque el código SQL correcto en la siguiente base de datos que aparece en la figura:



- ○ SELECT * FROM tracks ORDER BY Name;
- ○ SELECT * FROM tracks ORDER BY Name, AlbumId DESC;
- ○ SELECT * FROM tracks ORDER BY Name, AlbumId;
- ◉ SELECT * FROM tracks ORDER BY Name DESC;
- ○ SELECT * FROM tracks ORDER BY AlbumId DESC;

Si quisiera ver toda la población mundial cual seria el código SQL que debo escribir en la base de datos world: *0/1

Seleccione la opción correcta:

SELECT Count(ciudad.Population) FROM `city` ciudad JOIN `country` pais ON ciudad.CountryCode = pais.Code;

Típicamente, operaciones de agregado, borrado, modificación y recuperación de los datos de la base. *0/1

Seleccione cual es la definición correcta:

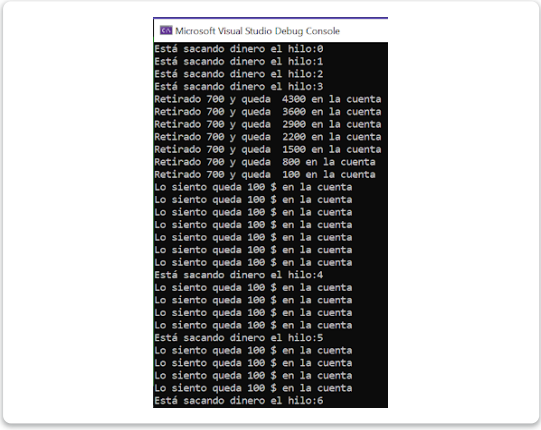
Las restricciones de integridad

*.../1

```
namespace Hilo
{
   internal class Program
   {
      static void Main(string[] args)
      {
         CuentaBancaria CuentaFamilia = new CuentaBancaria(5000);
         Thread[] hilosPersonas = new Thread[15];

         for (int i = 0; i < 15; i++)
         {
            Thread t = new Thread(CuentaFamilia.VamosRetirarEfectivo);
            t.Name = i.ToString();
            hilosPersonas[i] = t;
         }
         for (int i = 0; i < 15; i++) hilosPersonas[i].Start();
      }

      class CuentaBancaria
      {
         private Object bloqueaSaldoPositivo = new Object();
         double Saldo { set; get; }
         public CuentaBancaria(double Saldo)
         {
            this.Saldo = Saldo;
         }
         public double RetirarEfectivo(double Cantidad)
         {
            if ((Saldo - Cantidad) < 0)
            {
               Console.WriteLine($"Lo siento queda {Saldo} $ en la cuenta");
               return Saldo;
            }
            lock (bloqueaSaldoPositivo)
            {
               if (Saldo >= Cantidad)
               {
                  Console.WriteLine("Retirado {0} y queda  {1} en la cuenta",
Cantidad, (Saldo - Cantidad), Thread.CurrentThread.Name);
                  Saldo -= Cantidad;
               }
               return Saldo;
```

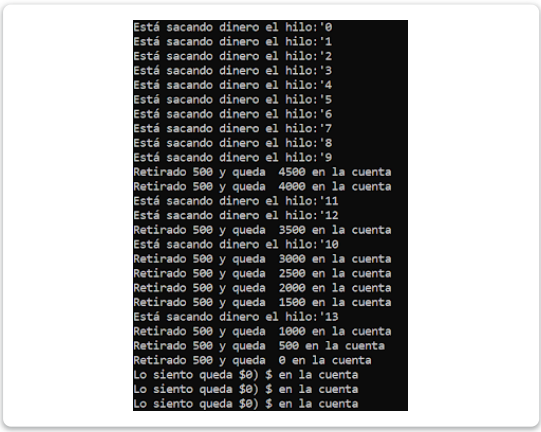
```
            }
        }
        public void VamosRetirarEfectivo()
        {
            Console.WriteLine("Está sacando dinero el hilo:{0}",
Thread.CurrentThread.Name);
            for (int i = 0; i < 4; i++) RetirarEfectivo(700);
        }
    }
  }
}
```

En el siguiente código fuente, indique cual es la respuesta correcta:

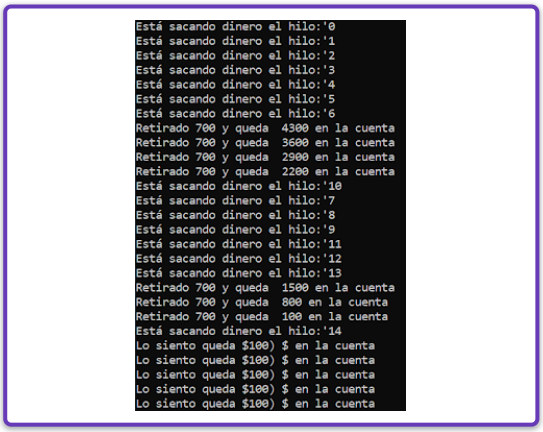


○ Opción 2

○ Ninguna Opción



○ Opción 3



◉ Opción 1

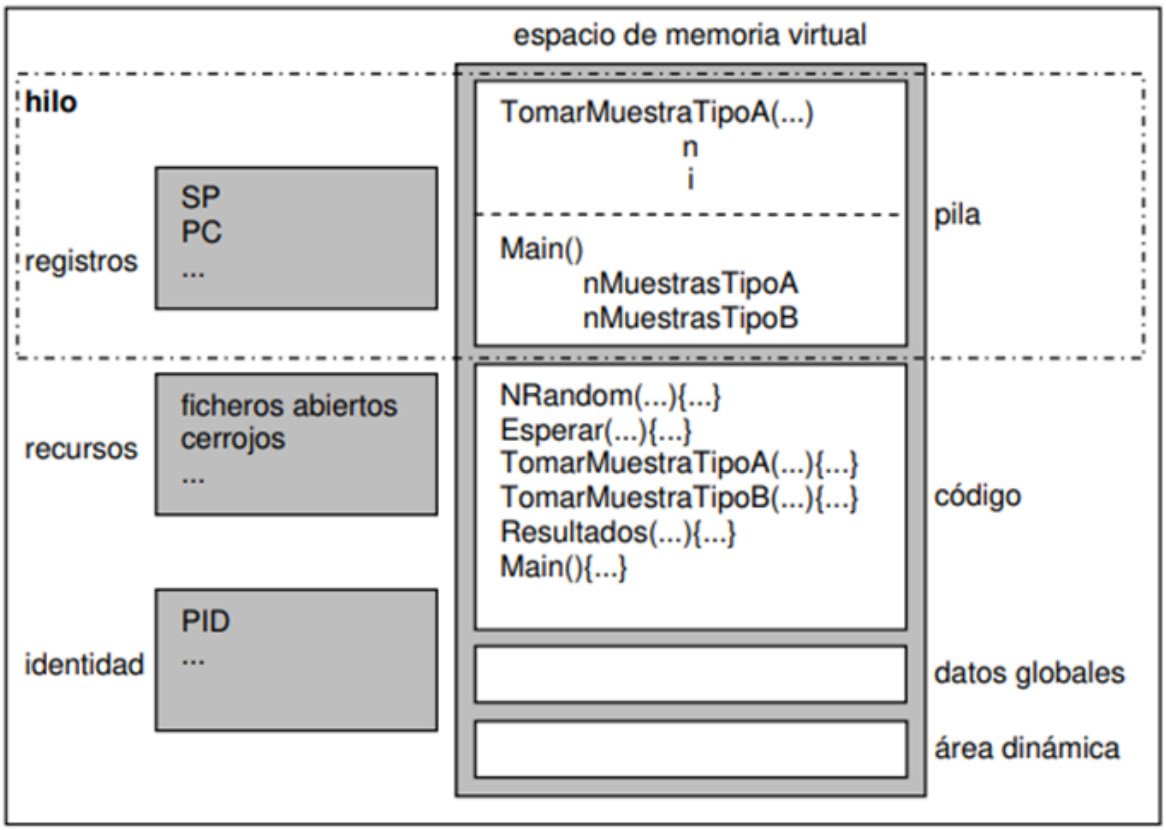
Borras el atributo nota de la tabla estudiantes: * 1/1

Debe escribir la sentencia correcta en SQL:

ALTER TABLE estudiantes DROP COLUMN nota;

En la siguiente figura puede observar que existen varias regiones de memoria: *

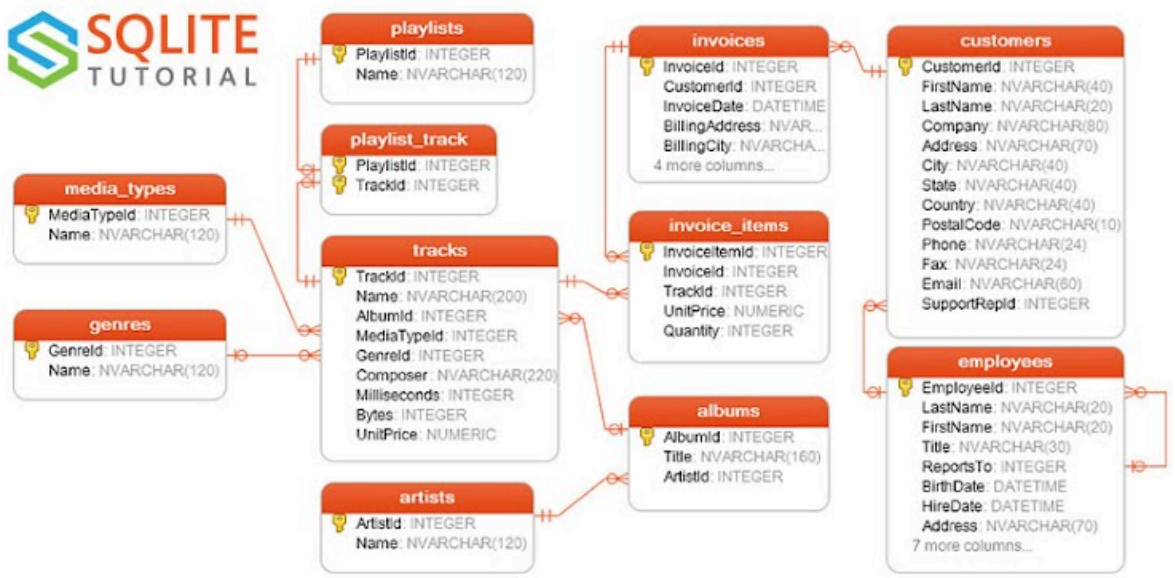
Seleccione las definiciones correctas:



	área de memoria donde se mantienen las variables automáticas y la información necesaria para continuar con la ejecución del método que define el hilo.	área de memoria donde se mantiene el código del programa.	área de memoria donde se mantienen los datos globales.	área de memoria para asignación dinámica.	Puntuación
Área dinámica:	☑	☐	☐	☐	0/1
Datos globales:	☐	☑	☐	☐	0/1
Pila:	☐	☐	☑	☐	0/1
Código:	☐	☐	☐	☑	0/1

Encontrar todas las facturas con un total entre 5 y 14 dólares. (invoices) * 0/1

Coloque el código SQL correcto en la siguiente base de datos que aparece en la figura:



- ○ SELECT * FROM invoices WHERE total >= 5 AND total <= 14;
- ○ SELECT * FROM invoices WHERE total > 5 AND total < 14;
- ◉ SELECT * FROM invoices WHERE total > 5 AND total <= 14;
- ○ SELECT * FROM invoices WHERE total >= 5 AND total < 14;
- ○ Ninguna

Es un tipo que representa una firma de método particular que devuelve verdadero o falso. Una instancia de este tipo se refiere a un método particular con una firma coincidente. *0/1

Seleccione cual es la definición correcta:

Hilos ▾