

Ejercicios de Programación en Consola de C#

Ing. Mg. Patricio Medina

Primera Edición

A dark blue vertical bar on the left side of the page. A blue arrow-shaped graphic points to the right, containing the date.

12-6-2015

Ejercicios de Programación en Consola de C#

A series of thin, curved lines in shades of blue and grey that originate from the bottom left and sweep upwards and to the right, creating a dynamic, abstract graphic element.

Ing. Mg. Patricio Medina

Profesor de la Universidad Técnica de Ambato

Profesor de la Pontificia Universidad Católica del Ecuador Sede Ambato

Asesor Informático

Primera Edición

Editorial Independiente

Autor: Ing. Mg. Patricio Medina

Capacitación realizada en: Cuba, Brasil y Ecuador

Contacto: ricardopmedina@uta.edu.ec, pmedina@pucesa.edu.ec, medina_patricio@yahoo.es

ISBN- 978-9942-21-144-6

Certificado Instituto de Propiedad Intelectual: QUI-044947

Primera Edición

Ambato – Ecuador

2015

Todos los nombres propios de productos y servicios que aparecen en este texto son marcas registradas de sus respectivas compañías u organizaciones. Denotamos éstos tan solo con fines de divulgación.

Las posiciones expresadas en este libro reflejan exclusivamente la opinión de su autor y por lo tanto no representan en ningún caso la posición de la UTA y PUCESA.

Publicación de distribución gratuita. Los contenidos pueden difundirse y reproducirse con fines educativos y con la condición de reconocer los créditos correspondientes.

Contenido

INVERTIR NÚMERO DE DOS CIFRAS	1
INVERTIR NÚMERO DE TRES CIFRAS	2
OPERACIONES BÁSICAS	3
COMPRA EN RESTAURANT	5
FUNCIONES BÁSICAS LIBRERÍA MATH.....	6
FORMATOS DE SALIDA	7
EJERCICIO PROPUESTO.....	9
MAYOR DE DOS NÚMEROS	10
MAYOR DE TRES NÚMEROS	12
DESGLOSE DE BILLETES	13
BONO DEL EMPLEADO POR HIJO	15
NÚMERO INTERMEDIO	16
TARIFA TELEFÓNICA	18
TRIÁNGULOS	19
EJERCICIO PROPUESTO.....	20
DÍA DE LA SEMANA	22
ESTADO CIVIL.....	24
CALIFICACIÓN	25
EJERCICIO PROPUESTO.....	27
TABLA DE MULTIPLICAR	29
PRESUPUESTO ANUAL EN AREAS HOSPITAL	30
SUMA DE N NÚMEROS PARES E IMPARES	32
TABLAS DE MULTIPLICAR	33
SUMA DE N NÚMEROS.....	35
MAYOR Y MENOR DE N NÚMEROS.....	36
SERIE DE FIBONACCI.....	38
CALIFICACIONES DE UN GRUPO DE ESTUDIANTES.....	39
NÚMEROS ALEATORIOS Y CARACTERES ASCII	41
FUNCIONES DE CADENA.....	42
SIMULACIÓN DE UN RELOJ DIGITAL.....	44
CANTIDAD DE VOCAL CERRADAS	46
ESTADISTICA POR VOCAL	47

FACTORIAL DE UN NÚMERO	49
SERIE DE UN NÚMERO	50
SUCESIÓN DE N CUADROS	52
MOVIMIENTO DE UNA FRASE	53
MOVIMIENTO CARÁCTER POR CARÁCTER HORIZONTAL.....	55
MOVIMIENTO CARÁCTER POR CARÁCTER VERTICAL.....	57
EJERCICIO PROPUESTO.....	59
TABLA DE MULTIPLICAR CON WHILE	60
TABLAS DE MULTIPLICAR CON WHILE.....	62
COMPROBAR SI ES NÚMERO PRIMO	63
FACTORES PRIMOS DE UN NÚMERO	65
GENERAR N NÚMEROS PRIMOS.....	66
EJERCICIO PROPUESTOS.....	68
VERIFICACIÓN DE UNA CLAVE 3 OPORTUNIDADES	69
GENERAR UN NÚMERO ALEATORIO ENTRE 10 Y 30	71
JUEGO ADIVINA UN NÚMERO.....	72
CONTROL DE UNA FACTURA	74
VOTACIONES POR SECTOR	76
PROMEDIO DE SUELDOS CERO O NEGATIVO SALE	77
FRASE INVERTIDA CON WHILE	79
EJERCICIO PROPUESTO.....	80
TEMA: MANIPULACIÓN DE UN ARREGLO	82
INTERCALACIÓN MAYUSCULAS Y MINUSCULAS	84
GENERAR NÚMEROS ALETORIOS EN UN ARREGLO	86
PASAR UNA FRASE A UN ARREGLO	88
SUMA ELEMENTOS PARES E IMPARES EN UN ARREGLO	89
MAYOR Y MENOR DE UN ARREGLO DE N ELEMENTOS.....	91
OBTENER EL DÍGITO VERIFICADOR DE LA CÉDULA	93
INSERTAR UN ELEMENTO EN UN ARREGLO	95
ELIMINAR UN ELEMENTO EN UN ARREGLO.....	97
SUMA DE DOS ARREGLOS DE 5 ELEMENTOS	99
SUMA DE DOS ARREGLOS DE 5 ELEMENTOS INTERCALADO	101
EJERCICIO PROPUESTO.....	103

EJERCICIO PROPUESTO.....	105
NÚMERO DECIMAL A BINARIO.....	107
NÚMERO DECIMAL A OCTAL.....	109
NÚMERO DECIMAL A HEXADECIMAL.....	111
EJERCICIO PROPUESTO.....	114
ORDENAMIENTO DE UN ARREGLO MÉTODO BURBUJA	117
BÚSQUEDA DE UN ELEMENTO EN UN ARREGLO	119
BÚSQUEDA BINARIA DE UN ELEMENTO EN UN ARREGLO.....	121
EJERCICIO PROPUESTO.....	123
TABLAS DE MULTIPLICAR EN UNA MATRIZ DE NxN	125
GENERAR ALEATORIOS EN UNA MATRIZ DE 5x5	127
SUMAR ELEMENTOS DE UNA MATRIZ DE NxN	128
SUMAR ELEMENTOS DE FILA Y UNA COLUMNA MATRIZ DE 5x5.....	130
SUMAR ELEMENTOS DE DIAGONAL PRINCIPAL Y SECUNDARIA	131
MATRIZ DE NxN.....	131
FORMAR LA LETRA A EN UNA MATRIZ DE NxN.....	133
EJERCICIO PROPUESTO.....	135
FORMAR LA LETRA R EN UNA MATRIZ DE NxN	138
SUMAR ELEMENTOS DE DIAGONAL PRINCIPAL Y SECUNDARIA	140
MATRIZ DE NxN.....	140
NÚMERO MAYOR Y MENOR EN UNA MATRIZ DE NxN	142
ORDENAMIENTO DE UNA MATRIZ DE NxN.....	144
SUMA DE MATRICES DE 5x5.....	146
MULTIPLICACIÓN DE MATRICES DE 4x4.....	148
EJERCICIO PROPUESTO.....	150
GENERACIÓN DEL TRÍANGULO DE PASCAL FORMA 1	152
GENERACIÓN DEL TRÍANGULO DE PASCAL FORMA 2	153
MATRIZ TRANSPUESTA DE NxN.....	155
MAYORES DE CADA FILA DE UNA MATRIZ NxN EN UN VECTOR.....	157
MENORES DE CADA COLUMNA DE UNA MATRIZ NxN EN UN VECTOR	158
PROMEDIOS DE CADA COLUMNA DE UNA MATRIZ NxN EN UN.....	160
VECTOR.....	160
VOTACIONES. SUMA DE CADA COLUMNA REPRESENTA A UN.....	162

CANDIDATO OBTENER EL CANDIDATO GANADOR.....	162
CUBO DE UN NÚMERO.....	164
MAYOR DE TRES NÚMEROS	165
VALOR ABSOLUTO DE UN NÚMERO	167
FACTORIAL DE UN NÚMERO	168
INVERTIR UNA FRASE	169
COMPROBAR SI UN NÚMERO ES MÚLTIPLO DE OTRO	171
NÚMERO A QUE DÍA DE LA SEMANA CORRESPONDE.....	172
NÚMERO COMPROBAR SI ES PRIMO	174
MENOR EN UN ARREGLO	176
TRANSFORMAR NÚMERO DECIMAL A BINARIO	177
EJERCICIO PROPUESTO.....	179
OBTENER EL DÍGITO VERIFICADOR DE LA CÉDULA	181
EJERCICIOS PROPUESTOS DIFERENTES Y NO RESUELTOS	183
GUÍA DE PROGRAMACIÓN DE C#.....	186
-Palabras clave de C#.....	186
-Tipos de instrucciones.....	191
-Instrucciones inalcanzables.....	192
Referencias.....	193

DEDICATORIA

Señor! Lo que tenga que hacer; lo haré lo mejor que pueda.

(Anónimo)

- A mi hermosa familia
- A mis estudiantes esperando que este nuevo material contribuya en su formación, ya que aprendo mucho junto a ustedes y es mi deber ser recíproco.

AGRADECIMIENTO

Deseo dejar constancia de la gran ayuda brindada por dos excelentes estudiantes David Bassante y David Mora, que realizaron su contribución con nuevos ejercicios, codificación, dedicando varias horas de trabajo.

INTRODUCCIÓN

*Me he dado cuenta que mientras
más trabajo, más suerte tengo
(Anónimo)*

Javier Olivar ¹ hace una reflexión sobre que: toda persona que pretenda construir un programa que de solución a determinada problemática, se enfrenta con dos grandes tareas.

La primera es el QUÉ, es decir, qué acciones debe realizar para poder resolver el problemas al cual se está enfrentando y el cual se realiza con papel y lápiz, como parte del trabajo de mesa previo a toda actividad de programación; la segunda es el CÓMO, o sea, de qué instrucciones se va a valer para escribir el código que realice las acciones determinadas en el QUÉ, las cuales están determinadas por el lenguaje seleccionado.

Actualmente existe una tendencia internacional a la enseñanza de lenguajes de programación, pasando por alto los elementos básicos de la Lógica de Programación, que constituyen a su vez la base sobre la cual se sustenta la programación en sí.

El principal problema al que se tiene para enfrentar la programación bien como profesión o como oficio es la poca fundamentación en Lógica.

De manera similar al hecho de que a un artista le enseñen a manejar muy bien un programa para dibujar (por ejemplo PhotoShop o algo así) y con ello consideren que es suficiente para que tenga claras todas las bases de la teoría del color y sus

¹ Blog <http://logicaunellezfs.blogspot.com/2011/07/actividad-de-logica.html>

diferentes vertientes. O se le enseña a un músico a manejar los menús de un sintetizador y se crea que con eso se le enseñó música, no podemos pensar que a un programador se le enseñe a programar tan solo con manejar las sentencias de un lenguaje

Considero que si a un programador primero se le entrena en Lógica de Programación podrá enfrentar cualquier lenguaje de programación al punto que, incluso, podría no enseñársele ningún lenguaje y el estudiante con la herramienta de la Lógica de Programación podrá utilizarlo sin ningún problema.

La lógica y el estudio de las estructuras de programación fundamentales definidas en la Programación Estructurada, contribuye al desarrollo del pensamiento lógico formal del individuo para acometer la programación de diversos algoritmos (estrategias de solución).

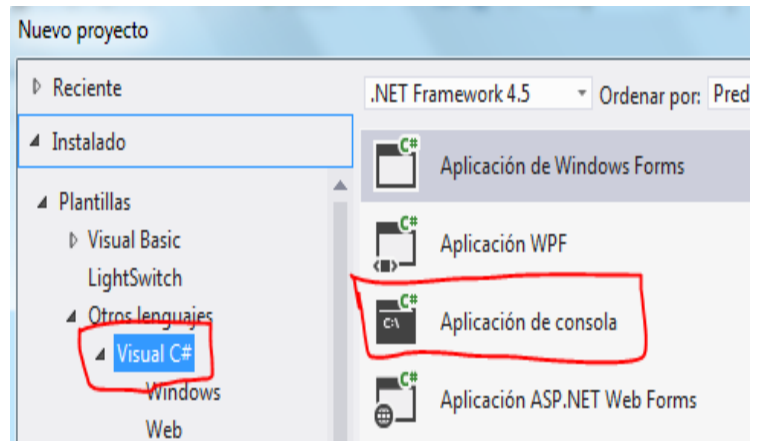
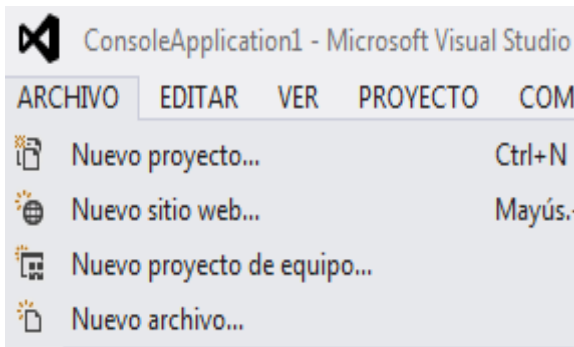
Si bien es cierto la programación actual se enfoca a la programación visual, de objetos, de agentes, de hilos, entre otras, no se puede dejar de lado la formación del estudiante en el manejo de estructuras de programación, por ello este texto ha sido diseñado como un recurso de auto aprendizaje y refuerzo fuera de clase, con contiene un variado conjunto de ejercicios que ilustran el uso de dichas estructuras, fortaleciendo el manejo de las mismas y creando una experiencia importantísima en el alumno

Para obtener los mejores resultados en cada Taller, recomiendo al lector analizar detenidamente cada indicación.

Esperando recibir sus comentarios y sugerencias serán muy bienvenidas a: medina_patricio@yahoo.es

INVERTIR NÚMERO DE DOS CIFRAS

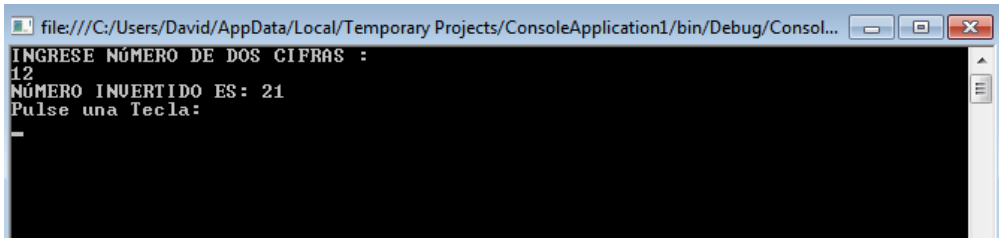
1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

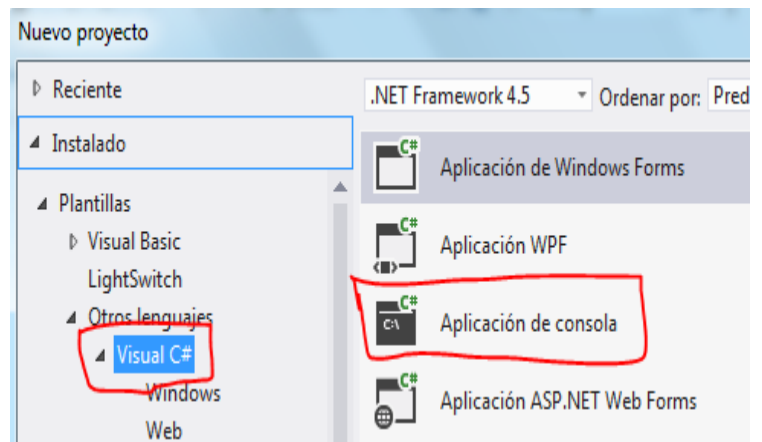
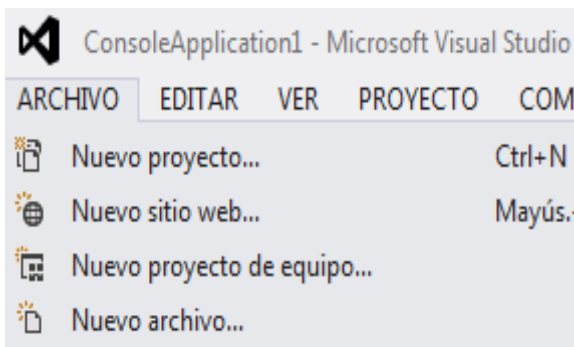
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int NUM, AUX, DEC, UNI;
            string linea;
            Console.WriteLine("INGRESE NÚMERO DE DOS CIFRAS :");
            linea = Console.ReadLine();
            NUM = int.Parse(linea);
            DEC = NUM/10;
            UNI = NUM % 10;
            AUX = (UNI * 10) + DEC;
            Console.WriteLine("NÚMERO INVERTIDO ES: " + AUX);
            Console.WriteLine("Pulse una Tecla:"); Console.ReadLine();
        }
    }
}
```

4. GRABAR Y EJECUTAR



INVERTIR NÚMERO DE TRES CIFRAS

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

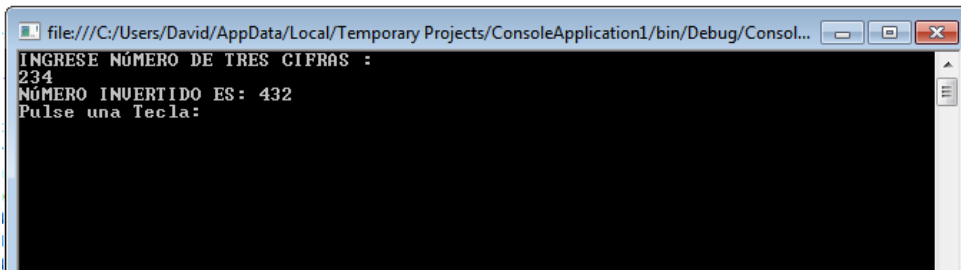
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int NUM, AUX, DEC, UNI, CEN;
            string linea;
            Console.WriteLine ("INGRESE NÚMERO DE TRES CIFRAS :");
            linea = Console.ReadLine();
            NUM = int.Parse(linea);
            CEN = NUM /100;
            NUM = NUM % 100;
```

```

    DEC = NUM/10;
    UNI = NUM % 10;
    AUX = (UNI * 100) + (DEC*10) + CEN;
    Console.WriteLine("NÚMERO INVERTIDO ES: " + AUX);
    Console.WriteLine("Pulse una Tecla:"); Console.ReadLine();
}
}
}
}
}

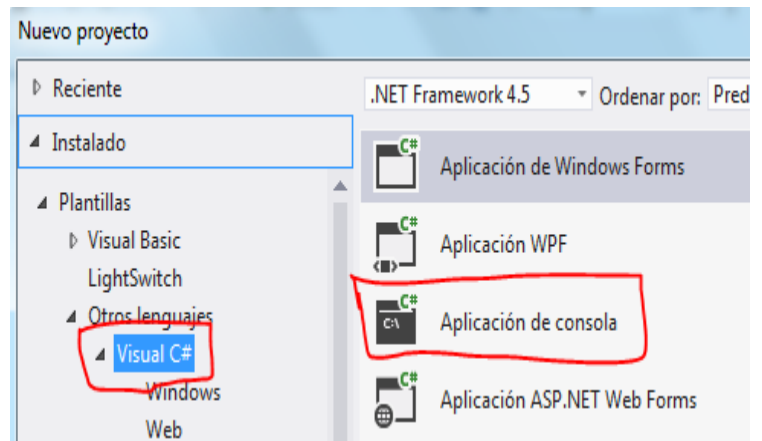
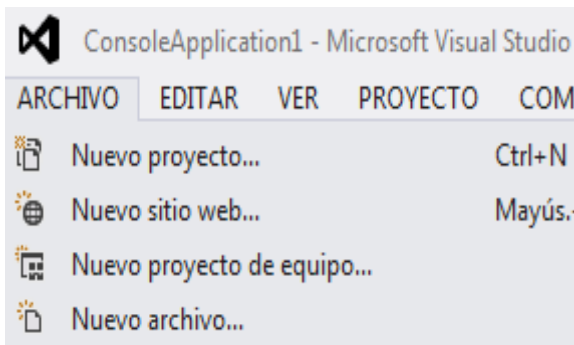
```

4. GRABAR Y EJECUTAR



OPERACIONES BÁSICAS

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

```

using System;
using System.Collections.Generic;

```

```

using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int NUM1, NUM2, RESULT;
            string linea;
            Console.Write("PRIMER NÚMERO :");
            linea = Console.ReadLine();
            NUM1 = int.Parse(linea);
            Console.Write("SEGUNDO NÚMERO :");
            linea = Console.ReadLine();
            NUM2 = int.Parse(linea);
            Console.WriteLine();
            RESULT = NUM1 + NUM2;
            Console.WriteLine("LA SUMA ES {0}: ", RESULT);
            RESULT = NUM1 - NUM2;
            Console.WriteLine("LA RESTA ES: {0} - {1} = {2} ", NUM1, NUM2, RESULT);
            RESULT = NUM1 * NUM2;
            Console.WriteLine("LA MULTIPLICACIÓN ES: " + RESULT);
            RESULT = NUM1 / NUM2;
            Console.WriteLine("LA DIVISIÓN ES: " + RESULT);
            RESULT = NUM1 % NUM2;
            Console.WriteLine("EL RESIDUO ES: " + RESULT);
            Console.Write("Pulse una Tecla:"); Console.ReadLine();
        }
    }
}

```

4. GRABAR Y EJECUTAR

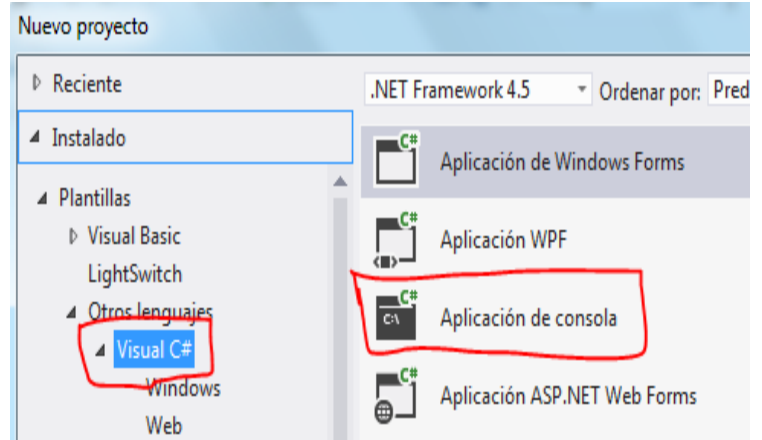
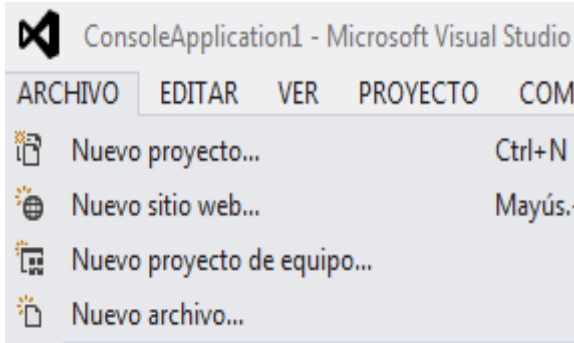
```

file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/Consol...
PRIMER NÚMERO :32
SEGUNDO NÚMERO :12
LA SUMA ES 44:
LA RESTA ES: 32 - 12 = 20
LA MULTIPLICACIÓN ES: 384
LA DIVISIÓN ES: 2
EL RESIDUO ES: 8
Pulse una Tecla:_

```

COMPRA EN RESTAURANT

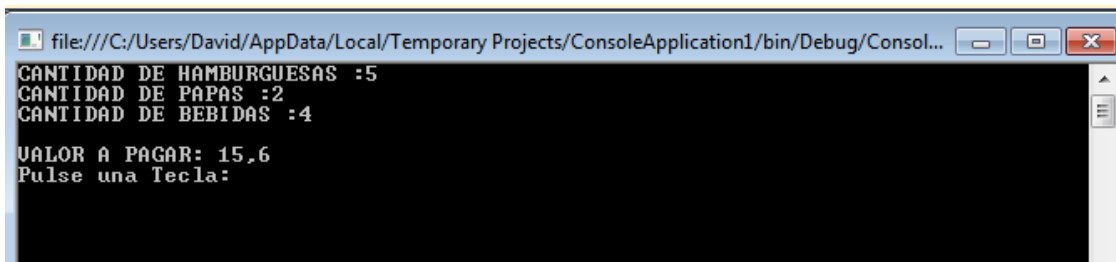
1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

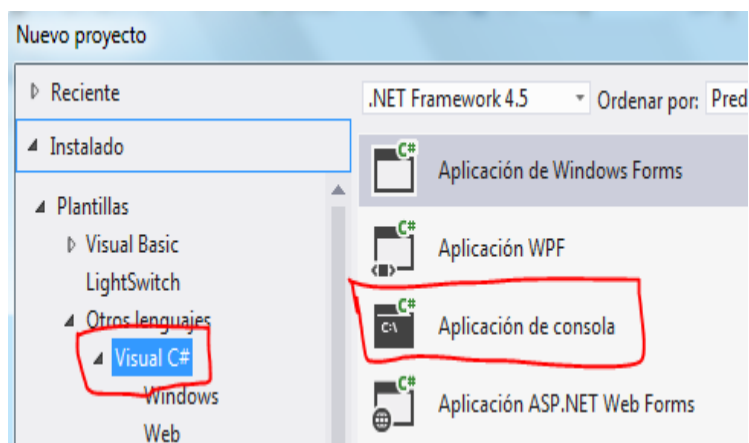
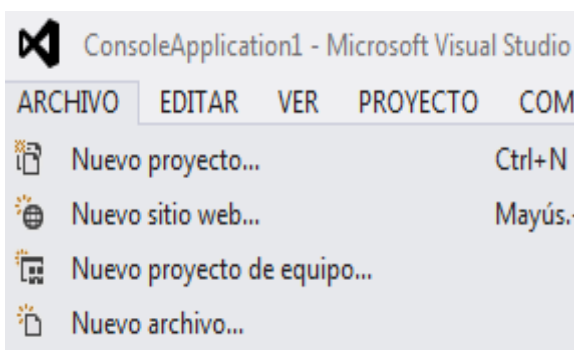
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            byte CANB, CANH, CANP;
            double APAGAR;
            string linea;
            const double PRECIOB = 0.8;
            const double PRECIOH = 2;
            const double PRECIOP = 1.2;
            Console.WriteLine("CANTIDAD DE HAMBURGUESAS :");linea = Console.ReadLine();
            CANH = byte.Parse (linea);
            Console.WriteLine("CANTIDAD DE PAPAS :");linea = Console.ReadLine();
            CANP = byte.Parse (linea);
            Console.WriteLine("CANTIDAD DE BEBIDAS :");linea = Console.ReadLine();
            CANB = byte.Parse (linea);
            Console.WriteLine();
            APAGAR = (CANH * PRECIOH) + (CANP * PRECIOP) + (CANB * PRECIOB);
            Console.WriteLine("VALOR A PAGAR: " + APAGAR);
            Console.WriteLine("Pulse una Tecla.");Console.ReadLine();
        }
    }
}
```

4. GRABAR Y EJECUTAR



FUNCIONES BÁSICAS LIBRERÍA MATH

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int NUM1;
            string linea;
            long RESULT;
            Console.Write("DIGITE UN NÚMERO :");
            linea = Console.ReadLine();
            NUM1 = int.Parse(linea);
            RESULT = Math.Abs(NUM1);
            Console.WriteLine("VALOR ABSOLUTO : " + RESULT);
        }
    }
}
```

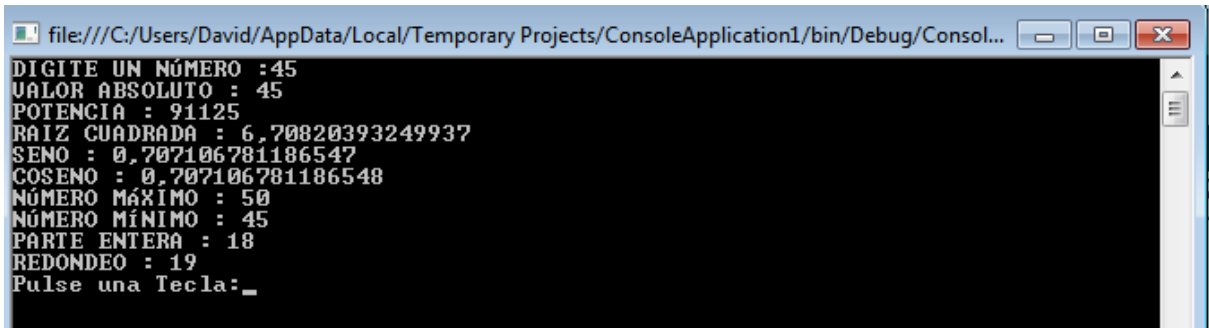


```

        Console.WriteLine("POTENCIA : " + Math.Pow(NUM1, 3));
        Console.WriteLine("RAIZ CUADRADA : " + Math.Sqrt(NUM1));
        Console.WriteLine("SENO : " + Math.Sin(NUM1 * Math.PI / 180));
        Console.WriteLine("COSENO : " + Math.Cos(NUM1 * Math.PI / 180));
        Console.WriteLine("NÚMERO MÁXIMO : " + Math.Max(NUM1, 50));
        Console.WriteLine("NÚMERO MÍNIMO : " + Math.Min(NUM1, 50));
        Console.WriteLine("PARTE ENTERA : " + Math.Truncate(18.78));
        Console.WriteLine("REDONDEO : " + Math.Round(18.78));
        Console.WriteLine("Pulse una Tecla:"); Console.ReadLine();
    }
}
}
}

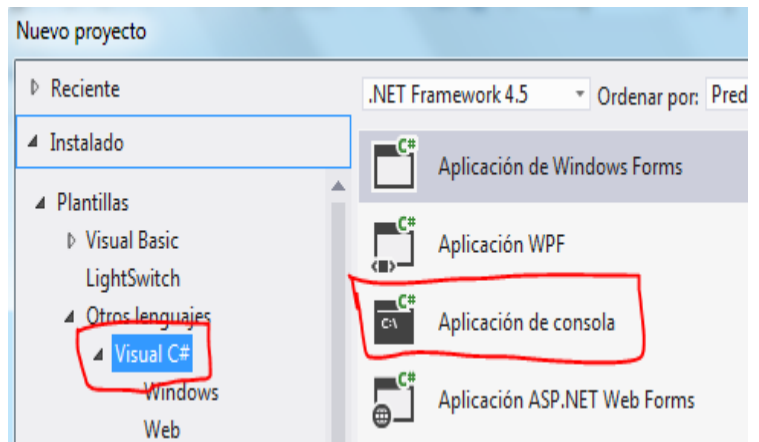
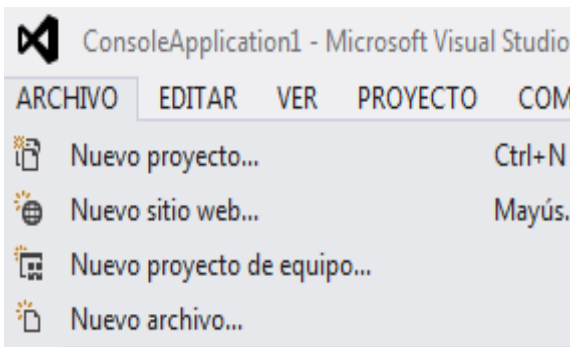
```

4. GRABAR Y EJECUTAR



FORMATOS DE SALIDA

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

```

using System;
using System.Collections.Generic;
using System.Linq;

```

```

using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            double BASE, ALTURA, RESULT;
            string linea;
            Console.Write("DIGITE LA BASE :"); linea = Console.ReadLine();
            BASE = double.Parse (linea);
            Console.Write("DIGITE LA ALTURA:"); linea = Console.ReadLine();
            ALTURA= double.Parse (linea);
            RESULT = (BASE * ALTURA) / 2;
            Console.WriteLine("AREA TRIANGULO :" + String.Format("{0:####.00}", RESULT));
            Console.WriteLine("AREA TRIANGULO :" + String.Format("{0:c}", RESULT));
            Console.WriteLine("AREA TRIANGULO :" + String.Format("{0:f}", RESULT));
            Console.WriteLine("AREA TRIANGULO :" + String.Format("{0:g}", RESULT));
            Console.WriteLine();
            Console.WriteLine("HOY ES: " + String.Format("Hoy es {0:F}", DateTime.Now));
            Console.WriteLine("HOY ES: " + String.Format("Hoy es {0:dddd}{0:dd/MM/yyyy}",
DateTime.Now));
            Console.Write("Pulse una Tecla:"); Console.ReadLine();
        }
    }
}

```

4. GRABAR Y EJECUTAR

```

file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/Consol...
DIGITE LA BASE :3
DIGITE LA ALTURA:6
AREA TRIANGULO :9,00
AREA TRIANGULO :$ 9,00
AREA TRIANGULO :9,00
AREA TRIANGULO :9
HOY ES: Hoy es domingo, 16 de marzo de 2014 18:30:19
HOY ES: Hoy es domingo16/03/2014
Pulse una Tecla:

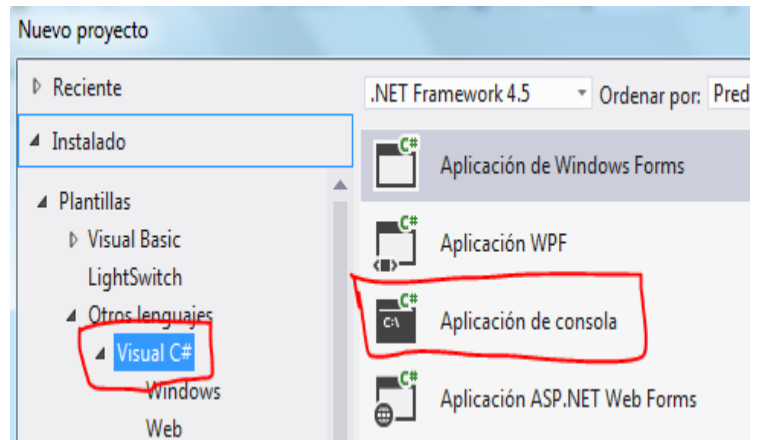
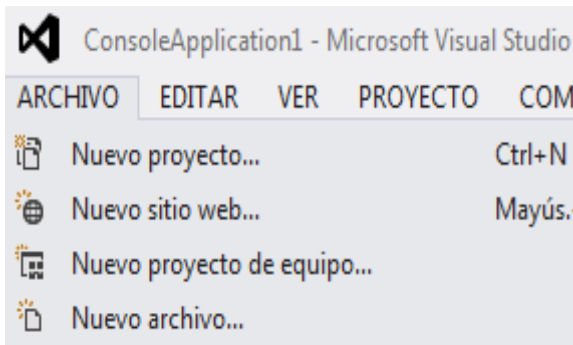
```

EJERCICIO PROPUESTO

Concepto:

El usuario debe ingresar dos números y el programa mostrará el resultado de la operación $(a+b)*(a-b)$

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola

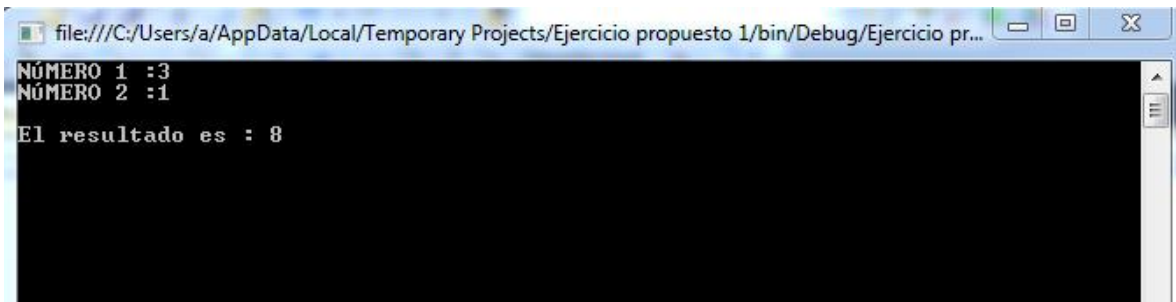


3. CÓDIGO

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

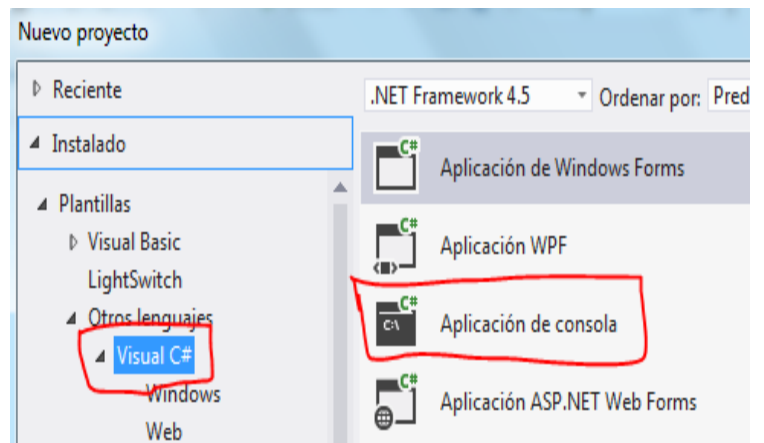
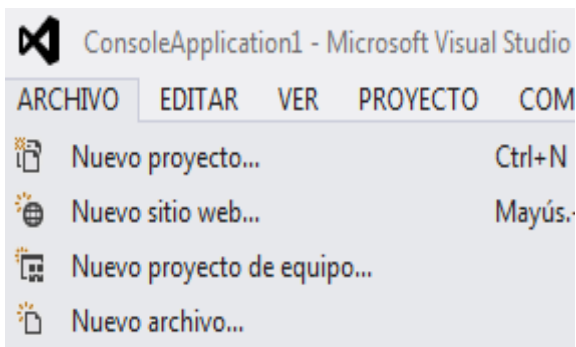
namespace Ejercicio_propuesto_1
{
    class Program
    {
        static void Main(string[] args)
        {
            int NUM1, NUM2;
            double RESULT;
            string linea;
            Console.WriteLine("NÚMERO 1 :"); linea = Console.ReadLine();
            NUM1 = int.Parse(linea);
            Console.WriteLine("NÚMERO 2 :"); linea = Console.ReadLine();
            NUM2 = int.Parse(linea);
            RESULT = (NUM1 + NUM2) * (NUM1 - NUM2);
            Console.WriteLine();
            Console.WriteLine("El resultado es : " +RESULT );
            Console.ReadLine();
        }
    }
}
```

4. GRABAR Y EJECUTAR



MAYOR DE DOS NÚMEROS

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int NUM1, NUM2;
            string linea;

```

```

Console.WriteLine("NÚMERO 1 :"); linea = Console.ReadLine();
NUM1 = int.Parse(linea);
Console.WriteLine("NÚMERO 2 :"); linea = Console.ReadLine();
NUM2 = int.Parse(linea);
if ((NUM1 > NUM2))
{
    Console.WriteLine("{0} ES MAYOR QUE {1}", NUM1, NUM2);
}
else
{
    if ((NUM1 == NUM2))
    {
        Console.WriteLine("{0} ES IGUAL A {1}", NUM1, NUM2);
    }
    else
    {
        Console.WriteLine("{0} ES MENOR QUE {1}", NUM1, NUM2);
    }
}
Console.WriteLine();
Console.WriteLine("OTRA MANERA");
string RESULT;
if (NUM1 > NUM2 )
{
    RESULT = "MAYOR";
}
else
if (NUM1 == NUM2 )
{
    RESULT = "IGUAL";
}
else
{
    RESULT = "MENOR";
}

Console.WriteLine("{0} ES {1} QUE {2}", NUM1, RESULT, NUM2);
Console.WriteLine("Pulse una Tecla:"); Console.ReadLine();
}
}
}

```

4. GRABAR Y EJECUTAR

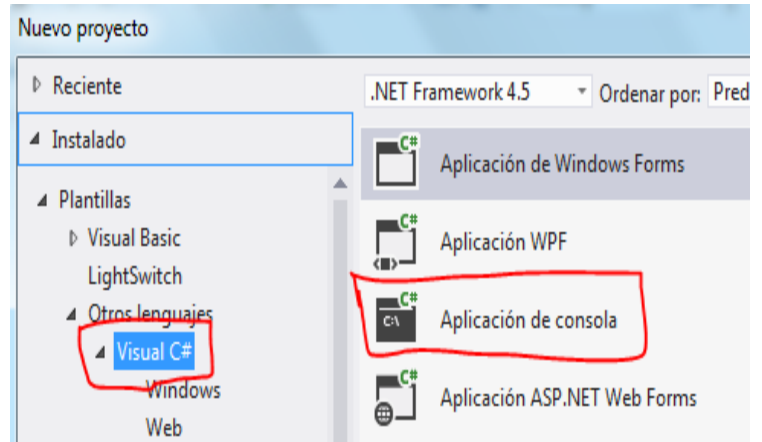
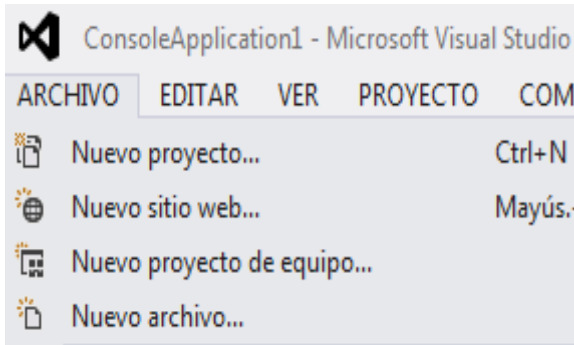
```

file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/Consol...
NÚMERO 1 :5
NÚMERO 2 :-5
5 ES MAYOR QUE -5
OTRA MANERA
5 ES MAYOR QUE -5
Pulse una Tecla:

```

MAYOR DE TRES NÚMEROS

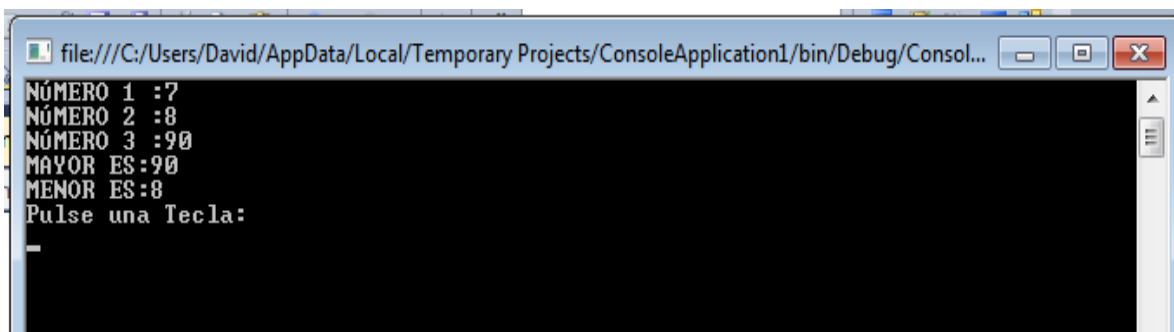
1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

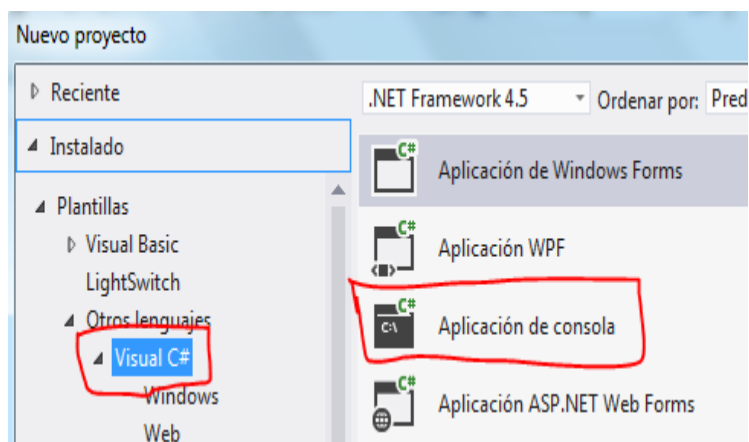
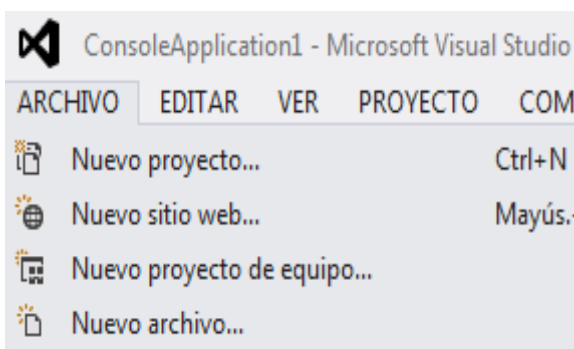
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            byte MAY, MEN, NUM1, NUM2, NUM3;
            string linea;
            Console.Write("NÚMERO 1 :"); linea = Console.ReadLine();
            NUM1 = byte.Parse(linea);
            Console.Write("NÚMERO 2 :"); linea = Console.ReadLine();
            NUM2 = byte.Parse(linea);
            Console.Write("NÚMERO 3 :"); linea = Console.ReadLine();
            NUM3 = byte.Parse(linea);
            MAY = NUM1; MEN = NUM1;
            if ((NUM2 > MAY)) MAY = NUM2;
            if ((NUM3 > MAY)) MAY = NUM3;
            if ((NUM2 > MEN)) MEN = NUM2;
            if ((NUM3 < MEN)) MEN = NUM3;
            Console.WriteLine("MAYOR ES:" + MAY);
            Console.WriteLine("MENOR ES:" + MEN);
            Console.WriteLine("Pulse una Tecla:"); Console.ReadLine();
        }
    }
}
```

4. GRABAR Y EJECUTAR



DESGLOSE DE BILLETES

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int CAN, C100, C50, C20, C10, C5;
            C100 = 0;
            C50 = 0;
            C20 = 0;
            C10 = 0;
        }
    }
}
```

```

C5 = 0;
string linea;
Console.Write("DIGITE UNA CANTIDAD :");linea = Console.ReadLine();
CAN = int.Parse (linea);
if( (CAN >= 100) )
{
    C100 = (CAN / 100);
    CAN = CAN - (C100 * 100);
}
if( (CAN >= 50) )
{
    C50 = (CAN / 50);
    CAN = CAN - (C50 * 50);
}
if( (CAN >= 20) )
{
    C20 = (CAN / 20);
    CAN = CAN - (C20 * 20);
}
if( (CAN >= 10) )
{
    C10 = (CAN / 10);
    CAN = CAN - (C10 * 10);
}
if( (CAN >= 5) )
{
    C5 = (CAN / 5);
    CAN = CAN - (C5 * 5);
}
Console.WriteLine("BILLETES DE A 100: " + C100);
Console.WriteLine("BILLETES DE A 50 : " + C50);
Console.WriteLine("BILLETES DE A 20 : " + C20);
Console.WriteLine("BILLETES DE A 10 : " + C10);
Console.WriteLine("BILLETES DE A 5 : " + C5);
Console.WriteLine("BILLETES DE A 1 : " + CAN);
Console.Write("Pulse una Tecla:"); Console.ReadLine();
}
}
}

```

4. GRABAR Y EJECUTAR

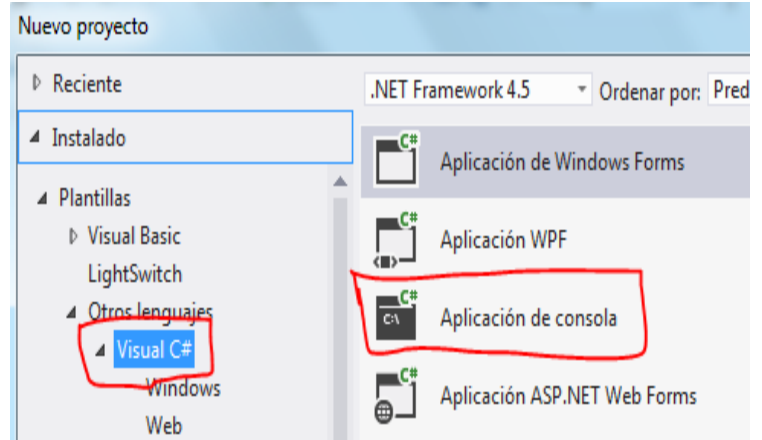
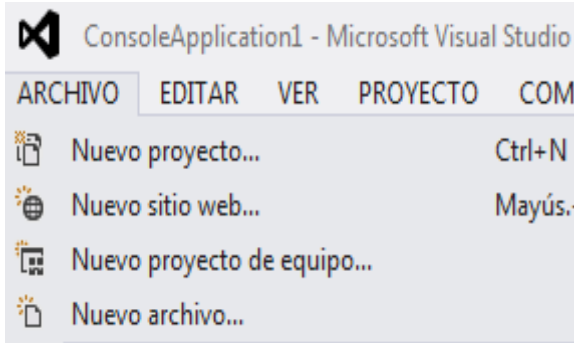
```

file:///C:/Users/David/AppData/Local/Temporary
DIGITE UNA CANTIDAD :45
BILLETES DE A 100: 0
BILLETES DE A 50 : 0
BILLETES DE A 20 : 2
BILLETES DE A 10 : 0
BILLETES DE A 5 : 1
BILLETES DE A 1 : 0
Pulse una Tecla: _

```


BONO DEL EMPLEADO POR HIJO

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

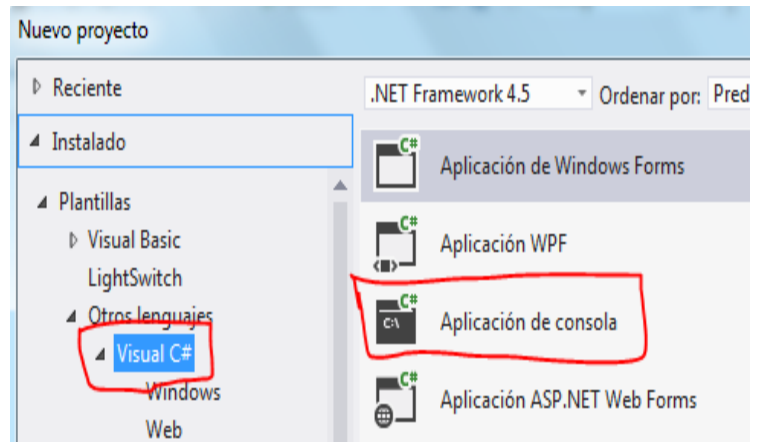
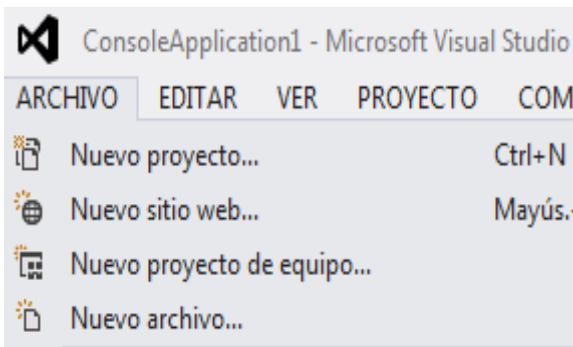
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication2
{
    class Program
    {
        static void Main(string[] args)
        {
            double SUELDO, ARECIBIR;
            int NHIJO, BONO;
            string NOM, linea;
            Console.WriteLine("NOMBRE EMPLEADO :"); NOM = Console.ReadLine();
            Console.WriteLine("SUELDO EMPLEADO :"); linea = Console.ReadLine();
            SUELDO = double.Parse(linea);
            Console.WriteLine("NÚMERO DE HIJOS :"); linea = Console.ReadLine();
            NHIJO = int.Parse(linea);
            if (NHIJO >= 3) { BONO = NHIJO * 10; } else { BONO = NHIJO * 20; }
            ARECIBIR = SUELDO + BONO;
            Console.WriteLine("RECIBE : " + String.Format("{0:c}", ARECIBIR));
            Console.WriteLine("Pulse una Tecla:"); Console.ReadLine();
        }
    }
}
```

4. GRABAR Y EJECUTAR

```
file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/Consol...
NOMBRE EMPLEADO :juan
SUELDO EMPLEADO :2000
NÚMERO DE HIJOS :30
RECIBE :$ 2.300,00
Pulse una Tecla:
```

NÚMERO INTERMEDIO

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola

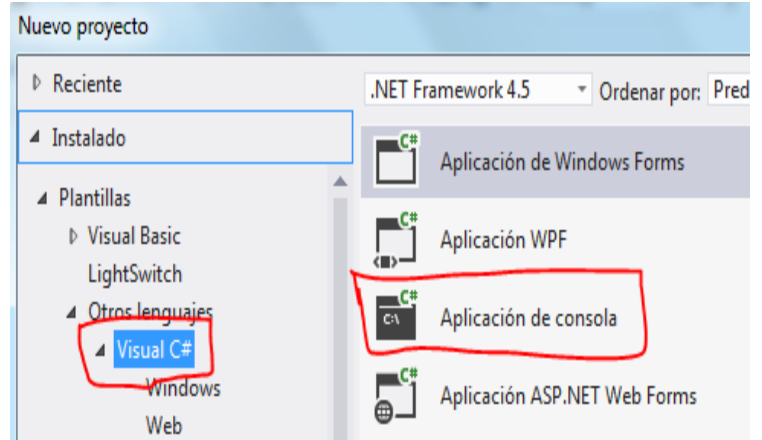
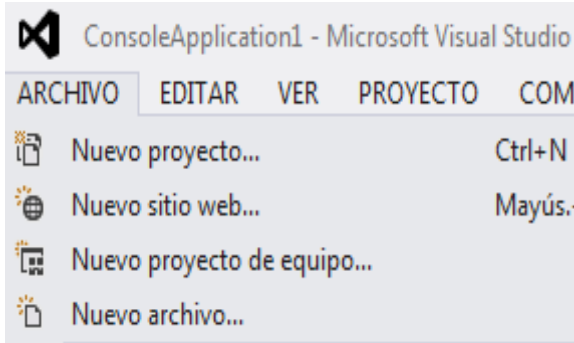


3. CÓDIGO

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int NUM1, NUM2, NUM3;
            string linea;
            Console.WriteLine("PRIMER NÚMERO :"); linea = Console.ReadLine();
            NUM1 = int.Parse(linea);
            Console.WriteLine("SEGUNDO NÚMERO :"); linea = Console.ReadLine();
            NUM2 = int.Parse(linea);
            Console.WriteLine("TERCER NÚMERO :"); linea = Console.ReadLine();
            NUM3 = int.Parse(linea);
            Console.WriteLine();
            Console.WriteLine("EL INTERMEDIO ES: ");
        }
    }
}
```


TARIFA TELEFÓNICA

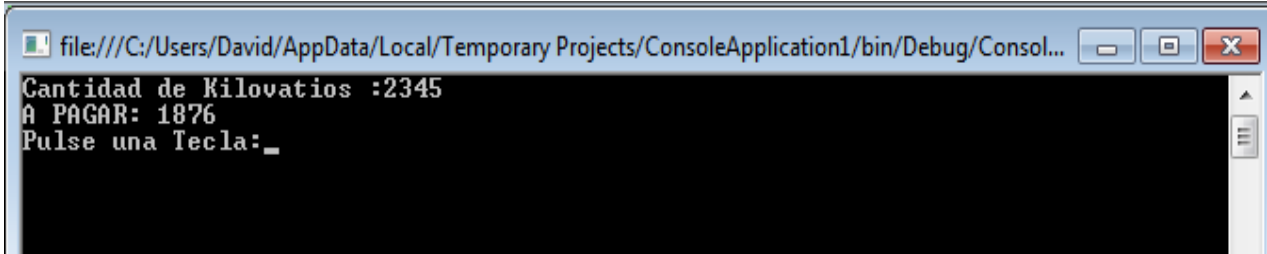
1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

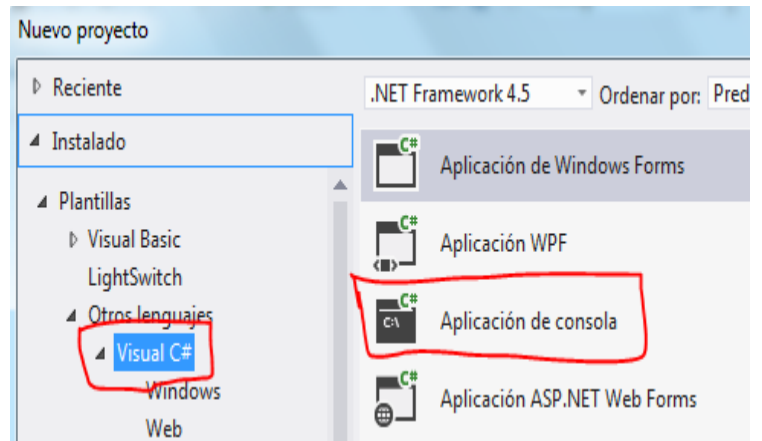
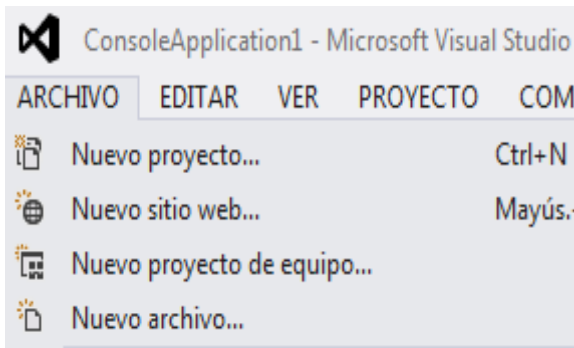
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int CANKV;
            double TOT, COSKV;
            COSKV = 0;
            string linea;
            Console.WriteLine("Cantidad de Kilovattios :"); linea = Console.ReadLine();
            CANKV = int.Parse(linea);
            if ((CANKV <= 1000)) COSKV = 0.14;
            if (((CANKV > 1000) & (CANKV <= 1800))) COSKV = 0.12;
            if ((CANKV > 1800)) COSKV = 0.8;
            TOT = CANKV * COSKV;
            Console.WriteLine("A PAGAR: " + TOT);
            Console.WriteLine("Pulse una Tecla:"); Console.ReadLine();
        }
    }
}
```

4. GRABAR Y EJECUTAR



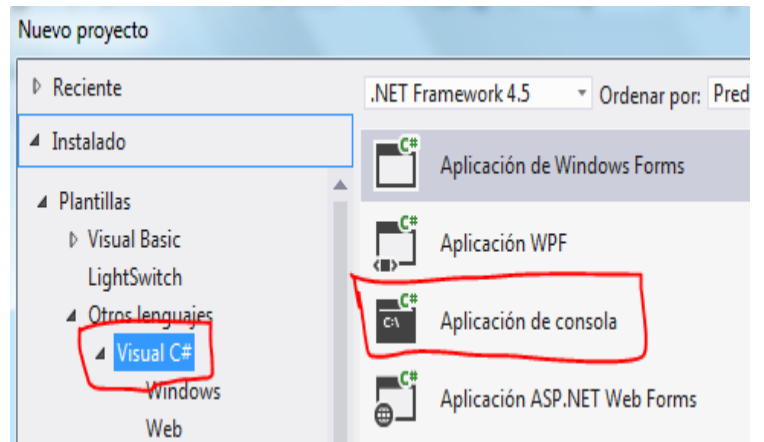
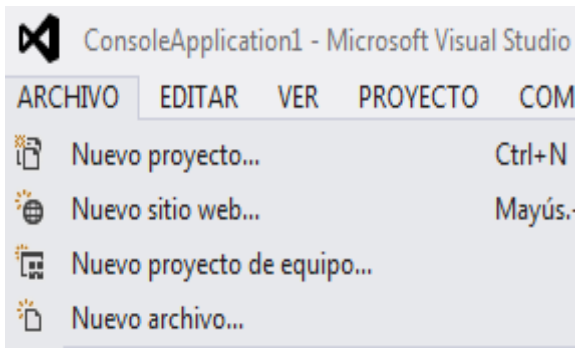
TRIÁNGULOS

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int LADO1, LADO2, LADO3;
            string linea;
            Console.WriteLine("DIGITE LADO 1 :"); linea = Console.ReadLine();
            LADO1 = int.Parse(linea);
            Console.WriteLine("DIGITE LADO 2 :"); linea = Console.ReadLine();
            LADO2 = int.Parse(linea);
            Console.WriteLine("DIGITE LADO 3 :"); linea = Console.ReadLine();
            LADO3 = int.Parse(linea);
        }
    }
}
```

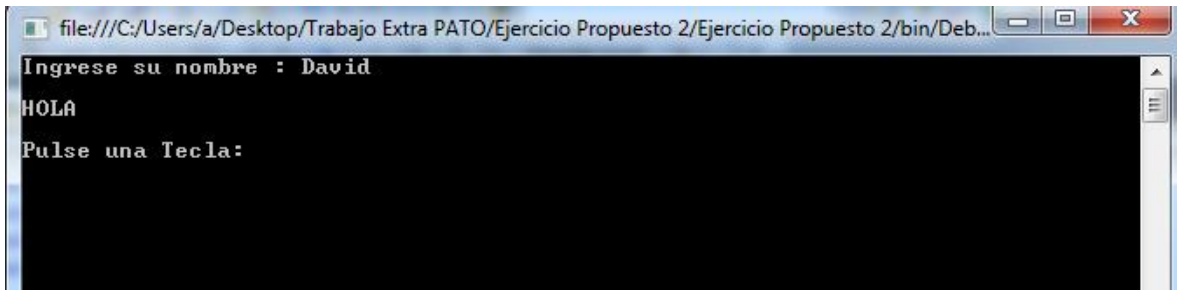



3. CÓDIGO

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

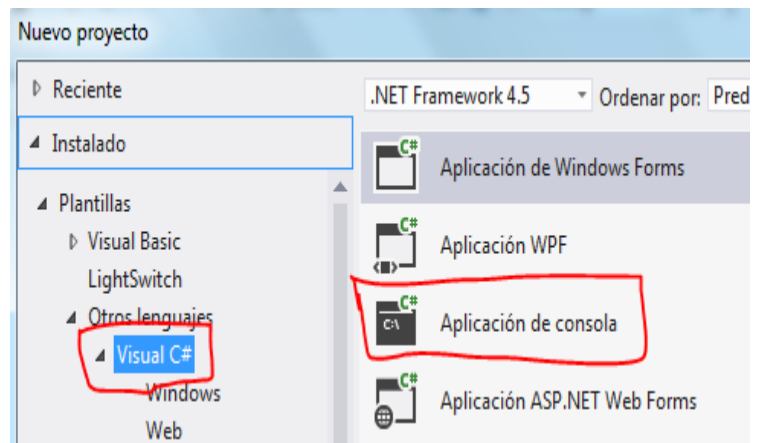
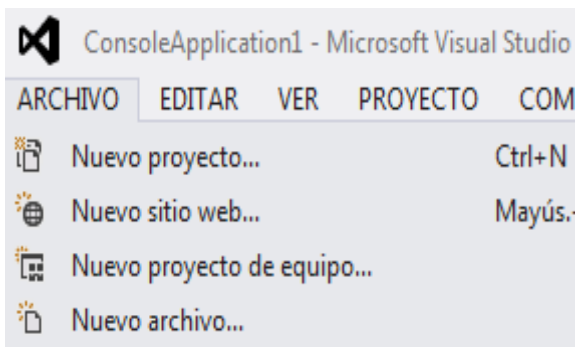
namespace Ejercicio_Propuesto_2
{
    class Program
    {
        static void Main(string[] args)
        {
            string NOMBRE;
            Console.WriteLine("Ingrese su nombre : "); NOMBRE = Console.ReadLine();
            if (NOMBRE == "David")
            {
                Console.WriteLine();
                Console.WriteLine("HOLA");
            }
            else
            {
                Console.WriteLine();
                Console.WriteLine("NO TE CONOZCO");
            }
            Console.WriteLine();
            Console.WriteLine("Pulse una Tecla."); Console.ReadLine();
        }
    }
}
```

4. GRABAR Y EJECUTAR



DÍA DE LA SEMANA

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace Dias_de_la_Semana
{
    class Program
    {
        static void Main(string[] args)
        {
            int num;
            string linea;
            Console.WriteLine();
            Console.WriteLine("DIAS DE LA SEMANA");
            Console.WriteLine();
            Console.Write("Ingrese un numero del 1 al 7 :");
            linea = Console.ReadLine();
            num = int.Parse(linea);
        }
    }
}
```

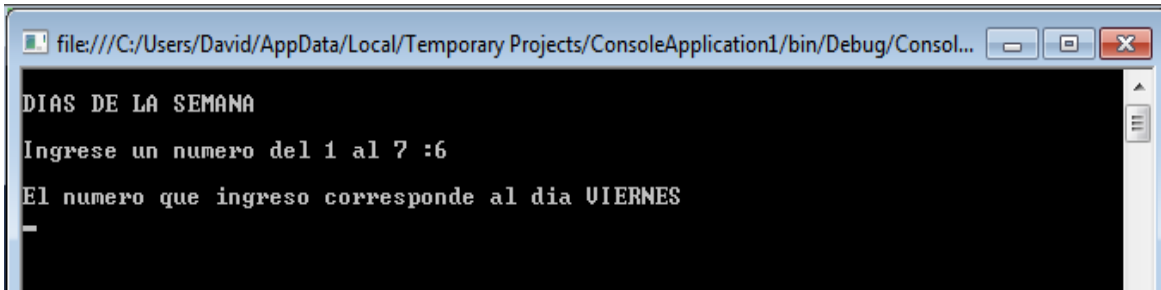


```

switch (num)
{
    case 1:
        Console.WriteLine();
        Console.WriteLine("El numero que ingreso corresponde al dia
DOMINGO");
        break;
    case 2:
        Console.WriteLine();
        Console.WriteLine("El numero que ingreso corresponde al dia LUNES");
        break;
    case 3:
        Console.WriteLine();
        Console.WriteLine("El numero que ingreso corresponde al dia MARTES");
        break;
    case 4:
        Console.WriteLine();
        Console.WriteLine("El numero que ingreso corresponde al dia
MIERCOLES");
        break;
    case 5:
        Console.WriteLine();
        Console.WriteLine("El numero que ingreso corresponde al dia JUEVES");
        break;
    case 6:
        Console.WriteLine();
        Console.WriteLine("El numero que ingreso corresponde al dia
VIERNES");
        break;
    case 7:
        Console.WriteLine();
        Console.WriteLine("El numero que ingreso corresponde al dia SABADO");
        break;
    default:
        Console.WriteLine();
        Console.WriteLine("El numero que ingreso esta fuera de rango");
        break;
}
Console.ReadKey();
}
}
}

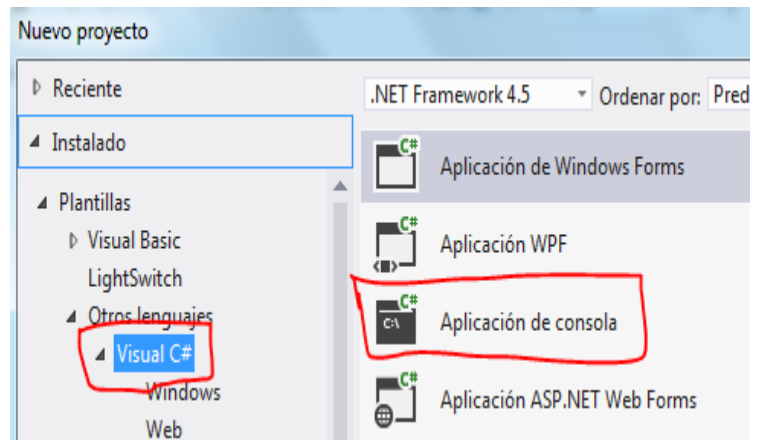
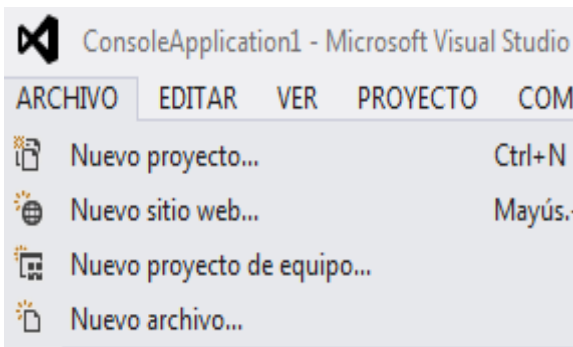
```

4. GRABAR Y EJECUTAR



ESTADO CIVIL

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            char ECIVIL;
            string linea;
            Console.Write("DIGITE C,S,V,D :"); linea = Console.ReadLine();
            ECIVIL = char.Parse(linea);
            switch (ECIVIL)
            {
                case 'C': ; Console.WriteLine("CASADO");
                    break;
                case 'S': ; Console.WriteLine("SOLTERO");
```

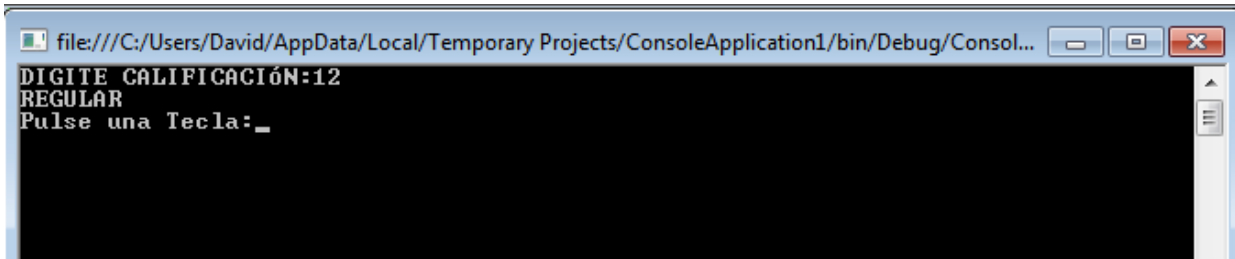


```

using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int NOTA;
            string linea;
            Console.Write("DIGITE CALIFICACIÓN:");linea = Console.ReadLine();
            NOTA= byte.Parse (linea);
            switch(NOTA)
            {
                case 19:
                case 20: ; Console.WriteLine("SOBRESALIENTE");
                    break;
                case 16:
                case 17:
                case 18:;Console.WriteLine("MUY BUENA");
                    break;
                case 14:
                case 15;; Console.WriteLine("BUENA");
                    break;
                case 12:
                case 13;;Console.WriteLine("REGULAR");
                    break;
                case 1:
                case 2:
                case 3:
                case 4:
                case 5:
                case 6:
                case 7:
                case 8:
                case 9:
                case 10:
                case 11:
                    Console.WriteLine("INSUFICIENTE");
                    break;
                default:
                    Console.WriteLine("FUERA DE RANGO");
                    break;
            }
            Console.Write("Pulse una Tecla:");Console.ReadLine();
        }
    }
}

```

4. GRABAR Y EJECUTAR

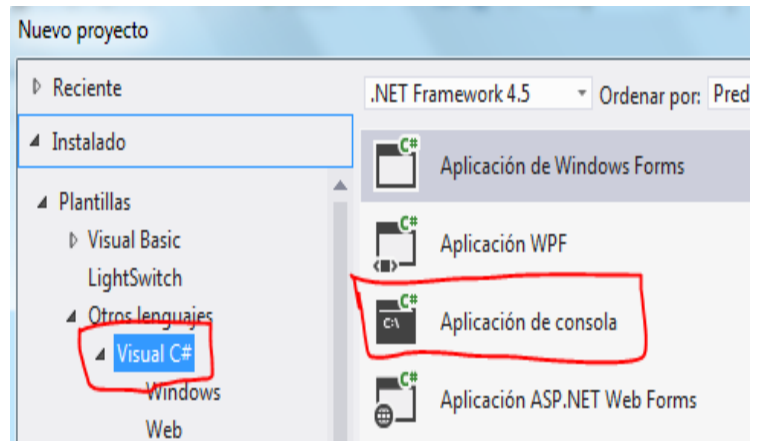
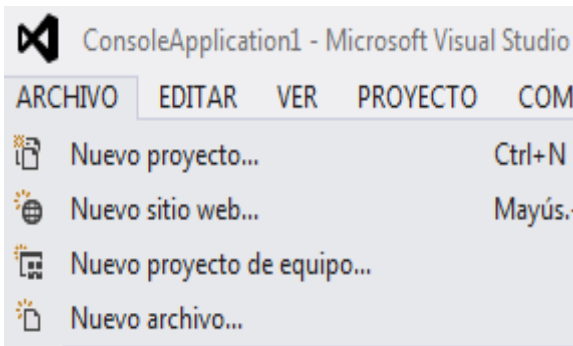


EJERCICIO PROPUESTO

Concepto:

Crear un programa que lea una letra tecleada por el usuario y diga si se trata de una vocal, una cifra numérica o una consonante (pista: habrá que usar un dato de tipo "char").

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

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

namespace Ejercicio_Propuesto_3
{
    class Program
    {
        static void Main(string[] args)
        {
            char TECLA;
        }
    }
}
```

```

string linea;
Console.Write("DIGITE UNA SOLA TECLA");
Console.WriteLine();
linea = Console.ReadLine();
TECLA = char.Parse(linea);
switch (TECLA)
{
    case '1':
    case '2':
    case '3':
    case '4':
    case '5':
    case '6':
    case '7':
    case '8':
    case '9': ; Console.WriteLine("ES UNA CIFRA NUMERICA");
        break;
    case 'a':
    case 'e':
    case 'i':
    case 'o':
    case 'u': ; Console.WriteLine("ES UNA VOCAL");
        break;
    default:
        Console.WriteLine("ES UNA CONSONANTE");
        break;
}
Console.Write("Pulse una Tecla:"); Console.ReadLine();
}
}
}
}
}

```

4. GRABAR Y EJECUTAR

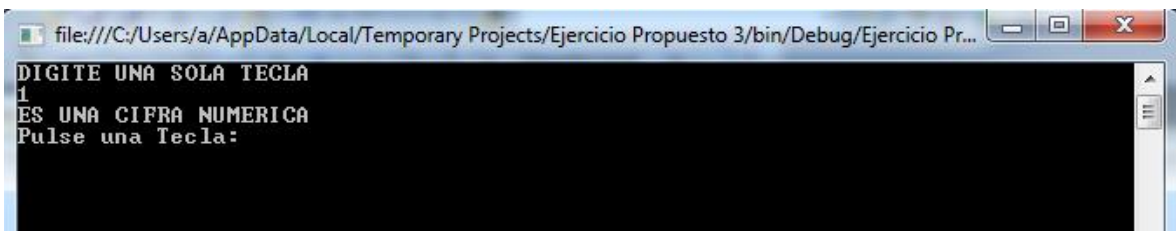
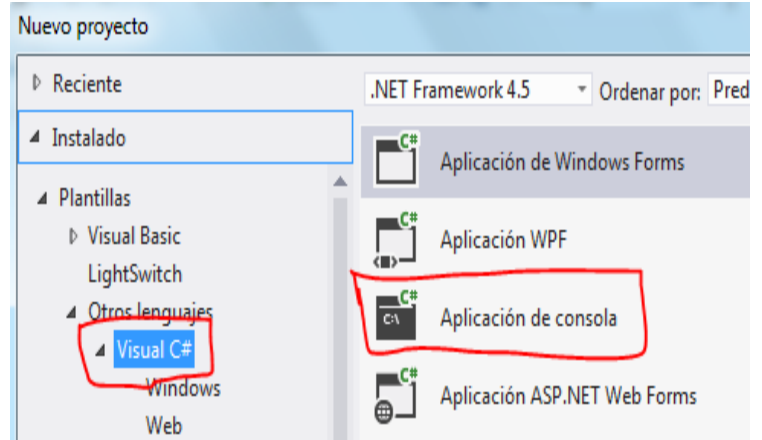
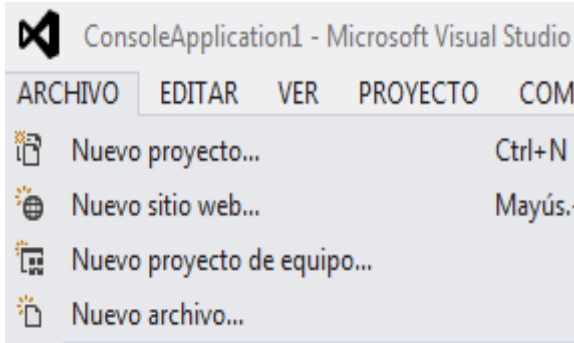


TABLA DE MULTIPLICAR

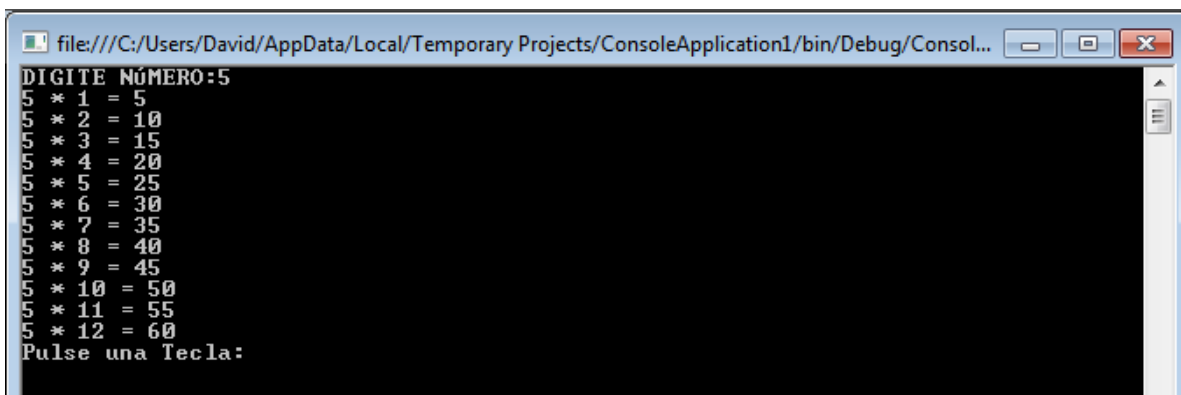
1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

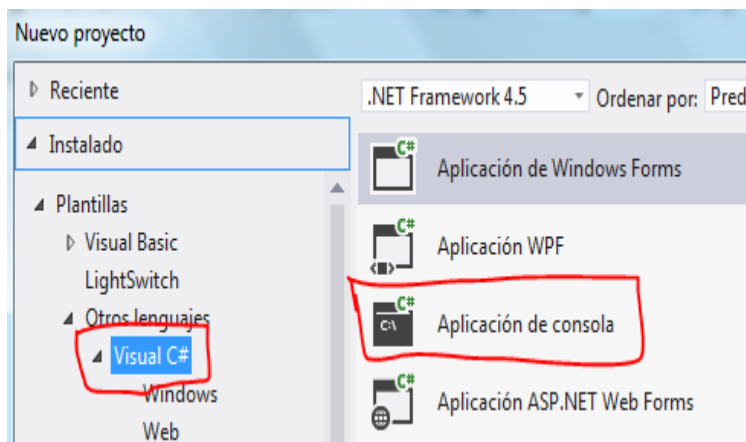
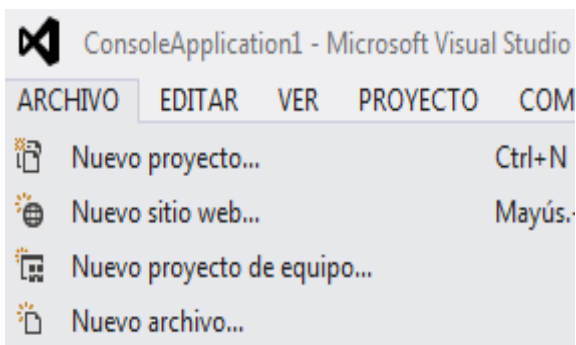
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace TABLA_DE_MULTIPLICAR
{
    class Program
    {
        static void Main(string[] args)
        {
            byte NUM, I;
            int RESULT;
            string linea;
            Console.Write("DIGITE NÚMERO:"); linea = Console.ReadLine();
            NUM = byte.Parse(linea);
            for (I = 1; I <= 12; I++)
            {
                RESULT = NUM * I;
                Console.WriteLine("{0} * {1} = {2}", NUM, I, RESULT);
            }
            Console.Write("Pulse una Tecla:"); Console.ReadLine();
        }
    }
}
```

4. GRABAR Y EJECUTAR



PRESUPUESTO ANUAL EN AREAS HOSPITAL

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola

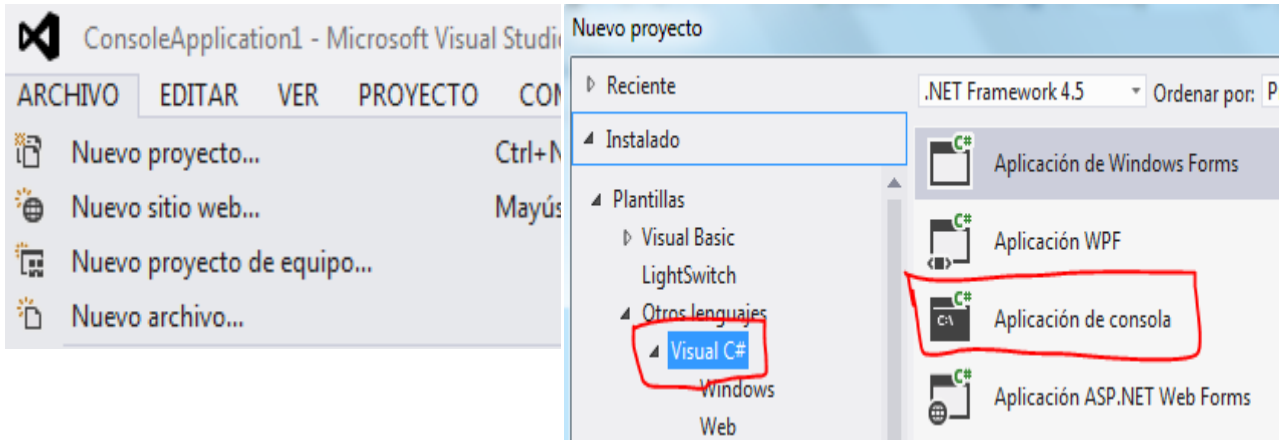


3. CÓDIGO

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace PRESUPUESTO_ANUAL_EN_AREAS_HOSPITAL
{
    class Program
    {
        static void Main(string[] args)
        {
            double CANP, TOT, PORC;
            string AREA, linea;
            byte CAN, I;
        }
    }
}
```


SUMA DE N NÚMEROS PARES E IMPARES

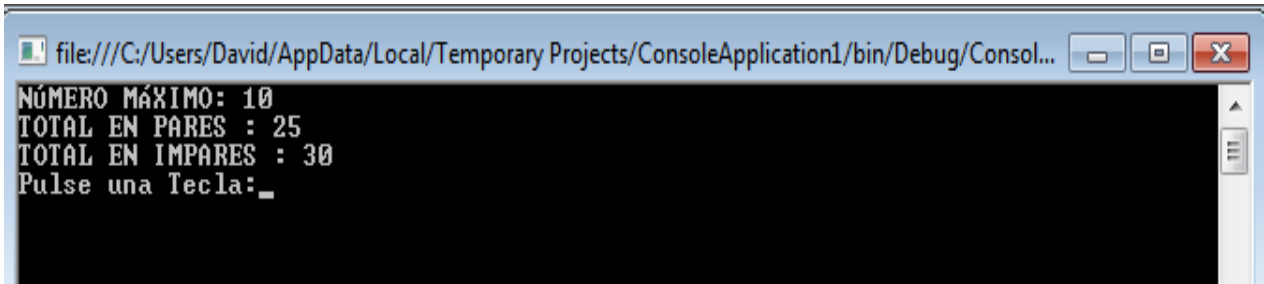
1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

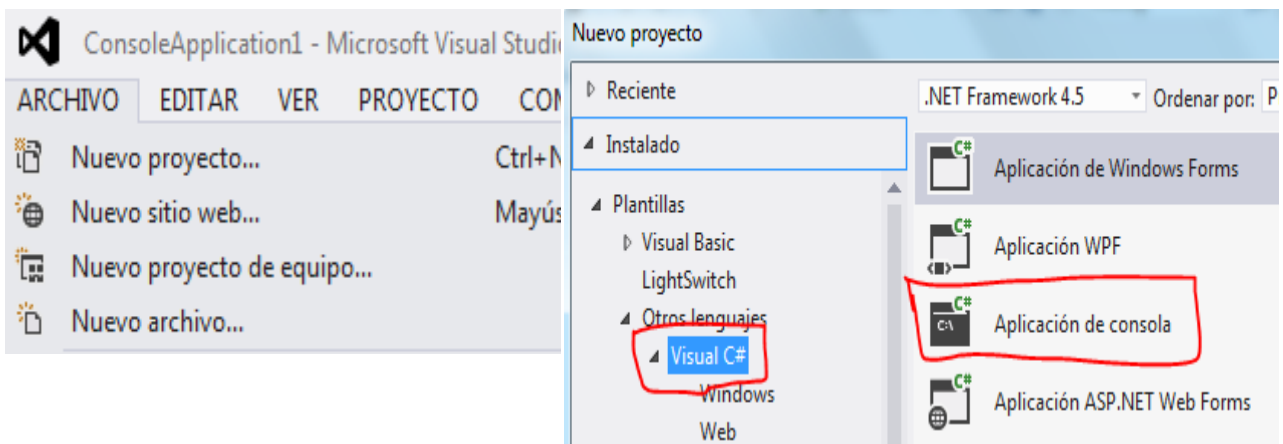
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace SUMA_DE_N_NÚMEROS_PARES_E_IMPARES
{
    class Program
    {
        static void Main(string[] args)
        {
            byte NUM, I;
            int SUMP = 0;
            int SUMI = 0;
            string linea;
            Console.WriteLine("NÚMERO MÁXIMO: "); linea = Console.ReadLine();
            NUM = byte.Parse(linea);
            for (I = 1; I <= NUM; I += 2)
            {
                SUMP = SUMP + I;
            }
            for (I = 2; I <= NUM; I += 2)
            {
                SUMI = SUMI + I;
            }
            Console.WriteLine("TOTAL EN PARES : " + SUMP);
            Console.WriteLine("TOTAL EN IMPARES : " + SUMI);
            Console.WriteLine("Pulse una Tecla:"); Console.ReadLine();
        }
    }
}
```

4. GRABAR Y EJECUTAR



TABLAS DE MULTIPLICAR

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola

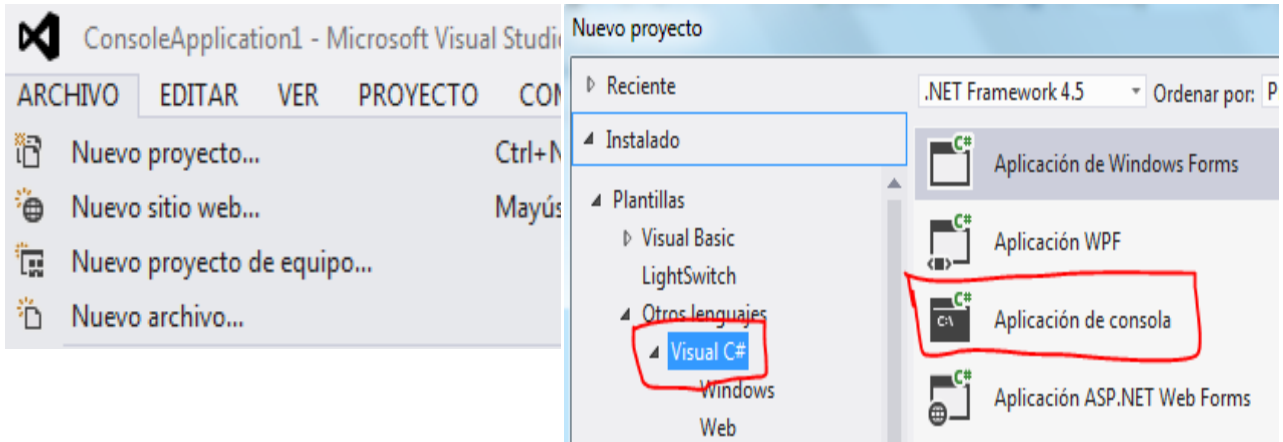


3. CÓDIGO

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace TABLAS_DE_MULTIPLICAR
{
    class Program
    {
        static void Main(string[] args)
        {
            int NUM, RESULT, T, I;
            string linea;
            Console.WriteLine("CUANTAS TABLAS: "); linea = Console.ReadLine();
            NUM = int.Parse(linea);
            for (T = 1; T <= NUM; T++)
            {
                for (I = 10; I >= 1; I--)
                {
                    RESULT = T * I;
                }
            }
        }
    }
}
```


SUMA DE N NÚMEROS

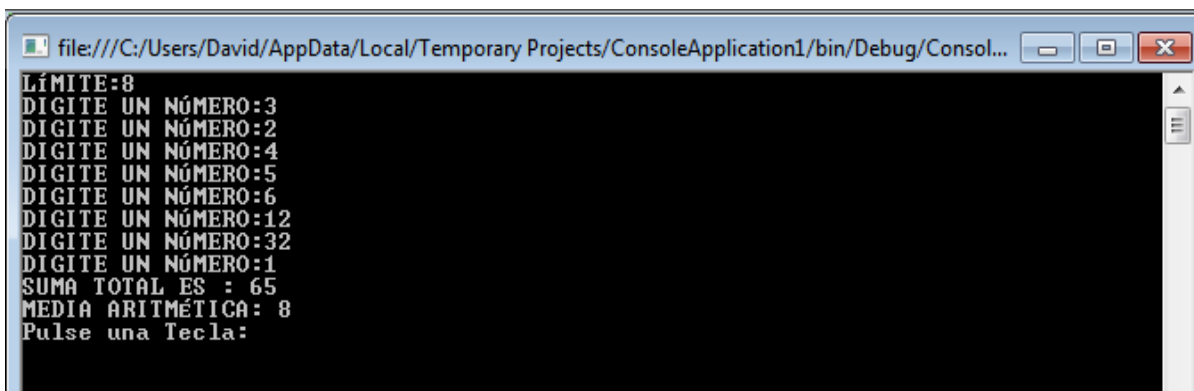
1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

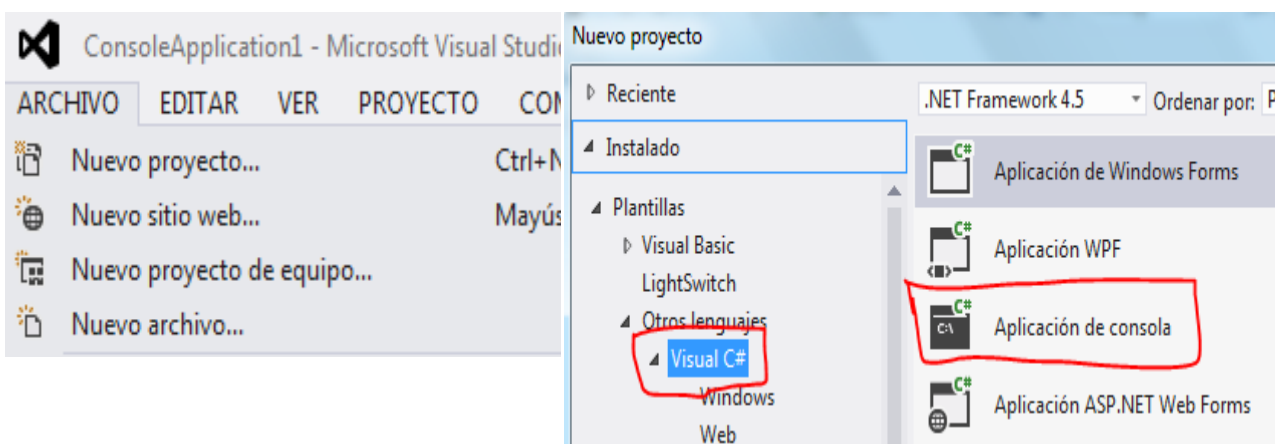
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace SUMA_DE_N_NÚMEROS
{
    class Program
    {
        static void Main(string[] args)
        {
            byte CAN, K;
            int NUM;
            int SUM = 0;
            string linea;
            Console.WriteLine("LÍMITE:"); linea = Console.ReadLine();
            CAN = byte.Parse(linea);
            for (K = 1; K <= CAN; K++)
            {
                Console.WriteLine("DIGITE UN NÚMERO:"); linea = Console.ReadLine();
                NUM=int.Parse (linea);
                SUM += NUM;
            }
            Console.WriteLine("SUMA TOTAL ES : " + SUM);
            Console.WriteLine("MEDIA ARITMÉTICA: " + SUM / CAN);
            Console.WriteLine("Pulse una Tecla:"); Console.ReadLine();
        }
    }
}
```

4. GRABAR Y EJECUTAR



MAYOR Y MENOR DE N NÚMEROS

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            byte CAN, K;
            int MAY, MEN, NUM;
        }
    }
}
```

```

string linea;
Console.WriteLine("LÍMITE:"); linea = Console.ReadLine();
CAN = byte.Parse(linea);
Console.WriteLine("DIGITE UN NÚMERO:"); linea = Console.ReadLine();
MAY = int.Parse(linea);
MEN = MAY;
for (K = 2; K <= CAN; K++)
{
    Console.WriteLine("DIGITE UN NÚMERO:"); linea = Console.ReadLine();
    NUM = int.Parse(linea);
    if (NUM > MAY) MAY = NUM;
    if (NUM < MEN) MEN = NUM;
}
Console.WriteLine("EL MAYOR ES : " + MAY);
Console.WriteLine("EL MENOR ES : " + MEN);
Console.WriteLine("DIFERENCIA ES : " + (MAY - MEN));
Console.WriteLine("Pulse una Tecla:"); Console.ReadLine();
}
}
}

```

4. GRABAR Y EJECUTAR

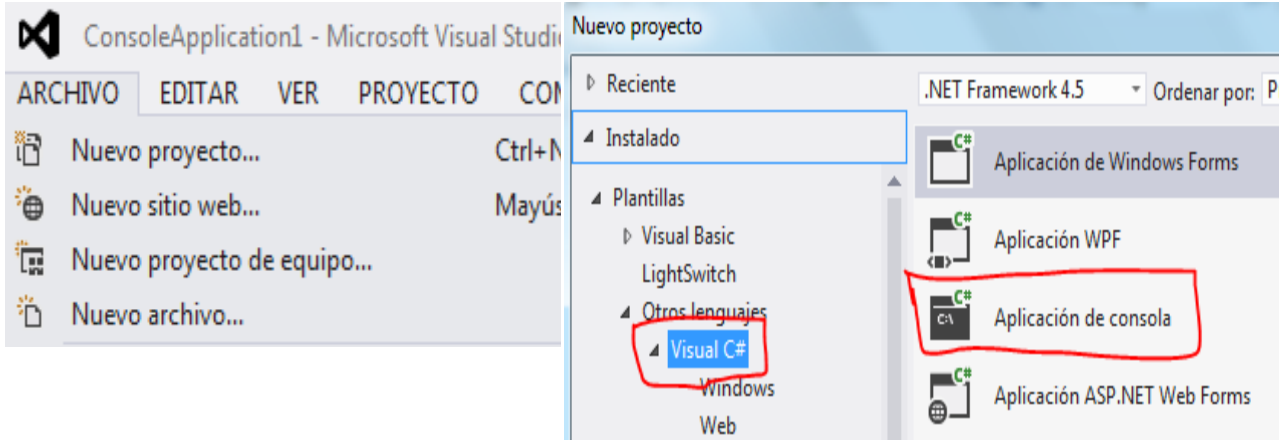
```

file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/Consol...
LÍMITE:4
DIGITE UN NÚMERO:21
DIGITE UN NÚMERO:32
DIGITE UN NÚMERO:1
DIGITE UN NÚMERO:234
EL MAYOR ES : 234
EL MENOR ES : 1
DIFERENCIA ES : 233
Pulse una Tecla:_

```

SERIE DE FIBONACCI

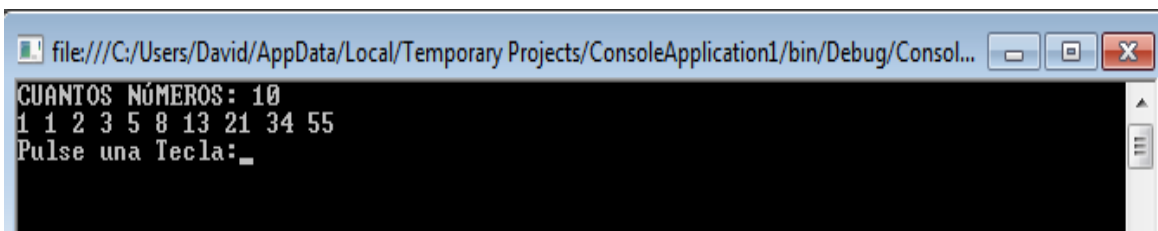
1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

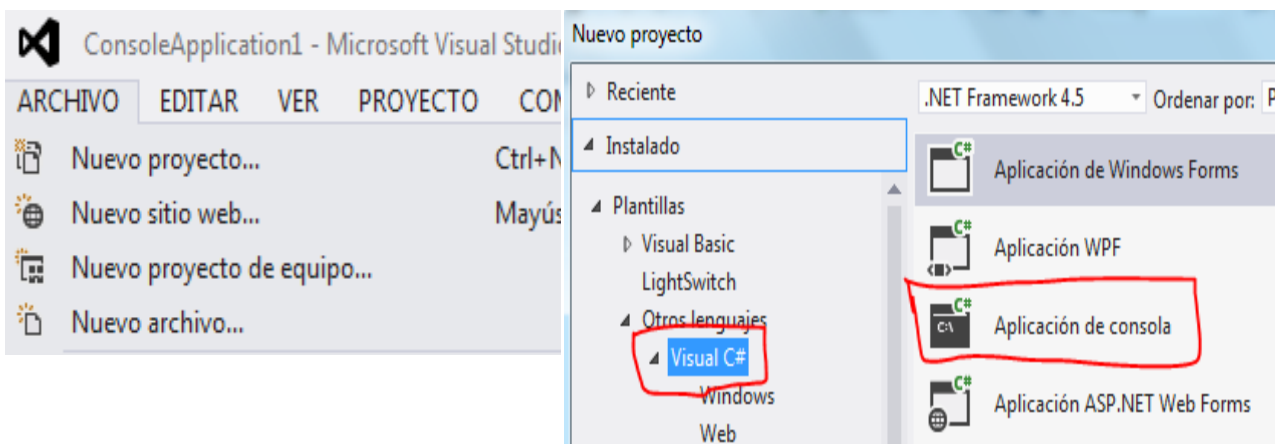
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace SERIE_DE_FIBONACCI
{
    class Program
    {
        static void Main(string[] args)
        {
            byte CAN, K;
            int A, B, C;
            string linea;
            Console.Write("CUANTOS NÚMEROS: "); linea = Console.ReadLine();
            CAN = byte.Parse(linea);
            A = 1; B = 1;
            Console.Write(A + " " + B + " ");
            for(K = 3; K <= CAN; K++){
                C = A + B;
                Console.Write(C + " ");
                //INTERCAMBIO DE VALORES
                A = B;
                B = C;
            }
            Console.WriteLine();
            Console.Write("Pulse una Tecla:"); Console.ReadLine();
        }
    }
}
```


4. GRABAR Y EJECUTAR



CALIFICACIONES DE UN GRUPO DE ESTUDIANTES

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola

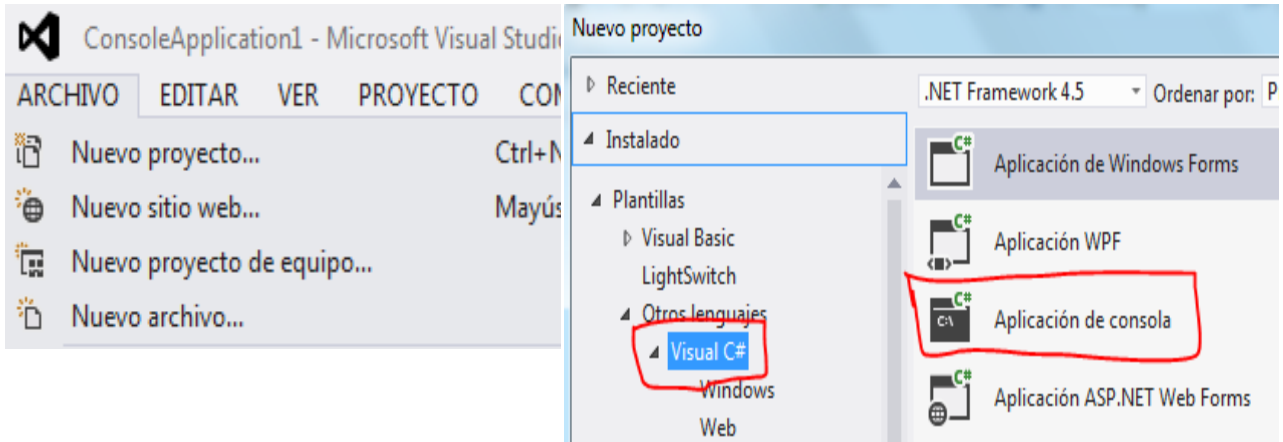


3. CÓDIGO

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace CALIFICACIONES_DE_UN_GRUPO_DE_ESTUDIANTES
{
    class Program
    {
        static void Main(string[] args)
        {
            byte CAN, K;
            double NOTA1, NOTA2, PROM, SUM;
            string NOM, linea;
            Console.WriteLine("CUANTOS ESTUDIANTES: "); linea = Console.ReadLine();
            CAN = byte.Parse(linea);
            SUM = 0;
            for (K = 1; K <= CAN; K++)
            {
```


NÚMEROS ALEATORIOS Y CARACTERES ASCII

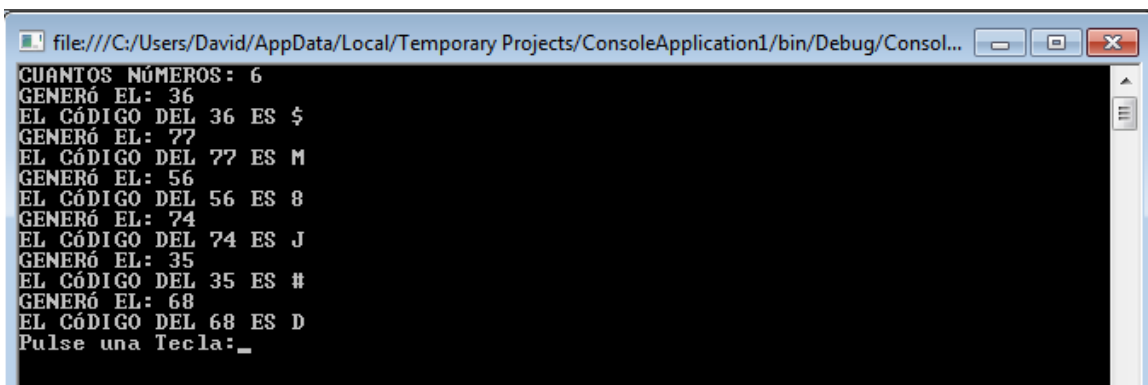
1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace NÚMEROS_ALEATORIOS_Y_CARACTERES_ASCII
{
    class Program
    {
        static void Main(string[] args)
        {
            byte CAN, K;
            int NUM;
            string linea;
            Console.Write("CUANTOS NÚMEROS: "); linea = Console.ReadLine();
            CAN = byte.Parse(linea);
            //INICIALIZA EL GENERADOR DE ALEATORIOS
            Random ran = new Random();
            for(K = 1; K <= CAN; K++){
                NUM = Convert.ToInt32(ran.Next(100));
                Console.WriteLine("GENERÓ EL: " + NUM);
                if( ((NUM > 0) & (NUM < 256)) ){
                    Console.WriteLine("EL CÓDIGO DEL " + NUM + " ES " +
                    Convert.ToString((char)NUM));
                }
            }
            Console.Write("Pulse una Tecla:"); Console.ReadLine();
        }
    }
}
```

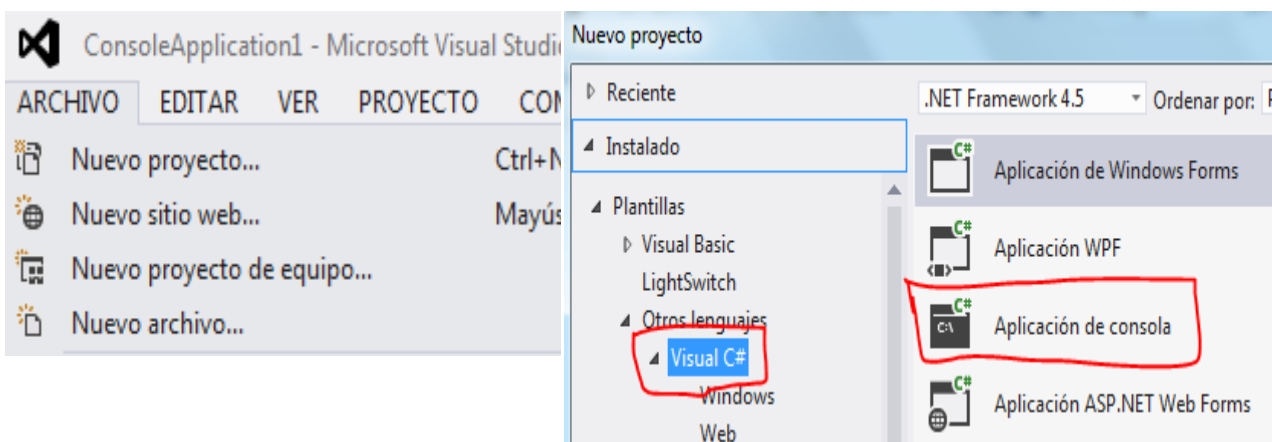
4. GRABAR Y EJECUTAR



```
file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/Consol...
CUANTOS NÚMEROS : 6
GENERó EL: 36
EL CódIGO DEL 36 ES $
GENERó EL: 77
EL CódIGO DEL 77 ES M
GENERó EL: 56
EL CódIGO DEL 56 ES 8
GENERó EL: 74
EL CódIGO DEL 74 ES J
GENERó EL: 35
EL CódIGO DEL 35 ES #
GENERó EL: 68
EL CódIGO DEL 68 ES D
Pulse una Tecla: _
```

FUNCIONES DE CADENA

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace FUNCIONES_DE_CADENA
{
    class Program
    {
        static void Main(string[] args)
        {
            string NOM;
            byte K;
            Console.Write("DIGITE SU NOMBRE : "); NOM = Console.ReadLine();
```



```

file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/Consol...
EN MINUSCULAS : jonatan
EN MAYÚSCULAS : JONATAN
REMOVER 4 LETRAS : JON
EXTRAER 4 LETRAS : ATAN
IZQUIERDA 4 LETRAS : JONA
DERECHA 4 LETRAS : ATAN
Pulse una Tecla:

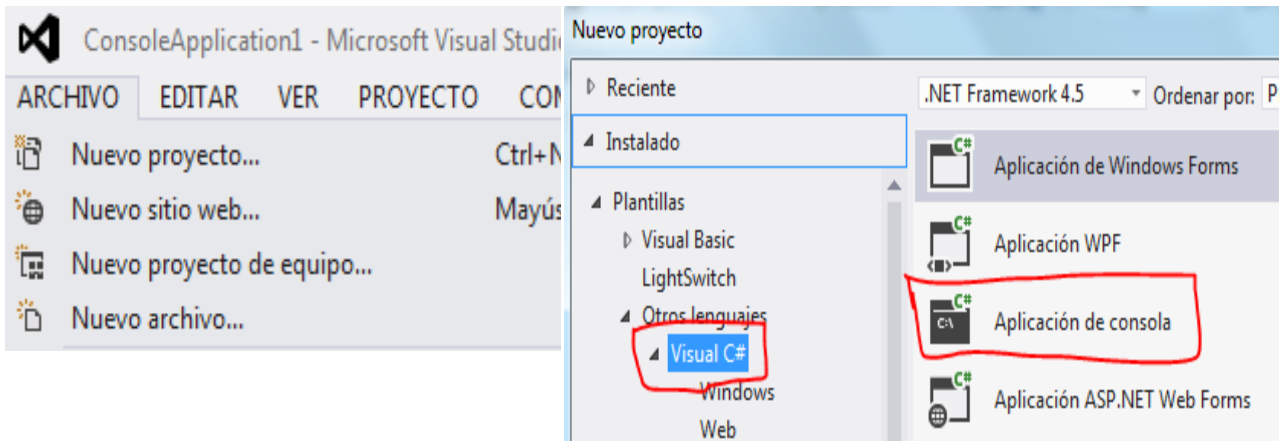
DESDE LA IZQUIERDA
J
JO
JON
JONA
JONAT
JONATA
JONATAN

DESDE LA DERECHA
N
AN
TAN
ATAN
NATAN
ONATAN
JONATAN
Pulse una Tecla:

```

SIMULACIÓN DE UN RELOJ DIGITAL

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace SIMULACIÓN_DE_UN_RELOJ_DIGITAL
{
    class Program
    {
        static void Main(string[] args)

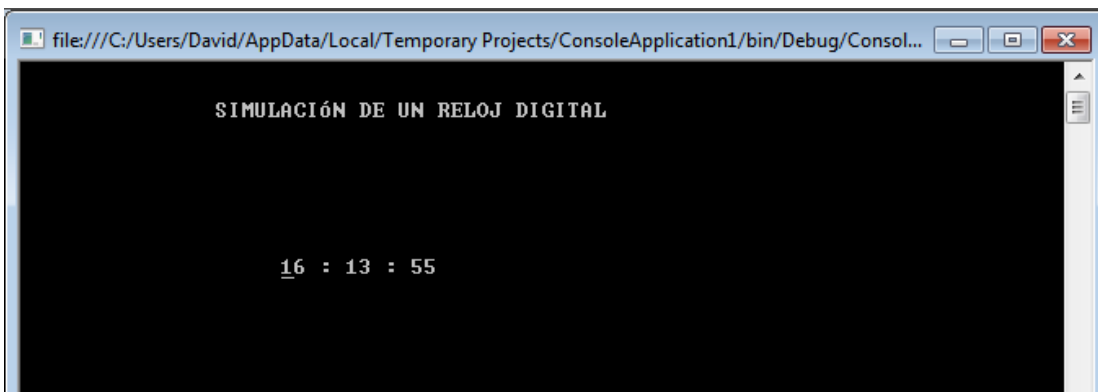
```

```

{
    byte H, M, S;
    Console.SetCursorPosition(15, 2);
    Console.WriteLine("SIMULACIÓN DE UN RELOJ DIGITAL");
    for (H = 0; H <= 24; H++)
    {
        for (M = 0; M <= 59; M++)
        {
            for (S = 0; S <= 59; S++)
            {
                Console.SetCursorPosition(20, 10);
                Console.WriteLine("{0} : {1} : {2}", H, M, S);
            }
        }
    }
    Console.SetCursorPosition(25, 15);
    Console.WriteLine("Pulse una Tecla:"); Console.ReadLine();
}
}
}

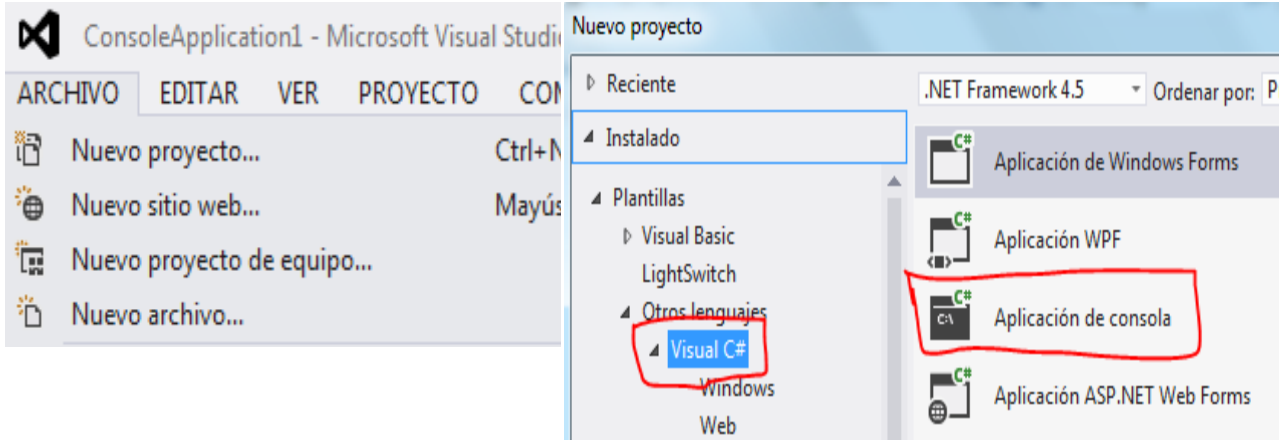
```

4. GRABAR Y EJECUTAR



CANTIDAD DE VOCALES CERRADAS

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace CANTIDAD_DE_VOCALES_CERRADAS
{
    class CANTIDAD_DE_VOCALES_CERRADAS
    {
        static void Main(string[] args)
        {
            string NOM;
            int K, SUM;
            string LE;
            Console.WriteLine("DIGITE UNA FRASE : "); NOM = Console.ReadLine();
            SUM = 0;
            NOM = NOM.ToUpper();
            Console.WriteLine(NOM);
            for (K = 1; K < NOM.Length; K++)
            {
                LE = Mid(NOM, K, 1);
                if (LE.Contains("U") || LE.Contains("I"))
                {
                    SUM = SUM + 1;
                }
            }
            Console.WriteLine("EXISTEN {0} VOCALES CERRADAS", SUM);
            Console.WriteLine("Pulse una Tecla:"); Console.ReadLine();
        }
        public static string Mid(string param, int startIndex, int length)
        {
            string result = param.Substring(startIndex, length);
            return result;
        }
    }
}
```

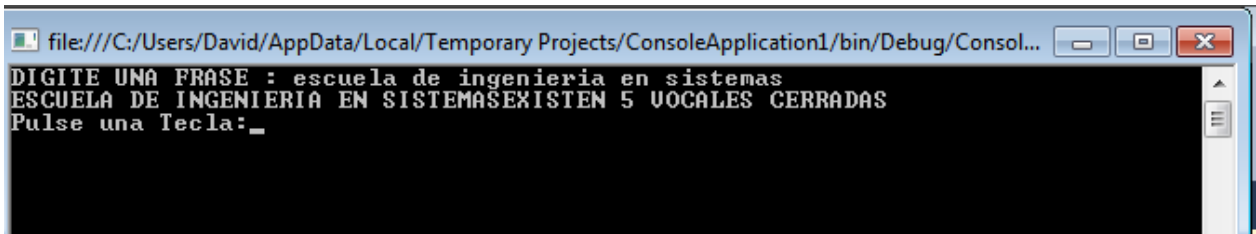


```

    }
    public static string Mid(string param, int startIndex)
    {
        string result = param.Substring(startIndex);
        return result;
    }
}

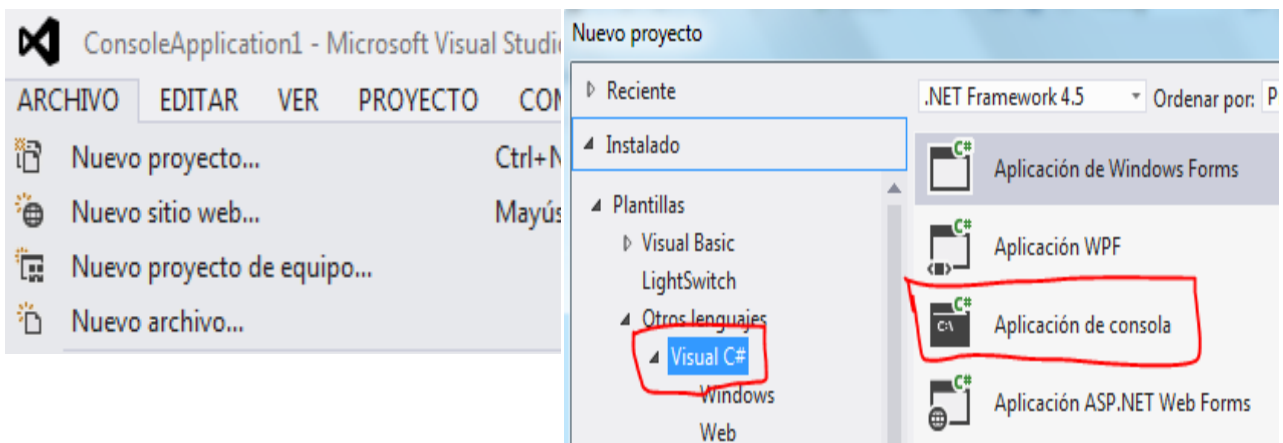
```

4. GRABAR Y EJECUTAR



ESTADISTICA POR VOCAL

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

```

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 ;
    int K = 0;
    int A = 0;
    int E = 0;
    int I = 0;
    int O = 0;
    int U = 0;
    string LE ;
    Console.Write("DIGITE UNA FRASE : ");
    NOM = Console.ReadLine();
    A = 0;
    E = 0;
    I = 0;
    O = 0;
    U = 0;
    for (K = 1; K < NOM.Length; K++)
    {
        NOM = NOM.ToUpper();
        LE = Mid(NOM, K, 1);
        switch (LE)
        {
            case "A":
                A = A + 1;
                break;
            case "E":
                E = E + 1;
                break;
            case "I":
                I = I + 1;
                break;
            case "O":
                O = O + 1;
                break;
            case "U":
                U = U + 1;
                break;
        }
    }
    Console.WriteLine("EXISTEN {0} VOCALES A", A);
    Console.WriteLine("EXISTEN {0} VOCALES E", E);
    Console.WriteLine("EXISTEN {0} VOCALES I", I);
    Console.WriteLine("EXISTEN {0} VOCALES O", O);
    Console.WriteLine("EXISTEN {0} VOCALES U", U);
    Console.Write("Pulse una Tecla:");
    Console.ReadLine();
}
}
}

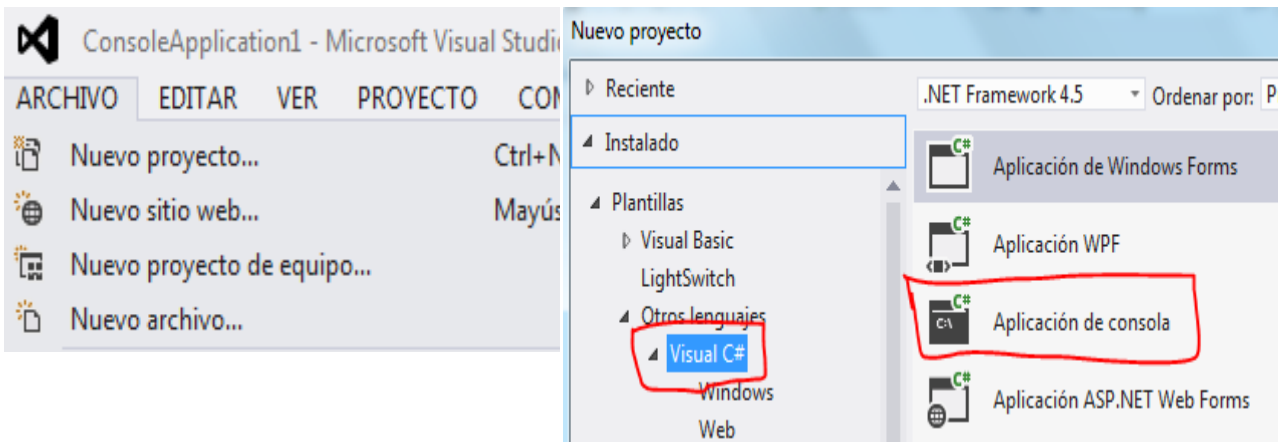
```

4. GRABAR Y EJECUTAR

```
file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/Consol...
DIGITE UNA FRASE : escuela de ingenieria en sistemas
EXISTEN 3 UOCALES A
EXISTEN 6 UOCALES E
EXISTEN 4 UOCALES I
EXISTEN 0 UOCALES O
EXISTEN 1 UOCALES U
Pulse una Tecla: _
```

FACTORIAL DE UN NÚMERO

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

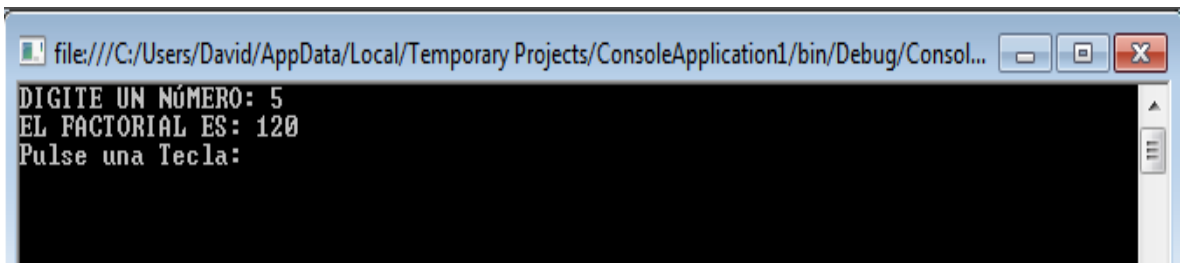
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace FACTORIAL_DE_UN_NÚMERO
{
    class Program
    {
        static void Main(string[] args)
        {
            byte NUM, K;
            long RESULT = 1;
            string linea;
            Console.WriteLine("DIGITE UN NÚMERO: "); linea = Console.ReadLine();
            NUM = byte.Parse(linea);
            for (K = 2; K <= NUM; K++)
            {
                RESULT = RESULT * K;
            }
        }
    }
}
```

```

        Console.WriteLine("EL FACTORIAL ES: " + RESULT);
        Console.Write("Pulse una Tecla:"); Console.ReadLine();
    }
}
}

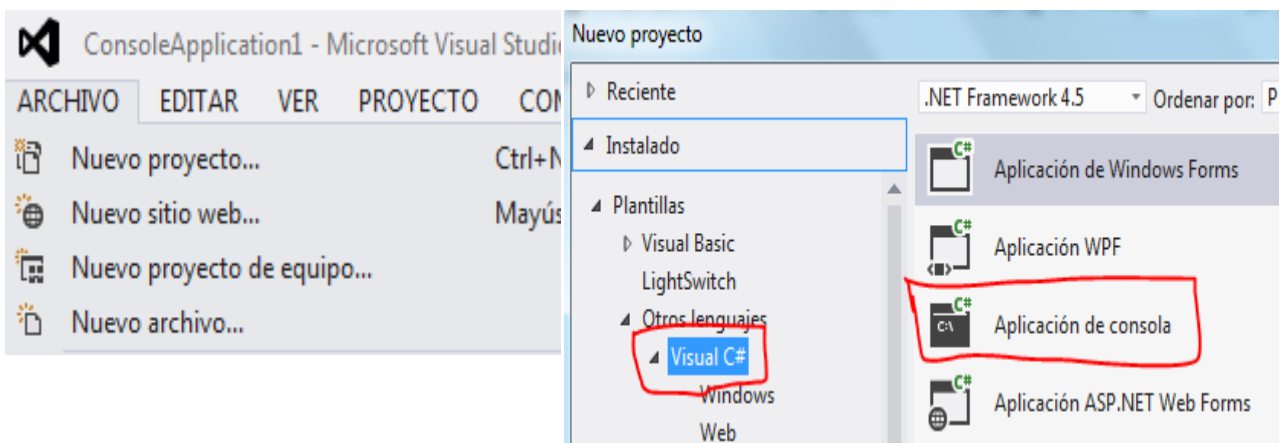
```

4. GRABAR Y EJECUTAR



SERIE DE UN NÚMERO

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

```

```

namespace SERIE_DE_UN_NÚMERO
{
    class Program
    {
        static void Main(string[] args)
        {
            byte NUM, K, F;
            long FAC;
            double RESULT = 0;
            string linea;
            Console.Write("DIGITE UN NÚMERO: "); linea = Console.ReadLine();
            NUM = byte.Parse(linea);
            for (K = 1; K <= NUM; K++)
            {
                FAC = 1;
                for (F = 1; F <= K; F++)
                {
                    FAC = FAC * F;
                }
                RESULT = RESULT + (K / FAC);
            }
            Console.WriteLine("RESULTADO DE LA SERIE ES: " + RESULT);
            Console.Write("Pulse una Tecla:"); Console.ReadLine();
        }
    }
}

```

4. GRABAR Y EJECUTAR

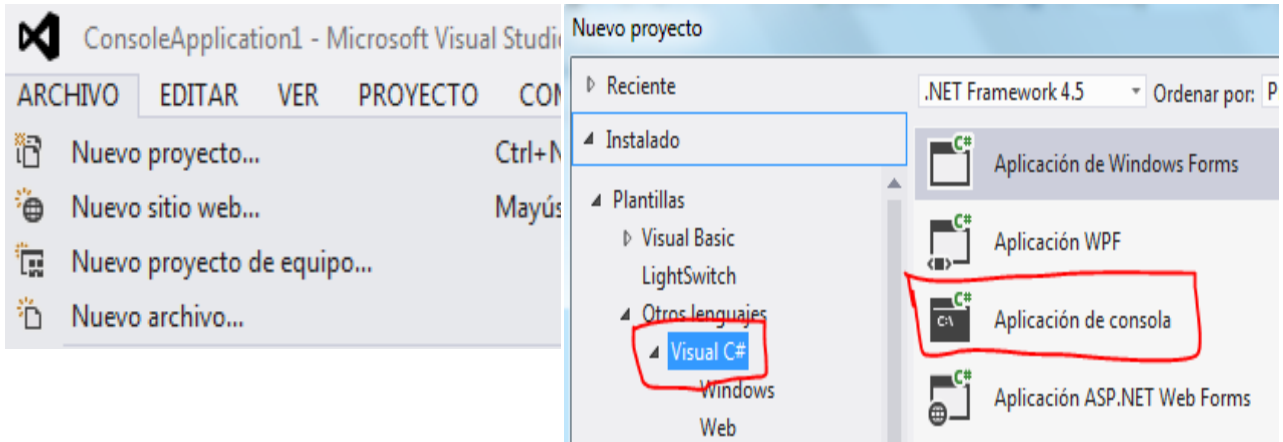
```

file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/Consol...
DIGITE UN NÚMERO: 2
RESULTADO DE LA SERIE ES: 2
Pulse una Tecla: _

```

SUCESIÓN DE N CUADROS

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola

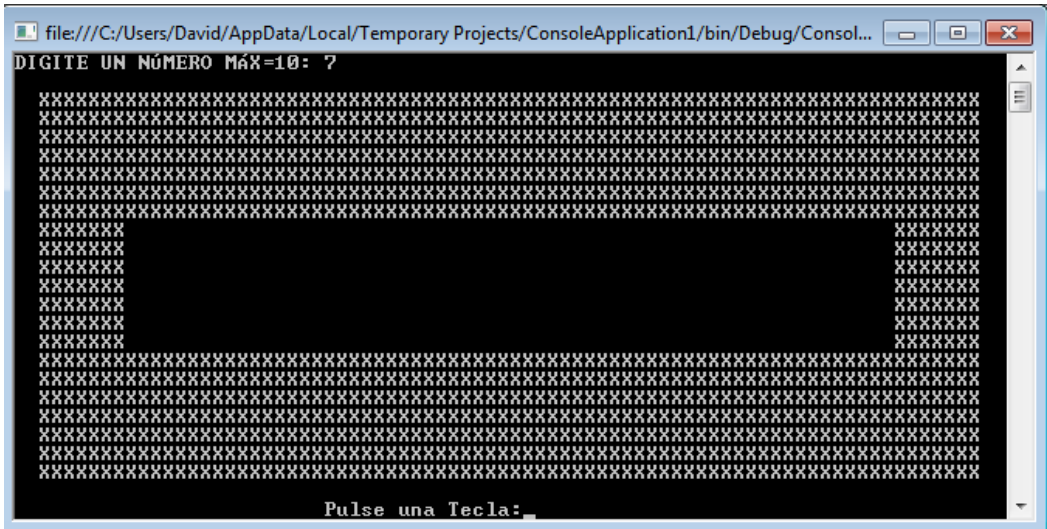


3. CÓDIGO

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace SUCESIÓN_DE_N_CUADROS
{
    class Program
    {
        static void Main(string[] args)
        {
            int CAN, F, C, T, FI, FF, CI, CF;
            string linea;
            Console.WriteLine("DIGITE UN NÚMERO MÁX=10: "); linea = Console.ReadLine();
            CAN = byte.Parse(linea);
            FI = 2; FF = 22;
            CI = 2; CF = 77;
            for (T = 1; T <= CAN; T++)
            {
                for (F = CI; F <= CF; F++)
                {
                    Console.SetCursorPosition(F, FF); Console.WriteLine("X");
                    Console.SetCursorPosition(F, FI); Console.WriteLine("X");
                }
                for (C = FI; C <= FF; C++)
                {
                    Console.SetCursorPosition(CI, C); Console.WriteLine("X");
                    Console.SetCursorPosition(CF, C); Console.WriteLine("X");
                }
                FI = FI + 1; FF = FF - 1;
                CI = CI + 1; CF = CF - 1;
            }
            Console.WriteLine();
        }
    }
}
```

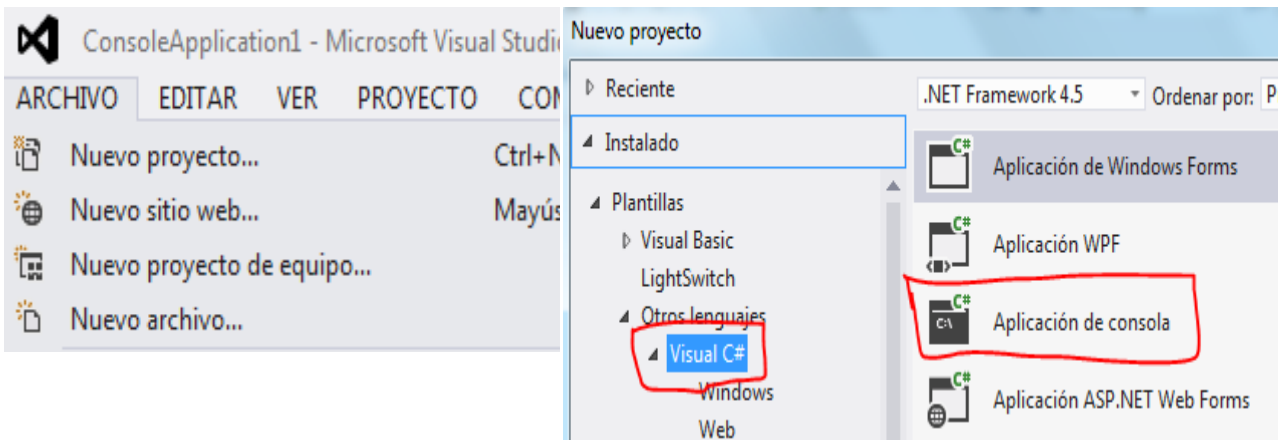
```
        Console.SetCursorPosition(25, 24); Console.Write("Pulse una Tecla:");  
        Console.ReadLine();  
    }  
}
```

4. GRABAR Y EJECUTAR



MOVIMIENTO DE UNA FRASE

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



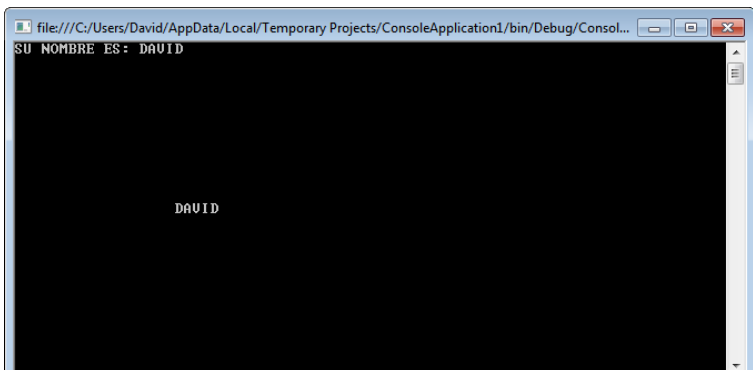
3. CÓDIGO

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace MOVIMIENTO_DE_UNA_FRASE
{
    class Program
    {
        static void Main(string[] args)
        {
            string NOM;
            byte K;
            Console.Write("SU NOMBRE ES: ");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();
        }
    }
}

```

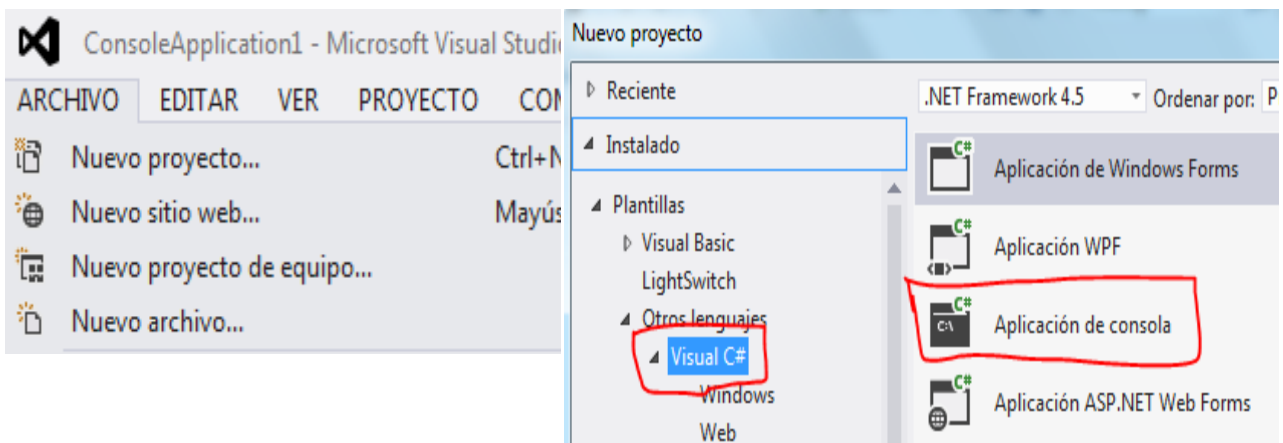
4. GRABAR Y EJECUTAR





MOVIMIENTO CARÁCTER POR CARÁCTER HORIZONTAL

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

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

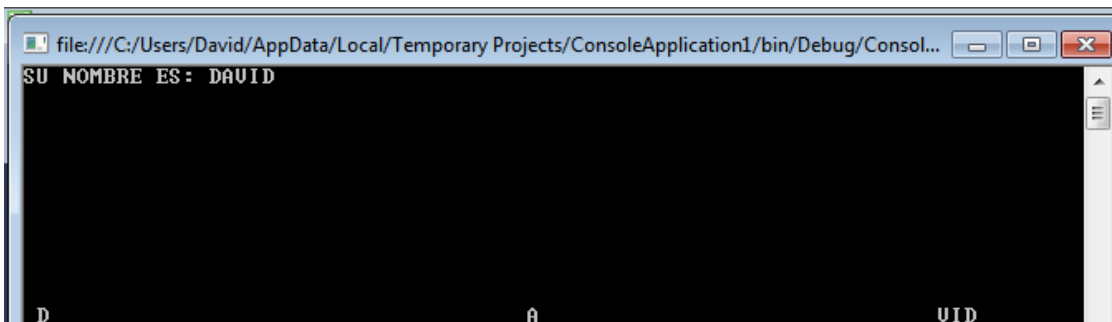
namespace ConsoleApplication1
{
    class Program
    {
        public static string Mid(string param, int startIndex, int length)
        {
            string result = param.Substring(startIndex, length);
            return result;
        }
        static void Main(string[] args)
```

```

{
    string NOM ;
    string CAR ;
    int K = 0;
    int P = 0;
    int CI = 0;
    int CF = 0;
    Console.Write("SU NOMBRE ES: ");
    NOM = Console.ReadLine();
    Console.SetCursorPosition(1, 12);
    Console.Write(NOM);
    CI = NOM.Length;
    CF = 70;
    NOM = NOM.ToUpper();
    for (P = NOM.Length; P >= 1; P--)
    {
        CAR = Mid(NOM, P-1, 1);
        for (K = CI; K <=CF; K++)
        {
            Console.SetCursorPosition(K, 12);
            Console.Write(" " + CAR);
            // REALIZAMOS UNA PAUSA
            System.Threading.Thread.Sleep(50);
        }
        CF = CF - 1;
        CI = CI - 1;
    }
    Console.WriteLine();
    Console.Write("Pulse una Tecla:");
    Console.ReadKey ();
}
}
}

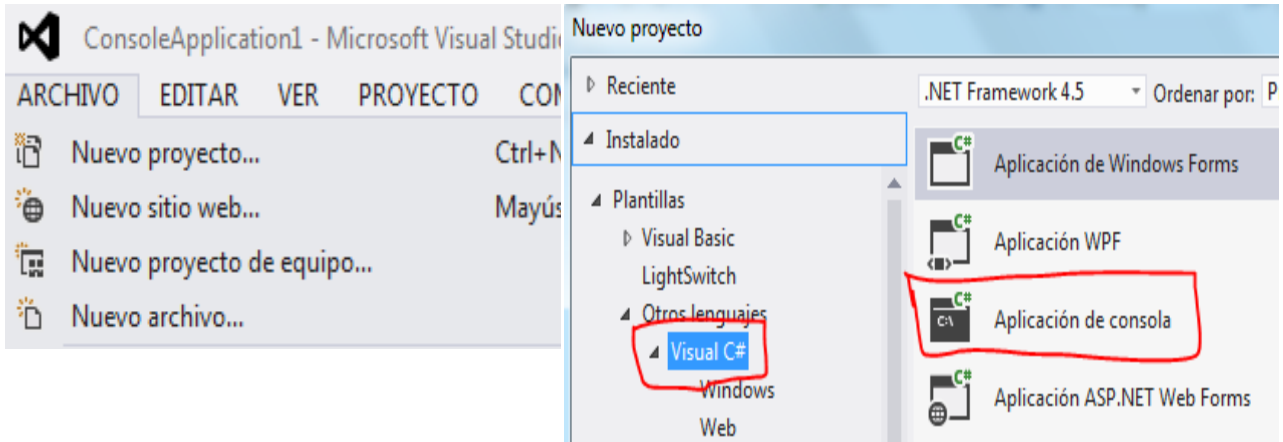
```

4. GRABAR Y EJECUTAR



MOVIMIENTO CARÁCTER POR CARÁCTER VERTICAL

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

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

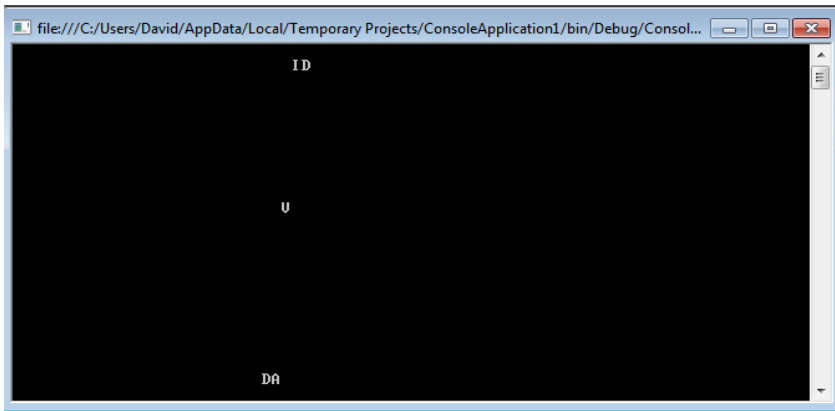
namespace ConsoleApplication1
{
    class Program
    {
        public static string Mid(string param, int startIndex, int length)
        {
            string result = param.Substring(startIndex, length);
            return result;
        }
        static void Main(string[] args)
        {
            string NOM ;
            string CAR ;
            int K = 0;
            int P = 0;
            int CI = 0;
            Console.Write("SU NOMBRE ES: ");
            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();
}
}
}
}
}

```

4. GRABAR Y EJECUTAR

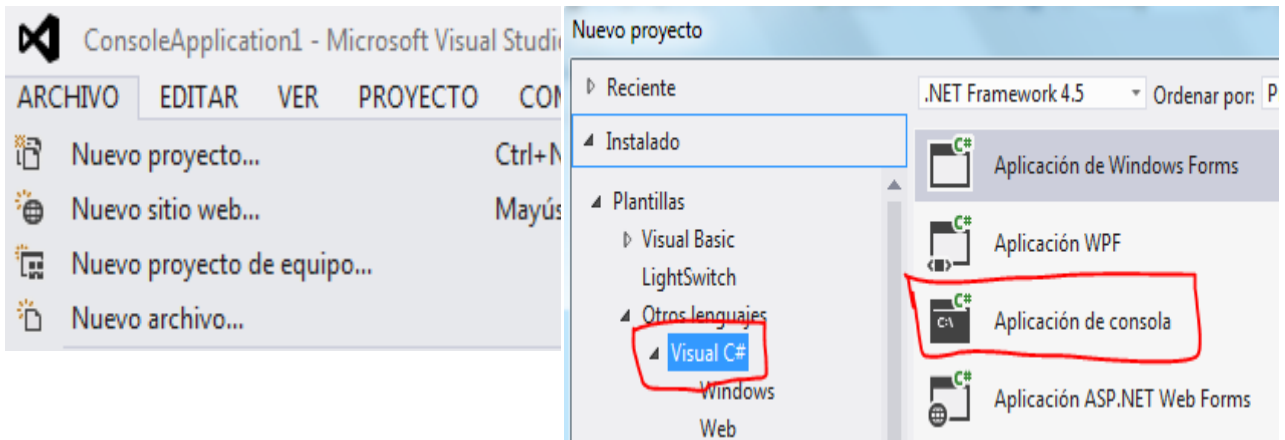


EJERCICIO PROPUESTO

Concepto:

Crear un programa que muestre las letras de la Z (mayúscula) a la A (mayúscula, descendiendo).

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



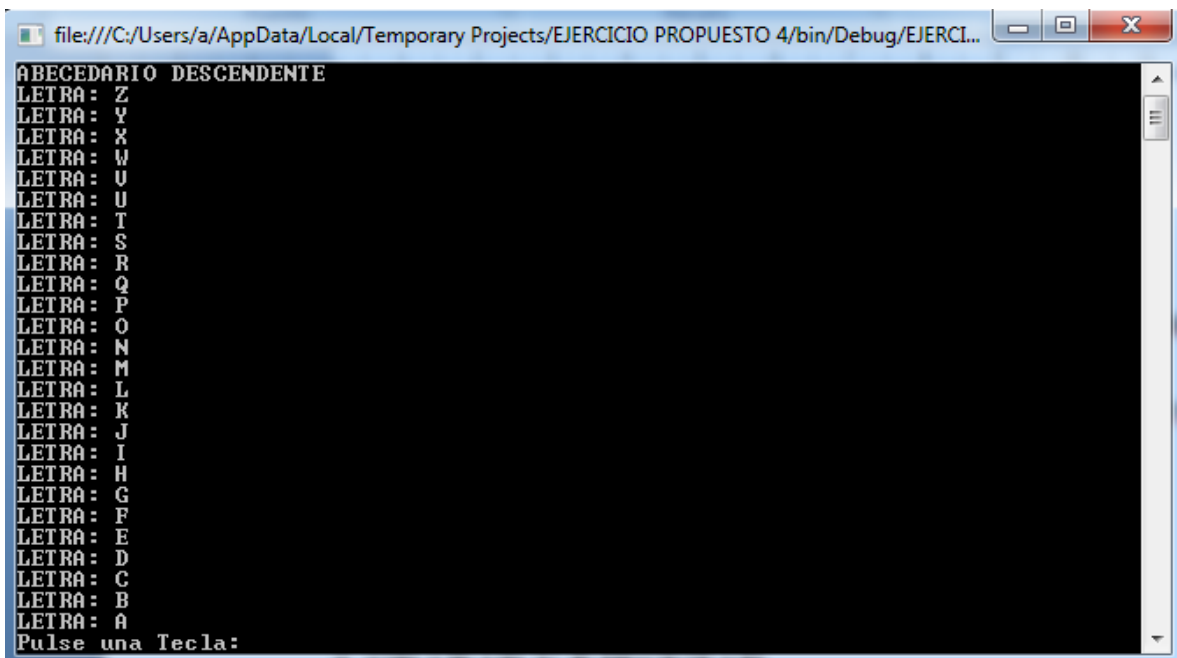
3. CÓDIGO

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

namespace EJERCICIO_PROPUESTO_4
{
    class Program
    {
        static void Main(string[] args)
        {
            int i;
            Console.Write("ABECEDARIO DESCENDENTE");
            Console.WriteLine();
            for (i = 90; i >= 65 ; i--)
            {
                Console.WriteLine("LETRA: " + Convert.ToString((char)i));
            }

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

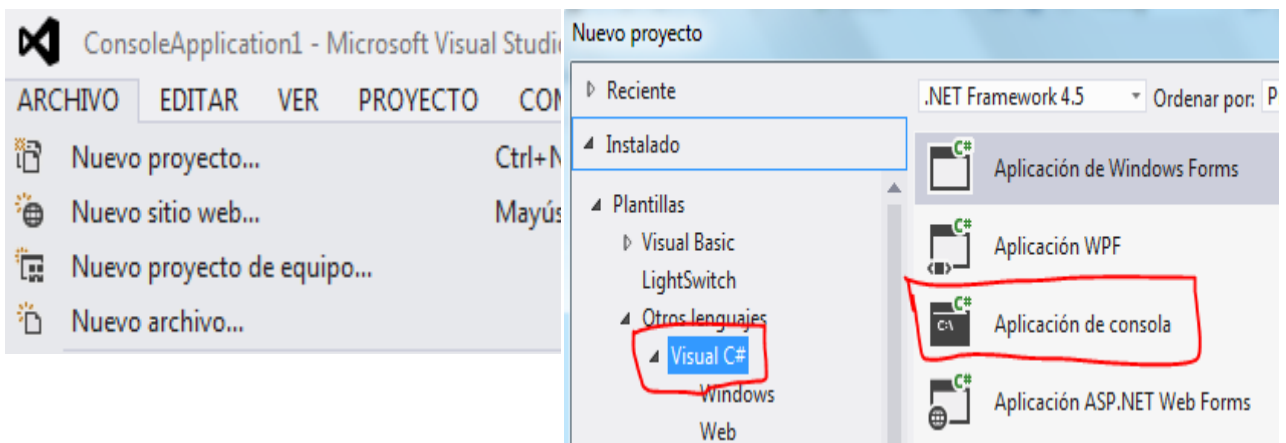
4. GRABAR Y EJECUTAR



```
file:///C:/Users/a/AppData/Local/Temporary Projects/EJERCICIO PROPUESTO 4/bin/Debug/EJERCI...
ABECEDARIO DESCENDENTE
LETRA : Z
LETRA : Y
LETRA : X
LETRA : W
LETRA : U
LETRA : U
LETRA : T
LETRA : S
LETRA : R
LETRA : Q
LETRA : P
LETRA : O
LETRA : N
LETRA : M
LETRA : L
LETRA : K
LETRA : J
LETRA : I
LETRA : H
LETRA : G
LETRA : F
LETRA : E
LETRA : D
LETRA : C
LETRA : B
LETRA : A
Pulse una Tecla:
```

TABLA DE MULTIPLICAR CON WHILE

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace TABLA_DE_MULTIPLICAR_CON_WHILE
{
    class Program
    {
        static void Main(string[] args)
        {
            int NUM, I, RESULT;
            string linea;
            Console.Write("DIGITE NÚMERO:"); linea = Console.ReadLine();
            NUM = int.Parse(linea);
            I = 1;
            while ((I <= 12))
            {
                RESULT = NUM * I;
                Console.WriteLine("{0} * {1} = {2}", NUM, I, RESULT);
                I = I + 1;
            }
            Console.Write("Pulse una Tecla:"); Console.ReadLine();
        }
    }
}

```

4. GRABAR Y EJECUTAR

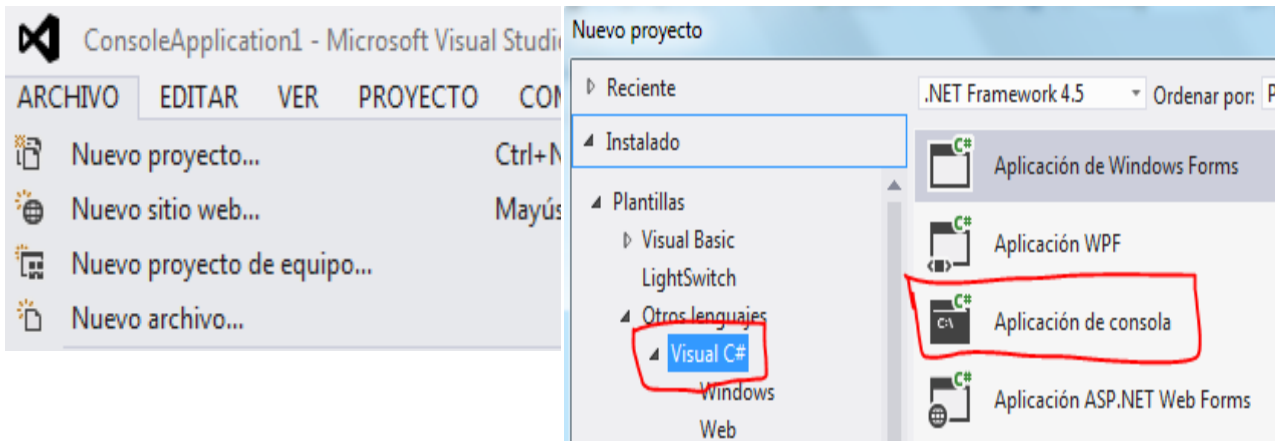
```

file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplicati
DIGITE NÚMERO:5
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50
5 * 11 = 55
5 * 12 = 60
Pulse una Tecla:

```

TABLAS DE MULTIPLICAR CON WHILE

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

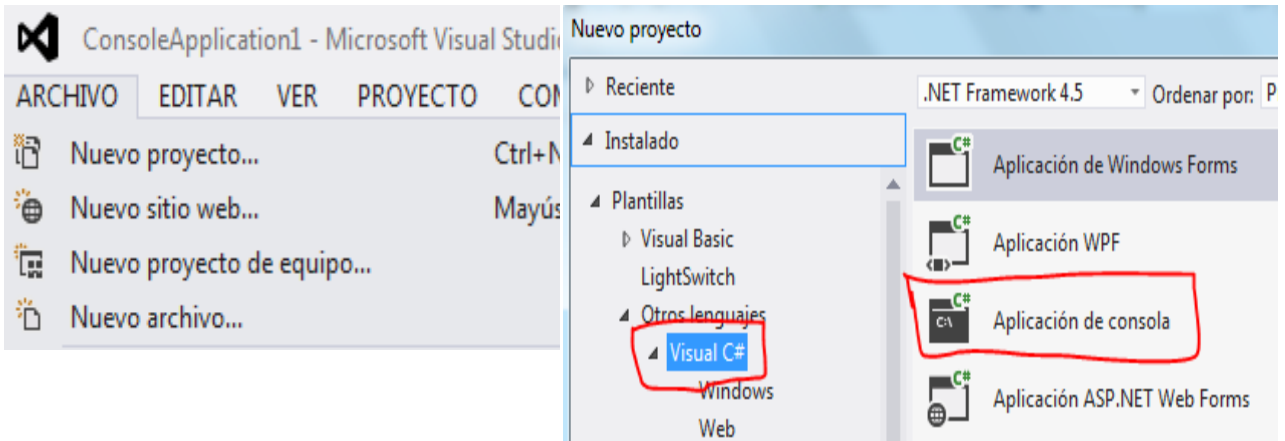
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace TABLAS_DE_MULTIPLICAR_CON_WHILE
{
    class Program
    {
        static void Main(string[] args)
        {
            int NUM, RESULT, T, I;
            string linea;
            Console.WriteLine("CUANTAS TABLAS: "); linea = Console.ReadLine();
            NUM = int.Parse(linea);
            T = 1;
            while ((T <= NUM))
            {
                I = 10;
                while (!(I < 1))
                {
                    RESULT = T * I;
                    Console.WriteLine("{0} * {1} = {2}", T, I, RESULT);
                    I = I - 1;
                }
                Console.WriteLine("Pulse una Tecla:"); Console.ReadLine();
                T = T + 1;
            }
        }
    }
}
```

4. GRABAR Y EJECUTAR


```
file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/Conso...
1 * 4 = 4
1 * 3 = 3
1 * 2 = 2
1 * 1 = 1
Pulse una Tecla:
2 * 10 = 20
2 * 9 = 18
2 * 8 = 16
2 * 7 = 14
2 * 6 = 12
2 * 5 = 10
2 * 4 = 8
2 * 3 = 6
2 * 2 = 4
2 * 1 = 2
Pulse una Tecla:
3 * 10 = 30
3 * 9 = 27
3 * 8 = 24
3 * 7 = 21
3 * 6 = 18
3 * 5 = 15
3 * 4 = 12
3 * 3 = 9
3 * 2 = 6
3 * 1 = 3
Pulse una Tecla: _
```

COMPROBAR SI ES NÚMERO PRIMO

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace Comprobar_si_es_Numero_Primo
{
    class Program
    {
```

```

static void Main(string[] args)
{
    int n, x, sw, resi;
    string linea;
    x = 2;
    sw = 0;
    Console.WriteLine("NUMERO PRIMO");
    Console.WriteLine();
    Console.Write("Ingrese el numero:");
    linea = Console.ReadLine();
    n = int.Parse(linea);
    while (x < n && sw == 0)
    {
        resi = n % x;
        if (resi == 0)
        {
            sw = 1;
        }
        else
        {
            x = x + 1;
        }
    }
    if (sw == 0)
    {
        Console.WriteLine();
        Console.WriteLine("El numero es PRIMO");
    }
    else
    {
        Console.WriteLine();
        Console.WriteLine("El numero no es PRIMO");
    }
    Console.ReadKey();
}
}
}

```

4. GRABAR Y EJECUTAR

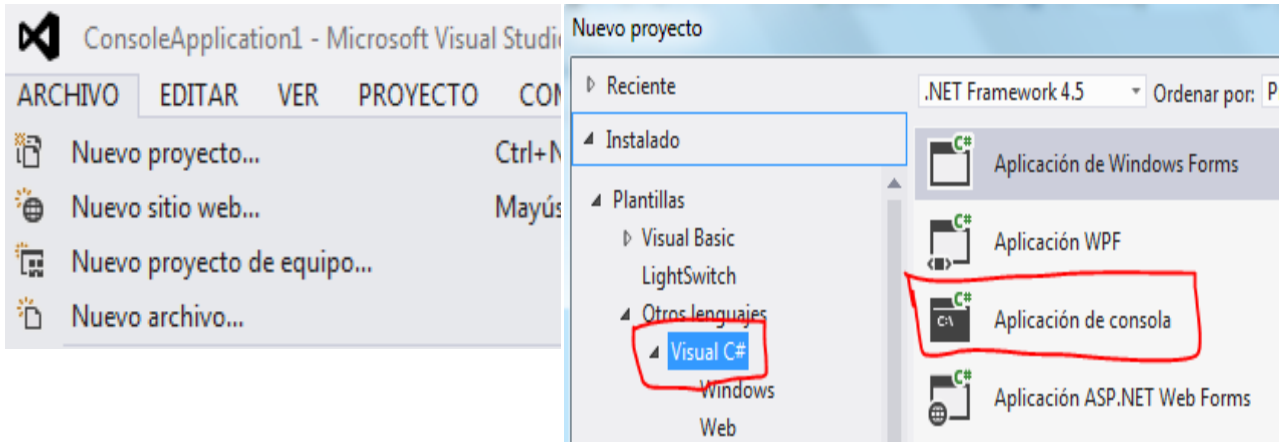
```

file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/Consol...
NUMERO PRIMO
Ingrese el numero:12
El numero no es PRIMO
_

```

FACTORES PRIMOS DE UN NÚMERO

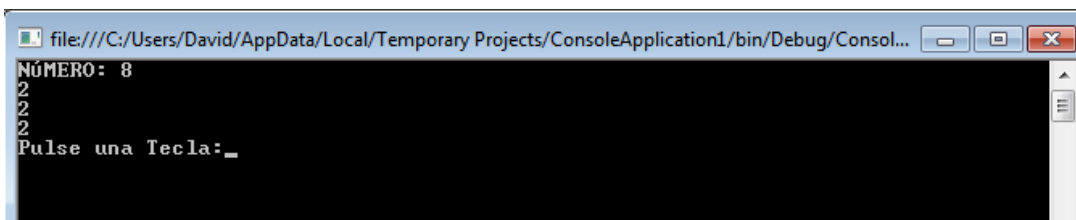
1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

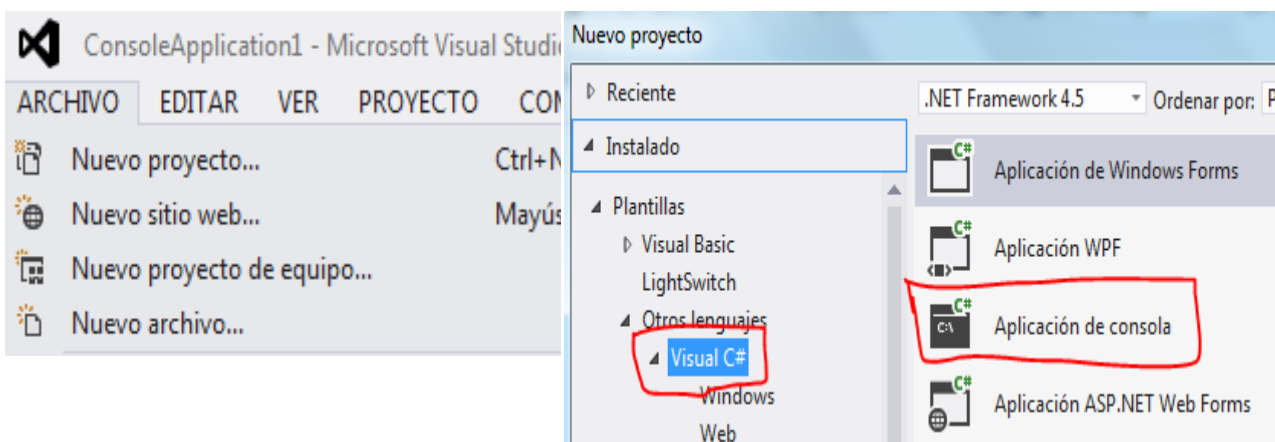
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace FACTORES_PRIMOS_DE_UN_NÚMERO
{
    class Program
    {
        static void Main(string[] args)
        {
            int NUM, RESI, K;
            string linea;
            Console.WriteLine("NÚMERO: "); linea = Console.ReadLine();
            NUM= int.Parse(linea);
            K = 2;
            while ((NUM != 1))
            {
                RESI = NUM % K;
                if ((RESI == 0))
                {
                    Console.WriteLine(K);
                    NUM = NUM / K;
                }
                else
                {
                    K = K + 1;
                }
            }
            Console.WriteLine("Pulse una Tecla:"); Console.ReadLine();
        }
    }
}
```

4. GRABAR Y EJECUTAR



GENERAR N NÚMEROS PRIMOS

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

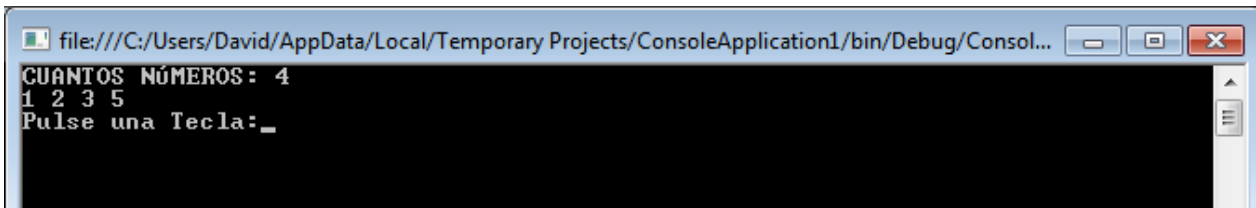
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace GENERAR_N_NÚMEROS_PRIMOS
{
    class Program
    {
        static void Main(string[] args)
        {
            int NUM, RESI, K, SW, CAN, X;
            string linea;
            Console.WriteLine("CUANTOS NÚMEROS: ");linea = Console.ReadLine();
            CAN = int.Parse(linea);
            NUM = 1;X = 0;
            while( (X < CAN)){
                K = 2;SW = 0;
                while( ((K < NUM) & (SW == 0))){
```

```

RESI = NUM % K;
if( (RESI == 0) ){
SW = 1;
}else{
K = K + 1;
}
}
if( (SW == 0) ){
Console.Write(NUM + " ");
X = X + 1;
}
NUM = NUM + 1;
}
Console.WriteLine();
Console.Write("Pulse una Tecla:");Console.ReadLine();
}
}
}

```

4. GRABAR Y EJECUTAR

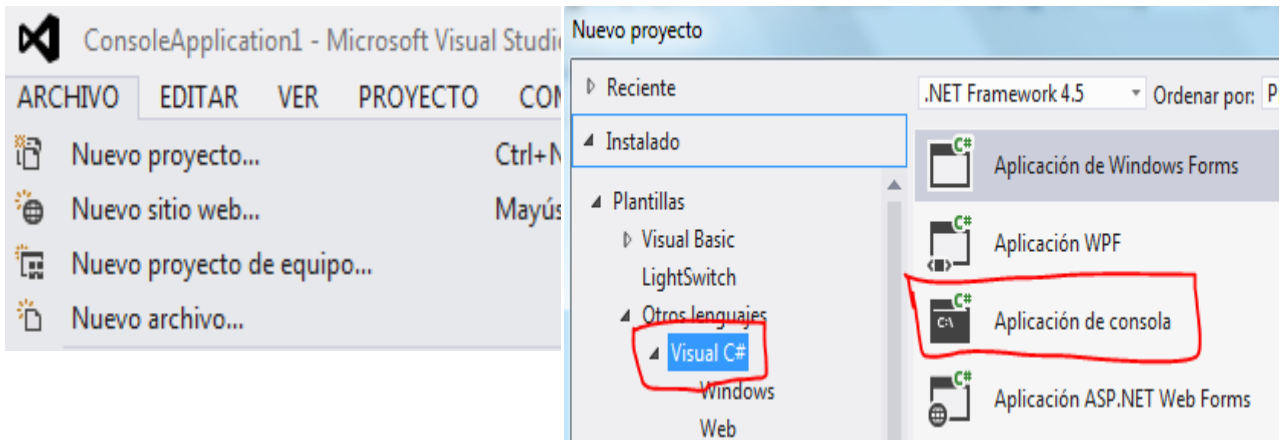


EJERCICIO PROPUESTOS

Concepto:

Crear un programa calcule cuantas cifras tiene un número entero positivo (pista: se puede hacer dividiendo varias veces entre 10).

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



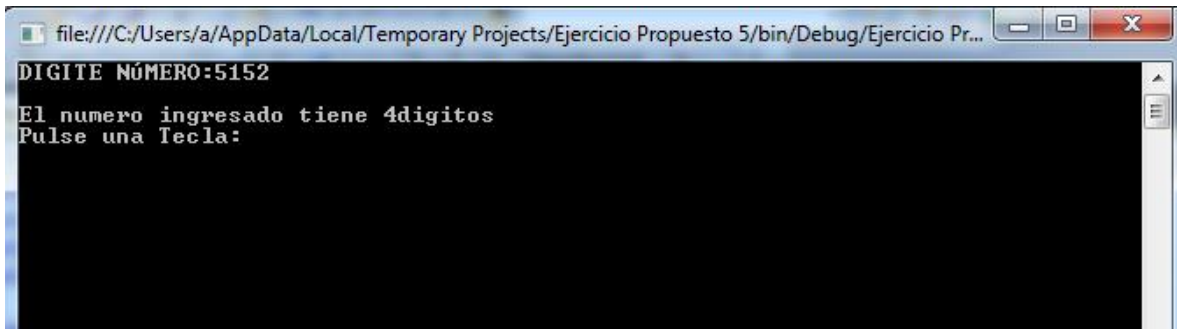
3. CÓDIGO

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

namespace Ejercicio_Propuesto_5
{
    class Program
    {
        static void Main(string[] args)
        {
            int NUM, I, Z, cont=0;
            string linea;
            Console.Write("DIGITE NÚMERO:"); linea = Console.ReadLine();
            NUM = int.Parse(linea);
            Z = linea.Length;
            I = Z;
            while ((I >= 1))
            {
                I--;
                cont++;
            }
            Console.WriteLine();
            Console.WriteLine("El numero ingresado tiene " + cont + " digitos");
            Console.Write("Pulse una Tecla:"); Console.ReadLine();
        }
    }
}
```

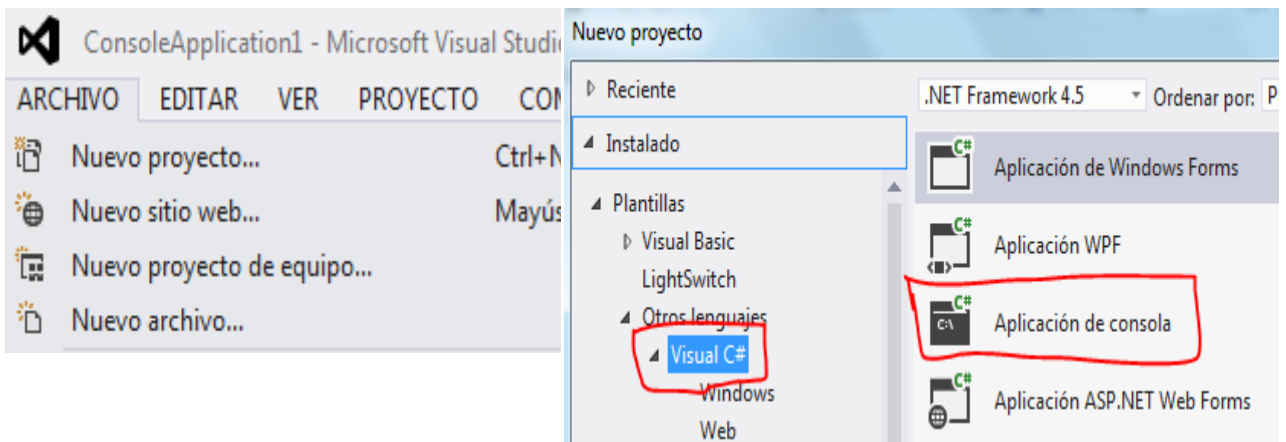
```
}  
    }  
}
```

4. GRABAR Y EJECUTAR



VERIFICACIÓN DE UNA CLAVE 3 OPORTUNIDADES

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
namespace VERIFICACIÓN_DE_UNA_CLAVE_3_OPORTUNIDADES  
{  
    class Program  
    {  
        static void Main(string[] args)
```

```

{
    byte K, SW;
    string CLAVE;
    SW = 0; K = 0;
    do
    {
        Console.WriteLine("DIGITE CLAVE: "); CLAVE = Console.ReadLine();
        if ((CLAVE.ToUpper() == "ARIEL"))
        {
            SW = 1;
        }
        else
        {
            K++;
        }
    } while (((K < 3) & (SW == 0)));
    if (SW == 1)
    {
        Console.WriteLine("BIENVENIDO " + CLAVE);
    }
    else
    {
        Console.WriteLine("OPORTUNIDADES TERMINADAS");
    }
    Console.WriteLine("Pulse una Tecla:"); Console.ReadLine();
}
}
}
}

```

4. GRABAR Y EJECUTAR

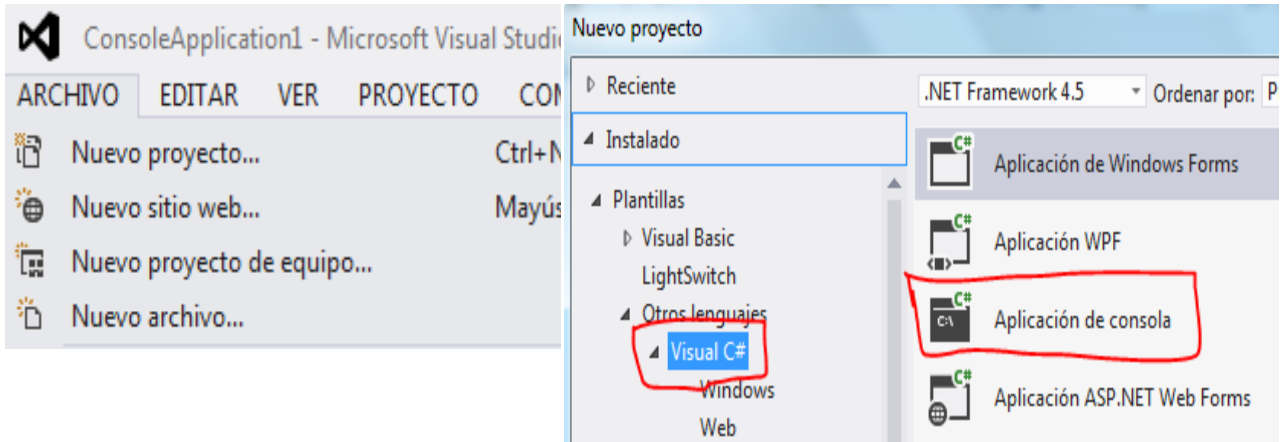
```

file:///C:/Users/David/AppData/Local/Temporary P
DIGITE CLAVE: docencia
DIGITE CLAVE: installer
DIGITE CLAVE: ariel
BIENVENIDO ariel
Pulse una Tecla:

```


GENERAR UN NÚMERO ALEATORIO ENTRE 10 Y 30

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

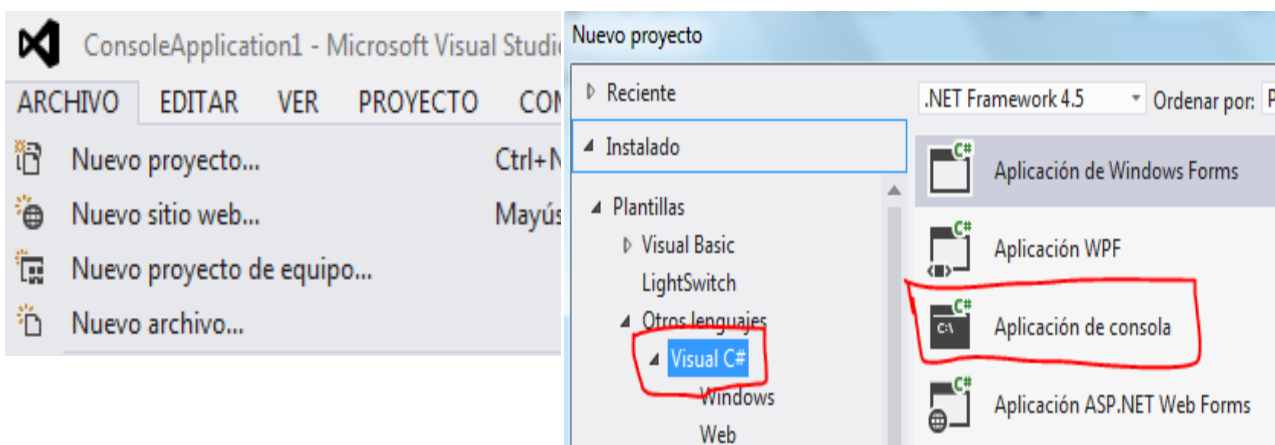
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int NUM;
            string OP;
            Random ran = new Random();
            do{
            do{
                NUM = Convert.ToInt32(ran.Next(100));
            }while(!( ((NUM >= 10) & (NUM <= 30))));
            Console.WriteLine("SE GENERO EL " + NUM);
            Console.Write("GENERA OTRO NÚMERO S/N:");OP = Console.ReadLine();
            OP = OP.ToUpper();
            }while( OP == "S");
            Console.Write("Pulse una Tecla:");Console.ReadLine();
        }
    }
}
```

4. GRABAR Y EJECUTAR

```
file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/
SE GENERO EL 20
GENERA OTRO NÚMERO S/N:s
SE GENERO EL 18
GENERA OTRO NÚMERO S/N:s
SE GENERO EL 20
GENERA OTRO NÚMERO S/N:n
Pulse una Tecla:
```

JUEGO ADIVINA UN NÚMERO

1. Abrir un nuevo Proyecto
2. Seleccione Aplicación de Consola



3. CÓDIGO

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace JUEGO_ADIVINA_UN_NÚMERO
{
    class Program
    {
        static void Main(string[] args)
        {
            const int OPOR = 4;
            int NUM, ADI;
            byte I, SW;
            string linea;
            //GENERO EL NÚMERO ENTRE 1 AL 20;
        }
    }
}
```

```

Random ran = new Random();
do{
NUM = Convert.ToInt32(ran.Next(100));
}while(!( (NUM >= 1) & (NUM <= 20)));
//PROCESO;
I = 1;SW = 0;
do{
    Console.Write("PIENSA QUÉ NÚMERO SERÁ:"); linea = Console.ReadLine();
    ADI = int.Parse(linea);
if( (ADI == NUM) ){
    Console.WriteLine("ADIVINASTE ERES GENIAL");
SW = 1;
}
else{

    if( (ADI > NUM) ){
        Console.WriteLine("TE PASASTE");

    }

    else{
        Console.WriteLine("ESTÁS BAJO");

    }

}
I++;
}while( ((I <= OPOR) & (SW == 0)));
if( (SW == 0) ){
    Console.Write("EL NÚMERO FUE: " + NUM);
}
Console.WriteLine();
Console.Write("Pulse una Tecla:"); Console.ReadLine();
}
}
}

```

4. GRABAR Y EJECUTAR

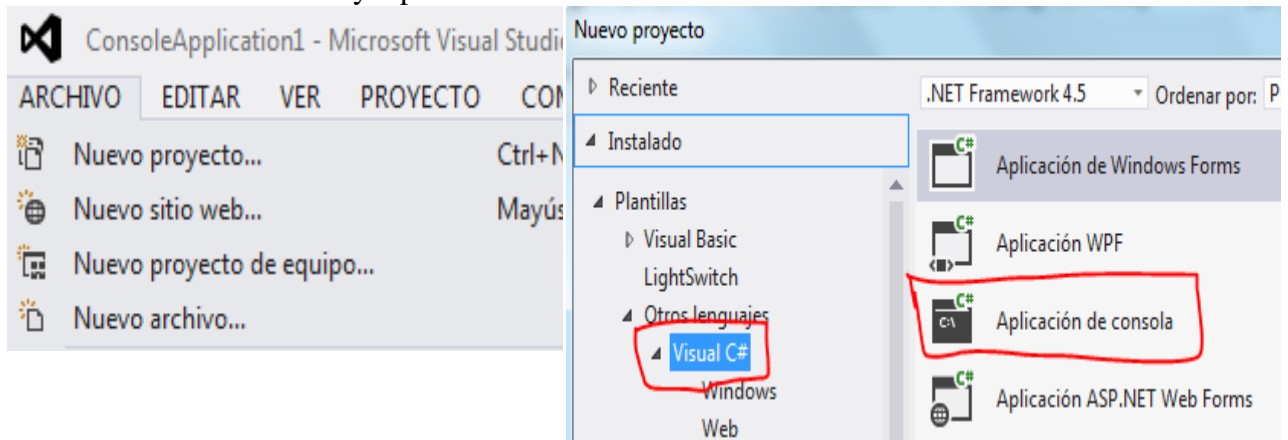
```

file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/Cons
PIENSA QUÉ NÚMERO SERÁ:23
TE PASASTE
PIENSA QUÉ NÚMERO SERÁ:20
TE PASASTE
PIENSA QUÉ NÚMERO SERÁ:10
ESTÁS BAJO
PIENSA QUÉ NÚMERO SERÁ:15
ESTÁS BAJO
EL NÚMERO FUE: 17
Pulse una Tecla:

```

CONTROL DE UNA FACTURA

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola

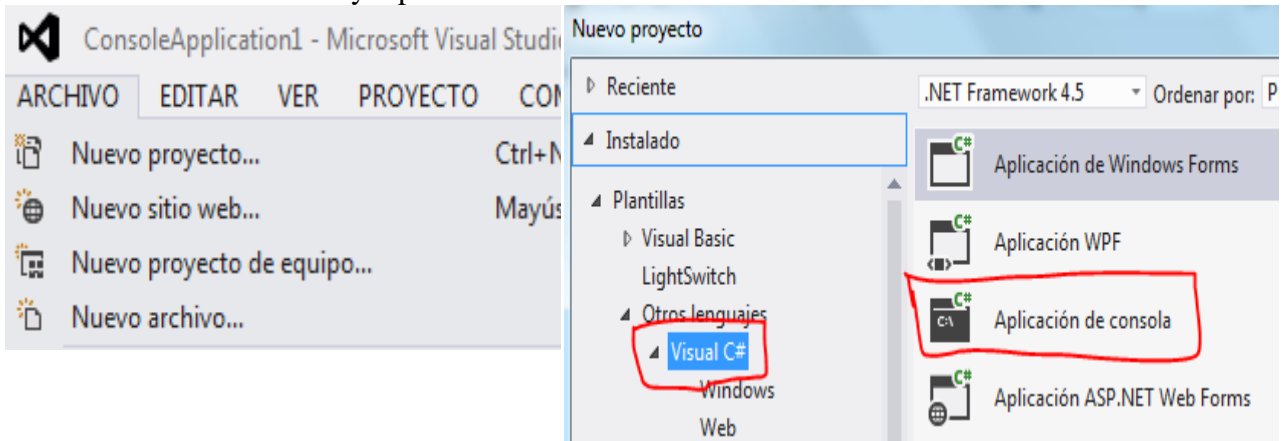


3. CÓDIGO

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            string PROD ;
            int CAN = 0;
            int FI = 0;
            double PRE = 0;
            double TOT = 0;
            double PAGA = 0;
            char OP ;
            string linea;
            FI = 6;
            PAGA = 0;
            Console.SetCursorPosition(5, 4);
            Console.WriteLine("PRODUCTO CANTIDAD PRECIO TOTAL");
            do
            {
                Console.SetCursorPosition(5, FI);
                PROD = Console.ReadLine();
                Console.SetCursorPosition(17, FI);
                linea = Console.ReadLine();
                CAN = int.Parse(linea);
                Console.SetCursorPosition(27, FI);
                linea = Console.ReadLine();
                PRE = int.Parse(linea);
                TOT = CAN * PRE;
                Console.SetCursorPosition(33, FI);
```


VOTACIONES POR SECTOR

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

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

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int CANN = 0;
            int CANS = 0;
            int CANC = 0;
            char OP ;
            string ZONA ,linea;
            CANN = 0;
            CANC = 0;
            CANS = 0;
            do
            {
                Console.Write("QUE SECTOR NORTE,CENTRO,SUR:");
                ZONA = Console.ReadLine();
                switch (ZONA.ToUpper())
                {
                    case "NORTE":
                        CANN = CANN + 1;
                        break;
                    case "CENTRO":
                        CANC = CANC + 1;
                        break;
                    case "SUR":
                        CANS = CANS + 1;
                        break;
                }
            }
        }
    }
}
```



```

using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            double SUMT = 0;
            double SUE = 0;
            double PROM = 0;
            int CAN = 0;
            string linea;
            SUMT = 0;
            do
            {
                Console.Write("DIGITE SUELDO:");
                linea=Console.ReadLine();
                SUE = double.Parse(linea);
                if (SUE > 0)
                {
                    SUMT = SUMT + SUE;
                    CAN = CAN + 1;
                }
            } while (SUE > 0);
            PROM = SUMT / CAN;
            Console.WriteLine();
            Console.WriteLine("TOTAL SUELDOS: " + SUMT);
            Console.WriteLine("EMPLEADOS : " + CAN);
            Console.WriteLine("PROMEDIO : " + PROM);
            Console.Write("Pulse una Tecla:");
            Console.ReadLine();
        }
    }
}

```

4. GRABAR Y EJECUTAR

```

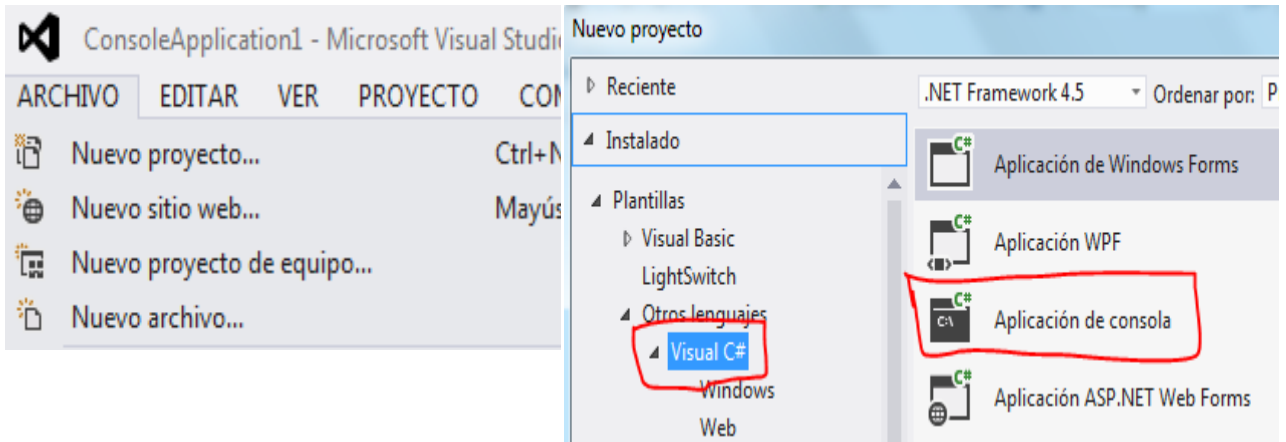
file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/Consol...
DIGITE SUELDO:23324
DIGITE SUELDO:12
DIGITE SUELDO:32213
DIGITE SUELDO:1333333333333333
DIGITE SUELDO:1241
DIGITE SUELDO:1
DIGITE SUELDO:2
DIGITE SUELDO:0

TOTAL SUELDOS: 133333333390126
EMPLEADOS : 7
PROMEDIO : 19047619055732,3
Pulse una Tecla:

```


FRASE INVERTIDA CON WHILE

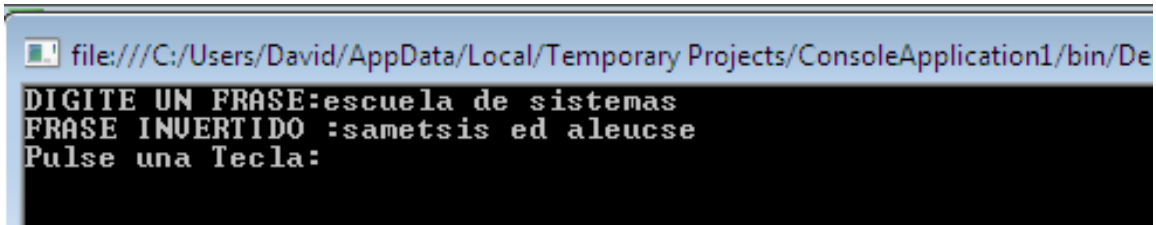
1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

```
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 AUX ;
            int POS = 0;
            AUX = "";
            Console.Write("DIGITE UN FRASE:");
            NOM = Console.ReadLine();
            POS = NOM.Length ;
            while ((POS > 0))
            {
                AUX = AUX + Mid(NOM, POS-1, 1);
                POS = POS - 1;
            }
            Console.WriteLine("FRASE INVERTIDO :" + AUX);
            Console.Write("Pulse una Tecla:");
            Console.ReadLine();
        }
    }
}
```

4. GRABAR Y EJECUTAR

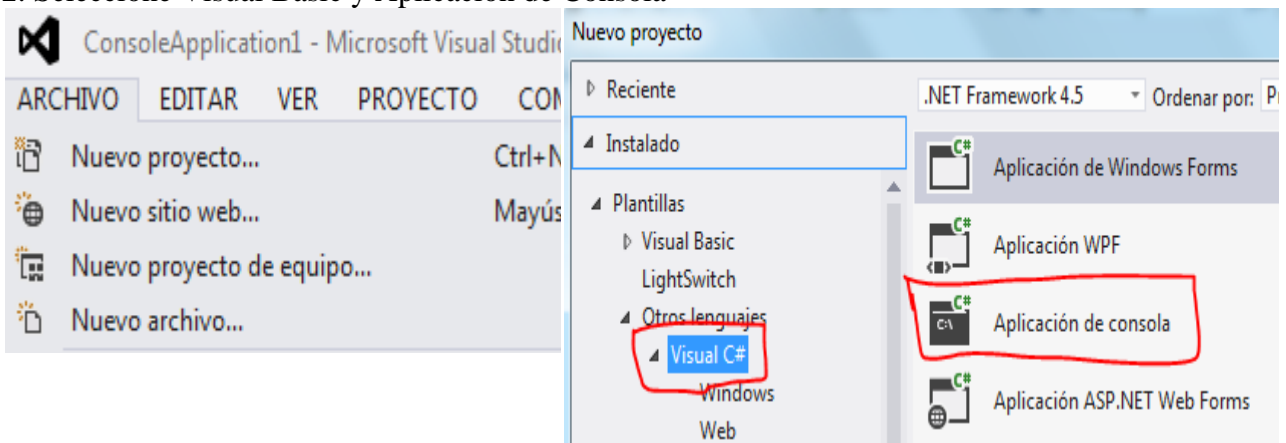


EJERCICIO PROPUESTO

Concepto:

Crear un programa que pida números positivos al usuario, y vaya calculando la suma de todos ellos (terminará cuando se teclea un número negativo o cero).

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

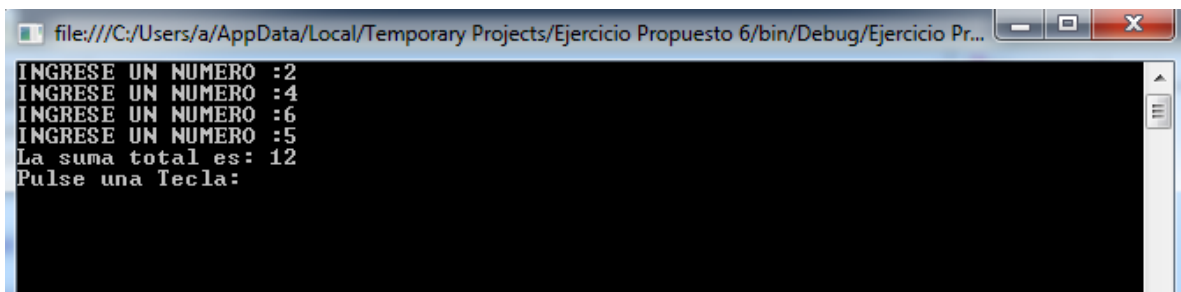
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Ejercicio_Propuesto_6
{
    class Program
    {
        static void Main(string[] args)
        {
            int NUM, RES, TOT=0 ;
            string linea;
            do{
                Console.Write("INGRESE UN NUMERO :");
                linea = Console.ReadLine();
                NUM = int.Parse(linea);
            }
        }
    }
}
```

```
        RES = NUM % 2;
        if (RES == 0 && NUM != 0)
            TOT = TOT + NUM;
        else
        {
        }
    }while (RES == 0 && NUM != 0);

    Console.WriteLine ("La suma total es: " +TOT);
    Console.Write("Pulse una Tecla:");
    Console.ReadLine();
}
}
}
```

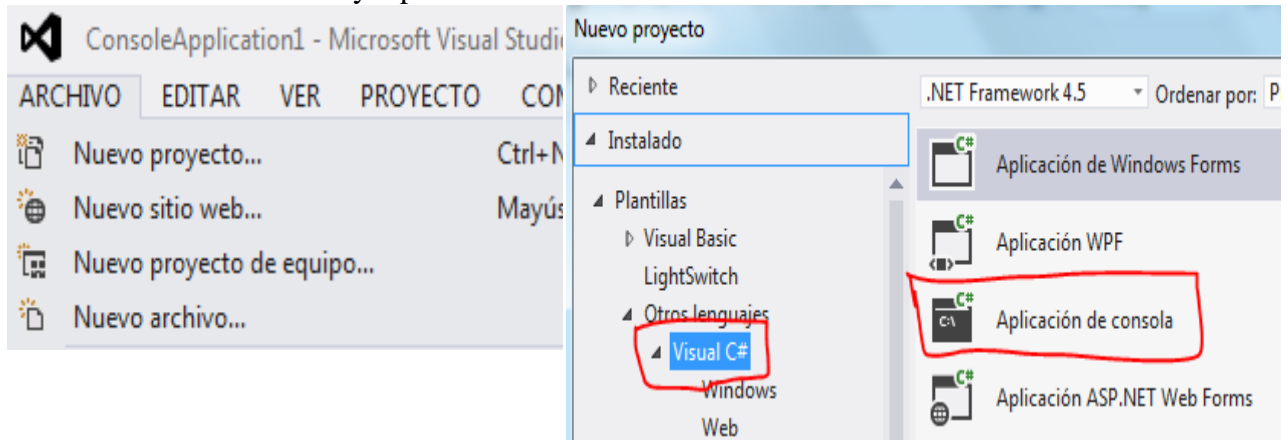
4. GRABAR Y EJECUTAR



```
file:///C:/Users/a/AppData/Local/Temporary Projects/Ejercicio Propuesto 6/bin/Debug/Ejercicio Pr...
INGRESE UN NUMERO :2
INGRESE UN NUMERO :4
INGRESE UN NUMERO :6
INGRESE UN NUMERO :5
La suma total es: 12
Pulse una Tecla:
```

TEMA: MANIPULACIÓN DE UN ARREGLO

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola

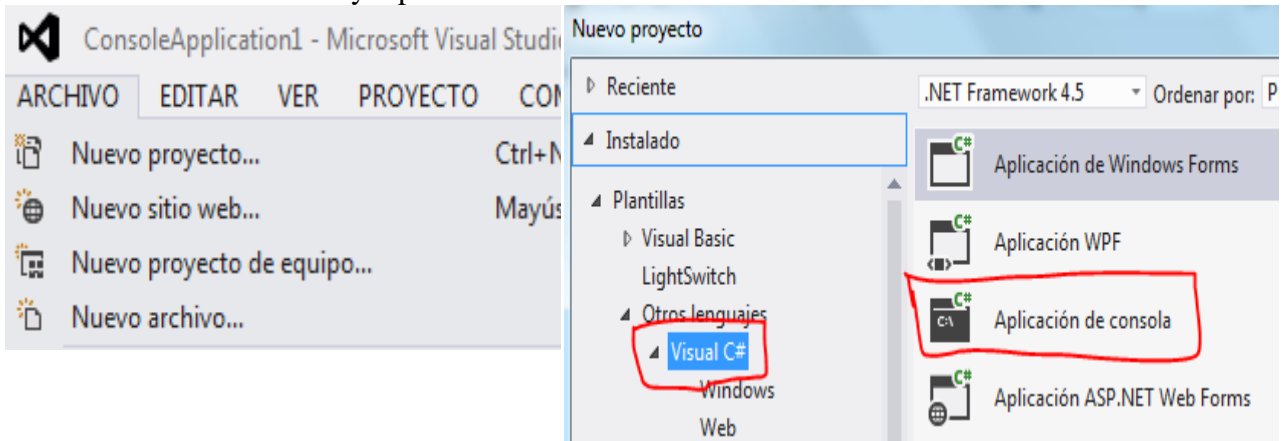


3. CÓDIGO

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            // FORMA 1
            // asignando valores al array al mismo tiempo que se declara,
            // la lista de valores debe ir encerrada entre llaves
            {
                string[] VECTOR = {
                    "LORENA",
                    "ESTEBAN",
                    "ARIEL"
                };
                byte K = 0;
                for (K = 0; K <= 2; K++)
                {
                    Console.WriteLine(VECTOR[K]);
                }
                Console.WriteLine();
                // FORMA 2
                // declarar primero la variable que contendrá el array,
                // asignar valores al array al mismo tiempo que se instancia
                // la lista de valores debe ir encerrada entre llaves
                int[] MILISTA = null;
                MILISTA = new int[] {
                    15,
```


INTERCALACIÓN MAYUSCULAS Y MINUSCULAS

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola

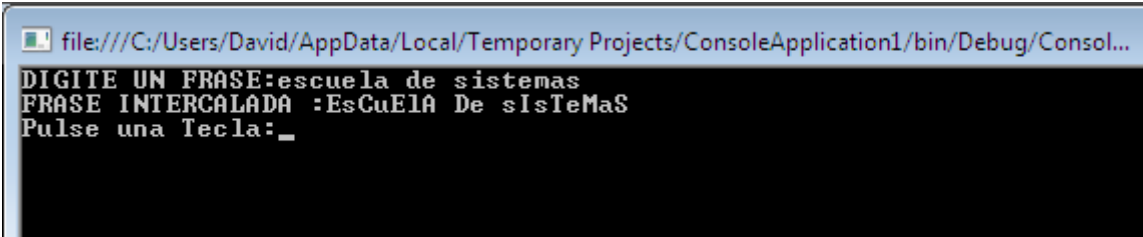


3. CÓDIGO

```
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 = null;
            string AUX = null;
            string LE = null;
            int POS = 0;
            AUX = "";
            Console.Write("DIGITE UN FRASE:");
            NOM = Console.ReadLine();
            POS = 1;
            while ((POS <= NOM.Length))
            {
                LE = Mid(NOM, POS-1, 1);
                if ((POS % 2 == 0))
                {
                    AUX = AUX + LE.ToLower();
                }
                else
                {
                    AUX = AUX + LE.ToUpper();
                }
                POS = POS + 1;
            }
        }
    }
}
```

```
        Console.WriteLine("FRASE INTERCALADA :" + AUX);  
        Console.Write("Pulse una Tecla:");  
        Console.ReadLine();  
    }  
}
```

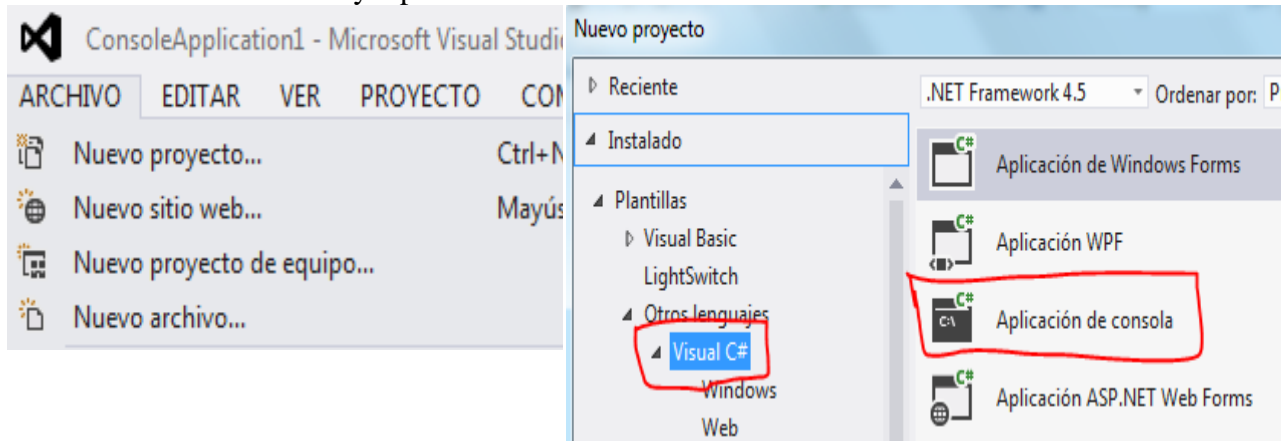
4. GRABAR Y EJECUTAR



```
file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/Consol...  
DIGITE UN FRASE:escuela de sistemas  
FRASE INTERCALADA :EsCuEIA De sIsTeMaS  
Pulse una Tecla:_
```

GENERAR NÚMEROS ALEATORIOS EN UN ARREGLO

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



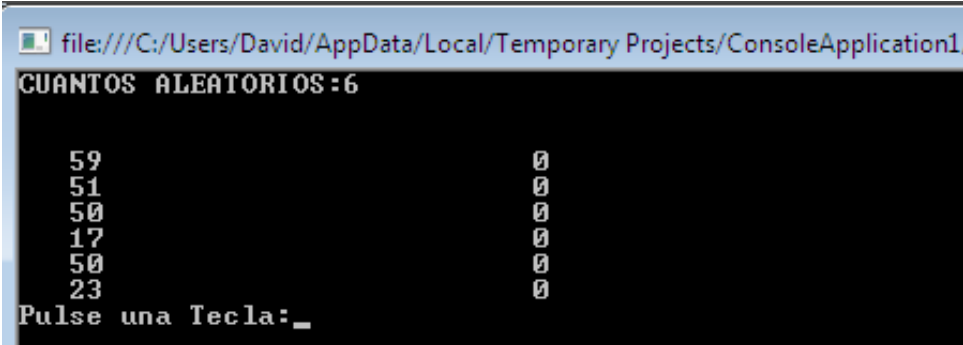
3. CÓDIGO

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int CAN = 0;
            int POS = 0;
            string linea;
            Random rnd = new Random();
            Console.Write("CUANTOS ALEATORIOS:");
            linea=Console.ReadLine();
            CAN = int.Parse(linea);
            double[] VEC = new double[CAN + 1];
            byte[] VEC1 = new byte[CAN + 1];
            // GENERACIÓN DE ALEATORIOS
            for (POS = 1; POS <= CAN; POS++)
            {
                VEC[POS] = rnd.Next(0, 99);
                // OBTENEMOS UN ENTERO DE 2 CIFRAS
            }
            // SALIDA DEL ARREGLO
            for (POS = 1; POS <= CAN; POS++)
            {
                Console.SetCursorPosition(3, POS + 2);
                Console.WriteLine(VEC[POS]);
                Console.SetCursorPosition(30, POS + 2);
                Console.WriteLine(VEC1[POS]);
            }
            Console.Write("Pulse una Tecla.");
            Console.ReadLine();
        }
    }
}
```



```
}  
}
```

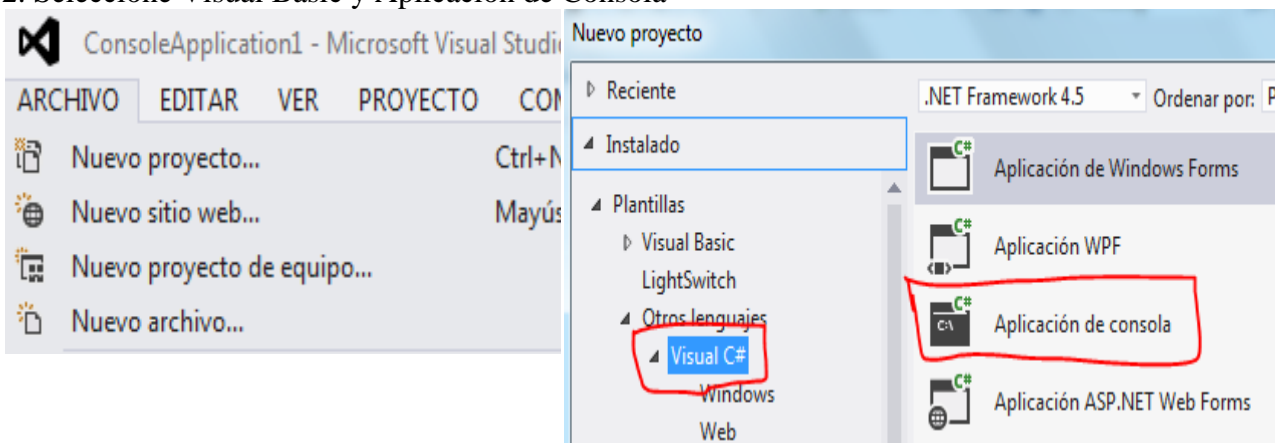
4. GRABAR Y EJECUTAR



```
file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1  
CUANTOS ALEATORIOS:6  
  
59          0  
51          0  
50          0  
17          0  
50          0  
23          0  
Pulse una Tecla: _
```

PASAR UNA FRASE A UN ARREGLO

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola

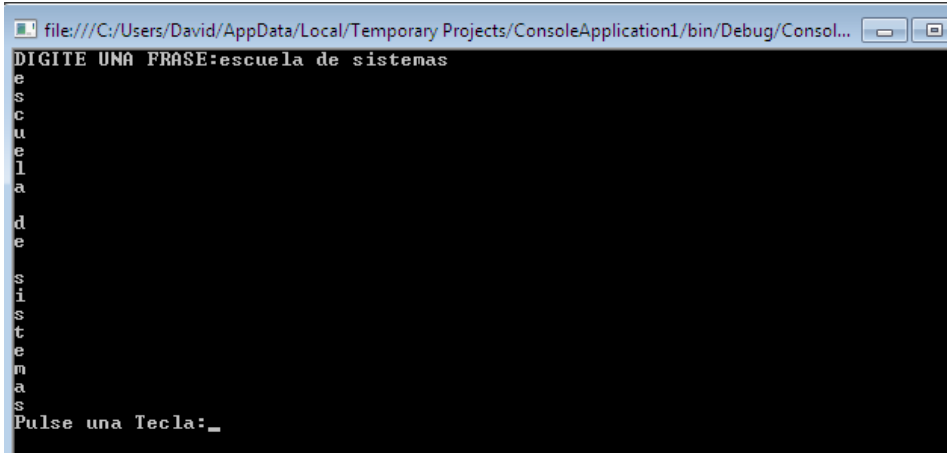


3. CÓDIGO

```
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 CAD ;
            int POS = 0;
            int LEN = 0;
            Console.Write("DIGITE UNA FRASE:");
            CAD = Console.ReadLine();
            LEN = CAD.Length;
            string [] VEC = new string [LEN + 1];
            // PASAMOS CARACTER A CARACTER AL ARREGLO
            for (POS = 1; POS <= LEN; POS++)
            {
                VEC[POS] = Mid(CAD, POS-1, 1);
            }
            // SALIDA
            for (POS = 1; POS <= LEN; POS++)
            {
                Console.WriteLine(VEC[POS]);
            }
            Console.Write("Pulse una Tecla:");
            Console.ReadLine();
        }
    }
}
```

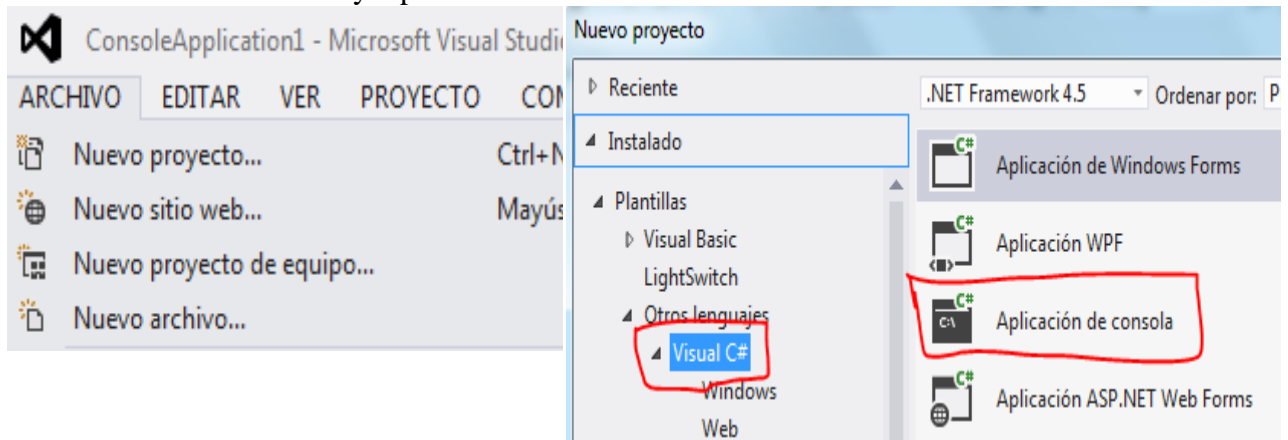
```
}
```

4. GRABAR Y EJECUTAR



SUMA ELEMENTOS PARES E IMPARES EN UN ARREGLO

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int I = 0;
            int CAN = 0;
            int SUMP = 0;
            int SUMI = 0;
            int[] VEC = new int[13];
        }
    }
}
```

```

string linea;
Console.Write("CUANTOS ELEMENTOS MÁX=12:");
linea=Console.ReadLine();
CAN = int.Parse(linea);
// INGRESO
for (I = 1; I <= CAN; I++)
{
    Console.Write("POSICIÓN {0} ==>", I);
    linea=Console.ReadLine();
    VEC[I] = int.Parse(linea);
}
// PROCESO
SUMP = 0;
SUMI = 0;
for (I = 1; I <= CAN; I++)
{
    if ((VEC[I] % 2) == 0)
    {
        SUMP = SUMP + VEC[I];
    }
    else
    {
        SUMI = SUMI + VEC[I];
    }
}
// SALIDA
Console.WriteLine();
Console.WriteLine("SUMA VALORES PARES: " + SUMP);
Console.WriteLine("SUMA VALORES IMPARES: " + SUMI);
Console.Write("Pulse una Tecla:");
Console.ReadLine();
}
}
}

```

4. GRABAR Y EJECUTAR

```

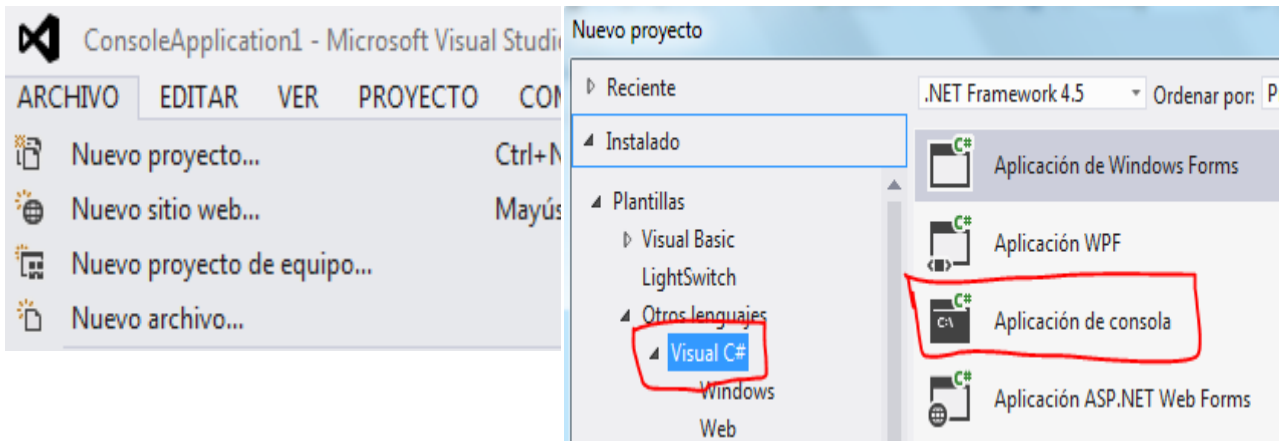
file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1
CUANTOS ELEMENTOS MÁX=12:5
POSICIÓN 1 ==>2
POSICIÓN 2 ==>3
POSICIÓN 3 ==>1
POSICIÓN 4 ==>12
POSICIÓN 5 ==>32

SUMA VALORES PARES: 46
SUMA VALORES IMPARES: 4
Pulse una Tecla:

```

MAYOR Y MENOR DE UN ARREGLO DE N ELEMENTOS

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola

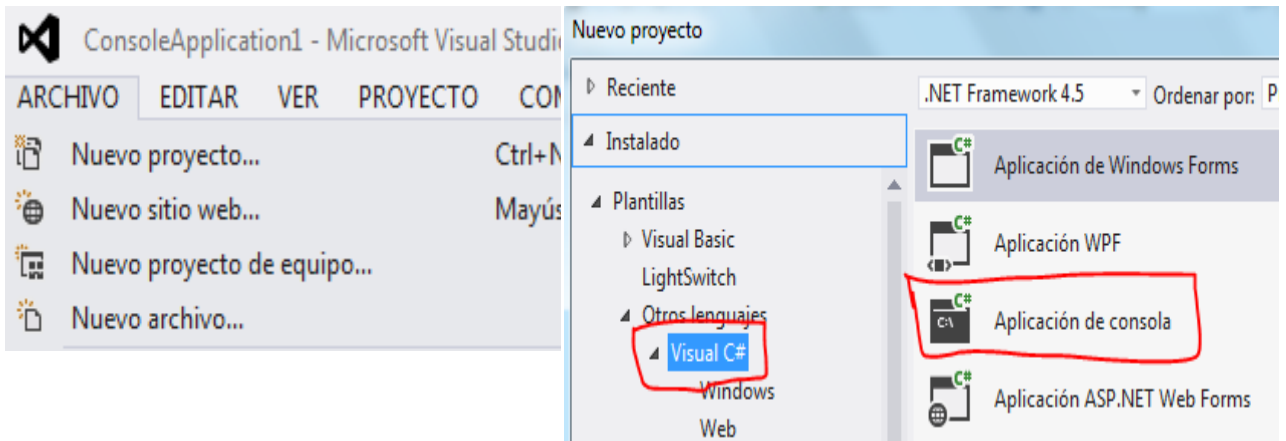


3. Código

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace taller_54
{
    class Program
    {
        static void Main(string[] args)
        {
            byte I = 0;
            byte CAN = 0;
            int MAY = 0;
            int MEN = 0;
            int POS = 0;
            String cad;
            Console.Write("CUANTOS ELEMENTOS:");
            cad=Console.ReadLine();
            CAN = byte.Parse(cad)
            int[] VEC = new int[CAN + 1];
            // INGRESO
            for (I = 1; I <= CAN; I++)
            {
                Console.Write("POSICIÓN {0} ==>", I);
                cad=Console.ReadLine();
                VEC[I] = byte.Parse(cad);
            }
            // PROCESO
            MAY = VEC[1];
            POS = 1;
            MEN = VEC[1];
            for (I = 2; I <= CAN; I++)
            {
                if ((VEC[I] > MAY))
```


OBTENER EL DÍGITO VERIFICADOR DE LA CÉDULA

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

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

namespace taller_55
{
    class Program
    {
        static void Main(string[] args)
        {
            bool VALIDO = false;
            int SUMP = 0;
            int SUMI = 0;
            int ST = 0;
            int P = 0;
            int AUX = 0;
            int DV = 0;
            String cad;
            int[] VEC = new int[10];
            // INGRESO
            for (P = 1; P <= 9; P++)
            {
                Console.Write("POSICIÓN {0} ==>", P);
                cad = Console.ReadLine();
                VEC[P] = int.Parse(cad);
            }
            // SUMATORIA DE PARES
            SUMP = 0;
            SUMI = 0;
            for (P = 2; P <= 8; P += 2)
            {
                SUMP = SUMP + VEC[P];
            }
        }
    }
}
```

```

// SUMATORIA DE IMPARES
for (P = 1; P <= 9; P += 2)
{
    AUX = VEC[P] * 2;
    if (AUX > 9)
        AUX = AUX - 9;
    SUMI = SUMI + AUX;
}
// OBTENCIÓN DE DÍGITO
ST = SUMP + SUMI;
DV = 10 - (ST % 10);
if (DV == 10)
    DV = 0;
Console.WriteLine("EL DÍGITO VERIFICADOR ES: " + DV);
Console.Write("Pulse una Tecla:");
Console.ReadLine();
}
}
}

```

4. GRABAR Y EJECUTAR

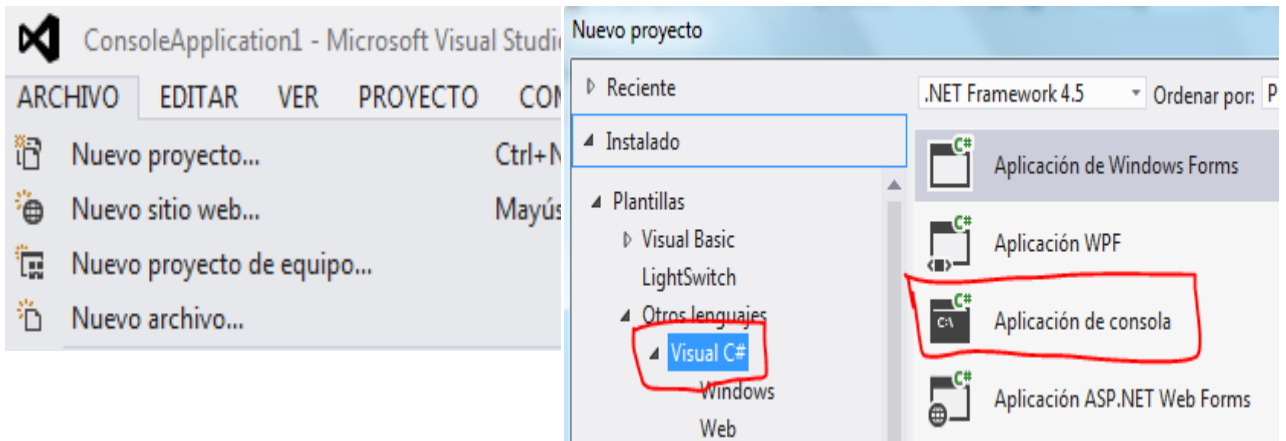
```

file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApp
POSICIÓN 1 ==>1
POSICIÓN 2 ==>8
POSICIÓN 3 ==>0
POSICIÓN 4 ==>4
POSICIÓN 5 ==>7
POSICIÓN 6 ==>5
POSICIÓN 7 ==>2
POSICIÓN 8 ==>6
POSICIÓN 9 ==>2
EL DÍGITO UERIFICADOR ES: 2
Pulse una Tecla:

```


INSERTAR UN ELEMENTO EN UN ARREGLO

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



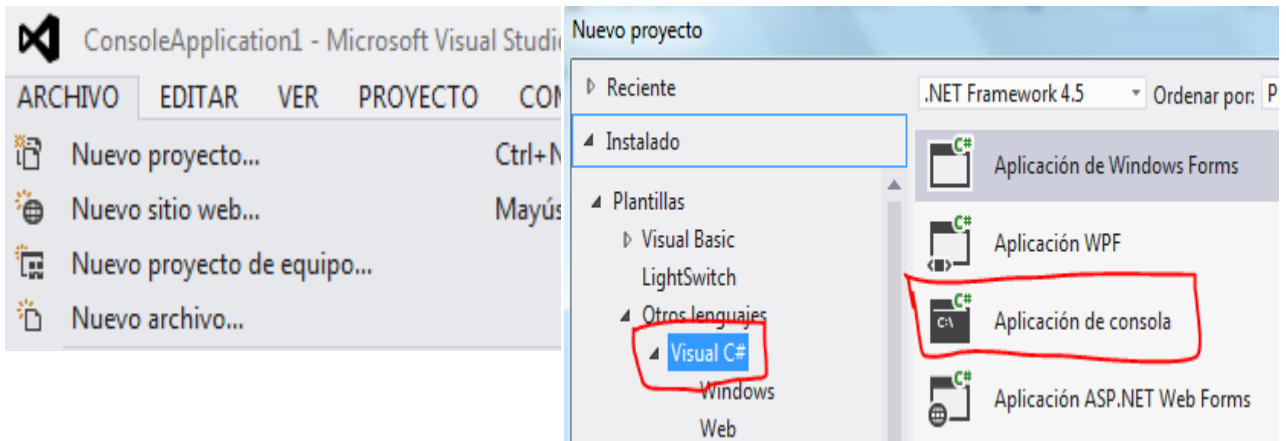
3. CÓDIGO

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

namespace taller_56
{
    class Program
    {
        static void Main(string[] args)
        {
            byte I = 0;
            byte CAN = 0;
            byte POS = 0;
            int ELE = 0;
            String cad;
            Console.Write("CUANTOS ELEMENTOS :");
            cad = Console.ReadLine();
            CAN = byte.Parse(cad);
            int[] VEC = new int[CAN+1];
            int[] NUEVO = new int[CAN + 2];
            // INGRESO
            for (I = 1; I <= CAN; I++) {
                Console.Write("POSICIÓN {0} ==>", I);
                cad = Console.ReadLine();
                VEC[I] = int.Parse(cad);
            }
            Console.Write("ELEMENTO A INSERTAR:");
            cad = Console.ReadLine();
            ELE = int.Parse(cad);
            do {
                Console.Write("POSICIÓN A INSERTAR:");
                cad = Console.ReadLine();
                POS = byte.Parse(cad);
            }
        }
    }
}
```


ELIMINAR UN ELEMENTO EN UN ARREGLO

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



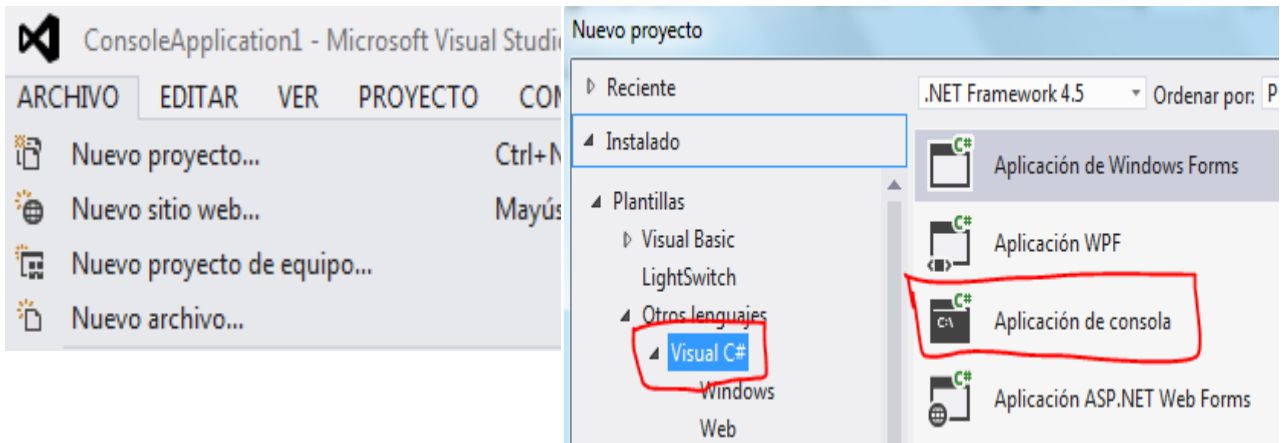
3. CÓDIGO

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

namespace taller_57
{
    class Program
    {
        static void Main(string[] args)
        {
            int I = 0;
            int CAN = 0;
            int POS = 0;
            String cad;
            Console.Write("CUANTOS ELEMENTOS :");
            cad = Console.ReadLine();
            CAN = int.Parse(cad);
            int[] VEC = new int[CAN+1];
            int[] NUEVO = new int[CAN];
            // INGRESO
            for (I = 1; I <= CAN; I++) {
                Console.Write("POSICIÓN {0} ==>", I);
                cad = Console.ReadLine();
                VEC[I] = int.Parse(cad);
            }
            do {
                Console.Write("POSICIÓN A ELIMINAR:");
                cad = Console.ReadLine();
                POS = int.Parse(cad);
            } while (((POS < 1) | (POS > CAN)));
            // PROCESO
            // TRANSLADAMOS DATOS ANTES DE LA POSICIÓN
            for (I = 1; I <= POS - 1; I++) {
```


SUMA DE DOS ARREGLOS DE 5 ELEMENTOS

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola

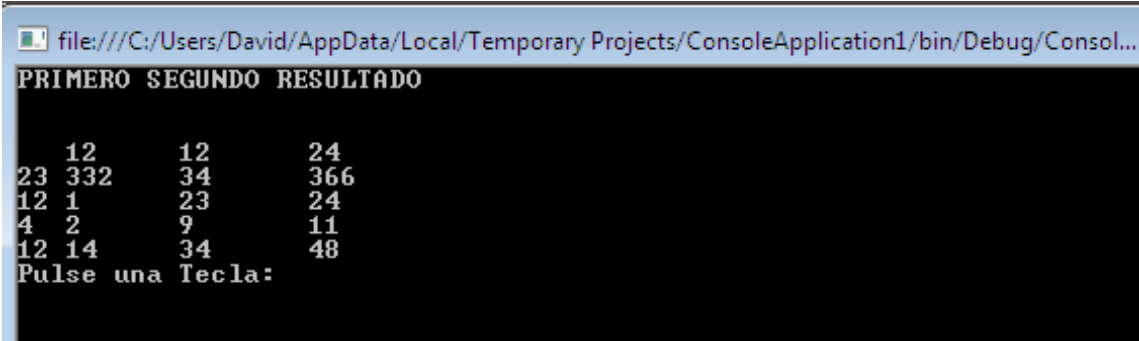


3. CÓDIGO

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            byte I = 0;
            int[] A = new int[10];
            int[] B = new int[10];
            int[] S = new int[6];
            string linea;
            // INGRESO
            Console.WriteLine("PRIMERO SEGUNDO RESULTADO");
            for (I = 1; I <= 5; I++) {
                // ARREGLO 1
                Console.SetCursorPosition(3, I + 2);
                linea=Console.ReadLine();
                A[I] = int.Parse (linea);
                // ARREGLO 2
                Console.ReadLine(); Console.SetCursorPosition(10, I + 2);
                linea=Console.ReadLine();
                B[I] = int.Parse (linea);
                S[I] = A[I] + B[I];
            }
            // SALIDA
            for (I = 1; I <= 5; I++) {
                Console.SetCursorPosition(18, I + 2);
                Console.WriteLine(S[I]);
            }
        }
    }
}
```

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

4. GRABAR Y EJECUTAR

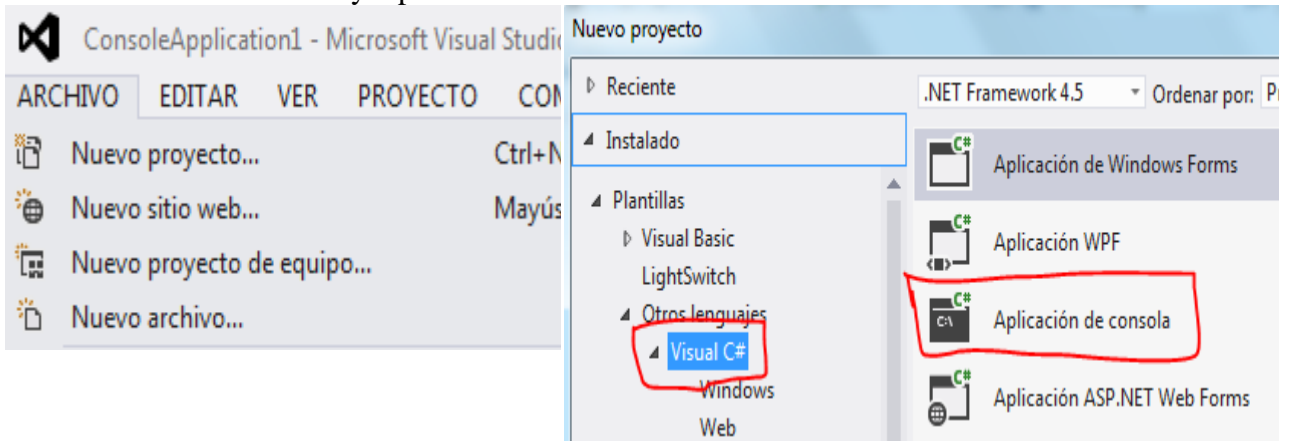


The screenshot shows a console window with the following content:

```
file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/Consol...  
PRIMERO SEGUNDO RESULTADO  
  
    12    12    24  
23 332    34    366  
12 1      23    24  
4  2      9     11  
12 14     34    48  
Pulse una Tecla:
```

SUMA DE DOS ARREGLOS DE 5 ELEMENTOS INTERCALADO

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

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

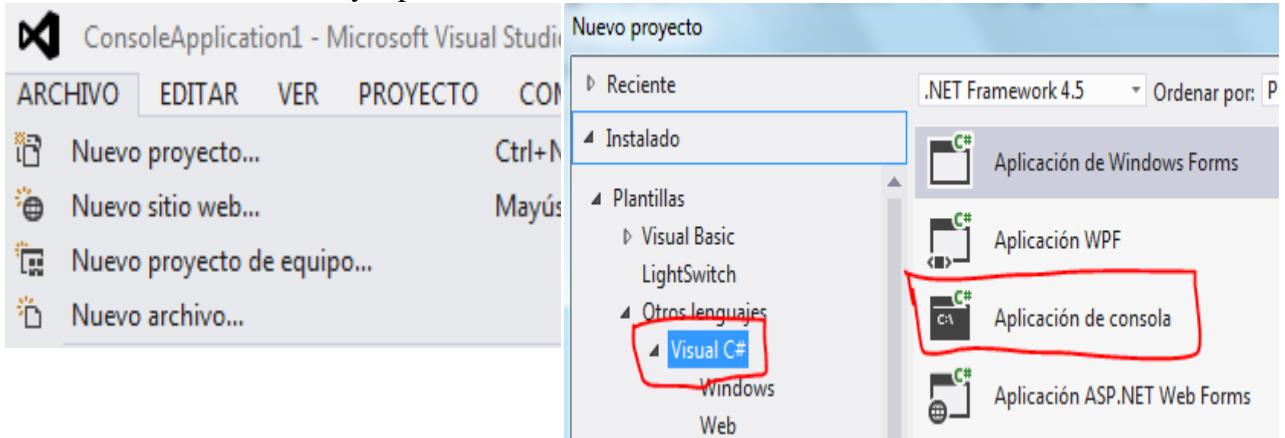
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int I = 0;
            int F = 0;
            int[] A = new int[10];
            int[] B = new int[10];
            int[] S = new int[6];
            string linea;
            // INGRESO
            Console.WriteLine("PRIMERO SEGUNDO RESULTADO");
            for (I = 1; I <= 5; I++)
            {
                // ARREGLO 1
                Console.SetCursorPosition(3, I + 2);
                linea = Console.ReadLine();
                A[I] = int.Parse(linea);
                // ARREGLO 2
                Console.SetCursorPosition(10, I + 2);
                linea = Console.ReadLine();
                B[I] = int.Parse(linea);
                // PROCESO
                F = 5;
                // POSICIÓN DE ARREGLO B
                for (I = 1; I <= 5; I++)
                {
```


EJERCICIO PROPUESTO

Concepto:

Desarrollar un programa que permita cargar 5 nombres de personas y sus edades respectivas. Luego de realizar la carga por teclado de todos los datos imprimir los nombres de las personas mayores de edad (mayores o iguales a 18 años)

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

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

namespace PruebaVector10
{
    class PruebaVector10
    {
        private string[] nombres;
        private int[] edades;

        public void Cargar()
        {
            nombres = new string[5];
            edades = new int[5];
            for (int f = 0; f < nombres.Length; f++)
            {
                Console.Write("Ingrese nombre:");
                nombres[f] = Console.ReadLine();
                Console.Write("Ingrese edad:");
                string linea;
                linea = Console.ReadLine();
                edades[f] = int.Parse(linea);
            }
        }

        public void MayoresEdad()
        {

```

```

        Console.WriteLine("Personas mayores de edad.");
        for (int f = 0; f < nombres.Length; f++)
        {
            if (edades[f] >= 18)
            {
                Console.WriteLine(nombres[f]);
            }
        }
        Console.ReadKey();
    }

    static void Main(string[] args)
    {
        PruebaVector10 pv = new PruebaVector10();
        pv.Cargar();
        pv.MayoresEdad();
    }
}

```

4. GRABAR Y EJECUTAR

```

file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/Consol...
Ingreso nombre:David
Ingreso edad:19
Ingreso nombre:Gabriel
Ingreso edad:22
Ingreso nombre:Estefany
Ingreso edad:14
Ingreso nombre:Patricio
Ingreso edad:50
Ingreso nombre:Mario
Ingreso edad:18
Personas mayores de edad.
David
Gabriel
Patricio
Mario
-

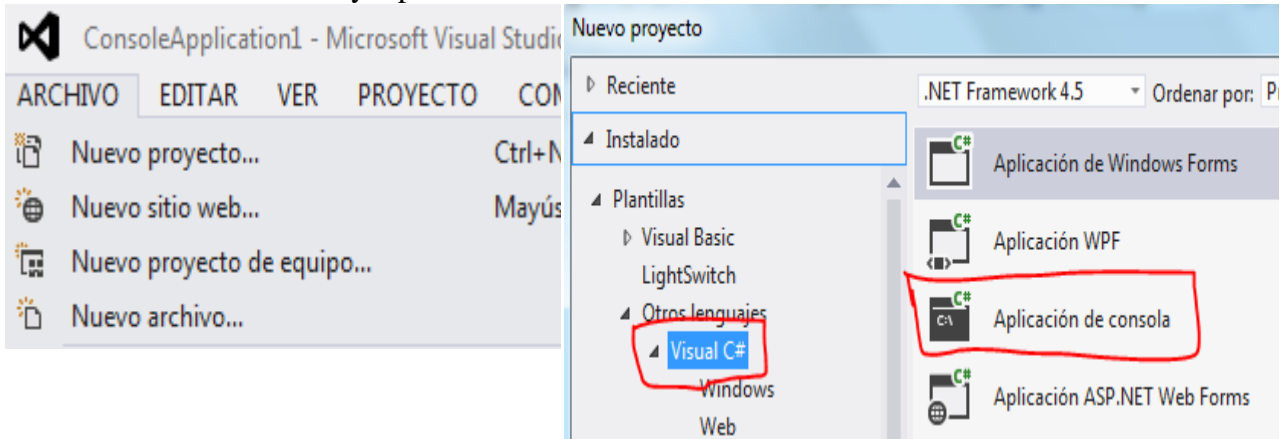
```

EJERCICIO PROPUESTO

Concepto:

Confeccionar un programa que permita cargar los nombres de 5 operarios y sus sueldos respectivos. Mostrar el sueldo mayor y el nombre del operario.

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

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

namespace PruebaVector11
{
    class PruebaVector11
    {
        private string[] nombres;
        private float[] sueldos;

        public void Cargar()
        {
            nombres = new string[5];
            sueldos = new float[5];
            for (int f = 0; f < nombres.Length; f++)
            {
                Console.WriteLine("Ingrese el nombre del empleado:");
                nombres[f] = Console.ReadLine();
                Console.WriteLine("Ingrese el sueldo:");
                string linea;
                linea = Console.ReadLine();
                sueldos[f] = float.Parse(linea);
            }
        }

        public void MayorSueldo()
        {

```

```

float mayor;
int pos;
mayor = sueldos[0];
pos = 0;
for (int f = 1; f < nombres.Length; f++)
{
    if (sueldos[f] > mayor)
    {
        mayor = sueldos[f];
        pos = f;
    }
}
Console.WriteLine("El empleado con sueldo mayor es " + nombres[pos]);
Console.WriteLine("Tiene un sueldo:" + mayor);
Console.ReadKey();
}

static void Main(string[] args)
{
    PruebaVector11 pv = new PruebaVector11();
    pv.Cargar();
    pv.MayorSueldo();
}
}
}

```

4. GRABAR Y EJECUTAR

The screenshot shows a console window with the following text:

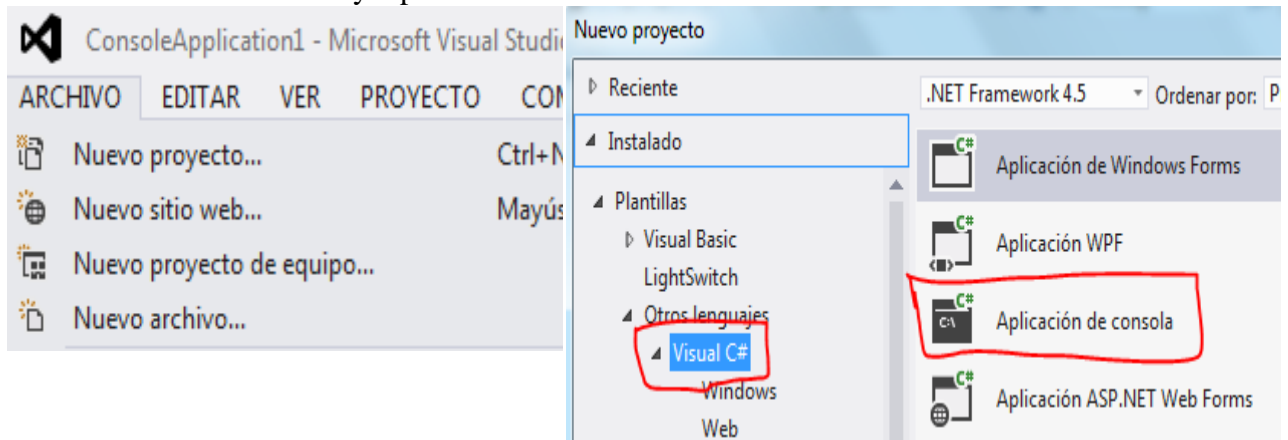
```

file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/Consol...
Ingrese el nombre del empleado:David
Ingrese el sueldo:2000
Ingrese el nombre del empleado:Gabriel
Ingrese el sueldo:2222
Ingrese el nombre del empleado:Mario
Ingrese el sueldo:3333
Ingrese el nombre del empleado:Alfonso
Ingrese el sueldo:4444
Ingrese el nombre del empleado:Armando
Ingrese el sueldo:5555
El empleado con sueldo mayor es Armando
Tiene un sueldo:5555

```

NÚMERO DECIMAL A BINARIO

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



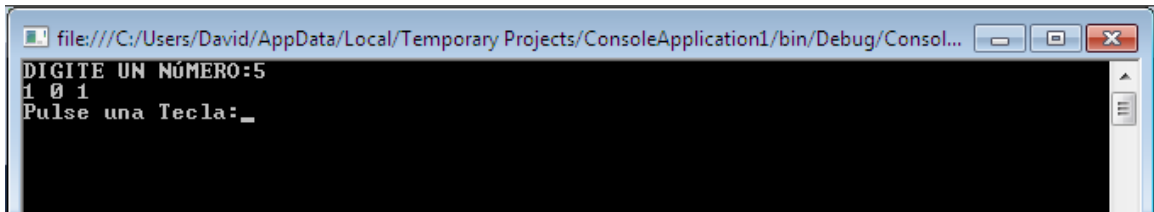
3. CÓDIGO

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

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            const int BASE =2;
            int NUM = 0;
            int AUX = 0;
            int POS = 0;
            int I = 0;
            int [] VEC = new int[21];
            string linea;
            Console.Write("DIGITE UN NÚMERO:");
            linea = Console.ReadLine();
            NUM = int.Parse(linea);
            POS = 1;
            while ((NUM >= BASE))
            {
                AUX = NUM % BASE;
                VEC[POS] = AUX;
                POS = POS + 1;
                NUM = NUM / BASE;
            }
            VEC[POS] = NUM;
            // SALIDA
            for (I = POS; I >= 1; I += -1)
            {
```

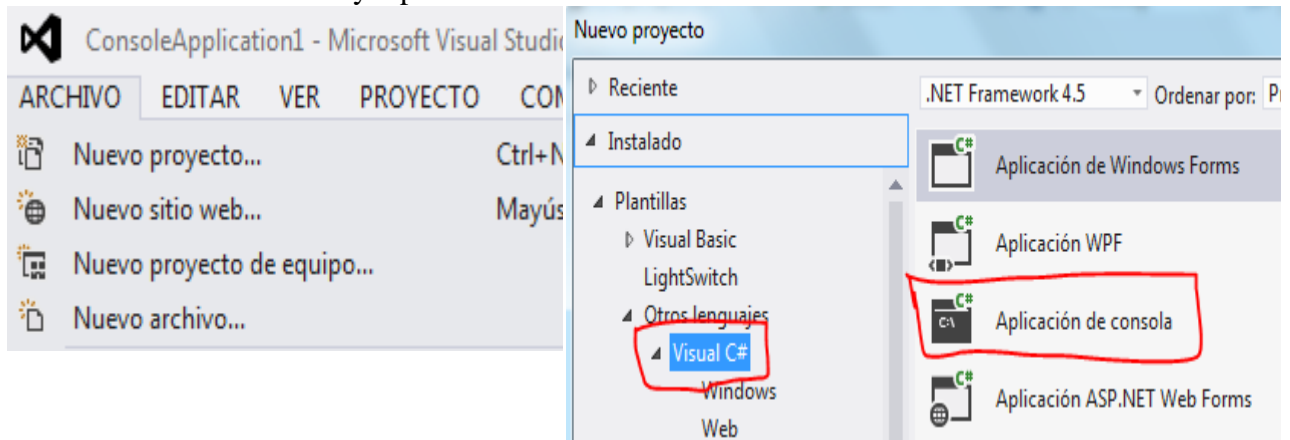
```
        Console.WriteLine(VEC[I] + " ");
    }
    Console.WriteLine();
    Console.WriteLine("Pulse una Tecla:");
    Console.ReadLine();
}
}
```

4. GRABAR Y EJECUTAR



NÚMERO DECIMAL A OCTAL

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



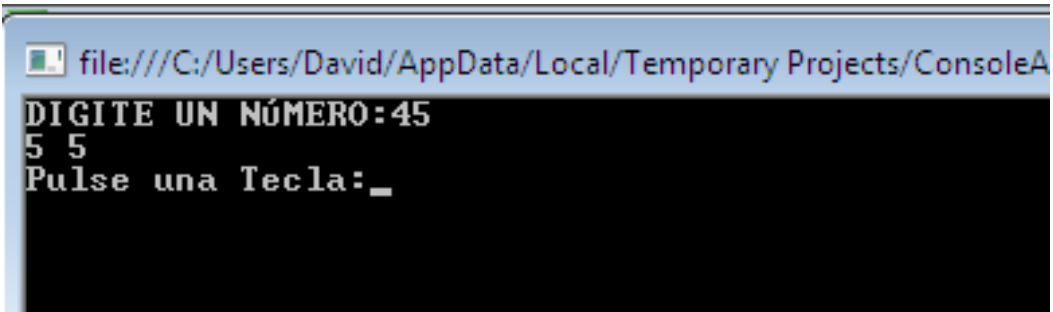
3. CÓDIGO

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

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            const int BASE = 8;
            int NUM = 0;
            int AUX = 0;
            int POS = 0;
            int I = 0;
            string linea;
            int [] VEC = new int[21];
            Console.Write("DIGITE UN NÚMERO:");
            linea = Console.ReadLine();
            NUM = int.Parse(linea);
            POS = 1;
            while ((NUM >= BASE))
            {
                AUX = NUM % BASE;
                VEC[POS] = AUX;
                POS = POS + 1;
                NUM = NUM / BASE;
            }
            VEC[POS] = NUM;
            // SALIDA
            for (I = POS; I >= 1; I += -1)
            {
```

```
        Console.Write(VEC[I] + " ");  
    }  
    Console.WriteLine();  
    Console.Write("Pulse una Tecla:");  
    Console.ReadLine();  
} }  
}
```

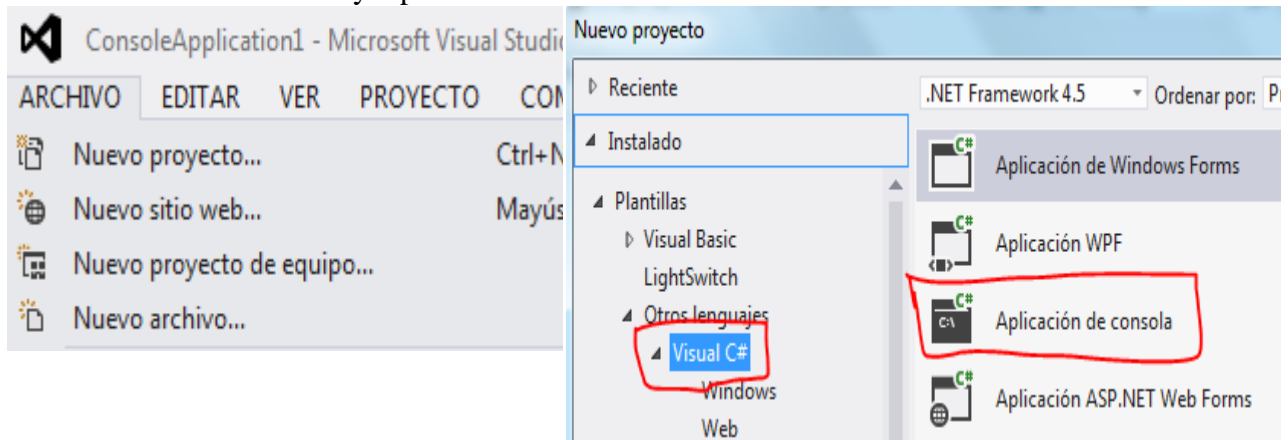
4. GRABAR Y EJECUTAR



```
file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleA  
DIGITE UN NÚMERO:45  
5 5  
Pulse una Tecla:_
```


NÚMERO DECIMAL A HEXADECIMAL

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

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

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            const int BASE = 16;
            int NUM = 0;
            int AUX = 0;
            int POS = 0;
            int I = 0;
            string [] VEC = new string[21];
            string DAT;
            string linea;
            Console.Write("DIGITE UN NÚMERO:");
            linea = Console.ReadLine();
            NUM = int.Parse(linea);
            // PROCESO
            POS = 1;
            while ((NUM >= BASE))
            {
                AUX = NUM % BASE;
                switch (AUX)
                {
                    case 10:
                        DAT = "A";
                        break;
                    case 11:

```

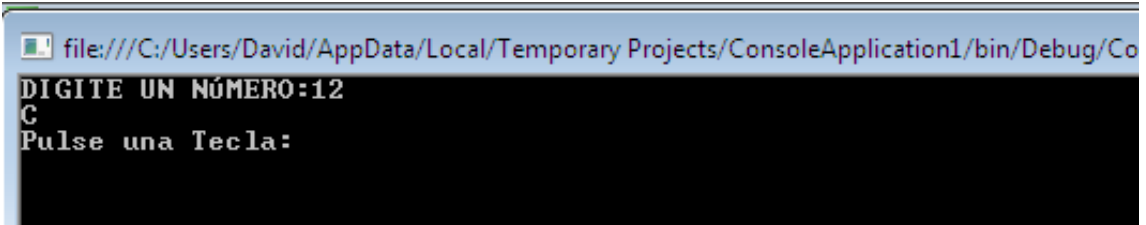
```

        DAT = "B";
        break;
    case 12:
        DAT = "C";
        break;
    case 13:
        DAT = "D";
        break;
    case 14:
        DAT = "E";
        break;
    case 15:
        DAT = "F";
        break;
    default:
        DAT = AUX.ToString();
        break;
    }
    VEC[POS] = DAT;
    POS = POS + 1;
    NUM = NUM / BASE;
}
// FIN DEL WHILE
switch (NUM)
{
    case 10:
        DAT = "A";
        break;
    case 11:
        DAT = "B";
        break;
    case 12:
        DAT = "C";
        break;
    case 13:
        DAT = "D";
        break;
    case 14:
        DAT = "E";
        break;
    case 15:
        DAT = "F";
        break;
    default:
        DAT = Convert.ToString(NUM);
        break;
}
VEC[POS] = DAT;
// SALIDA
for (I = POS; I >= 1; I += -1)
{
    Console.Write(VEC[I] + " ");
}
Console.WriteLine();
Console.Write("Pulse una Tecla:");
Console.ReadLine();
}
}

```

}

4. GRABAR Y EJECUTAR



The screenshot shows a console window with a title bar containing the file path: file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/Co. The console output consists of three lines: "DIGITE UN NÚMERO:12", "C", and "Pulse una Tecla:". The text is displayed in a monospaced font on a black background.

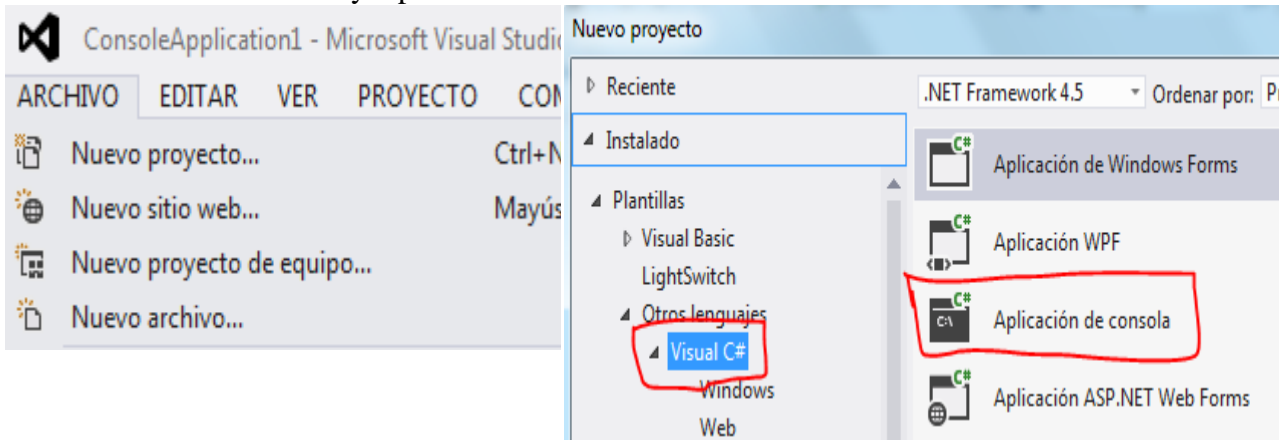
```
file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/Co
DIGITE UN NÚMERO:12
C
Pulse una Tecla:
```

EJERCICIO PROPUESTO

Concepto:

Confeccionar un programa que permita obtener una serie de sumas ejemplo primer número=2, segundo número=4; suma1=2, suma2=6; es decir las sumas son resultado de la suma anterior mas el numero ingresado correspondiente.

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

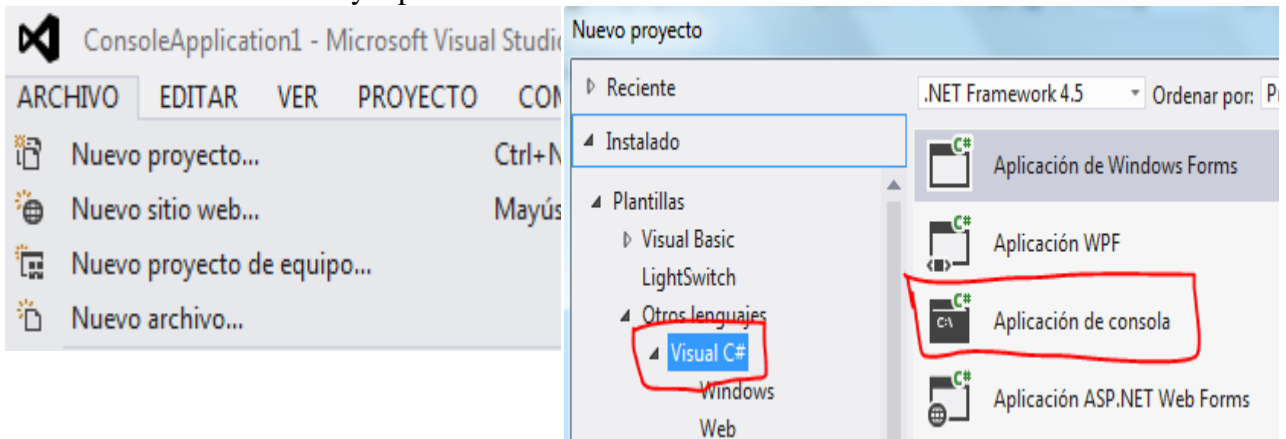
```
using System;
public class Ejemplo36
{
    public static void Main()
    {
        int[] numero = new int[5]; /* Un array de 5 números enteros */
        int suma; /* Un entero que será la suma */
        int i; /* Para recorrer los elementos */
        for (i = 0; i <= 4; i++) /* Pedimos los datos */
        {
            Console.WriteLine("Introduce el dato numero {0}: ", i + 1);
            numero[i] = Convert.ToInt32(Console.ReadLine());
        }
        suma = 0; /* Valor inicial de la suma */
        for (i = 0; i <= 4; i++) /* Y hallamos la suma repetitiva */
        {
            suma += numero[i];
            Console.WriteLine("Su suma es {0}", suma);
        }
        Console.ReadKey();
    }
}
```

4. GRABAR Y EJECUTAR

```
file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/Consol...
Introduce el dato numero 1: 12
Introduce el dato numero 2: 32
Introduce el dato numero 3: 43
Introduce el dato numero 4: 23
Introduce el dato numero 5: 21
Su suma es 12
Su suma es 44
Su suma es 87
Su suma es 110
Su suma es 131
```

ORDENAMIENTO DE UN ARREGLO

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

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

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int I = 0;
            int K = 0;
            int CAN = 0;
            int AUX = 0;
            string linea;
            Console.WriteLine("CUANTOS ELEMENTOS MÁX=12:");
            linea=Console.ReadLine();
            CAN = int.Parse(linea);
        }
    }
}
```

```

int[] VEC = new int[CAN + 1];
// INGRESO
for (I = 1; I <= CAN; I++)
{
    Console.Write("POSICIÓN {0} ==>", I);
    linea = Console.ReadLine();
    VEC[I] = int.Parse(linea);
}
// PROCESO
for (I = 1; I <= CAN; I++)
{
    for (K = I; K <= CAN; K++)
    {
        if ((VEC[K] > VEC[I]))
        {
            AUX = VEC[K];
            VEC[K] = VEC[I];
            VEC[I] = AUX;
        }
    }
}
Console.WriteLine();
// ARREGLO
Console.WriteLine("ARREGLO ORDENADO DESCENDENTE");
for (I = 1; I <= CAN; I++)
{
    Console.WriteLine(VEC[I]);
}
Console.Write("Pulse una Tecla:");
Console.ReadLine();
}
}
}

```

4. GRABAR Y EJECUTAR

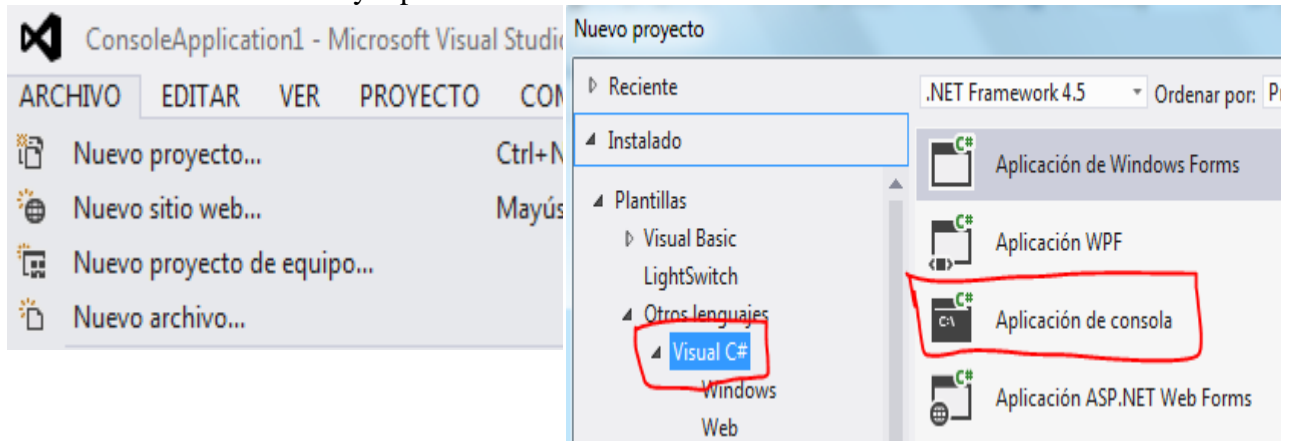
```

file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/Consol
CUANTOS ELEMENTOS MÁX=12:4
POSICIÓN 1 ==>12
POSICIÓN 2 ==>32
POSICIÓN 3 ==>32
POSICIÓN 4 ==>1
ARREGLO ORDENADO DESCENDENTE
32
32
12
1
Pulse una Tecla:_

```

ORDENAMIENTO DE UN ARREGLO MÉTODO BURBUJA

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



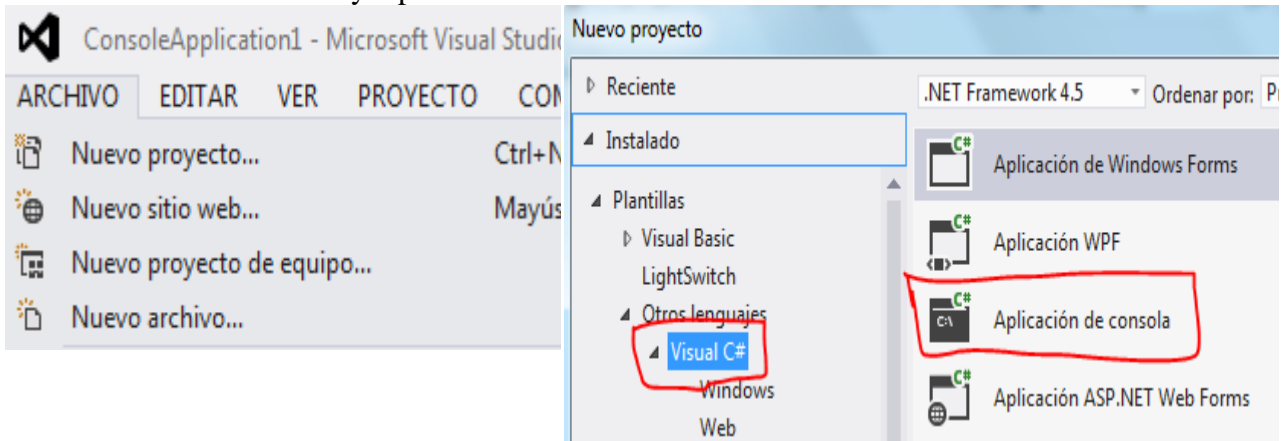
3. CÓDIGO

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

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int I = 0;
            int K = 0;
            int CAN = 0;
            int AUX = 0;
            string linea;
            Console.WriteLine("CUANTOS ELEMENTOS MÁX=12:");
            linea=Console.ReadLine();
            CAN = int.Parse(linea);
            int[] VEC = new int[CAN + 1];
            // INGRESO
            for (I = 1; I <= CAN; I++)
            {
                Console.WriteLine("POSICIÓN {0} ==>", I);
                linea = Console.ReadLine();
                VEC[I] = int.Parse(linea);
            }
            // PROCESO
            for (I = 1; I <= CAN - 1; I++)
            {
                for (K = I + 1; K <= CAN; K++)
                {
                    if ((VEC[I] < VEC[K]))
                    {
```


BÚSQUEDA DE UN ELEMENTO EN UN ARREGLO

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



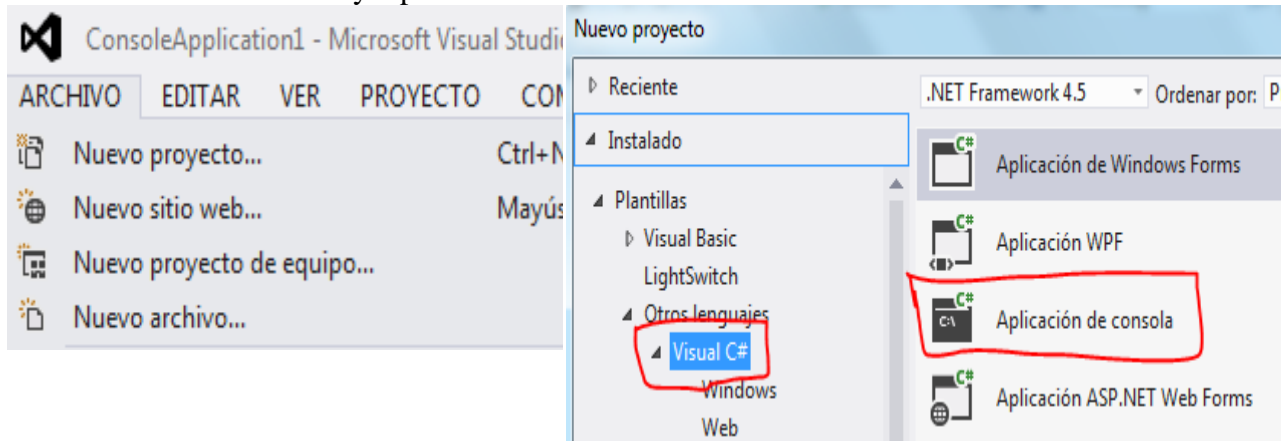
3. CÓDIGO

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

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int I = 0;
            int CAN = 0;
            int TOT = 0;
            int BUS = 0;
            string linea;
            Console.Write("CUANTOS ELEMENTOS:");
            linea = Console.ReadLine();
            CAN = int.Parse(linea);
            int[] VEC = new int[CAN + 1];
            // INGRESO
            for (I = 1; I <= CAN; I++)
            {
                Console.Write("POSICIÓN {0} ==>", I);
                linea = Console.ReadLine();
                VEC[I] = int.Parse(linea);
            }
            Console.Write("ELEMENTO A BUSCAR:");
            linea = Console.ReadLine();
            BUS = int.Parse(linea);
            // PROCESO
            TOT = 0;
            for (I = 1; I <= CAN; I++)
            {
```


BÚSQUEDA BINARIA DE UN ELEMENTO EN UN ARREGLO

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

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

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int I = 0;
            int CAN = 0;
            int J = 0;
            int ALTO = 0;
            int BAJO = 0;
            int CENTRAL = 0;
            int BUS = 0;
            int AUX = 0;
            string linea;
            Random rnd = new Random();
            Console.Write("CUANTOS ELEMENTOS:");
            linea=Console.ReadLine();
            CAN = int.Parse(linea);
            int[] VEC = new int[CAN + 1];
            bool ENCONTRADO = false;
            // INGRESO
            for (I = 1; I <= CAN; I++)
            {
                VEC[I] = rnd.Next(0, 99);
            }
            // SALIDA DEL ARREGLO ALEATORIO
            for (I = 1; I <= CAN; I++)
            {
```

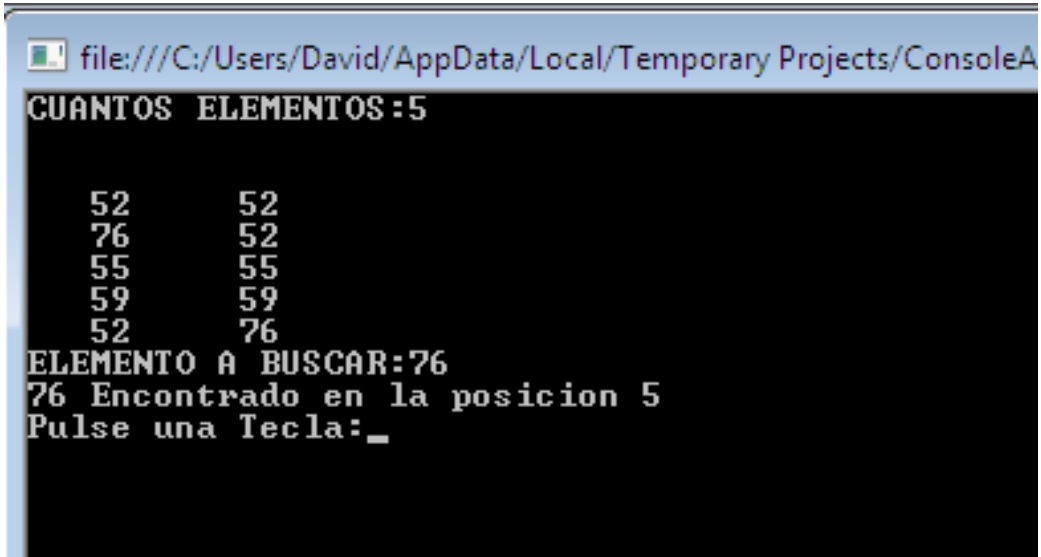
```

        Console.SetCursorPosition(3, I + 2);
        Console.WriteLine(VEC[I]);
    }
    // PRIMERO ORDENAMOS EL ARREGLO
    for (J = 1; J <= CAN; J++)
    {
        for (I = 1; I <= CAN - 1; I++)
        {
            if ((VEC[I] > VEC[I + 1]))
            {
                AUX = VEC[I];
                VEC[I] = VEC[I + 1];
                VEC[I + 1] = AUX;
            }
        }
    }
    // SALIDA DEL ARREGLO ORDENADO
    for (I = 1; I <= CAN; I++)
    {
        Console.SetCursorPosition(10, I + 2);
        Console.WriteLine(VEC[I]);
    }
    // AHORA SI LA BÚSQUEDA
    Console.Write("ELEMENTO A BUSCAR:");
    linea = Console.ReadLine();
    BUS = int.Parse(linea);
    BAJO = 1;
    ALTO = CAN;
    // CENTRAL = (BAJO + ALTO) / 2
    while ((BAJO <= ALTO) & (ENCONTRADO == false))
    {
        CENTRAL = (BAJO + ALTO) / 2;
        if ((VEC[CENTRAL] == BUS))
        {
            ENCONTRADO = true;
        }
        else
        {
            if ((VEC[CENTRAL] > BUS))
            {
                ALTO = CENTRAL - 1;
            }
            else
            {
                BAJO = CENTRAL + 1;
            }
        }
    }
    if ((ENCONTRADO))
    {
        Console.WriteLine("{0} Encontrado en la posicion {1}", BUS, CENTRAL);
    }
    else
    {
        Console.WriteLine("No existe {0} ", BUS);
    }
    Console.Write("Pulse una Tecla.");
    Console.ReadLine();

```

```
}  
}  
}
```

4. GRABAR Y EJECUTAR

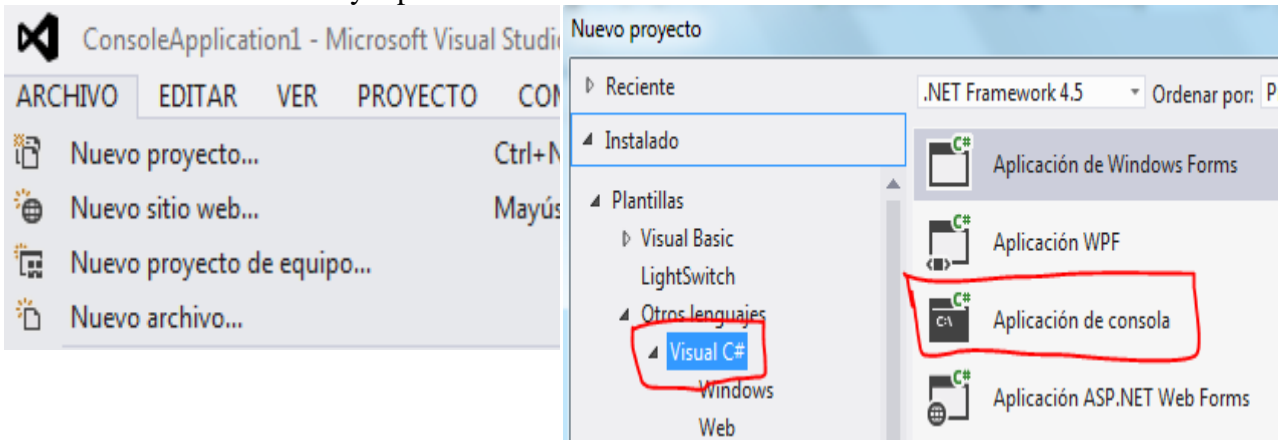


EJERCICIO PROPUESTO

Concepto:

Crear una matriz de $n * m$ filas (cargar n y m por teclado) Imprimir la matriz completa y la última fila.

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

```
using System;  
using System.Collections.Generic;
```

```

using System.Linq;
using System.Text;

namespace Matriz5
{
    class Matriz5
    {
        private int[,] mat;

        public void Cargar()
        {
            Console.WriteLine("Cuantas fila tiene la matriz:");
            string linea;
            linea = Console.ReadLine();
            int filas = int.Parse(linea);
            Console.WriteLine("Cuantas columnas tiene la matriz:");
            linea = Console.ReadLine();
            int columnas = int.Parse(linea);
            mat = new int[filas, columnas];
            for (int f = 0; f < mat.GetLength(0); f++)
            {
                for (int c = 0; c < mat.GetLength(1); c++)
                {
                    Console.WriteLine("Ingreso componente:");
                    linea = Console.ReadLine();
                    mat[f, c] = int.Parse(linea);
                }
            }
        }

        public void Imprimir()
        {
            for (int f = 0; f < mat.GetLength(0); f++)
            {
                for (int c = 0; c < mat.GetLength(1); c++)
                {
                    Console.WriteLine(mat[f, c] + " ");
                }
                Console.WriteLine();
            }
        }

        public void ImprimirUltimaFila()
        {
            Console.WriteLine("Ultima fila");
            for (int c = 0; c < mat.GetLength(1); c++)
            {
                Console.WriteLine(mat[mat.GetLength(0) - 1, c] + " ");
            }
        }

        static void Main(string[] args)
        {
            Matriz5 ma = new Matriz5();
            ma.Cargar();
            ma.Imprimir();
            ma.ImprimirUltimaFila();
            Console.ReadKey();
        }
    }
}

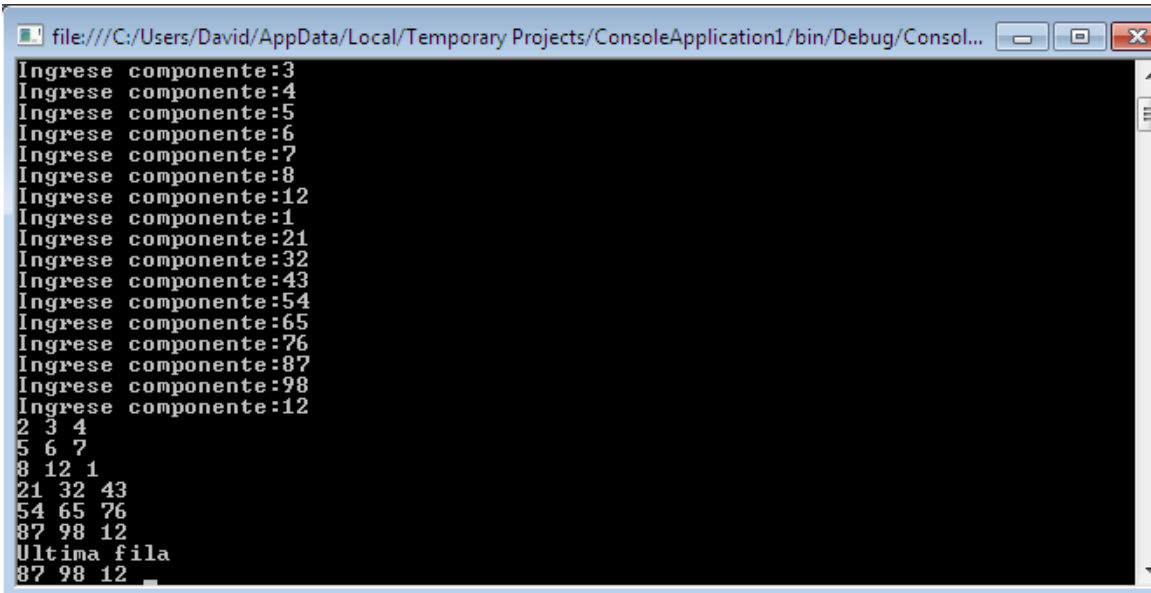
```

```

    }
}
}

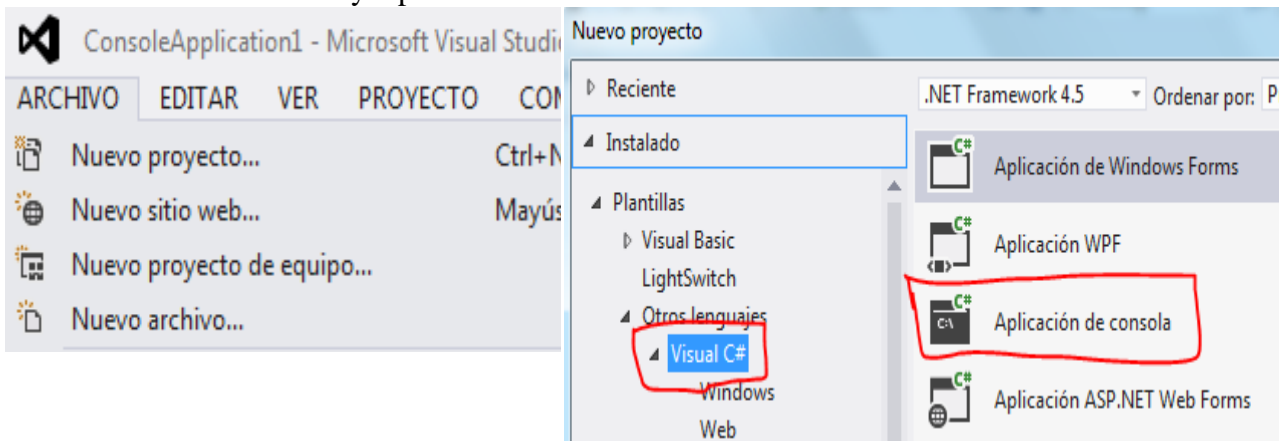
```

4. GRABAR Y EJECUTAR



TABLAS DE MULTIPLICAR EN UNA MATRIZ DE NxN

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

```

```

namespace ConsoleApplication1

```

```

{
    class Program
    {
        static void Main(string[] args)
        {
            int F = 0;
            int C = 0;
            int N = 0;
            int M = 0;
            string linea;
            // GENERO LOS NÚMEROS
            Console.Write("FILAS DE LA MATRIZ:");
            linea = Console.ReadLine();
            N = int.Parse(linea);
            Console.Write("COLUMNAS DE LA MATRIZ:");
            linea = Console.ReadLine();
            M = int.Parse(linea);
            int[,] MAT = new int[N + 1, M + 1];
            for (F = 1; F <= N; F++)
            {
                for (C = 1; C <= M; C++)
                {
                    MAT[F, C] = F * C;
                }
            }
            // SALIDA
            for (F = 1; F <= N; F++)
            {
                for (C = 1; C <= M; C++)
                {
                    // APROVECHAMOS CICLOS Y GENERAMOS POSICIÓN EN PANTALLA
                    // PARA COLUMNA Y FILA
                    Console.SetCursorPosition(C * 4, F + 1);
                    Console.Write(MAT[F, C]);
                }
            }
            Console.WriteLine();
            Console.Write("Pulse una Tecla:");
            Console.ReadLine();
        }
    }
}

```

4. GRABAR Y EJECUTAR

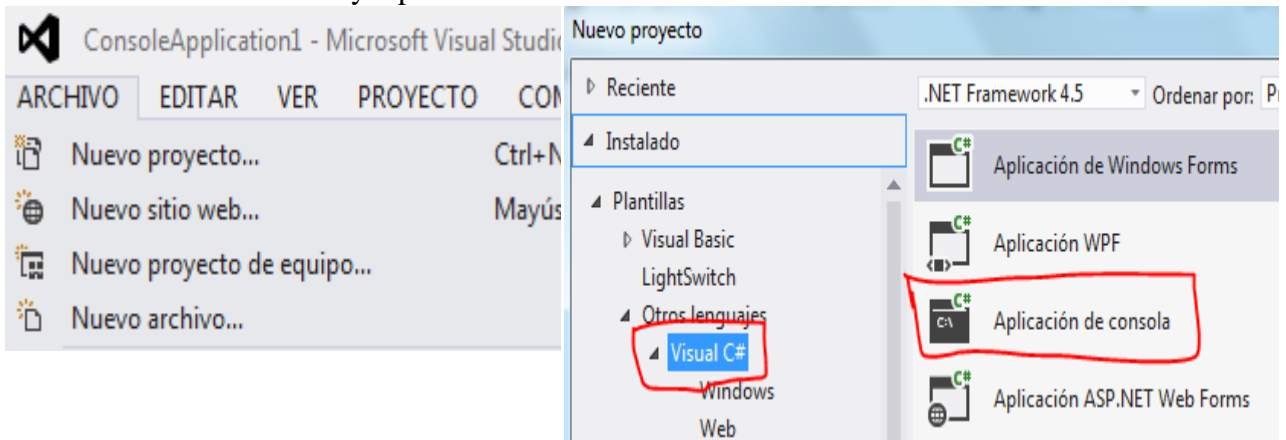
```

file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/
FILAS DE LA MATRIZ:3
COLUMNAS DE LA MATRIZ:2
 1  2
 2  4
 3  6
Pulse una Tecla:_

```


GENERAR ALEATORIOS EN UNA MATRIZ DE 5x5

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

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

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int F = 0;
            int C = 0;
            int[,] MAT = new int[6, 6];
            Random rnd = new Random();
            // GENERO LOS NÚMEROS
            for (F = 1; F <= 5; F++)
            {
                for (C = 1; C <= 5; C++)
                {
                    MAT[F, C] = rnd.Next(0, 99);
                }
            }
            // SALIDA
            for (F = 1; F <= 5; F++)
            {
                for (C = 1; C <= 5; C++)
                {
                    // APROVECHAMOS CICLOS Y GENERAMOS POSICIÓN EN PANTALLA
                    // PARA COLUMNA Y FILA
                    Console.SetCursorPosition(C * 4, F + 1);
                    Console.Write(MAT[F, C]);
                }
            }
        }
    }
}
```

```

    }
    Console.WriteLine();
    Console.WriteLine("Pulse una Tecla:");
    Console.ReadLine();
}
}
}

```

4. GRABAR Y EJECUTAR

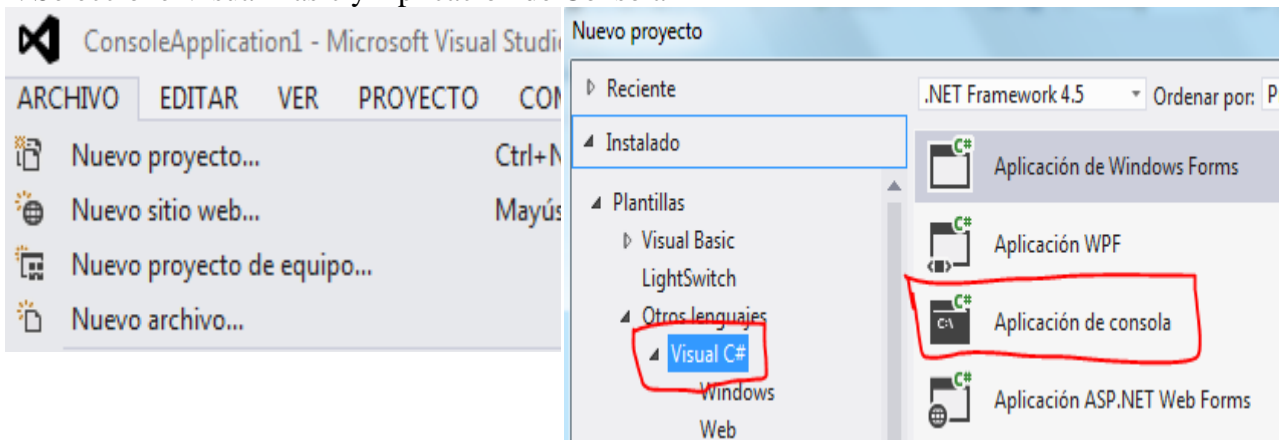
```

file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debu
97 28 51 20 51
37 58 79 8 30
52 95 8 46 93
11 31 98 94 55
27 35 2 25 12
Pulse una Tecla:

```

SUMAR ELEMENTOS DE UNA MATRIZ DE NxN

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)

```

```

    {
        int F = 0;
        int C = 0;
        int N = 0;
        int SUM = 0;
        string linea;
        // INGRESO
        Console.Write("TAMAÑO DE LA MATRIZ:");
        linea=Console.ReadLine();
        N = int.Parse(linea);
        int[,] MAT = new int[N + 1, N + 1];
        for (F = 1; F <= N; F++)
        {
            for (C = 1; C <= N; C++)
            {
                Console.SetCursorPosition(C * 4, F + 1);
                linea = Console.ReadLine();
                MAT[F, C] = int.Parse(linea);
            }
        }
        // PROCESO
        for (F = 1; F <= N; F++)
        {
            for (C = 1; C <= N; C++)
            {
                SUM = SUM + MAT[F, C];
            }
        }
        // SALIDA
        Console.WriteLine();
        Console.WriteLine("SUMA TOTAL ES:" + SUM);
        Console.WriteLine("PROMEDIO TOTAL ES:" + SUM / (N * N));
        Console.Write("Pulse una Tecla:");
        Console.ReadLine();
    }
}
}

```

4. GRABAR Y EJECUTAR

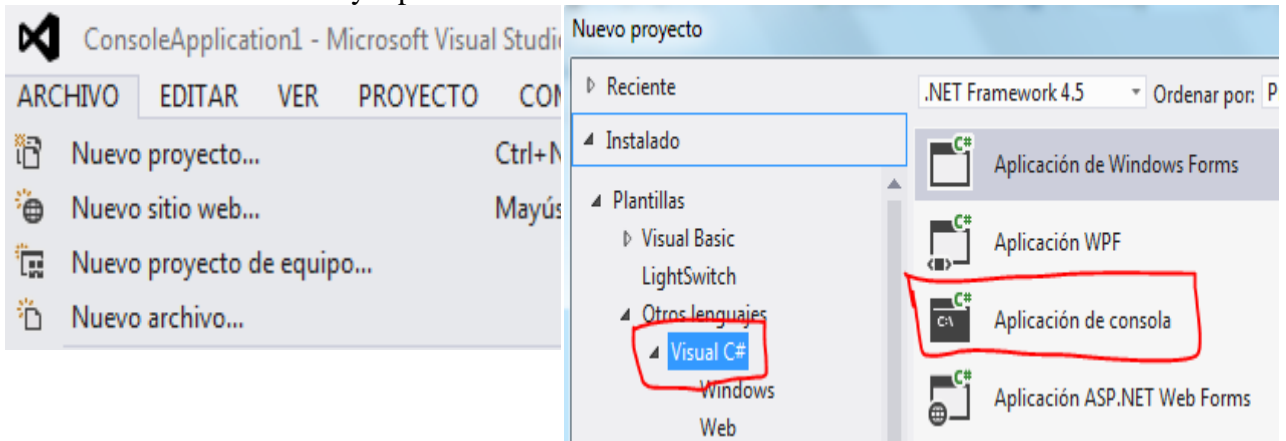
```

file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/Conso
TAMAÑO DE LA MATRIZ:3
    2  3  4
    5  2  1
   45  3  2
SUMA TOTAL ES:67
PROMEDIO TOTAL ES:7
Pulse una Tecla:_

```

SUMAR ELEMENTOS DE FILA Y UNA COLUMNA MATRIZ DE 5x5

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

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

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int F = 0;
            int C = 0;
            int SUMF = 0;
            int SUMC = 0;
            string linea;
            // INGRESO
            int[,] MAT = new int[6, 6];
            for (F = 1; F <= 5; F++)
            {
                for (C = 1; C <= 5; C++)
                {
                    Console.SetCursorPosition(C * 4, F + 1);
                    linea = Console.ReadLine();
                    MAT[F, C] = int.Parse (linea );
                }
            }
            // FILA 2
            SUMF = 0;
            for (C = 1; C <= 5; C++)
            {
                SUMF = SUMF + MAT[2, C];
            }
            // COLUMNNA 3
```

```

SUMC = 0;
for (F = 1; F <= 5; F++)
{
    SUMC = SUMC + MAT[F, 3];
}
// SALIDA
Console.WriteLine();
Console.WriteLine("SUMA FILA 2 ES:" + SUMF);
Console.WriteLine("SUMA COLUMNA 3 ES:" + SUMC);
Console.Write("Pulse una Tecla:");
Console.ReadLine();
}
}
}

```

4. GRABAR Y EJECUTAR

```

file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/ConsoleApplication1.exe
2 3 4 2 12
34 54 21 23 2
3 4 5 6 7
2 23 41 21 52
45 98 7 89 13

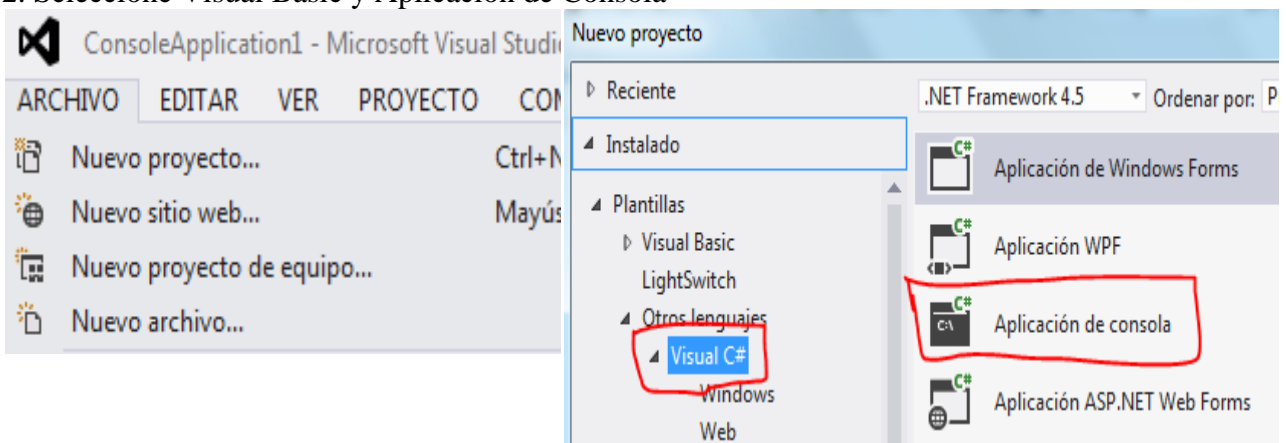
SUMA FILA 2 ES:134
SUMA COLUMNA 3 ES:78
Pulse una Tecla:

```

SUMAR ELEMENTOS DE DIAGONAL PRINCIPAL Y SECUNDARIA

MATRIZ DE NxN

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int F = 0;
            int C = 0;
            int N = 0;
            int SUMP = 0;
            int SUMS = 0;
            string linea;
            // INGRESO
            // PARA COMODIDAD GENERAMOS VALORES PARA LA MATRIZ
            Console.Write("TAMAÑO DE LA MATRIZ:");
            Random rnd = new Random();
            linea = Console.ReadLine();
            N = int.Parse(linea);
            int[,] MAT = new int[N + 1, N + 1];
            for (F = 1; F <= N; F++)
            {
                for (C = 1; C <= N; C++)
                {
                    MAT[F, C] = rnd.Next(0, 9) ;
                    Console.SetCursorPosition(C * 4, F + 1);
                    Console.Write(MAT[F, C]);
                }
            }
            // DIAGONAL PRINCIPAL
            SUMP = 0;
            for (F = 1; F <= N; F++)
            {
                SUMP = SUMP + MAT[F, F];
            }
            // DIAGONAL SECUNDARIA
            SUMS = 0;
            C = N;
            for (F = 1; F <= N; F++)
            {
                SUMS = SUMS + MAT[F, C];
                C = C - 1;
            }
            // SALIDA
            Console.WriteLine();
            Console.WriteLine("SUMA DIAGONAL PRINCIPAL ES :" + SUMP);
            Console.WriteLine("SUMA DIAGONAL SECUNDARIA ES:" + SUMS);
            Console.Write("Pulse una Tecla:");
            Console.ReadLine();
        }
    }
}

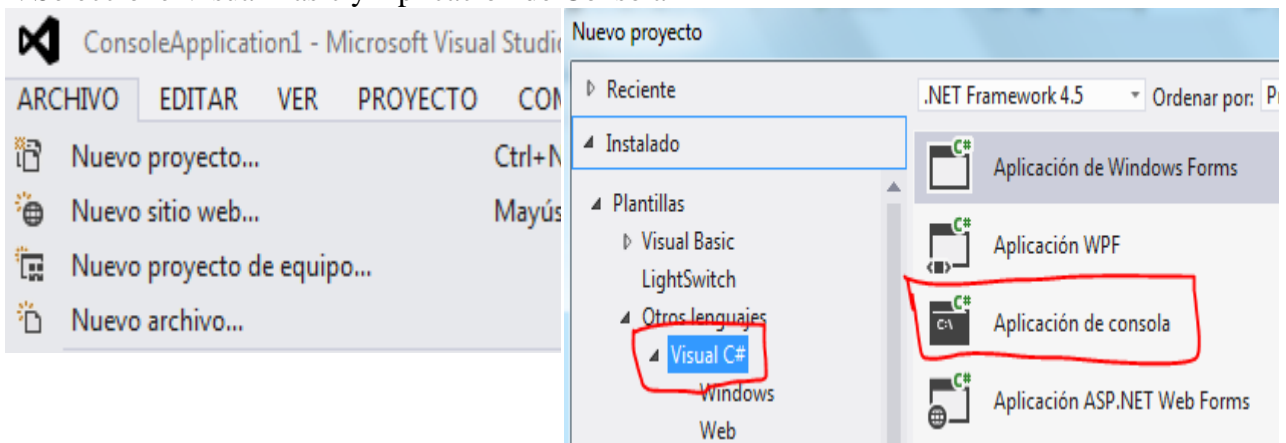
```

4. GRABAR Y EJECUTAR

```
file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/ConsoleApplication1.exe
TAMAÑO DE LA MATRIZ:4
  4  4  2  5
  3  0  3  4
  7  8  6  3
  3  1  6  6
SUMA DIAGONAL PRINCIPAL ES :16
SUMA DIAGONAL SECUNDARIA ES:19
Pulse una Tecla: _
```

FORMAR LA LETRA A EN UNA MATRIZ DE NxN

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

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

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int F = 0;
            int C = 0;
            int N = 0;
            int MI = 0;
            string linea;
            // MATRIZ EN BLANCO
        }
    }
}
```

```

Console.WriteLine("TAMAÑO DE LA MATRIZ IMPAR:");
linea = Console.ReadLine();
N = int.Parse(linea );
N = (N % 2 == 0 ? N + 1 : N);
string [,] MAT = new string [N + 1, N + 1];
// MATRIZ EN BLANCO
for (F = 1; F <= N; F++)
{
    for (C = 1; C <= N; C++)
    {
        MAT[F, C] = " ";
        // UN ESPACIO EN BLANCO
    }
}
// FORMAMOS LA LETRA
MI = N / 2 + 1;
for (F = 1; F <= N; F++)
{
    MAT[F, 1] = "A";
    MAT[F, N] = "A";
    MAT[MI, F] = "A";
    MAT[1, F] = "A";
}
// SALIDA
for (F = 1; F <= N; F++)
{
    for (C = 1; C <= N; C++)
    {
        Console.SetCursorPosition(C, F + 1);
        Console.Write(MAT[F, C]);
    }
}
Console.WriteLine();
Console.Write("Pulse una Tecla:");
Console.ReadLine();
}
}
}

```

4. GRABAR Y EJECUTAR

```

file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/Consol...
TAMAÑO DE LA MATRIZ IMPAR:5
AAAAA
A  A
AAAAA
A  A
AAAAA
Pulse una Tecla:

```


EJERCICIO PROPUESTO

Concepto:

Se tiene la siguiente información:

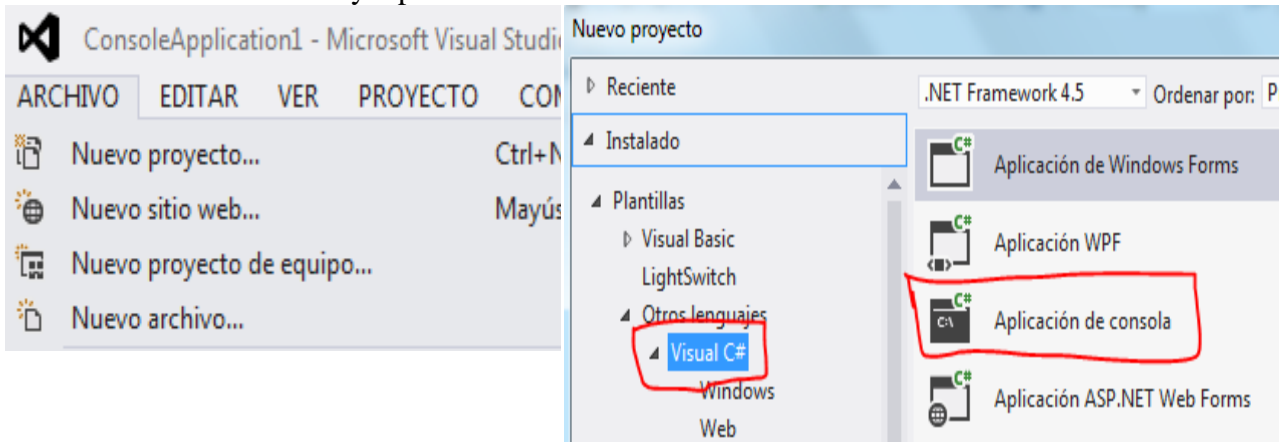
- Nombres de 4 empleados.
- Ingresos en concepto de sueldo, cobrado por cada empleado, en los últimos 3 meses.

Confeccionar el programa para:

- Realizar la carga de la información mencionada.
- Generar un vector que contenga el ingreso acumulado en sueldos en los últimos 3 meses para cada empleado.
- Mostrar por pantalla el total pagado en sueldos a todos los empleados en los últimos 3 meses
- Obtener el nombre del empleado que tuvo el mayor ingreso acumulado

1. Abrir un nuevo Proyecto

2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

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

namespace Matriz9
{
    class Matriz9
    {
        private string[] empleados;
        private int[,] sueldos;
        private int[] sueldostot;

        public void Cargar()
        {
            empleados = new String[4];
            sueldos = new int[4, 3];
            for (int f = 0; f < empleados.Length; f++)
            {
                Console.WriteLine("Ingrese el nombre del empleado:");
                empleados[f] = Console.ReadLine();
                for (int c = 0; c < sueldos.GetLength(1); c++)
```

```

        {
            Console.Write("Ingrese sueldo:");
            string linea;
            linea = Console.ReadLine();
            sueldos[f, c] = int.Parse(linea);
        }
    }
}

public void CalcularSumaSuealdos()
{
    sueldostot = new int[4];
    for (int f = 0; f < sueldos.GetLength(0); f++)
    {
        int suma = 0;
        for (int c = 0; c < sueldos.GetLength(1); c++)
        {
            suma = suma + sueldos[f, c];
        }
        sueldostot[f] = suma;
    }
}

public void ImprimirTotalPagado()
{
    Console.WriteLine("Total de sueldos pagados por empleado.");
    for (int f = 0; f < sueldostot.Length; f++)
    {
        Console.WriteLine(empleados[f] + " - " + sueldostot[f]);
    }
}

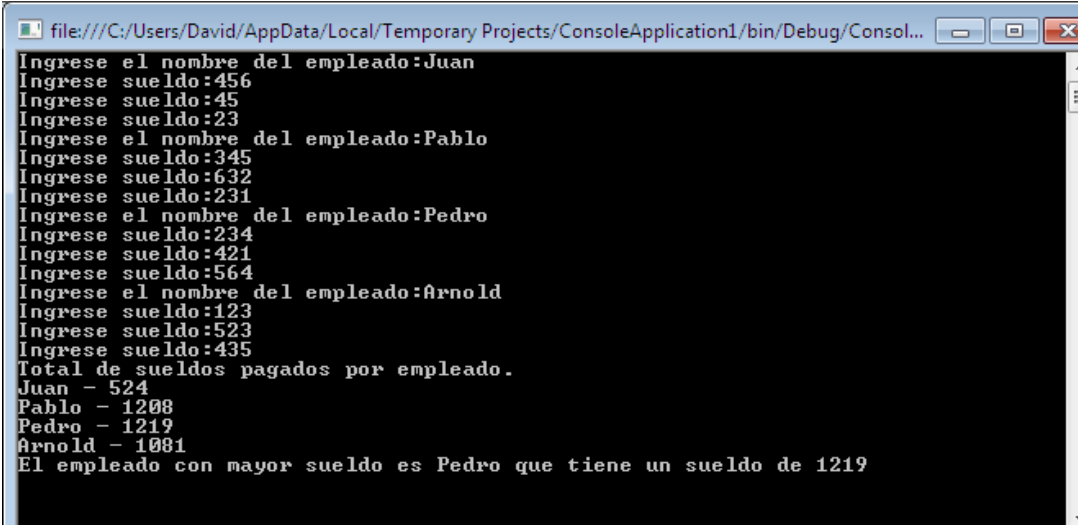
public void EmpleadoMayorSueldo()
{
    int may = sueldostot[0];
    string nom = empleados[0];
    for (int f = 0; f < sueldostot.Length; f++)
    {
        if (sueldostot[f] > may)
        {
            may = sueldostot[f];
            nom = empleados[f];
        }
    }
    Console.WriteLine("El empleado con mayor sueldo es " + nom + " que tiene un
sueldo de " + may);
}

static void Main(string[] args)
{
    Matriz9 ma = new Matriz9();
    ma.Cargar();
    ma.CalcularSumaSuealdos();
    ma.ImprimirTotalPagado();
    ma.EmpleadoMayorSueldo();
    Console.ReadKey();
}
}

```

}

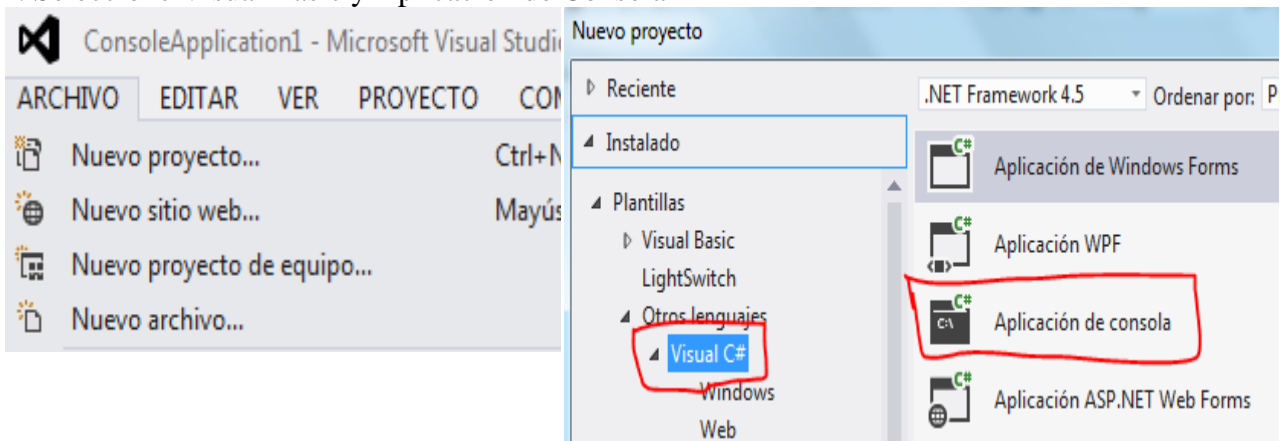
4. GRABAR Y EJECUTAR



```
file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/Consol...
Ingrese el nombre del empleado:Juan
Ingrese sueldo:456
Ingrese sueldo:45
Ingrese sueldo:23
Ingrese el nombre del empleado:Pablo
Ingrese sueldo:345
Ingrese sueldo:632
Ingrese sueldo:231
Ingrese el nombre del empleado:Pedro
Ingrese sueldo:234
Ingrese sueldo:421
Ingrese sueldo:564
Ingrese el nombre del empleado:Arnold
Ingrese sueldo:123
Ingrese sueldo:523
Ingrese sueldo:435
Total de sueldos pagados por empleado.
Juan - 524
Pablo - 1208
Pedro - 1219
Arnold - 1081
El empleado con mayor sueldo es Pedro que tiene un sueldo de 1219
```

FORMAR LA LETRA R EN UNA MATRIZ DE NxN

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

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

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int F = 0;
            int C = 0;
            int N = 0;
            int MI = 0;
            int FI = 0;
            string linea;
            // MATRIZ EN BLANCO
            Console.WriteLine("TAMAÑO DE LA MATRIZ IMPAR:");
            linea = Console.ReadLine();
            N = int.Parse(linea);
            N = (N % 2 == 0 ? N + 1 : N);
            string [,] MAT = new string [N + 1, N + 1];
            // MATRIZ EN BLANCO
            for (F = 1; F <= N; F++)
            {
                for (C = 1; C <= N; C++)
                {
                    MAT[F, C] = " ";
                    // UN ESPACIO EN BLANCO
                }
            }
            // FORMAMOS LA LETRA
            MI = N / 2 + 1;
        }
    }
}
```

```

for (F = 1; F <= N; F++)
{
    MAT[F, 1] = "A";
    MAT[MI, F] = "A";
    MAT[1, F] = "A";
}
FI = MI;
// PARA EMPEZAR DESDE FILA DE LA MITAD
for (F = 1; F <= MI; F++)
{
    MAT[F, N] = "A";
    MAT[FI, FI] = "A";
    FI = FI + 1;
}
// SALIDA
for (F = 1; F <= N; F++)
{
    for (C = 1; C <= N; C++)
    {
        Console.SetCursorPosition(C, F + 1);
        Console.Write(MAT[F, C]);
    }
    Console.WriteLine();
    Console.Write("Pulse una Tecla:");
    Console.ReadLine();
}
}
}
}

```

4. GRABAR Y EJECUTAR

```

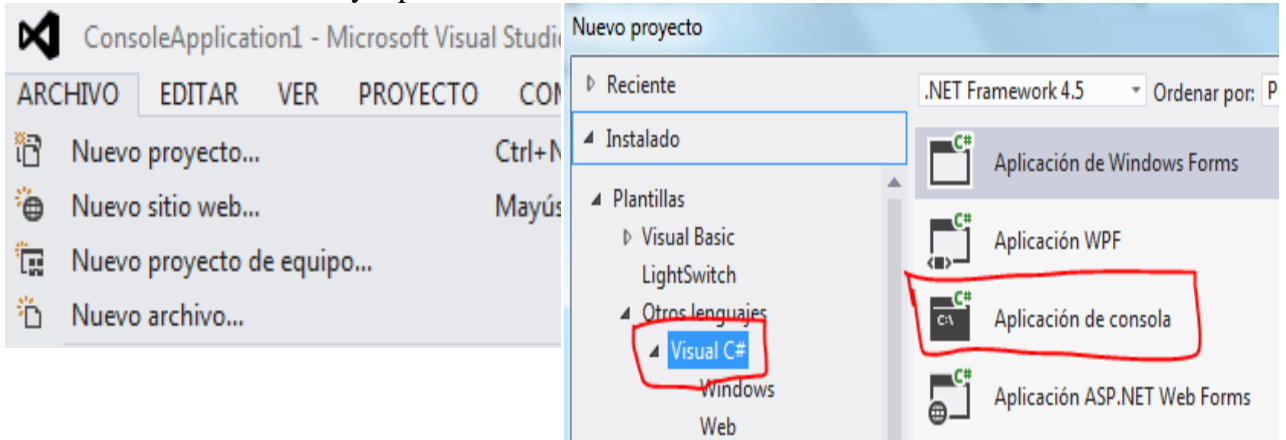
file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/Consol...
TAMAÑO DE LA MATRIZ IMPAR:5
AAAAA
A A
AAAAA
A A
A A
Pulse una Tecla:

```

SUMAR ELEMENTOS DE DIAGONAL PRINCIPAL Y SECUNDARIA

MATRIZ DE NxN

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

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

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int F = 0;
            int C = 0;
            int N = 0;
            int SUMP = 0;
            int SUMS = 0;
            string linea;
            // INGRESO
            // PARA COMODIDAD GENERAMOS VALORES PARA LA MATRIZ
            Console.WriteLine("TAMAÑO DE LA MATRIZ:");
            Random rnd = new Random();
            linea = Console.ReadLine();
            N = int.Parse(linea);
            int[,] MAT = new int[N + 1, N + 1];
            for (F = 1; F <= N; F++)
            {
                for (C = 1; C <= N; C++)
                {
                    MAT[F, C] = rnd.Next(0, 9);
                    Console.SetCursorPosition(C * 4, F + 1);
                    Console.Write(MAT[F, C]);
                }
            }
        }
    }
}
```

```

    }
    // DIAGONAL PRINCIPAL
    SUMP = 0;
    for (F = 1; F <= N; F++)
    {
        SUMP = SUMP + MAT[F, F];
    }
    // DIAGONAL SECUNDARIA
    SUMS = 0;
    C = N;
    for (F = 1; F <= N; F++)
    {
        SUMS = SUMS + MAT[F, C];
        C = C - 1;
    }
    // SALIDA
    Console.WriteLine();
    Console.WriteLine("SUMA DIAGONAL PRINCIPAL ES :" + SUMP);
    Console.WriteLine("SUMA DIAGONAL SECUNDARIA ES:" + SUMS);
    Console.Write("Pulse una Tecla:");
    Console.ReadLine();
}
}
}

```

4. GRABAR Y EJECUTAR

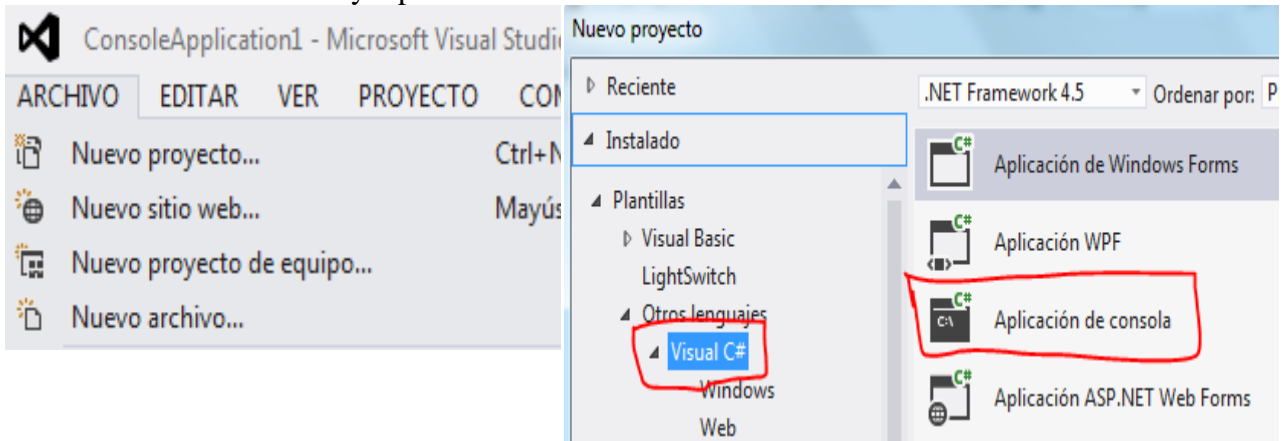
```

file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplic
TAMAÑO DE LA MATRIZ:6
  2  8  8  2  3  8
  3  2  3  4  8  8
  2  6  8  2  2  6
  8  0  0  0  4  2
  7  4  2  4  1  1
  4  4  8  6  3  4
SUMA DIAGONAL PRINCIPAL ES :17
SUMA DIAGONAL SECUNDARIA ES:26
Pulse una Tecla:

```

NÚMERO MAYOR Y MENOR EN UNA MATRIZ DE NxN

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

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

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int F = 0;
            int C = 0;
            int N = 0;
            int MAY = 0;
            int MEN = 0;
            string linea;
            // INGRESO
            // PARA COMODIDAD GENERAMOS VALORES PARA LA MATRIZ
            Console.WriteLine("TAMAÑO DE LA MATRIZ:");
            Random rnd = new Random();
            linea = Console.ReadLine();
            N = int.Parse(linea);
            int[,] MAT = new int[N + 1, N + 1];
            for (F = 1; F <= N; F++)
            {
                for (C = 1; C <= N; C++)
                {
                    MAT[F, C] = rnd.Next(0, 99);
                    Console.SetCursorPosition(C * 4, F + 1);
                    Console.Write(MAT[F, C]);
                }
            }
            // PROCESO
```



```

MAY = MAT[1, 1];
MEN = MAT[1, 1];
for (F = 1; F <= N; F++)
{
    for (C = 1; C <= N; C++)
    {
        if ((MAT[F, C] > MAY))
            MAY = MAT[F, C];
        if ((MAT[F, C] < MEN))
            MEN = MAT[F, C];
    }
}
// SALIDA
Console.WriteLine();
Console.WriteLine("NÚMERO MAYOR ES: " + MAY);
Console.WriteLine("NÚMERO MENOR ES: " + MEN);
Console.Write("Pulse una Tecla:");
Console.ReadLine();
}
}
}

```

4. GRABAR Y EJECUTAR

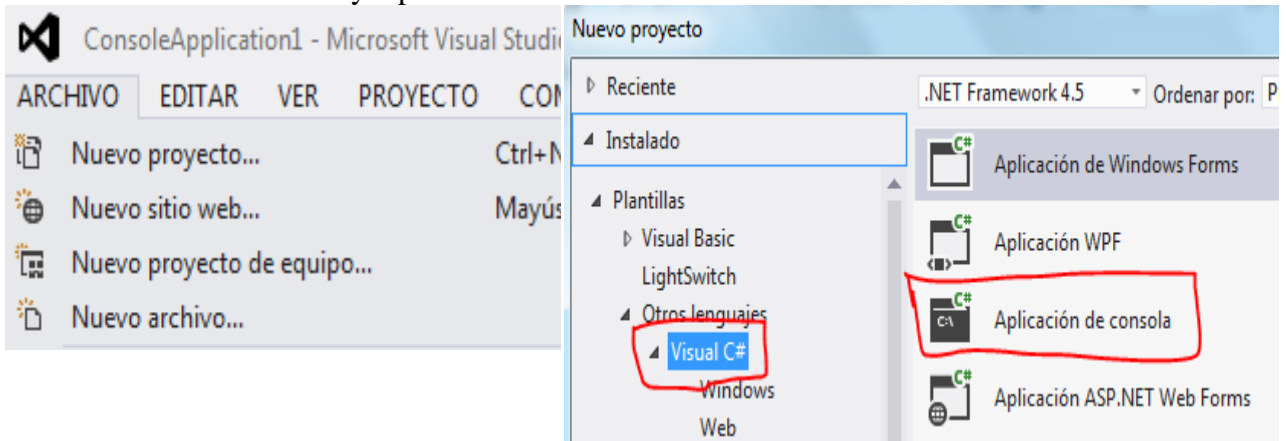
```

file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug
TAMAÑO DE LA MATRIZ:5
54 22 8 68 47
39 3 61 44 54
12 48 0 38 23
87 12 84 4 47
73 5 13 49 40
NÚMERO MAYOR ES: 87
NÚMERO MENOR ES: 0
Pulse una Tecla:

```

ORDENAMIENTO DE UNA MATRIZ DE N x N

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int F = 0;
            int C = 0;
            int I = 0;
            int K = 0;
            int N = 0;
            int AUX = 0;
            string cadena;
            // INGRESO
            // PARA COMODIDAD GENERAMOS VALORES PARA LA MATRIZ
            Console.WriteLine("TAMAÑO DE LA MATRIZ: ");
            cadena = Console.ReadLine();
            Random rnd = new Random();
            N = int.Parse(cadena);
            int[,] MAT = new int[N + 1, N + 1];
            for (F = 1; F <= N; F++)
            {
                for (C = 1; C <= N; C++)
                {
                    MAT[F, C] = rnd.Next(0, 100);
                    Console.SetCursorPosition(C * 4, F + 1);
                    Console.Write(MAT[F, C]);
                }
            }
            // PROCESO
            for (F = 1; F <= N; F++)
            {
```

```

    for (C = 1; C <= N; C++)
    {
        for (I = 1; I <= N; I++)
        {
            for (K = 1; K <= N; K++)
            {
                if ((MAT[F, C] < MAT[I, K]))
                {
                    AUX = MAT[F, C];
                    MAT[F, C] = MAT[I, K];
                    MAT[I, K] = AUX;
                }
            }
        }
    }
}
// SALIDA
for (F = 1; F <= N; F++)
{
    for (C = 1; C <= N; C++)
    {
        Console.SetCursorPosition(C * 4, F + 10);
        Console.Write(MAT[F, C]);
    }
    Console.WriteLine();
    Console.Write("Pulse una Tecla:");
    Console.ReadLine();
}
}
}

```

4. GRABAR Y EJECUTAR

```

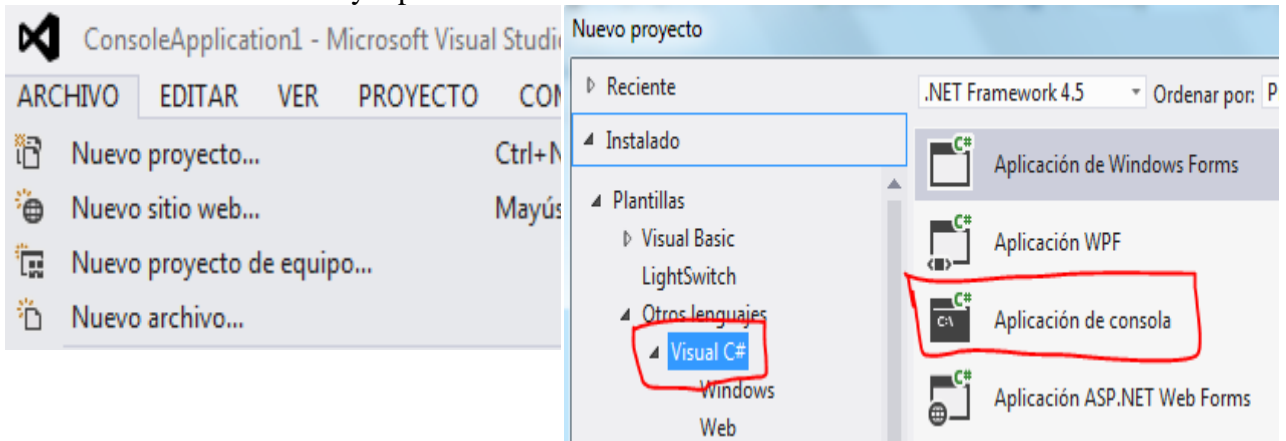
file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/Consol...
TAMAÑO DE LA MATRIZ: 3
    6   96  63
    4   71  22
    46   75   3

    3   4   6
    22  46  63
    71  75  96
Pulse una Tecla: _

```

SUMA DE MATRICES DE 5x5

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



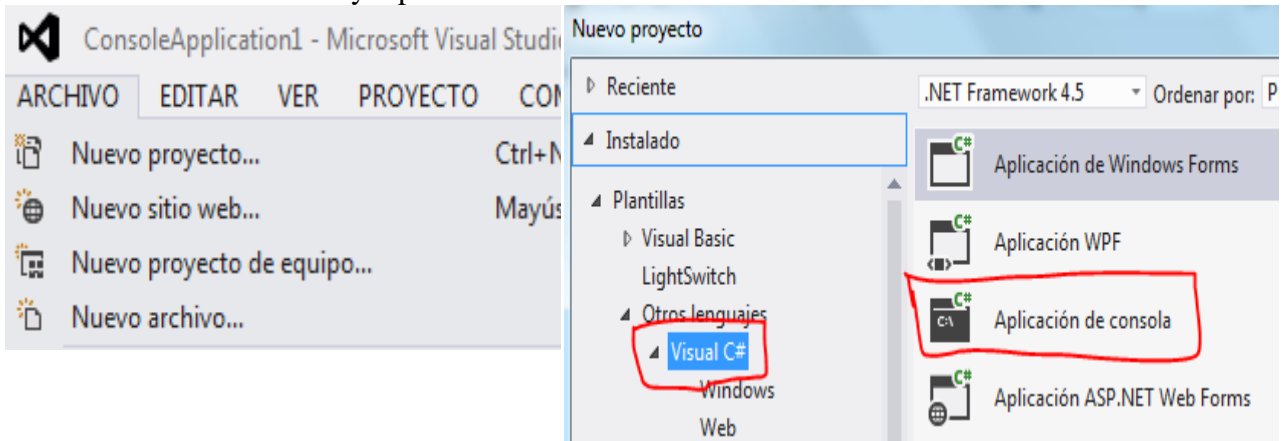
3. CÓDIGO

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

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            Random rnd = new Random();
            int F = 0;
            int C = 0;
            int[,] MATA = new int[100,100];
            int[,] MATB = new int[100,100];
            int[,] MATC = new int[6, 6];
            // INGRESO
            // PARA COMODIDAD GENERAMOS VALORES PARA LAS MATRICES
            for (F = 1; F <= 5; F++) {
                for (C = 1; C <= 5; C++) {
                    MATA[F, C] = rnd.Next(0, 10);
                    MATB[F, C] = rnd.Next(0, 10);
                    // SALIDA MATRIZ A
                    Console.SetCursorPosition(C * 4, F);
                    Console.Write(MATA[F, C]);
                    // SALIDA MATRIZ B
                    Console.SetCursorPosition(C * 4, F + 6);
                    Console.Write(MATB[F, C]);
                }
            }
            // PROCESO
            for (F = 1; F <= 5; F++) {
                for (C = 1; C <= 5; C++) {
```


MULTIPLICACIÓN DE MATRICES DE 4x4

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

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

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int N = 4;
            // TAMAÑO DE LA MATRIZ
            int F = 0;
            int C = 0;
            int K = 0;
            int[,] MATA = new int [100,100];
            int[,] MATB = new int [100,100];
            int[,] MATC = new int[N + 1, N + 1];
            // INGRESO
            // PARA COMODIDAD GENERAMOS VALORES PARA LAS MATRICES
            Random rnd = new Random();
            for (F = 1; F <= N; F++) {
                for (C = 1; C <= N; C++) {
                    MATA[F, C] = rnd.Next(0, 10);
                    MATB[F, C] = rnd.Next(0, 10);
                    // SALIDA MATRIZ A
                    Console.SetCursorPosition(C * 4, F);
                    Console.Write(MATA[F, C]);
                    // SALIDA MATRIZ B
                    Console.SetCursorPosition(C * 4, F + 6);
                    Console.Write(MATB[F, C]);
                }
            }
        }
    }
}
```

```

// PROCESO
for (F = 1; F <= N; F++) {
    for (C = 1; C <= N; C++) {
        for (K = 1; K <= N; K++) {
            MATC[F, C] = MATC[F, C] + (MATA[F, K] * MATB[K, C]);
        }
    }
}
// SALIDA
Console.WriteLine();
Console.WriteLine("MATRIZ RESULTANTE");
for (F = 1; F <= N; F++) {
    for (C = 1; C <= N; C++) {
        Console.SetCursorPosition(C * 4, F + 13);
        Console.Write(MATC[F, C]);
    }
}
Console.WriteLine();
Console.Write("Pulse una Tecla:");
Console.ReadLine();
}
}
}

```

4. GRABAR Y EJECUTAR

```

7 5 3 9
7 7 2 2
6 6 5 2
2 1 4 5

6 2 9 5
4 0 9 9
7 0 4 4
2 5 4 4
MATRIZ RESULTANTE

101 59 156 128
88 24 142 114
99 22 136 112
54 29 63 55
Pulse una Tecla:

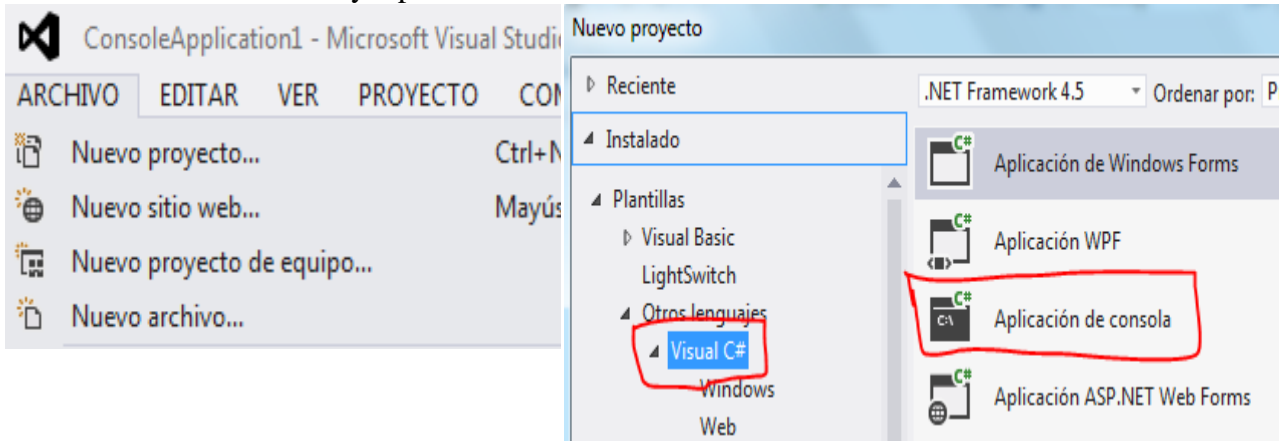
```

EJERCICIO PROPUESTO

Concepto:

Confeccionaremos un programa que permita crear una matriz irregular y luego imprimir la matriz en forma completa.

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

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

```
namespace MatrizIrregular1
{
    class MatrizIrregular1
    {
        private int[][] mat;

        public void Cargar()
        {
            Console.WriteLine("Cuantas fila tiene la matriz:");
            string linea = Console.ReadLine();
            int filas = int.Parse(linea);
            mat = new int[filas][];
            for (int f = 0; f < mat.Length; f++)
            {
                Console.WriteLine("Cuantas elementos tiene la fila " + f + ":");
                linea = Console.ReadLine();
                int elementos = int.Parse(linea);
                mat[f] = new int[elementos];
                for (int c = 0; c < mat[f].Length; c++)
                {
                    Console.WriteLine("Ingrese componente:");
                    linea = Console.ReadLine();
                    mat[f][c] = int.Parse(linea);
                }
            }
        }
    }
}
```



```

public void Imprimir()
{
    for (int f = 0; f < mat.Length; f++)
    {
        for (int c = 0; c < mat[f].Length; c++)
        {
            Console.Write(mat[f][c] + " ");
        }
        Console.WriteLine();
    }
    Console.ReadLine();
}

static void Main(string[] args)
{
    MatrizIrregular1 ma = new MatrizIrregular1();
    ma.Cargar();
    ma.Imprimir();
}
}

```

4. GRABAR Y EJECUTAR

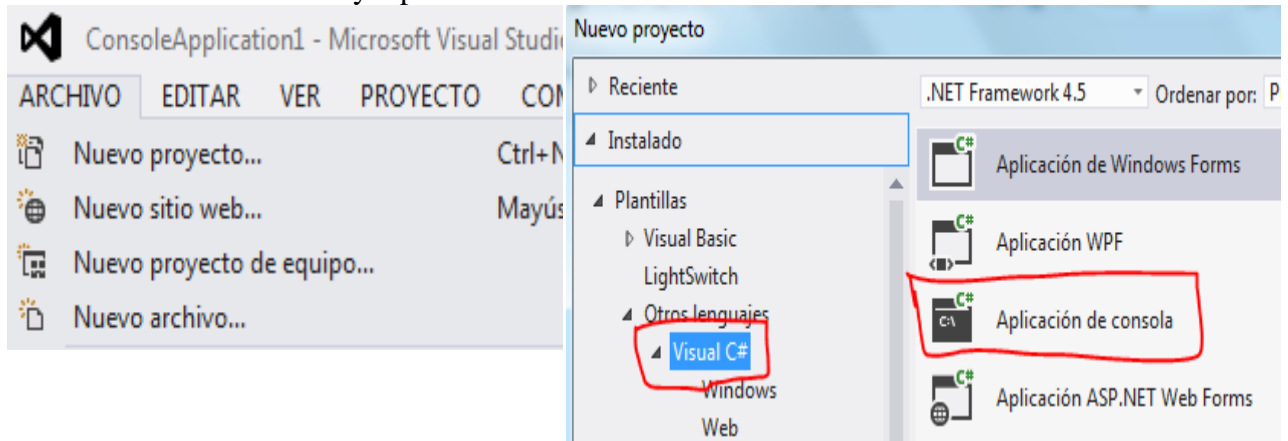
```

file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/Consol...
Cuantas fila tiene la matriz:4
Cuantas elementos tiene la fila 0:2
Ingrese componente:4
Ingrese componente:3
Cuantas elementos tiene la fila 1:2
Ingrese componente:2
Ingrese componente:3
Cuantas elementos tiene la fila 2:5
Ingrese componente:2
Ingrese componente:3
Ingrese componente:4
Ingrese componente:5
Ingrese componente:6
Cuantas elementos tiene la fila 3:1
Ingrese componente:2
4 3
2 3
2 3 4 5 6
2
-

```

GENERACIÓN DEL TRIÁNGULO DE PASCAL FORMA 1

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

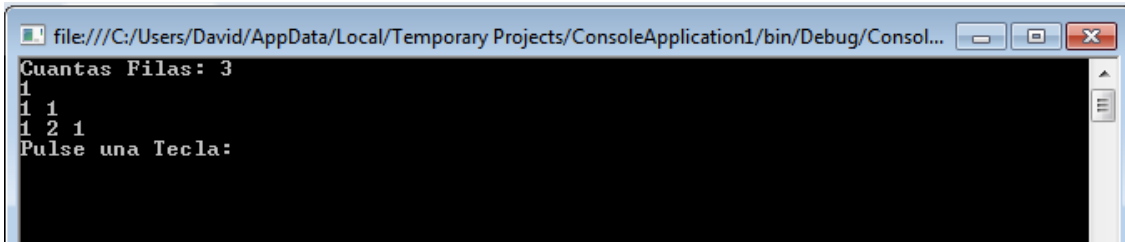
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int F = 0;
            int C = 0;
            int N = 0;
            string cadena;
            Console.WriteLine("Cuantas Filas: ");
            cadena = Console.ReadLine();
            N = int.Parse(cadena);
            int[,] MAT = new int[N + 1, N + 1];
            // PROCESO
            for (F = 1; F <= N; F++)
            {
                for (C = 1; C <= N; C++)
                {
                    if (((C == 1) | (F == C)))
                    {
                        MAT[F, C] = 1;
                    }
                    else
                    {
                        MAT[F, C] = MAT[F - 1, C] + MAT[F - 1, C - 1];
                    }
                }
            }
            // SALIDA
            for (F = 1; F <= N; F++)
            {
```

```

        for (C = 1; C <= N; C++)
        {
            if ((MAT[F, C] != 0))
            {
                Console.WriteLine("{0} ", MAT[F, C]);
            }
            Console.WriteLine();
        }
        Console.WriteLine("Pulse una Tecla:");
        Console.ReadLine();
    }
}
}
}

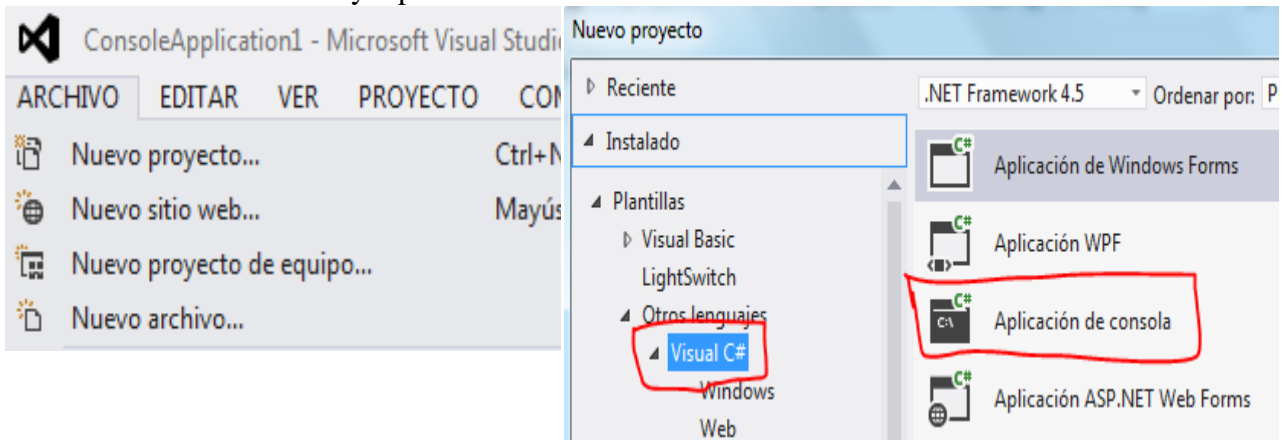
```

4. GRABAR Y EJECUTAR



GENERACIÓN DEL TRIÁNGULO DE PASCAL FORMA 2

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication1

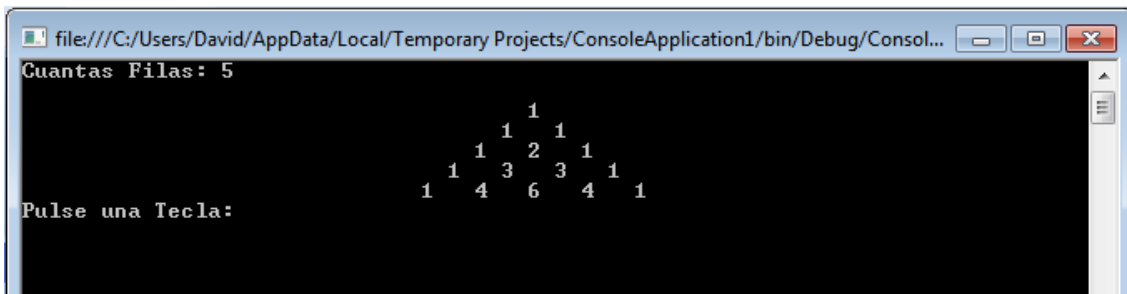
```

```

{
class Program
{
    static void Main(string[] args)
    {
        int F = 0;
        int C = 0;
        int N = 0;
        int COL = 0;
        int FI = 0;
        string cadena;
        Console.Write("Cuantas Filas: ");
        cadena= Console.ReadLine();
        N = int.Parse(cadena);
        int[,] MAT = new int[N + 1, N + 1];
        // PROCESO
        for (F = 1; F <= N; F++)
        {
            for (C = 1; C <= N; C++)
            {
                if (((C == 1) | (F == C)))
                {
                    MAT[F, C] = 1;
                }
                else
                {
                    MAT[F, C] = MAT[F - 1, C] + MAT[F - 1, C - 1];
                }
            }
        }
        // SALIDA
        FI = 2;
        for (F = 1; F <= N; F++)
        {
            COL = 40 - (F * 2);
            for (C = 1; C <= N; C++)
            {
                if ((MAT[F, C] != 0))
                {
                    Console.SetCursorPosition(COL, FI);
                    Console.Write(MAT[F, C]);
                    COL = COL + 4;
                }
            }
            FI = FI + 1;
            Console.WriteLine();
        }
        Console.Write("Pulse una Tecla:");
        Console.ReadLine();
    }
}
}

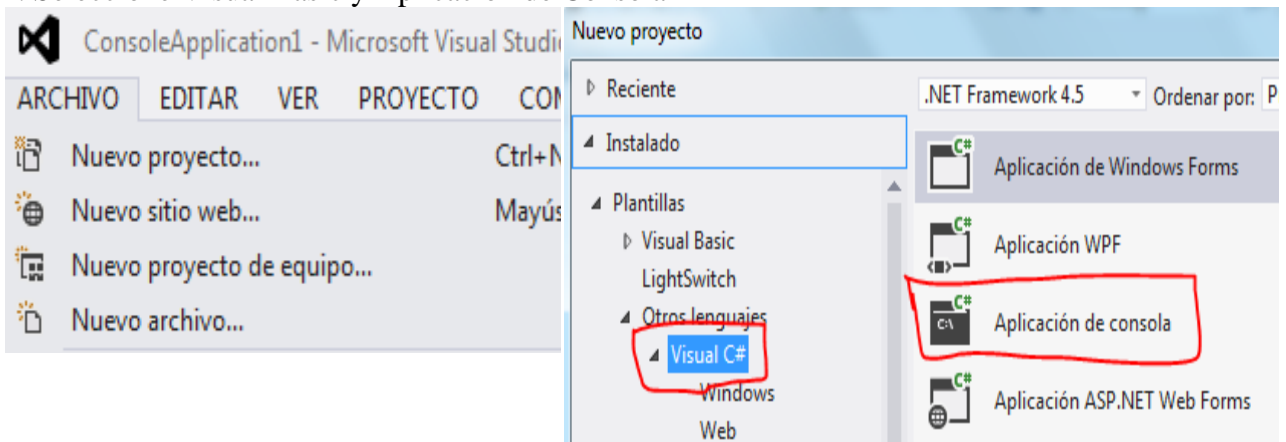
```

4. GRABAR Y EJECUTAR



MATRIZ TRANSPUESTA DE NxN

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

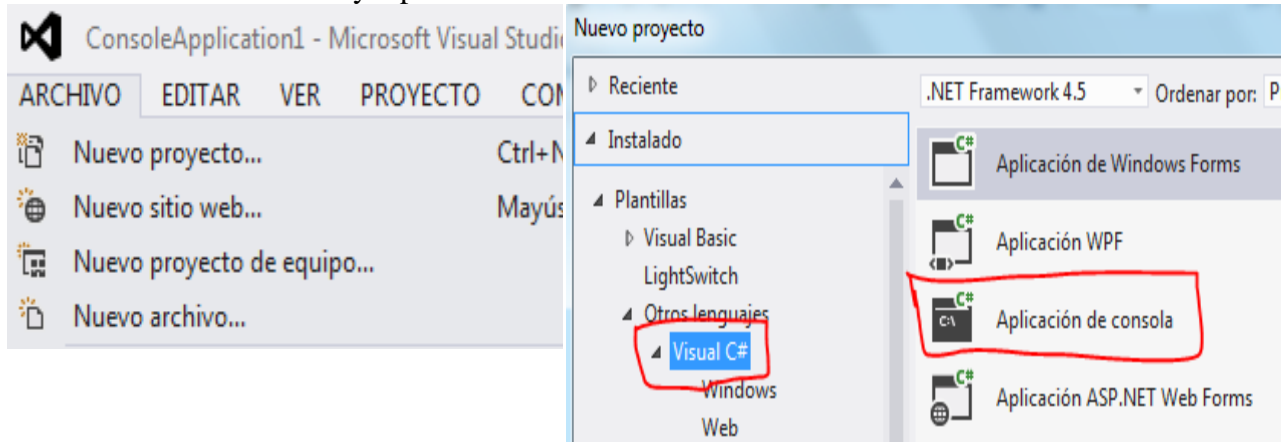
```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int F = 0;
            int C = 0;
            int N = 0;
            string cadena;
            // INGRESO
            // PARA COMODIDAD GENERAMOS VALORES PARA LA MATRIZ
            Console.WriteLine("TAMAÑO DE LA MATRIZ: ");
            cadena = Console.ReadLine();
            N = int.Parse(cadena);
            int[,] MAT = new int[100,100];
        }
    }
}

```


MAYORES DE CADA FILA DE UNA MATRIZ N×N EN UN VECTOR

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int F = 0;
            int C = 0;
            int N = 0;
            int MAY = 0;
            string cadena;
            // INGRESO
            // PARA COMODIDAD GENERAMOS VALORES PARA LA MATRIZ
            Console.WriteLine("TAMAÑO DE LA MATRIZ: ");
            cadena = Console.ReadLine();
            N = int.Parse(cadena);
            int[,] MAT = new int [100,100];
            int[] VEC = new int[N + 1];
            Random rnd = new Random();
            for (F = 1; F <= N; F++) {
                for (C = 1; C <= N; C++) {
                    MAT[F, C] = rnd.Next(0, 100);
                    Console.SetCursorPosition(C * 4, F + 1);
                    Console.Write(MAT[F, C]);
                }
            }
            // PROCESO
            for (F = 1; F <= N; F++) {
                MAY = MAT[F, 1];
            }
        }
    }
}
```



```

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int F = 0;
            int C = 0;
            int N = 0;
            int MEN = 0;
            string cadena;
            // INGRESO
            // PARA COMODIDAD GENERAMOS VALORES PARA LA MATRIZ
            Console.Write("TAMAÑO DE LA MATRIZ: ");
            cadena =Console.ReadLine();
            N = int.Parse (cadena );
            int[,] MAT = new int [100,100];
            int[] VEC = new int[N + 1];
            Random rnd = new Random();
            for (F = 1; F <= N; F++) {
                for (C = 1; C <= N; C++) {
                    MAT[F, C] = rnd.Next(0, 100);
                    Console.SetCursorPosition(C * 4, F + 1);
                    Console.Write(MAT[F, C]);
                }
            }
            // PROCESO
            for (C = 1; C <= N; C++) {
                MEN = MAT[1, C];
                for (F = 1; F <= N; F++) {
                    if ((MAT[F, C] < MEN))
                        MEN = MAT[F, C];
                }
                VEC[C] = MEN;
            }
            // SALIDA
            for (C = 1; C <= N; C++) {
                Console.SetCursorPosition(C * 4, 15);
                Console.Write(VEC[C]);
            }
            Console.WriteLine();
            Console.Write("Pulse una Tecla:");
            Console.ReadLine();
        }
    }
}

```

4. GRABAR Y EJECUTAR

```

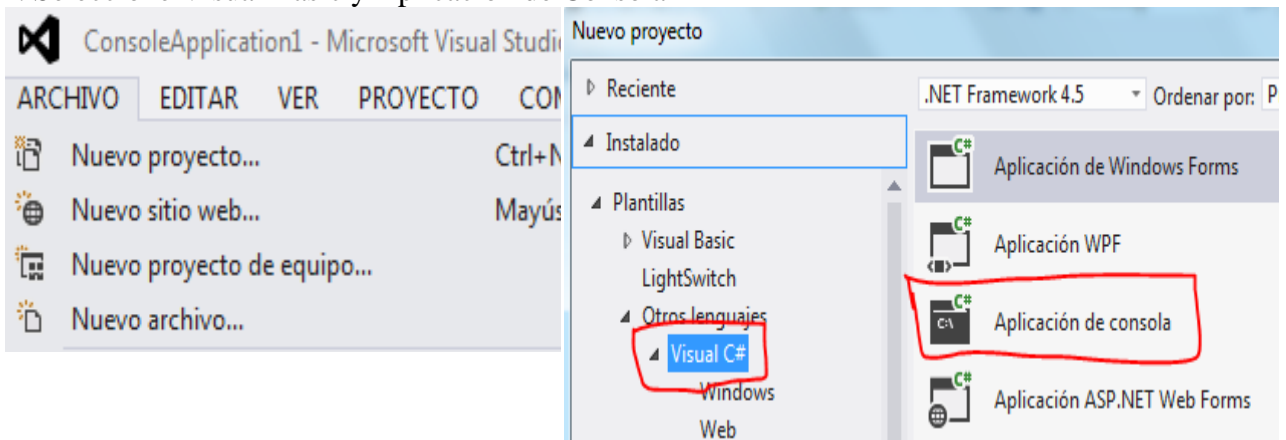
file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/Consol...
TAMAÑO DE LA MATRIZ: 7
84 92 52 83 28 35 27
88 72 20 4 1 85 29
65 10 42 15 30 10 31
4 58 40 37 52 57 45
25 30 42 2 47 52 73
14 69 0 72 77 75 34
22 51 62 48 85 23 87

4 10 0 2 1 10 27
Pulse una Tecla:

```

PROMEDIOS DE CADA COLUMNA DE UNA MATRIZ NxN EN UN VECTOR

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

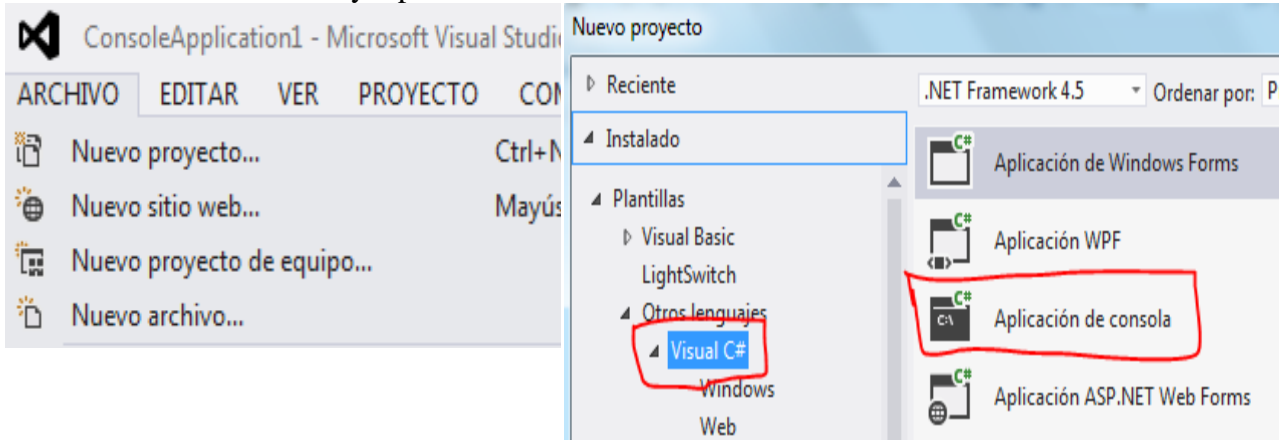
```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int F = 0;
            int C = 0;
            int N = 0;
            int SUM = 0;
            string cadena;

```


VOTACIONES. SUMA DE CADA COLUMNA REPRESENTA A UN CANDIDATO OBTENER EL CANDIDATO GANADOR

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola

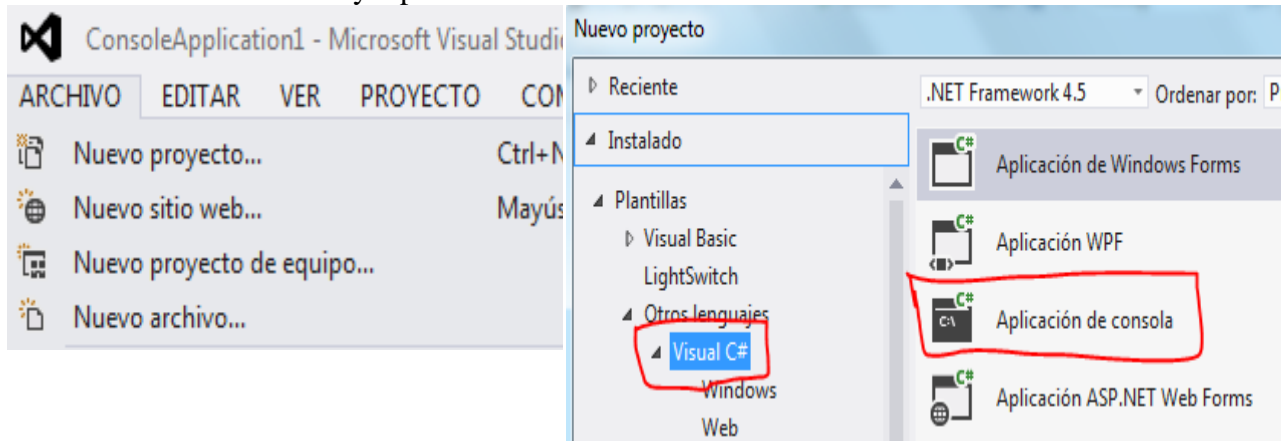


3. CÓDIGO

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int F = 0;
            int C = 0;
            int N = 0;
            int COL = 0;
            int MAY = 0;
            int SUM = 0;
            string cadena;
            // INGRESO
            // PARA COMODIDAD GENERAMOS VALORES PARA LA MATRIZ
            Console.WriteLine("TAMAÑO DE LA MATRIZ: ");
            cadena = Console.ReadLine();
            N = int.Parse(cadena);
            int[,] MAT = new int[100, 100];
            int[] VEC = new int[N + 1];
            Random rnd = new Random();
            for (F = 1; F <= N; F++) {
                for (C = 1; C <= N; C++) {
                    MAT[F, C] = rnd.Next(0, 10);
                    Console.SetCursorPosition(C * 6, F + 1);
                    Console.Write(MAT[F, C]);
                }
            }
            // PROCESO
```


CUBO DE UN NÚMERO

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



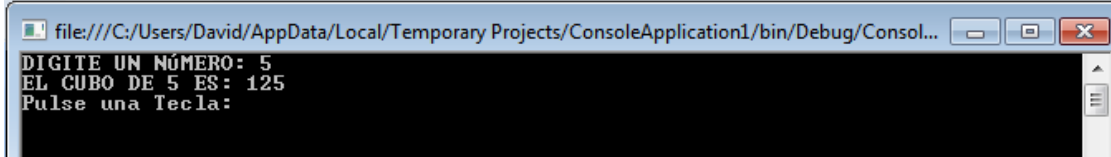
3. CÓDIGO

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

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int NUM = 0;
            int RESUL = 0;
            string cadena;
            Program ob =new Program();
            Console.Write("DIGITE UN NÚMERO: ");
            cadena = Console.ReadLine();
            NUM = int.Parse(cadena);
            RESUL = ob .CUBO (NUM );
            Console.Write("EL CUBO DE {0} ES: {1} ", NUM, RESUL);
            Console.WriteLine();
            Console.Write("Pulse una Tecla:");
            Console.ReadLine();

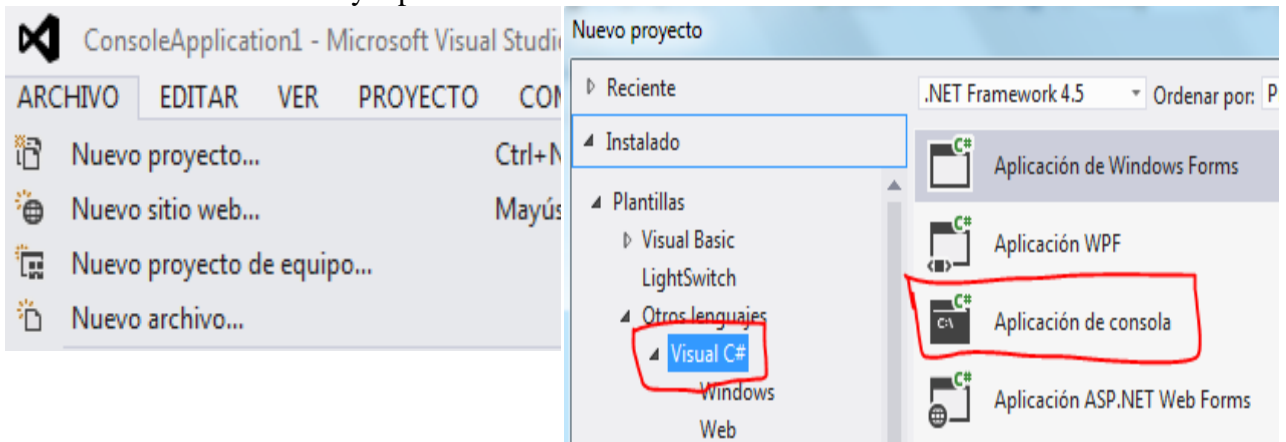
        }
        private int CUBO(int AUX)
        {
            return AUX * AUX * AUX;
        }
    }
}
```

4. GRABAR Y EJECUTAR



MAYOR DE TRES NÚMEROS

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

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

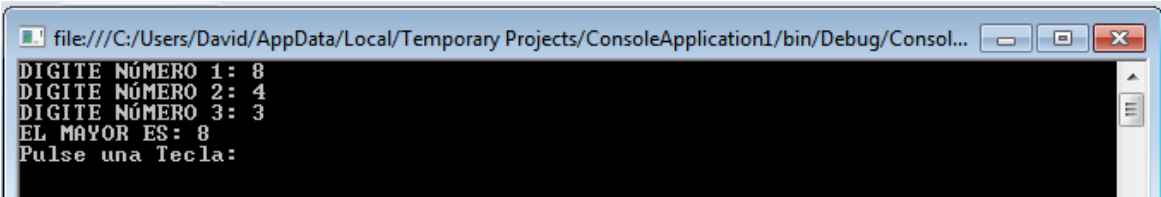
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            Program ob = new Program();
            int NUM1 = 0;
            int NUM2 = 0;
            int NUM3 = 0;
            string cadena;
            long RESULT = 0;
            Console.Write("DIGITE NÚMERO 1: ");
            cadena = Console.ReadLine();
            NUM1 = int.Parse(cadena);
            Console.Write("DIGITE NÚMERO 2: ");
```

```

        cadena = Console.ReadLine();
        NUM2 = int.Parse(cadena);
        Console.Write("DIGITE NÚMERO 3: ");
        cadena = Console.ReadLine();
        NUM3 = int.Parse(cadena);
        RESUL =ob. MAYOR(NUM1, NUM2, NUM3);
        Console.Write("EL MAYOR ES: " + RESUL);
        Console.WriteLine();
        Console.Write("Pulse una Tecla.");
        Console.ReadLine();
    }
    private int MAYOR(int N1, int N2,int N3)
    {
        int MAY ;
        MAY = N1;
        if (N2 > MAY)
        {
            MAY = N2;
        }
        if (N3 > MAY)
        {
            MAY = N3;
        }
        return MAY;
    }
}
}

```

4. GRABAR Y EJECUTAR



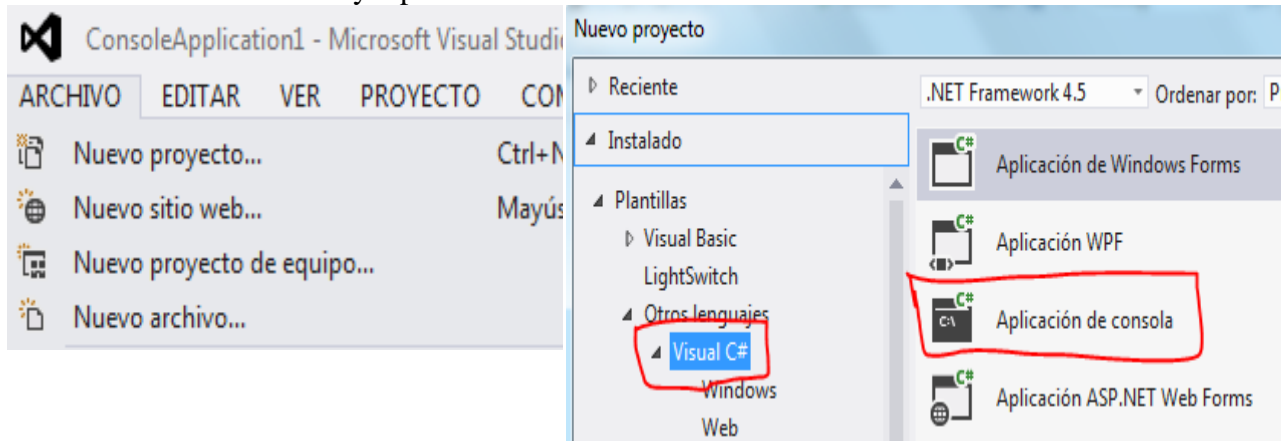
```

file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/Consol...
DIGITE NÚMERO 1: 8
DIGITE NÚMERO 2: 4
DIGITE NÚMERO 3: 3
EL MAYOR ES: 8
Pulse una Tecla:

```


VALOR ABSOLUTO DE UN NÚMERO

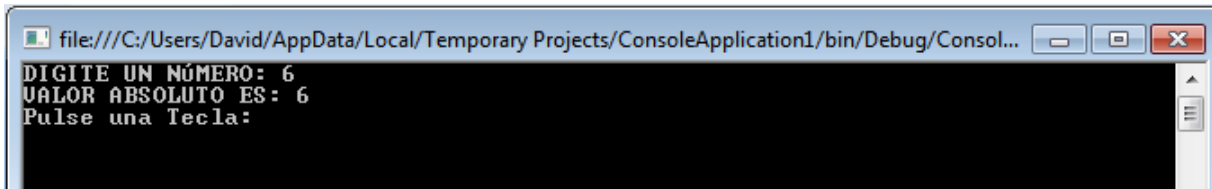
1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

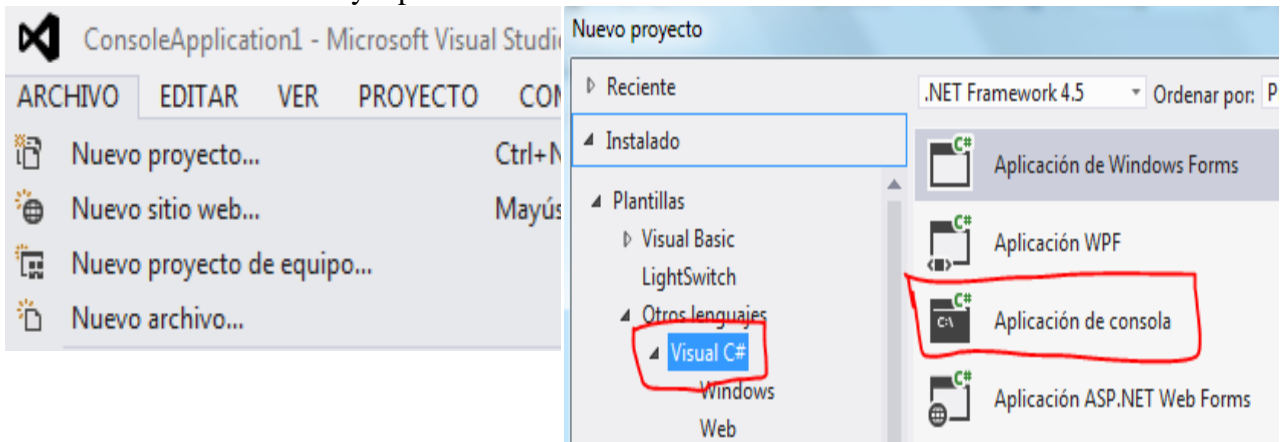
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            Program ob = new Program();
            int NUM = 0;
            long RESULT = 0;
            string cadena;
            Console.Write("DIGITE UN NÚMERO: ");
            cadena = Console.ReadLine();
            NUM = int.Parse(cadena);
            RESULT = ob.MIABS(NUM);
            Console.WriteLine("VALOR ABSOLUTO ES: " + RESULT);
            Console.WriteLine();
            Console.Write("Pulse una Tecla:");
            Console.ReadLine();
        }
        private long MIABS(int AUX)
        {
            if (AUX < 0)
            {
                AUX = AUX * -1;
            }
            return AUX;
        }
    }
}
```

4. GRABAR Y EJECUTAR



FACTORIAL DE UN NÚMERO

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

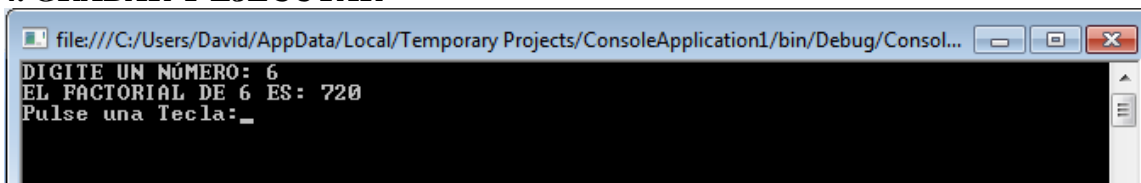
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            Program ob = new Program();
            int NUM = 0;
            long RESULT = 0;
            string cadena;
            Console.Write("DIGITE UN NÚMERO: ");
            cadena = Console.ReadLine();
            NUM = int.Parse(cadena);
            RESULT = ob.FACTORIAL(NUM);
            Console.Write("EL FACTORIAL DE {0} ES: {1} ", NUM, RESULT);
            Console.WriteLine();
            Console.Write("Pulse una Tecla:");
        }
    }
}
```

```

        Console.ReadLine();
    }
    private long FACTORIAL(int AUX)
    {
        long RES = 1;
        byte K = 0;
        for (K = 2; K <= AUX; K++)
        {
            RES = RES * K;
        }
        return RES;
    }
}
}
}

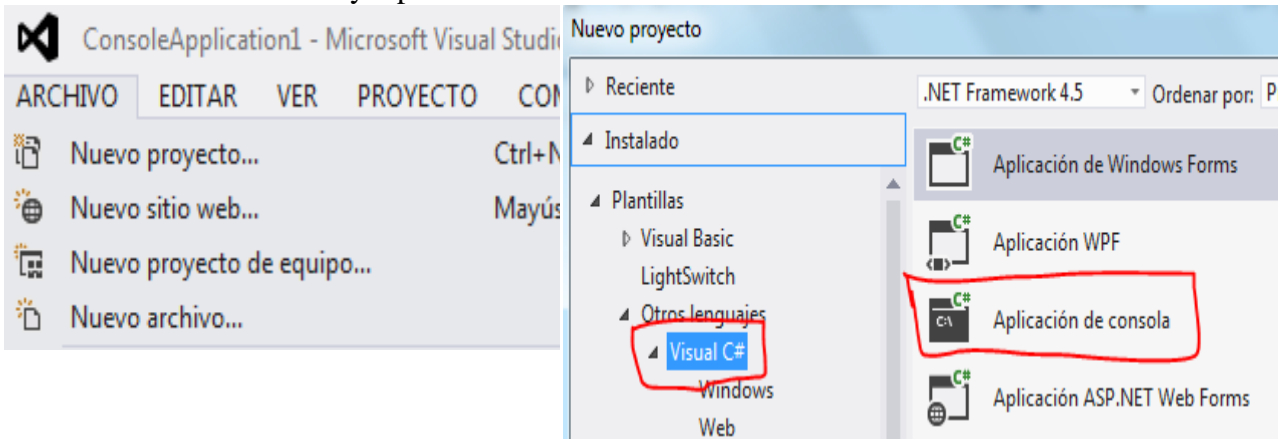
```

4. GRABAR Y EJECUTAR



INVERTIR UNA FRASE

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {

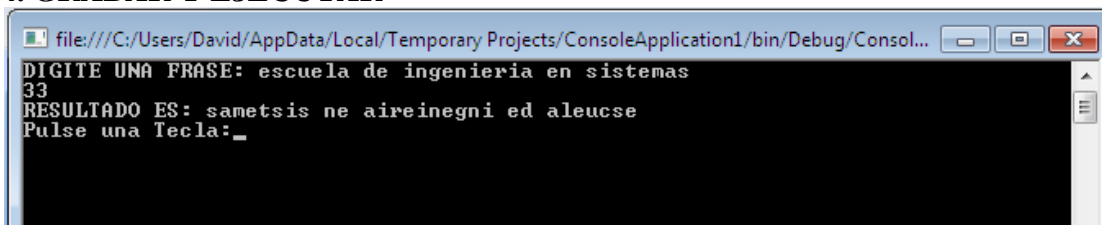
```

```

static void Main(string[] args)
{
    string NOM ;
    Program ob = new Program();
    Console.Write("DIGITE UNA FRASE: ");
    NOM = Console.ReadLine();
    Console.Write("RESULTADO ES: " + ob.INVERTIR(NOM));
    Console.WriteLine();
    Console.Write("Pulse una Tecla:");
    Console.ReadLine();
}
public string INVERTIR(string FRASE)
{
    string AUX ;
    string cadena;
    Program ob = new Program();
    int POS=FRASE.Length;
    Console.WriteLine (POS);
    AUX = "";
    while (POS >=1)
    {
        cadena = ob.Mid(FRASE, POS-1);
        AUX = AUX + cadena;
        POS--;
    }
    return AUX;
}
public string Mid(string param, int startIndex)
{
    return param.Substring(startIndex, 1);
}
}
}

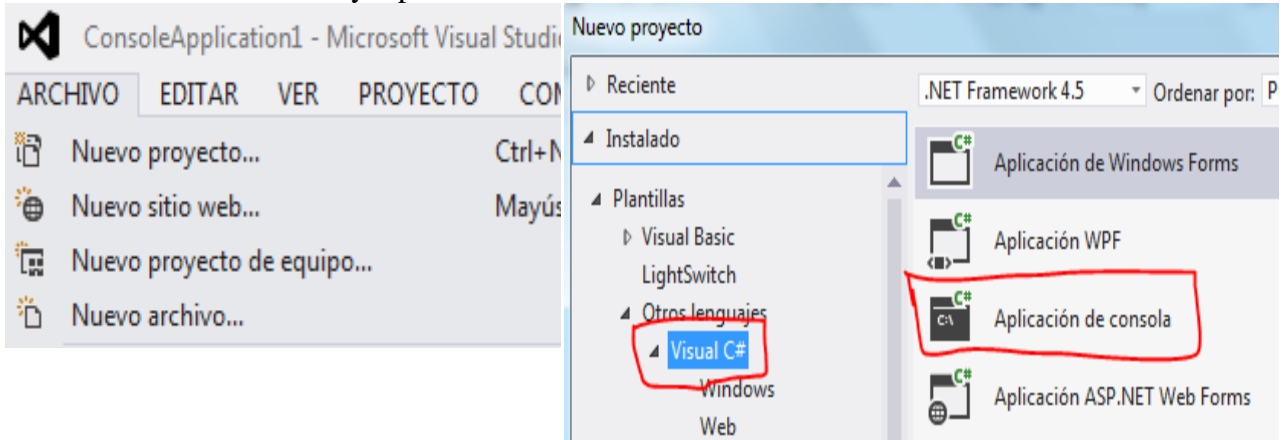
```

4. GRABAR Y EJECUTAR



COMPROBAR SI UN NÚMERO ES MÚLTIPLO DE OTRO

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

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

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int NUM1 = 0;
            int NUM2 = 0;
            string RESUL ;
            string cadena;
            Program ob=new Program ();
            Console.Write("DIGITE NÚMERO 1: ");
            cadena = Console.ReadLine();
            NUM1 = int.Parse(cadena);
            Console.Write("DIGITE NÚMERO 2: ");
            cadena = Console.ReadLine();
            NUM2 = int.Parse(cadena);
            RESUL = ob.MULTIPL0(NUM1, NUM2);
            Console.WriteLine();
            Console.WriteLine("{0} {1} DE {2}", NUM1, RESUL, NUM2);
            Console.Write("Pulse una Tecla:");
            Console.ReadLine();
        }
        private string MULTIPL0(int N1 ,int N2 )
        {
            string R ;
            int num;
            num = N1 % N2;
        }
    }
}
```

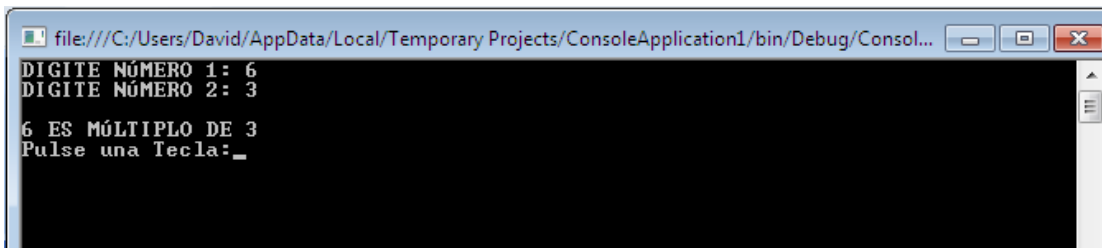
```

    if (num == 0)
    {
        R = "ES MÚLTIPLO";
    }
    else
    {
        R = "NO ES MÚLTIPLO";
    }

    return R;
}
}
}

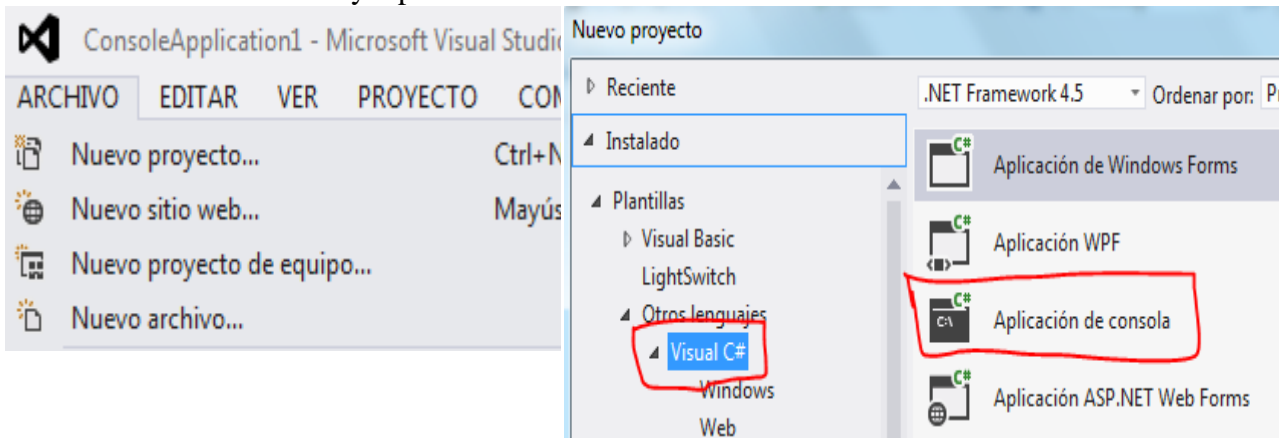
```

4. GRABAR Y EJECUTAR



NÚMERO A QUE DÍA DE LA SEMANA CORRESPONDE

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

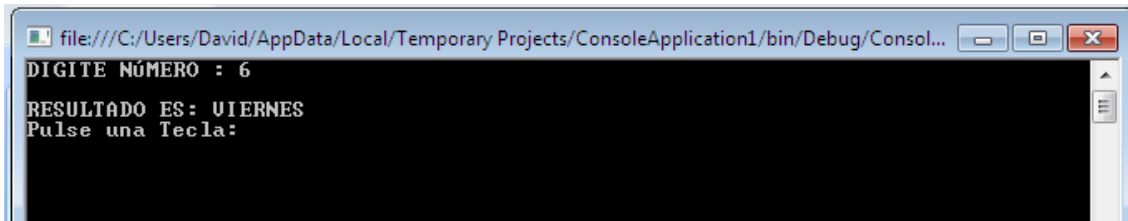
```

```

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int NUM = 0;
            string RESUL = null;
            string cadena;
            Program ob=new Program ();
            Console.Write("DIGITE NÚMERO : ");
            cadena = Console.ReadLine();
            NUM = int.Parse(cadena);
            RESUL = ob.DIASEMANA(NUM);
            Console.WriteLine();
            Console.WriteLine("RESULTADO ES: " + RESUL);
            Console.Write("Pulse una Tecla:");
            Console.ReadLine();
        }
        private string DIASEMANA(int DIA)
        {
            string R = null;
            switch (DIA)
            {
                case 1:
                    R = "DOMINGO";
                    break;
                case 2:
                    R = "LUNES";
                    break;
                case 3:
                    R = "MARTES";
                    break;
                case 4:
                    R = "MIÉRCOLES";
                    break;
                case 5:
                    R = "JUEVES";
                    break;
                case 6:
                    R = "VIERNES";
                    break;
                case 7:
                    R = "SÁBADO";
                    break;
                default:
                    R = "FUERA DE RANGO";
                    break;
            }
            return (R);
        }
    }
}

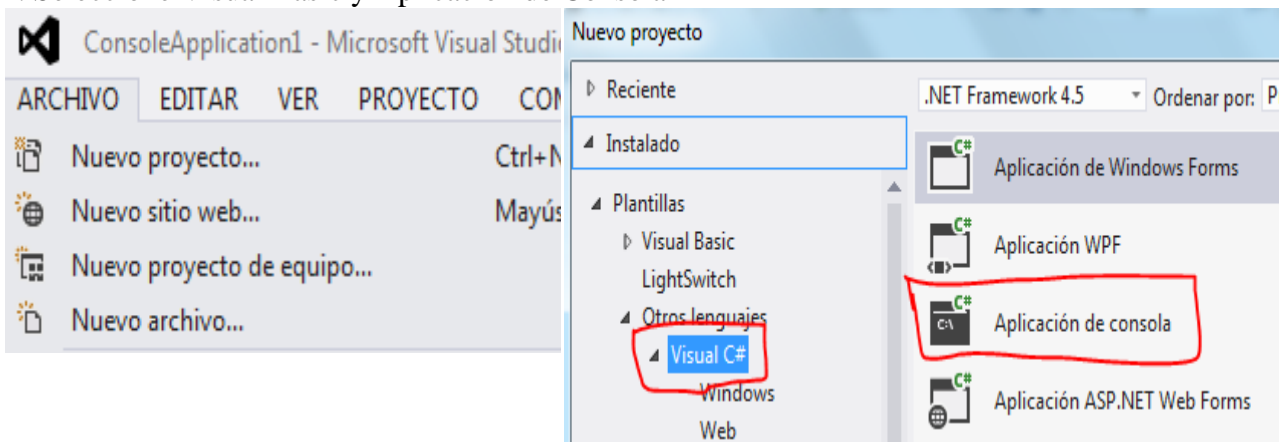
```

4. GRABAR Y EJECUTAR



NÚMERO COMPROBAR SI ES PRIMO

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



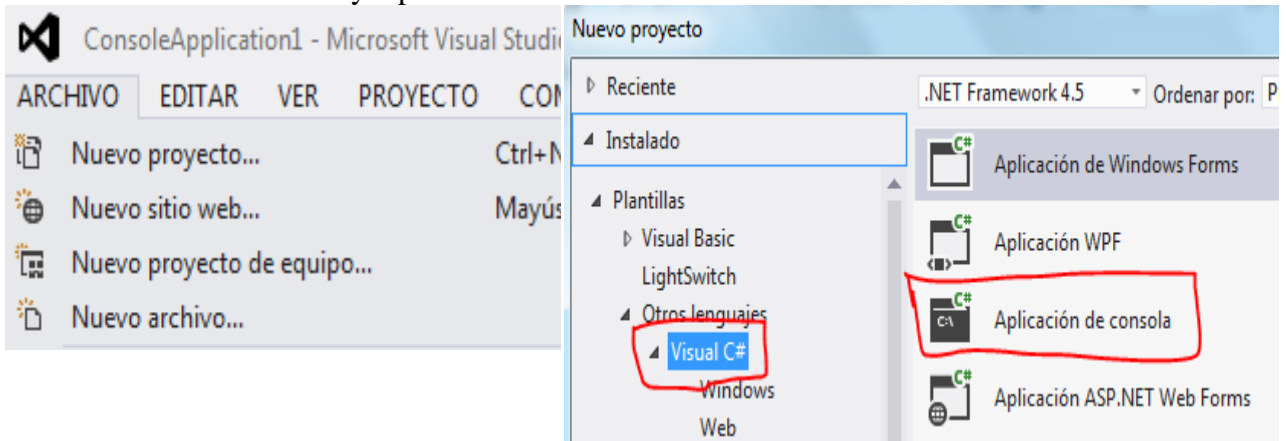
3. CÓDIGO

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

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int NUM = 0;
            string RESUL = null;
            string cadena;
            Program ob = new Program();
            Console.Write("DIGITE NÚMERO : ");
            cadena = Console.ReadLine();
            NUM = int.Parse(cadena);
            RESUL = ob.PRIMO(NUM);
            Console.WriteLine();
            Console.WriteLine("RESULTADO ES: " + RESUL);
            Console.Write("Pulse una Tecla:");
            Console.ReadLine();
        }
    }
}
```


MENOR EN UN ARREGLO

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

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

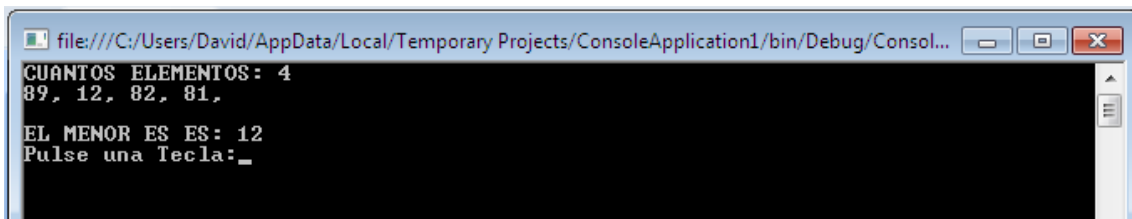
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int K = 0;
            int CAN = 0;
            int RESULT = 0;
            string cadena;
            Program ob = new Program();
            // INGRESO
            Console.Write("CUANTOS ELEMENTOS: ");
            cadena = Console.ReadLine();
            CAN = int.Parse(cadena);
            int[] VEC = new int[CAN + 1];
            Random rnd = new Random();
            for (K = 1; K <= CAN; K++)
            {
                VEC[K] = rnd.Next(1, 100);
                Console.Write(VEC[K] + ", ");
            }
            Console.WriteLine();
            RESULT = ob.VECTORMENOR(VEC, CAN);
            Console.WriteLine();
            Console.WriteLine("EL MENOR ES ES: " + RESULT);
            Console.Write("Pulse una Tecla:");
            Console.ReadLine();
        }
        private int VECTORMENOR(int[] NUEVO, int N)
        {
```

```

int MEN = 0;
int I = 0;
MEN = NUEVO[1];
for (I = 2; I <= N; I++)
{
    if ((NUEVO[I] < MEN))
        MEN = NUEVO[I];
}
return MEN;
}
}
}

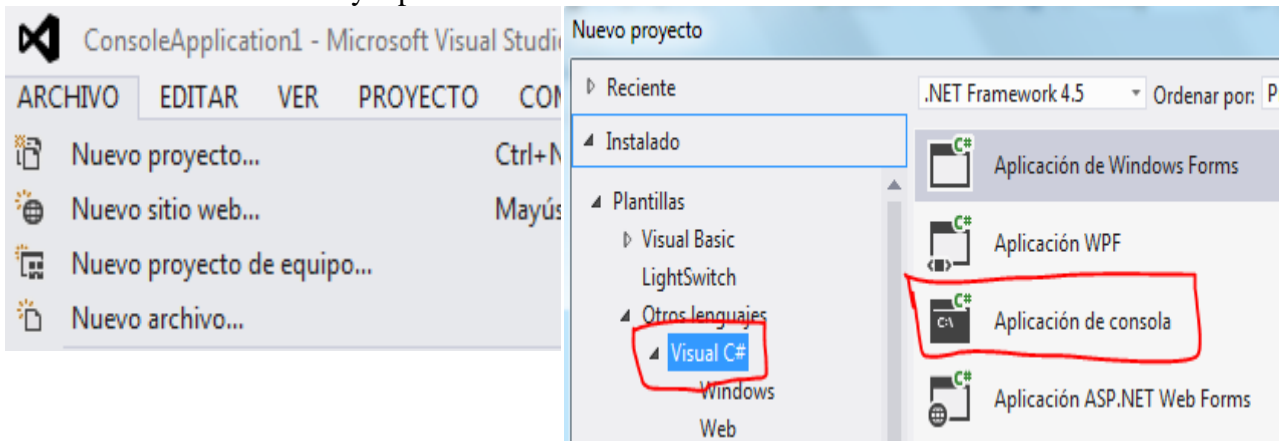
```

4. GRABAR Y EJECUTAR



TRANSFORMAR NÚMERO DECIMAL A BINARIO

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication1
{

```

```

class Program
{
    static void Main(string[] args)
    {
        int NUM1 = 0;
        string RESUL = null;
        string cadena;
        Program ob = new Program();
        Console.Write("DIGITE NÚMERO : ");
        cadena = Console.ReadLine();
        NUM1 = int.Parse(cadena);
        RESUL = ob.BINARIO(NUM1);
        Console.WriteLine();
        Console.WriteLine("RESULTADO ES: " + RESUL);
        Console.Write("Pulse una Tecla:");
        Console.ReadLine();
    }
    private string BINARIO(int NUM)
    {
        int K = 0;
        int AUX = 0;
        string R = null;
        string BIN = null;
        R = "";
        while ((NUM >= 2))
        {
            AUX = NUM % 2;
            R = R + AUX.ToString();
            NUM = NUM / 2;
        }
        R = R + NUM.ToString();
        BIN = "";
        for (K = R.Length; K >= 1; K += -1)
        {
            BIN = BIN + R.Substring(K-1, 1);
        }
        return BIN;
    }
}
}

```

4. GRABAR Y EJECUTAR

```

file:///C:/Users/David/AppData/Local/Temporary Projects/ConsoleApplication1/bin/Debug/Consol...
DIGITE NÚMERO : 6
RESULTADO ES: 110
Pulse una Tecla: _

```

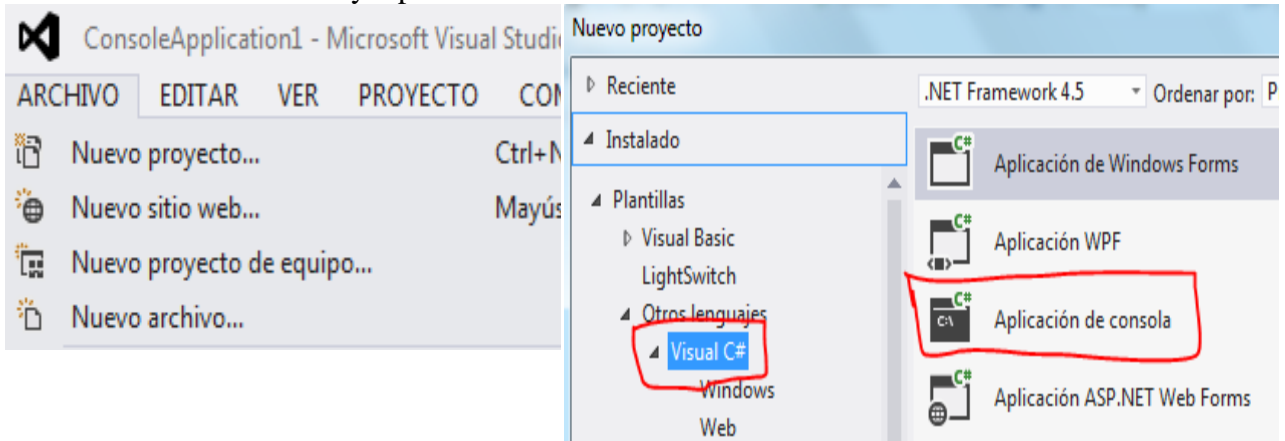
EJERCICIO PROPUESTO

Concepto:

Crear y cargar una matriz de 4 filas por 4 columnas. Imprimir la diagonal principal.

```
x - - -  
- x - -  
- - x -  
- - - x
```

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
  
namespace Matriz2  
{  
    class Matriz2  
    {  
        private int[,] mat;  
  
        public void Cargar()  
        {  
            mat = new int[4, 4];  
            for (int f = 0; f < 4; f++)  
            {  
                for (int c = 0; c < 4; c++)  
                {  
                    Console.Write("Ingrese componente:");  
                    string linea;  
                    linea = Console.ReadLine();  
                    mat[f, c] = int.Parse(linea);  
                }  
            }  
        }  
  
        public void ImprimirDiagonalPrincipal()  
        {
```

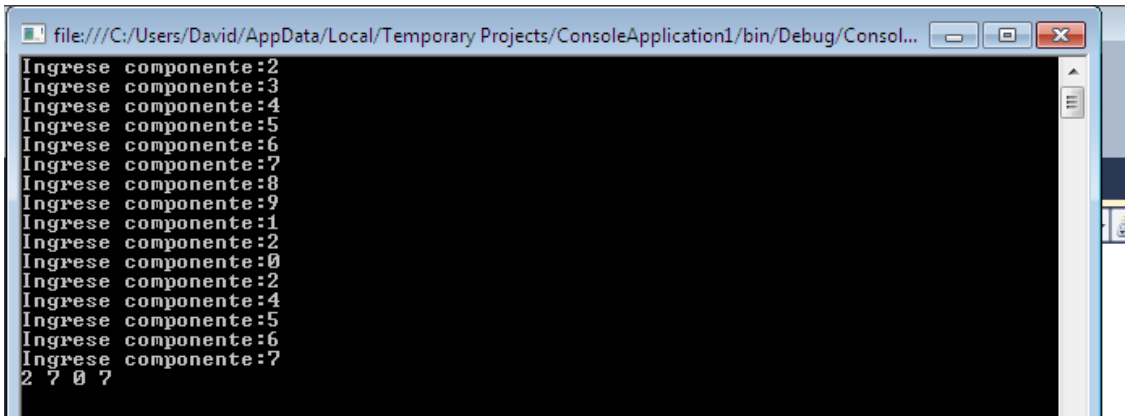
```

        for (int k = 0; k < 4; k++)
        {
            Console.Write(mat[k, k] + " ");
        }
        Console.ReadKey();
    }

    static void Main(string[] args)
    {
        Matriz2 ma = new Matriz2();
        ma.Cargar();
        ma.ImprimirDiagonalPrincipal();
    }
}

```

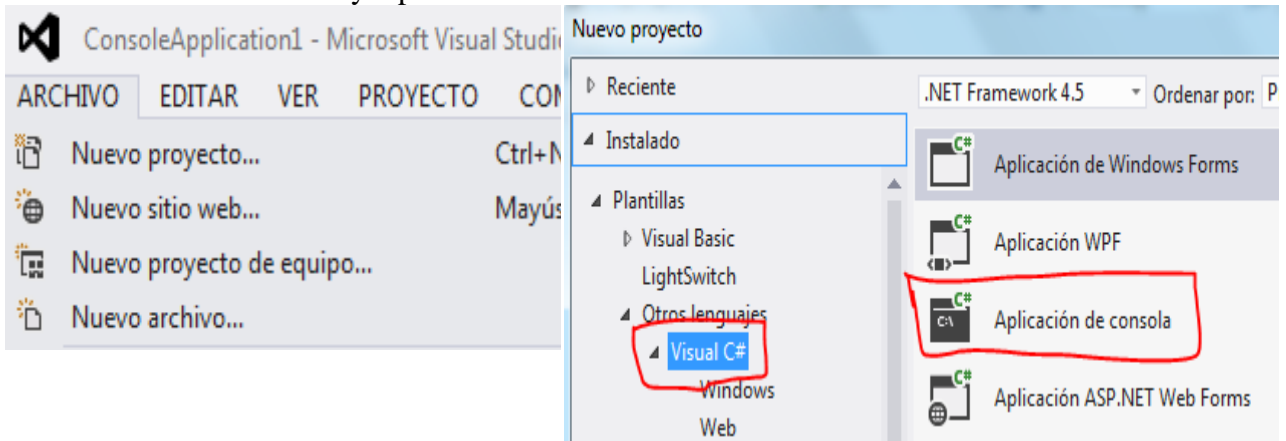
4. GRABAR Y EJECUTAR



Nota: el programa como adicional puede ser modificado para mostrar toda la matriz y comprobar el resultado del programa.

OBTENER EL DÍGITO VERIFICADOR DE LA CÉDULA

1. Abrir un nuevo Proyecto
2. Seleccione Visual Basic y Aplicación de Consola



3. CÓDIGO

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

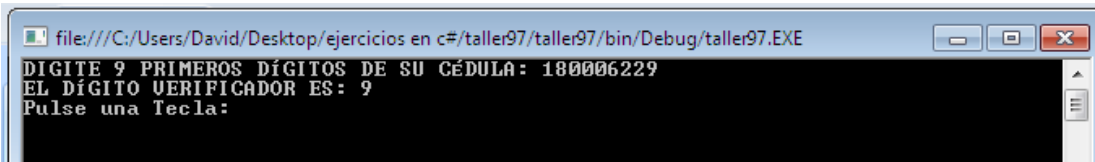
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int DV = 0;
            string MICEDU = null;
            Program ob=new Program ();
            Console.Write("DIGITE 9 PRIMEROS DÍGITOS DE SU CÉDULA: ");
            MICEDU = Console.ReadLine();
            DV = ob.ci(MICEDU);
            Console.WriteLine("EL DÍGITO VERIFICADOR ES: " + DV);
            Console.WriteLine();
            Console.WriteLine("Pulse una Tecla.");
            Console.ReadLine();
        }
        public int ci(string CED)
        {
            int SUMP = 0;
            int SUMI = 0;
            int ST = 0;
            int AUX = 0;
            int P = 0;
            int DIGITO = 0;
            // SUMATORIA DE PARES
            SUMP = 0;
            SUMI = 0;
            for (P = 2; P <=8; P += 2)
            {
                SUMP = SUMP + int.Parse(CED.Substring(P-1, 1));
            }
        }
    }
}
```

```

    }
    // SUMATORIA DE IMPARES
    for (P = 1; P <= 9; P += 2)
    {
        AUX = int.Parse (CED.Substring (P-1,1))* 2;
        if (AUX > 9)
            AUX = AUX - 9;
        SUMI = SUMI + AUX;
    }
    // OBTENCIÓN DE DÍGITO
    ST = SUMP + SUMI;
    DIGITO = 10 - (ST % 10);
    if (DIGITO == 10)
        DIGITO = 0;
    return DIGITO;
}
}
}

```

4. GRABAR Y EJECUTAR



```

file:///C:/Users/David/Desktop/ejercicios en c#/taller97/taller97/bin/Debug/taller97.EXE
DIGITE 9 PRIMEROS DÍGITOS DE SU CÉDULA: 180006229
EL DÍGITO VERIFICADOR ES: 9
Pulse una Tecla:

```


EJERCICIOS PROPUESTOS DIFERENTES Y NO RESUELTOS

- 1) El usuario tecleará dos números (x e y), y el programa deberá calcular cual es el resultado de su división y el resto de esa división.

- 2) Crear un programa que pida al usuario dos números enteros y diga si el primero es múltiplo del segundo (pista: igual que antes, habrá que ver si el resto de la división es cero: $a \% b == 0$).

- 3) Crear un programa que multiplique dos números enteros de la siguiente forma: pedirá al usuario un primer número entero. Si el número que se que teclee es 0, escribirá en pantalla "El producto de 0 por cualquier número es 0". Si se ha teclado un número distinto de cero, se pedirá al usuario un segundo número y se mostrará el producto de ambos.

- 4) Crear un programa que pida al usuario dos números reales. Si el segundo no es cero, mostrará el resultado de dividir entre el primero y el segundo. Por el contrario, si el segundo número es cero, escribirá "Error: No se puede dividir entre cero".

- 5) Crear un programa que pida una letra al usuario y diga si se trata de una vocal.

- 6) Crear un programa que pida al usuario dos números enteros y diga "Uno de los números es positivo", "Los dos números son positivos" o bien "Ninguno de los números es positivo", según corresponda.

- 7) Crear un programa que use el operador condicional para mostrar un el valor absoluto de un número de la siguiente forma: si el número es positivo, se mostrará tal cual; si es negativo, se mostrará cambiado de signo.

- 8) Crea un programa que escriba en pantalla los números del 1 al 10, usando "while".

- 9) Crea un programa que escriba en pantalla los números pares del 26 al 10 (descendiendo), usando "while".

- 10) Crea un programa que escriba en pantalla los números del 1 al 10, usando "do..while".

- 11) Crea un programa que escriba en pantalla los números pares del 26 al 10 (descendiendo), usando "do..while".
- 12) Crear un programa que muestre los números del 15 al 5, descendiendo (pista: en cada pasada habrá que descontar 1, por ejemplo haciendo $i=i-1$, que se puede abreviar $i--$).
- 13) Crear un programa que muestre los primeros ocho números pares (pista: en cada pasada habrá que aumentar de 2 en 2, o bien mostrar el doble del valor que hace de contador).
- 14) Crear un programa que escriba en pantalla los números del 1 al 50 que sean múltiplos de 3 (pista: habrá que recorrer todos esos números y ver si el resto de la división entre 3 resulta 0).
- 15) Cargar un vector de n elementos. imprimir el menor y un mensaje si se repite dentro del vector.
- 16) Se debe crear un vector donde almacenar 5 sueldos. Ordenar el vector sueldos de menor a mayor.
- 17) Definir un vector donde almacenar los nombres de 5 países. Confeccionar el algoritmo de ordenamiento alfabético.
- 18) Cargar en un vector los nombres de 5 países y en otro vector paralelo la cantidad de habitantes del mismo. Ordenar alfabéticamente e imprimir los resultados. Por último ordenar con respecto a la cantidad de habitantes (de mayor a menor) e imprimir nuevamente.
- 19) Crear una matriz de 2 filas y 5 columnas. Realizar la carga de componentes por columna (es decir primero ingresar toda la primer columna, luego la segunda columna y así sucesivamente)
Imprimir luego la matriz
- 20) Se desea saber la temperatura media trimestral de cuatro países. Para ello se tiene como dato las temperaturas medias mensuales de dichos países.
Se debe ingresar el nombre del país y seguidamente las tres temperaturas medias mensuales.
Seleccionar las estructuras de datos adecuadas para el almacenamiento de los datos en memoria.
a - Cargar por teclado los nombres de los países y las temperaturas medias mensuales.

b - Imprimir los nombres de los países y las temperaturas medias mensuales de las mismas.

c - Calcular la temperatura media trimestral de cada país.

e - Imprimir los nombres de las provincias y las temperaturas medias trimestrales.

f - Imprimir el nombre de la provincia con la temperatura media trimestral mayor.

21) Confeccionar una clase para administrar una matriz irregular de 5 filas y 1 columna la primer fila, 2 columnas la segunda fila y así sucesivamente hasta 5 columnas la última fila (crearla sin la intervención del operador)

Realizar la carga por teclado e imprimir posteriormente.

22) Confeccionar una clase para administrar los días que han faltado los 3 empleados de una empresa.

Definir un vector de 3 elementos de tipo string para cargar los nombres y una matriz irregular para cargar los días que han faltado cada empleado (cargar el número de día que faltó)

Cada fila de la matriz representan los días de cada empleado.

Mostrar los empleados con la cantidad de inasistencias.

Cuál empleado faltó menos días.

23) Crear una matriz de $n * m$ filas (cargar n y m por teclado) Intercambiar la primer fila con la segundo. Imprimir luego la matriz.

24) Crear una matriz de $n * m$ filas (cargar n y m por teclado) Imprimir los cuatro valores que se encuentran en los vértices de la misma ($mat[0][0]$ etc.)

-Palabras clave de C#

Abstract: El modificador `abstract` indica que el elemento objeto de la modificación tiene una implementación incompleta o que le falta una implementación. El modificador `abstract` se puede usar con clases, métodos, propiedades, indizadores y eventos. Use el modificador `abstract` en una declaración de clase para indicar que la clase sólo se puede utilizar como clase base de otras clases. Los miembros que están marcados como `abstract` o que se incluyen en una clase `abstract`, deben ser implementados por clases derivadas de la clase `abstract`.

Break: La instrucción `break` permite terminar el bucle envolvente más cercano o la instrucción `switch` en la cual aparece. El control se transfiere a la instrucción que sigue a la instrucción terminada, si existe alguna.

Char: La palabra clave `char` se utiliza para declarar una instancia de la estructura de `System.Char` que .NET Framework utiliza para representar un carácter Unicode. El valor de un objeto de `Char` es un valor (ordinal) numérico de 16 bits.

Continue: La instrucción `continue` pasa el control a la siguiente iteración de `while` que agrega, `do`, `for`, o la instrucción `foreach` en la que aparece.

Do: La instrucción `do` ejecuta una instrucción o un bloque de instrucciones repetidamente hasta que una determinada expresión se evalúa como `false`. El cuerpo del bucle se debe incluir entre llaves, `{}`, a menos que esté compuesto por una sola instrucción. En ese caso, las llaves son opcionales.

Event: La palabra clave `event` se utiliza para declarar un evento en una clase de editor.

Try-finally: Utilizando un bloque `finally`, puede limpiar los recursos que se asignan en un bloque `try`, y puede ejecutar código incluso si se produce una excepción en el bloque `try`. Normalmente, las instrucciones de un bloque `finally` se ejecuta cuando el control sale de una instrucción `try`. La transferencia del control puede producirse como resultado de la ejecución normal, de la ejecución de la instrucción `break`, `continue`, `goto` o `return`, o de la propagación de una excepción fuera de la instrucción `try`.

Dentro de una excepción controlada, se garantiza que el bloque asociado `finally` pueda ejecutarse. Sin embargo, si la excepción no se controla, la ejecución del bloque `finally` depende de cómo se desencadena la operación de desenredo de excepción. Esto, a su vez, depende de cómo se configurar el equipo. Para obtener más información, vea [Procesamiento de excepciones no controladas en CLR](#).

Normalmente, cuando una excepción no controlada finaliza una aplicación, si se ejecuta o no el bloque `finally` no es importante. Sin embargo, si tiene instrucciones en un `finally` bloqueado que debe ejecutarse incluso en esa situación, una solución es agregar un `catch` bloqueado a la instrucción `try-finally`. Alternativamente, puede detectar la excepción que se produzca en el bloque `try` de una instrucción `try-finally` situada más arriba de la pila de llamadas. Es decir, puede detectar la excepción en el método que llama al método que contiene la instrucción de `try-finally`, o en el método que llama a ese método, o en cualquier método de la pila de llamadas. Si la excepción no se detecta, la

ejecución del bloque de finally depende de si el sistema operativo elige desencadenar una operación de desenredo de excepción.

Internal: La palabra clave internal es modificador de acceso para los tipos y miembros de tipo. Los tipos o miembros internos sólo son accesibles dentro de los archivos del mismo ensamblado, como en este ejemplo:

```
public class BaseClass
{
    // Only accessible within the same assembly
    internal static int x = 0;
}
```

Espacio de nombres: La palabra clave namespace se utiliza para declarar un ámbito que contiene un conjunto de objetos relacionados. Puede utilizar un espacio de nombres para organizar elementos de código y crear global - tipos únicos. Ejemplo:

```
namespace SampleNamespace
{
    class SampleClass { }

    interface SampleInterface { }

    struct SampleStruct { }

    enum SampleEnum { a, b }

    delegate void SampleDelegate(int i);

    namespace SampleNamespace.Nested
    {
        class SampleClass2 { }
    }
}
```

Operator: Utilice la palabra clave operator para sobrecargar un operador integrado o proporcionar una conversión definida por el usuario en una declaración de clase o struct.

Params: Usando la palabra clave params, puede especificar un parámetro de método que acepta un número variable de argumentos.

Puede enviar una lista separada por comas de argumentos del tipo especificado en la declaración de parámetro o en una matriz de argumentos del tipo especificado. También puede no enviar ningún argumento. Si no se envía ningún argumento, la longitud de la lista de params es cero.

No se permiten parámetros adicionales después de la palabra clave params, ni varias palabras clave params en una misma declaración de método.

ReadOnly: La palabra clave readonly corresponde a un modificador que se puede utilizar en campos. Cuando una declaración de campo incluye un modificador readonly, las asignaciones a los campos que aparecen en la declaración sólo pueden tener lugar en la propia declaración o en un constructor de la misma clase. Ejemplo:

```
class Age
{
    readonly int _year;
    Age(int year)
    {
        _year = year;
    }
    void ChangeYear()
    {
        // _year = 1967; // Compile error if uncommented.
    }
}
```

Static: Utilice el modificador static para declarar un miembro estático, que pertenece al propio tipo en vez de a un objeto específico. El modificador static puede utilizarse con clases, campos, métodos, propiedades, operadores, eventos y constructores, pero no puede utilizarse con indizadores, destructores o tipos que no sean clases. Para obtener más información, vea Clases estáticas y sus miembros (Guía de programación de C#).

Ejemplo:

```
static class CompanyEmployee
{
    public static void DoSomething() { /*...*/ }
    public static void DoSomethingElse() { /*...*/ }
}
```

This: La palabra clave `this` hace referencia a la instancia actual de la clase y también se utiliza como modificador del primer parámetro de un método de extensión.

Void: Cuando se utiliza como tipo de valor devuelto para un método, `void` especifica que el método no devuelve un valor.

`void` no se permite en la lista de parámetros de un método. Un método que no utiliza parámetros y que no devuelve ningún valor se declara del siguiente modo:

as: Puede utilizar el operador `as` para realizar algunos tipos de conversiones entre los tipos de referencia compatibles o tipos que aceptan valores `NULL`.

Byte: La palabra clave `byte` denota un tipo entero que almacena valores según se muestra en la tabla siguiente.

Checked: La palabra clave `checked` se usa con el fin de habilitar explícitamente la comprobación de desbordamiento para operaciones aritméticas y conversiones de tipo integral.

Decimal: La palabra clave `decimal` indica un tipo de datos de 128 bits. Comparado con los tipos de punto flotante, el tipo `decimal` tiene una mayor precisión y un intervalo más reducido, lo que lo hace adecuado para cálculos financieros y monetarios. El intervalo aproximado y la precisión para el tipo `decimal` aparecen en la siguiente tabla.

Double: La palabra clave `double` denota un tipo simple que almacena valores de punto flotante de 64 bits. La siguiente tabla muestra la precisión y el intervalo de valores aproximado para el tipo `double`.

Goto: La instrucción `goto` transfiere el control del programa directamente a una instrucción identificada por una etiqueta. Un uso habitual de `goto` consiste en transferir el control a una etiqueta `switch-case` específica o a la etiqueta predeterminada de una instrucción `switch`. La instrucción `goto` también es útil para salir de bucles de varios niveles de anidamiento.

New: En C#, la palabra clave `new` se puede utilizar como operador, modificador o restricción.

Private: La palabra clave `private` es un modificador de acceso de miembros. El acceso de tipo `private` corresponde al nivel de acceso más restrictivo. Los miembros privados sólo son accesibles dentro del cuerpo de la clase o `struct` en los que se declaran.

Short: La palabra clave `short` denota un tipo de datos integral que almacena valores según el tamaño y el intervalo que se indican en la tabla siguiente.

String: El tipo `string` representa una secuencia de cero o más caracteres Unicode. `string` es un alias de `String` en .NET Framework.

Switch: La instrucción `switch` es una instrucción de control que selecciona una *sección switch* para ejecutarla desde una lista de candidatos.

Class: Las clases se declaran mediante la palabra clave class, como se muestra en el ejemplo siguiente:

```
class TestClass
{
    // Methods, properties, fields, events, delegates
    // and nested classes go here.
}
```

Default: La palabra clave default se puede utilizar en la instrucción switch o en código genérico.

if-else: Una instrucción if identifica que sentencia se tiene que ejecutar en función del valor de una expresión Boolean. En el ejemplo siguiente, la variable Booleanresult se establece en true y, a continuación, se comprueba en la instrucción if. La salida es The condition is true.

Float: La palabra clave float denota un tipo simple que almacena valores de punto flotante de 32 bits. La siguiente tabla muestra la precisión y el intervalo de valores aproximado para el tipo float.

Int: La palabra clave int denota un tipo entero que almacena valores según el tamaño y el intervalo que se indica en la tabla siguiente.

Protected: La palabra clave protected es un modificador de acceso a miembros. Un miembro protegido es accesible dentro de su clase y por instancias de clases derivadas. Encontrará una comparación de protected con los otros modificadores de acceso en Niveles de accesibilidad.

Return: La instrucción return termina la ejecución del método en el que aparece y devuelve el control al método que realizó la llamada. También puede devolver un valor opcional. Si el método es del tipo void, la instrucción return se puede omitir.

While: La instrucción while ejecuta una instrucción o un bloque de instrucciones repetidamente hasta que una expresión especificada se evalúa como false.

Const: Utilice la palabra clave const para declarar un campo de constante o un local de constante. Esta palabra clave especifica que el valor del campo o de la variable local es constante, o sea, que no se puede modificar.

Long: La palabra clave long denota un tipo entero que almacena valores según el tamaño y el intervalo que se indican en la tabla siguiente.

Object: El tipo object es un alias de Object en .NET Framework. En el sistema de tipos unificado de C#, todos los tipos (tipos de valor y de referencia predefinidos y definidos por el usuario) se heredan directa o indirectamente de Object. Las variables de tipo object pueden recibir valores de cualquier tipo. Cuando una variable de un tipo de valor se convierte en un objeto, se dice que se le ha aplicado

la *conversión boxing*. Cuando una variable de objeto de tipo se convierte en un tipo de valor, se dice que se le ha aplicado la *conversión unboxing*. Para obtener más información, vea Boxing y Unboxing.

Public: La palabra clave `public` es un modificador de acceso para tipos y miembros de tipos. El acceso de tipo `public` corresponde al nivel de acceso menos restrictivo. No existen restricciones para obtener acceso a los miembros públicos.

Switch: La instrucción `switch` es una instrucción de control que selecciona una *sección switch* para ejecutarla desde una lista de candidatos.

-Tipos de instrucciones

Instrucciones de declaración: Una instrucción de declaración introduce una nueva variable o constante. Una declaración de variable puede asignar opcionalmente un valor a la variable. En una declaración de constante, se requiere la asignación.

Instrucciones de expresión: Las instrucciones de expresión que calculan un valor deben almacenar el valor en una variable.

Instrucciones de selección: Las instrucciones de selección permiten crear bifurcaciones a diferentes secciones de código, en función de una o varias condiciones especificadas. Para obtener más información, vea los temas siguientes:

`if`, `else`, `switch`, `case`

Instrucciones de iteración: Las instrucciones de iteración permiten recorrer en bucle las colecciones como las matrices o bien realizar el mismo conjunto de instrucciones repetidamente hasta que se cumpla una condición especificada. Para obtener más información, vea los temas siguientes:

`do`, `for`, `foreach`, `in`, `while`

Instrucciones de salto: Las instrucciones de salto transfieren el control a otra sección de código. Para obtener más información, vea los temas siguientes:

`break`, `continue`, `default`, `goto`, `return`, `yield`

Instrucciones de control de excepciones: Las instrucciones de control de excepciones permiten recuperarse correctamente de condiciones excepcionales que se producen en tiempo de ejecución. Para obtener más información, vea los temas siguientes:

`throw`, `try-catch`, `try-finally`, `try-catch-finally`

Checked y unchecked: Las instrucciones `checked` y `unchecked` permiten especificar si las operaciones numéricas pueden producir un desbordamiento cuando el resultado se almacena en una variable demasiado pequeña para contener el valor resultante. Para obtener más información, vea `checked` y `unchecked`.

-Instrucciones inalcanzables

Si el compilador determina que el flujo de control nunca puede alcanzar una instrucción determinada bajo ninguna circunstancia, generará la advertencia CS0162, tal como se muestra en el ejemplo siguiente:

```
// An over-simplified example of unreachable code.  
const int val = 5;  
if (val < 4)  
{  
    System.Console.WriteLine("I'll never write anything."); //CS0162  
}
```

Referencias

Cabanes, N. *Microsoft Visual C# 2010* .

Jeff Ferguson, B. P. (2003). *La Biblia C#*. Madrid: Anaya Multimedia.

Medina, P. Prácticas de Consola en Java.

microsoft. (s.f.). *msdn.microsoft.com*. Obtenido de msdn.microsoft.com: <http://msdn.microsoft.com/es-es/library/x53a06bb.aspx>

Cerezo, Y., Pañalba, O., & Caballero, R. (2007). *Iniciación a la Programación en C# un enfoque práctico*. Madrid: Delta Publicaciones.

Canal Visual Basic;. (20 de Febrero de 2010). *Visual Basic*. Obtenido de Manuales: <http://www.canalvisualbasic.net/manual-net/vb-net/>

Duran, L. (2007). Bases de datos con Visual Basic. Marcombo.

Foros Aprender a programar. (21 de Enero de 2011). *Foros Aprender a programar*;. Obtenido de Visual Basic: <http://www.aprenderaprogramar.com/foros/>

Foros del Web. (10 de Marzo de 2010). *Foros*. Obtenido de Conectar a Bases de datos: <http://www.forosdelweb.com/>

Medina, P. (2012). *Guía práctica en VB Net*. Ambato.

Microsoft. (10 de Noviembre de 2014). *Microsoft Developer Network*. Obtenido de <http://msdn.microsoft.com/es-es/library/aa287558%28v=vs.71%29.aspx>

Microsoft. (11 de Marzo de 2009). *Msdn.Microsoft*. Obtenido de Librería: <http://msdn.microsoft.com/es-es/library/2x7h1hfk%28v=vs.80%29.aspx>

Esta obra pretende a través de la lógica y el estudio de las estructuras de programación fundamentales contribuir al desarrollo del pensamiento lógico formal del individuo para acometer la programación de diversos algoritmos.

Si bien es cierto la programación actual se enfoca a la programación visual, de objetos, de agentes, de hilos, entre otras, no se puede dejar de lado la formación del estudiante en el manejo de estructuras de programación, por ello este texto ha sido diseñado como un recurso de auto aprendizaje y refuerzo fuera de clase, contiene un variado conjunto de ejercicios que ilustran el uso de estructuras, fortalece su manejo y crea una experiencia autónoma enriquecedora en el estudiante, además de constituirse en un recurso didáctico para docentes de áreas afines.

ISBN- 978-9942-21-144-6

