



El futuro digital
es de todos

MinTIC



UNIVERSIDAD
EL BOSQUE

Misión
TIC 2022

Ciclo I



FUNDAMENTOS DE PROGRAMACIÓN

SEMANA 1

Ingresar



Rolando Fabián Junco



```

class MathHelp
{
    public static string BigNumbersSum(string n1, string n2)
    {
        string[] numbers = new string[2];
        int shortNumLength, longNumLength;

        if (isPrepared && number < prepared)
        {
            return preparedNumbers[number - 1];
        }
        else
        {
            InBigNumbersSum(n1, n2, numbers, out shortNumLength, longNumLength);
            return ComputeBigNumbersSum(numbers, shortNumLength, longNumLength);
        }
    }
}

```



Semana 1

Encuentro 1

Estructura Funcional del Computador (hardware y Software)

Representación de la Información en el computador

Lenguajes y Paradigmas de la Computación

Encuentro 2

Programación Estructurada y Modular

Metodología de la programación

Herramientas de programación

```

class MathHelp:
{
    public static string BigNumbersSum(string n1, string n2)
    {
        string[] numbers = new string[2];
        int shortNumLength, longNumLength;

        InBigNumbersSum(n1, n2, numbers, out shortNumLength, out longNumLength);
        return ComputeBigNumbersSum(numbers, shortNumLength, longNumLength);
    }
}

```



Metodología

50 horas sesiones sincrónicas orientadas por el formador específico de este curso.

120 horas de trabajo individual de programación, entre las cuales podrán solicitar apoyo de un tutor específico por ciclo de manera sincrónica a demanda, con previa solicitud del beneficiario, de acuerdo con las herramientas virtuales dispuestas para tal fin.

25 horas de formación de lectura en inglés y, 5 horas de trabajo en habilidades personales (coaching)

HORARIO Grupo 6 : lunes miércoles y viernes 7 am a 9:30 am

3 Ciclos de 7 Semanas

```
class MathHelp:
{
    public static string BigNumbersSum(string n1, string n2)
    {
        string[] numbers = n1.Split(' ');
        int shortNumLength, longNumLength;
        if (n1.Length < n2.Length)
        {
            return BigNumbersSum(n2, n1, numbers, out shortNumLength, longNumLength);
        }
        return ComputeBigNumbersSum(n1, n2, numbers, out shortNumLength, longNumLength);
    }
}
```



Cronograma de Trabajo



```

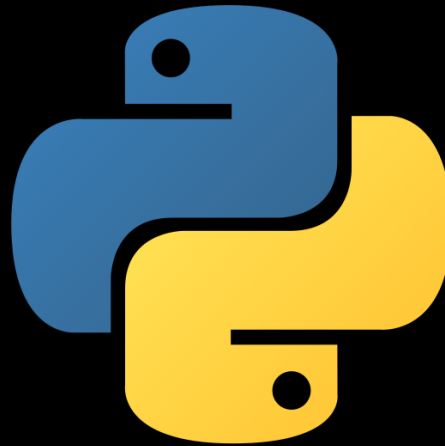
class MathHelper
{
    if (isPrepared && number < preparedNo)
    public static string BigNumbersSum(string n1, string n2)
    {
        string[] numbers = new string[2];
        int shortNumLength, longNumLength;
        if (n1.Length < n2.Length)
        {
            numbers[0] = n1;
            numbers[1] = n2;
            shortNumLength = n1.Length;
            longNumLength = n2.Length;
        }
        else
        {
            numbers[0] = n2;
            numbers[1] = n1;
            shortNumLength = n2.Length;
            longNumLength = n1.Length;
        }
        return ComputeBigNumbersSum(numbers, shortNumLength, longNumLength);
    }
}

```



Lenguaje

El ciclo tiene como objetivo la formación en Lenguaje Python (todas las estructuras de programación imperativa).





Evaluación y retroalimentación

El desarrollo de las actividades propuestas durante este ciclo será de carácter individual.

La evaluación de este ciclo 1 corresponde a: * Evaluación de los 5 retos y a la * Evaluación del curso de inglés.

Las evaluaciones se calificarán bajo la escala de 0 a 5. Al final de cada semana, como estudiante debe subir a la plataforma de evaluación la solución para el reto, en la cual, le generará la respectiva calificación.

Dicha calificación corresponde al número de requerimientos perfectamente resueltos, es decir, que pase todas las pruebas.

La plataforma reportará en caso de no aprobar satisfactoriamente los retos evaluados.


```

class MathHelp:
{
    public static string BigNumbersSum(string n1, string n2)
    {
        string[] numbers = new string[2];
        int shortNumLength, longNumLength;

        if (isPrepared && number < preparedNumbers)
        {
            return preparedNumbers[number - 1];
        }

        else
        {
            int sum = Sum(n1, n2, numbers, out shortNumLength, longNumLength);
            return ComputeBigNumbersSum(numbers, shortNumLength, longNumLength);
        }
    }
}

```



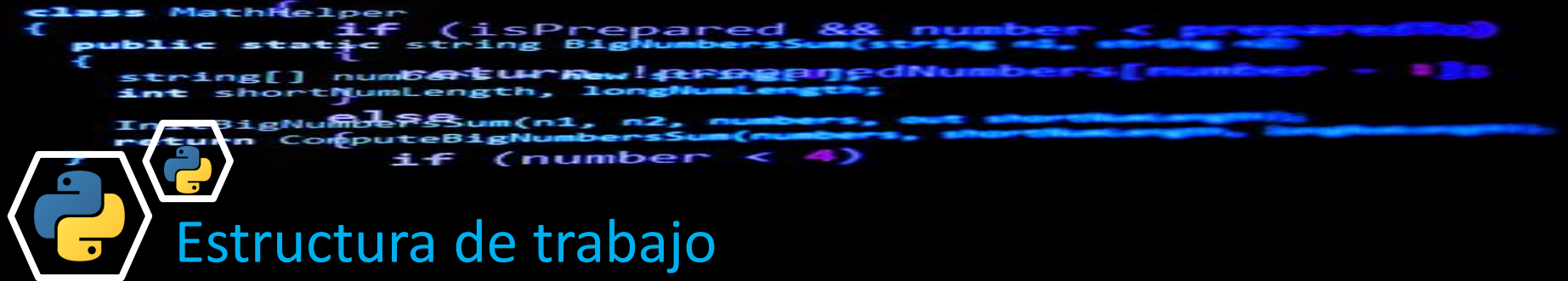
Evaluación y retroalimentación

Recuerda que la plataforma esta diseñada para comparar las soluciones entregadas, con el propósito de revisar y detectar semejanzas que puedan ser interpretadas como copia.

La forma de evaluación del curso de inglés lo define cada operador

La nota final del ciclo se calcula así: reto 1 (10%), reto 2 (10%), reto 3 (20%), reto 4(20%), reto 5 (20%), inglés (20%)

Para lograr aprobar el ciclo, deberás obtener una nota mínima de 3 sobre 5



Estructura de trabajo

Como estudiante tienes asignado un tutor al que le puedes hacer preguntas durante el tiempo dedicado a la solución del reto.

En la plataforma encontrarás un esquema de solicitudes de citas, que los tutores van atendiendo a lo largo del día. La plataforma llevará un registro del uso de este apoyo.

Cada grupo de estudiantes contará con un tutor, responsable de las sesiones de coaching, a su vez puede recurrir a ellos para buscar apoyo y consejo.

Para esto la plataforma cuenta con un esquema de solicitudes de citas, que el tutor va atendiendo a lo largo del día. La plataforma llevará un control de asistencia de los beneficiarios a las sesiones sincrónicas.

La plataforma guardará las soluciones de los retos entregadas por los estudiantes.


```

class MathHelper
{
    if (isPrepared && number < preparedNo)
    public static string BigNumbersSum(string n1, string n2)
    {
        string[] numbers = new string[2];
        int shortNumLength, longNumLength;
        if (n1.Length < n2.Length)
        {
            numbers[0] = n1;
            numbers[1] = n2;
            shortNumLength = n1.Length;
            longNumLength = n2.Length;
        }
        else
        {
            numbers[0] = n2;
            numbers[1] = n1;
            shortNumLength = n2.Length;
            longNumLength = n1.Length;
        }
        return ComputeBigNumbersSum(numbers, shortNumLength, longNumLength);
    }
}

```



Seguimiento

Cada semana, deberás diligenciar una encuesta sobre el proceso de aprendizaje, dificultades encontradas, calidad de las sesiones sincrónicas y el apoyo recibido por los tutores.

```

class MathHelp
{
    if (isPrepared && number < prepared)
    {
        public static string BigNumbersSum(string n1, string n2)
        {
            string[] numbers = new string[2];
            int shortNumLength, longNumLength;

            InBigNumbersSum(n1, n2, numbers, out shortNumLength, out longNumLength);
            return ComputeBigNumbersSum(numbers, shortNumLength, longNumLength);
        }
    }
}

```



Perfil de Egreso

- Desarrollar un programa monousuario para resolver los requerimientos planteados por un tercero
- Construir un programa utilizando un lenguaje imperativo (Python)
- Construir un programa con una interfaz de consola o gráfica simple
- Construir un programa que maneje archivos de texto para almacenar información persistente (formato JSON / CSV)

```

class MathHelper
{
    public static string BigNumbersSum(string n1, string n2)
    {
        string[] numbers = new string[n1.Length + n2.Length];
        int shortNumLength, longNumLength;
        if (n1.Length < n2.Length)
        {
            shortNumLength = n1.Length;
            longNumLength = n2.Length;
        }
        else
        {
            shortNumLength = n2.Length;
            longNumLength = n1.Length;
        }
        for (int i = 0; i < shortNumLength; i++)
        {
            numbers[i] = n1.Substring(i, shortNumLength - i);
        }
        for (int i = 0; i < longNumLength; i++)
        {
            numbers[i + shortNumLength] = n2.Substring(i, longNumLength - i);
        }
        return ComputeBigNumbersSum(numbers, shortNumLength, longNumLength);
    }
}

```



Atención de Inquietudes y realimentación

- Inquietudes ruta de aprendizaje y datos de contacto atención estudiantil:
<https://bit.ly/3e49eZf>
- Inquietudes relacionadas la formación: en encuentros y el correo interno.
- Correos electrónicos para reportar fallas académicas:
- inasistenciasmintic@unbosque.edu.co
- Correos electrónicos para radicar incapacidades
- incapacidadesmintic@unbosque.edu.co
- Correos electrónicos para solicitar apoyo en el manejo de la plataforma con la Mesa de servicio:
- soportemisiontic@unbosque.edu.co



Diagrama de aprendizaje basado en competencias. El diagrama muestra tres círculos que se superponen en un triángulo. El círculo superior es azul claro y contiene el texto "Conocimiento". El círculo inferior izquierdo es azul claro y contiene el texto "Integración" y "Propias experiencias". El círculo inferior derecho es gris claro y contiene el texto "Aplicación". En el centro, donde los tres círculos se superponen, hay un círculo más oscuro que contiene el texto "Aprendizaje".

```

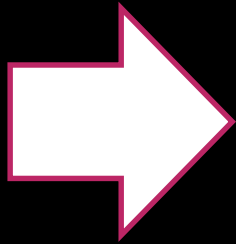
class MathHelp:
{
    public static string BigNumbersSum(string n1, string n2)
    {
        string[] numbers = new string[2];
        int shortNumLength, longNumLength;

        InBigNumbersSum(n1, n2, numbers, out shortNumLength, out longNumLength);
        return ComputeBigNumbersSum(numbers, shortNumLength, longNumLength);
    }
}

```

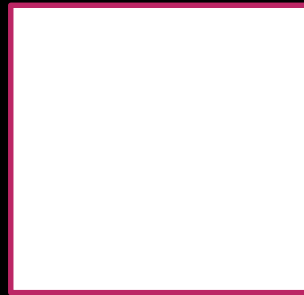


Entradas Procesos y salidas



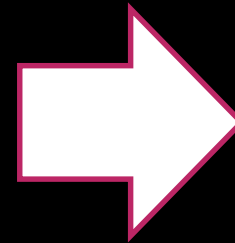
Input

BASE = 5
ALTURA = 4



Process

$(\text{BASE} * \text{ALTURA}) / 2$



Output

AREA = 10

```
class MathHelper
{
    public static string BigNumbersSum(string n1, string n2)
    {
        string[] numbers = new!preparedNumbers[number - 1];
        int shortNumLength, longNumLength;
        InBigNumbersSum(n1, n2, numbers, out shortNumLength, longNumLength);
        return ComputeBigNumbersSum(numbers, shortNumLength, longNumLength);
    }
    if (number < 4)
}
```



Hardware Dispositivos de Salida



Impresoras.

Parlantes.

Videobeams y proyectores.

Copiadoras de CD o DVD.

Módems.




```
class MathHelper
{
    public static string BigNumbersSum(string n1, string n2)
    {
        string[] numbers = new string[n1.Length + n2.Length];
        int shortNumLength, longNumLength;
        if (n1.Length < n2.Length)
        {
            shortNumLength = n1.Length;
            longNumLength = n2.Length;
        }
        else
        {
            shortNumLength = n2.Length;
            longNumLength = n1.Length;
        }
        return ComputeBigNumbersSum(n1, n2, numbers, shortNumLength, longNumLength);
    }
}
```



Hardware Dispositivos de Entrada



```
class MathHelper
{
    public static string BigNumbersSum(string n1, string n2)
    {
        string[] numbers = new string[n1.Length + n2.Length];
        int shortNumLength, longNumLength;
        if (n1.Length < n2.Length)
        {
            shortNumLength = n1.Length;
            longNumLength = n2.Length;
        }
        else
        {
            shortNumLength = n2.Length;
            longNumLength = n1.Length;
        }
        return ComputeBigNumbersSum(n1, n2, numbers, shortNumLength, longNumLength);
    }
}
```



Hardware Dispositivos de Entrada



```

class MathHelfer:
{
    public static string BigNumbersSum(string n1, string n2)
    {
        string[] numbers = new string[n1.Length + n2.Length];
        int shortNumLength, longNumLength;

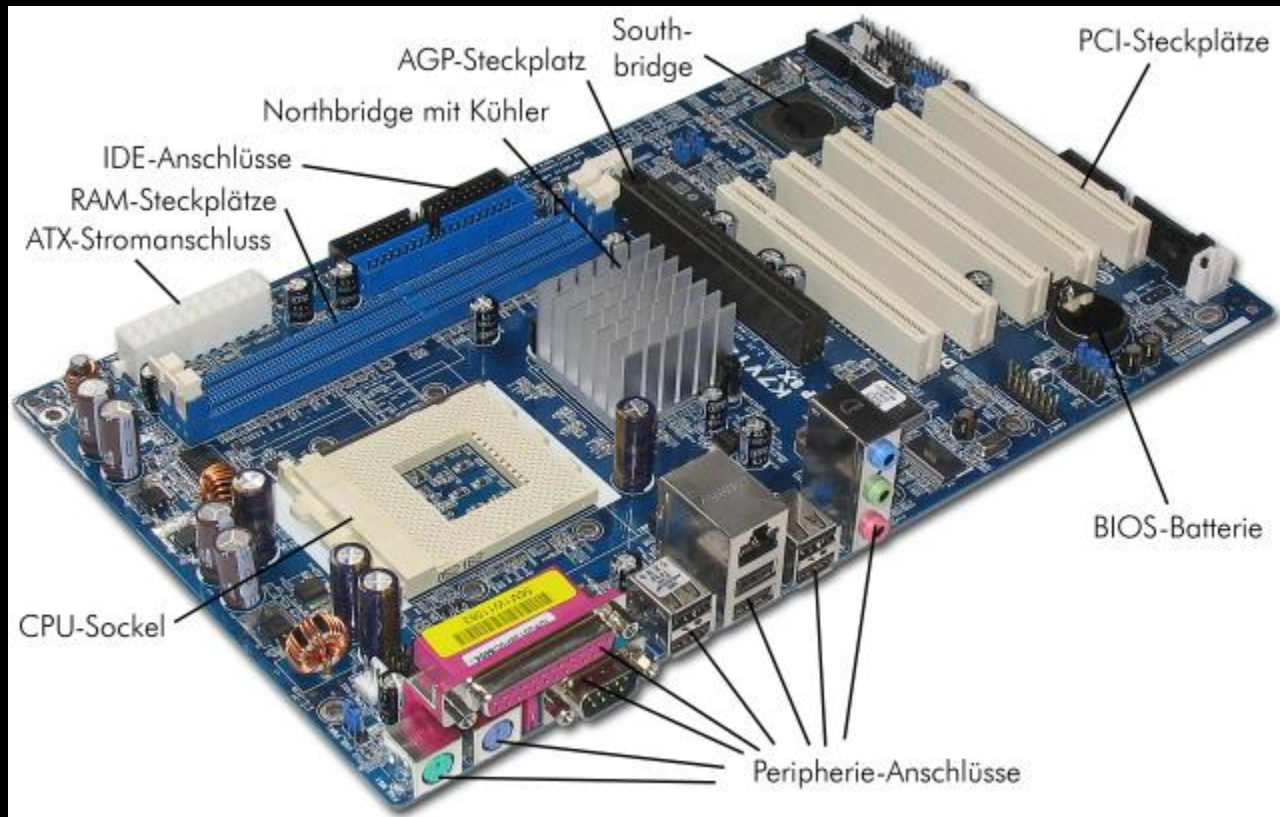
        if (n1.Length < n2.Length)
        {
            shortNumLength = n1.Length;
            longNumLength = n2.Length;
        }
        else
        {
            shortNumLength = n2.Length;
            longNumLength = n1.Length;
        }

        return ComputeBigNumbersSum(n1, n2, numbers, shortNumLength, longNumLength);
    }

    private static string ComputeBigNumbersSum(string n1, string n2, string[] numbers, int shortNumLength, int longNumLength)
    {
        if (number < 4)
    
```



Componentes Internos del Computador




```

class MathHelp
{
    public static string BigNumbersSum(string n1, string n2)
    {
        string[] numbers = new string[n1.Length + n2.Length];
        int shortNumLength, longNumLength;

        if (n1.Length < n2.Length)
        {
            shortNumLength = n1.Length;
            longNumLength = n2.Length;
        }
        else
        {
            shortNumLength = n2.Length;
            longNumLength = n1.Length;
        }

        return ComputeBigNumbersSum(n1, n2, numbers, shortNumLength, longNumLength);
    }

    private static string ComputeBigNumbersSum(string n1, string n2, string[] numbers, int shortNumLength, int longNumLength)
    {
        if (number < 4)
    
```



Componentes Internos del Computador



“RAM (Random Access Memory, memoria de acceso aleatorio): Memoria de almacenamiento primario. Almacena temporalmente instrucciones de programa y datos. El computador divide un chip de RAM en varias localidades de igual tamaño. ... La RAM ofrece al procesador un almacenamiento temporal para programas y datos”

Fuente.

http://www.iecov.edu.co/informatica/media_tecnica/documentos/computador.pdf

```
class MathHelp:
{
    public static string BigNumbersSum(string n1, string n2)
    {
        string[] num6 = new string[10];
        int shortNumLength, longNumLength;
        if (isPrepared && number < preparedNumbers)
        {
            return preparedNumbers[number - 1];
        }
        else
        {
            InBigNumbersSum(n1, n2, numbers, out shortNumLength, out longNumLength);
            return ComputeBigNumbersSum(numbers, shortNumLength, longNumLength);
        }
    }
}
```



Componentes Internos del Computador



El procesador (CPU, Central Processing Unit) es el componente más importante de nuestro PC. Su función es la de ser el cerebro de todo el funcionamiento del sistema, siendo el encargado de dirigir todas las tareas que lleva a cabo nuestro equipo.

```

class MathHelp:
{
    public static string BigNumbersSum(string n1, string n2)
    {
        string[] numbers = new string[2];
        int shortNumLength, longNumLength;

        if (isPrepared && number < preparedNumbers)
        {
            return preparedNumbers[number - 1];
        }

        InBigNumbersSum(n1, n2, numbers, out shortNumLength, out longNumLength);
        return ComputeBigNumbersSum(numbers, shortNumLength, longNumLength);
    }
}

```



Componentes Internos del Computador



Los discos duros:

Dispositivos de almacenamiento de datos para almacenar información digital. Como música, fotografías, vídeos, archivos de texto o programas informáticos, el disco duro es una de las partes más importantes de cualquier sistema informático.


```

class MathHelfer
{
    public static string BigNumbersSum(string n1, string n2)
    {
        string[] numbers = new string[2];
        int shortNumLength, longNumLength;

        if (isPrepared && number < preparedNumbers)
        {
            return preparedNumbers[number - 1];
        }

        else
        {
            InBigNumbersSum(n1, n2, numbers, out shortNumLength, longNumLength);
            return ComputeBigNumbersSum(numbers, shortNumLength, longNumLength);
        }
    }
}

```



Lenguajes

Python. Python

Java. ...

JavaScript. ...

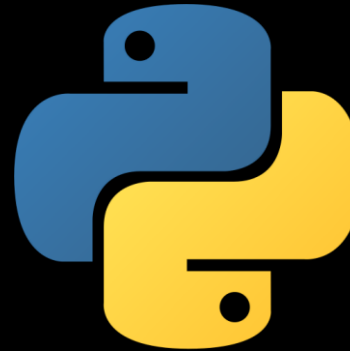
C# ...

PHP. ...

C/C++ ...

R. ...

Objective-C.





Representación de la Información en el computador

Bit: es la mínima unidad de memoria y se corresponde con uno de los valores posibles, en dígitos binarios 0 o 1.

Byte: representación de un solo carácter (letra, número, espacio o símbolo). Es la mínima unidad de información. Un byte equivale a 8 bits.

Kbyte (kilo byte): conjunto de 1024 bytes.

Mbyte (mega byte): es igual a 1024 kbytes.

Gbyte (giga byte): es igual a 1024 mbytes.

Tbyte (tera byte): es igual a 1024 gbytes.



Representación de la Información en el computador

DECIMAL	BINARIO	HEXADECIMAL
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F



¡GRACIAS POR SU
ATENCIÓN!