



UNIVERSIDAD CES

Un compromiso con la excelencia

Maestría en TIC en Salud

VIDEOCONFERENCIA 2

Programación Básica de
Computadores

Ing. Pedro Ortiz Tamayo

La Universidad CES es la propietaria y titular de todos los derechos de propiedad intelectual asociados al presente contenido. La comunicación pública del mismo se realiza, única y exclusivamente, con fines de divulgación e información. Por lo tanto, el material no se podrá usar para propósitos diferentes a los indicados. La presente divulgación no implica licencia, cesión o autorización de uso o explotación de ningún tipo de derechos de propiedad intelectual diferentes sobre el mismo. La copia, reproducción total o parcial, modificación, adaptación, traducción o distribución, infringe los derechos de la Universidad y causa daños por los que se podrá ser objeto de las acciones civiles y penales correspondientes y de las medidas cautelares que se consideren pertinentes o necesarias. Las opiniones expresadas por los autores o partícipes no constituyen ni comprometen la posición oficial o institucional de la Universidad CES.



"El secreto del cambio es centrar toda tu energía, no en luchar contra lo viejo, sino en construir lo nuevo."

SOCRATES

Consultoría

Alaphn Consulting asesora a empresas del sector salud en la planeación, montaje y ejecución de proyectos de tecnología de la información en salud, en áreas como Interoperabilidad, Imágenes Diagnósticas, Historia Clínica y Flujos de Información entre otros.

Proyectos de Radiología

Determinar la viabilidad económica y financiera y estructurar el proyecto de implementación de un sistema de RIS/PACS al interior de un hospital o de un centro radiológico, teniendo en cuenta los siguientes pasos:

- Análisis de propuestas.
- Definición de calificación de Proveedores.
- Elección del fabricante.
- Definición y validación del cronograma de proyecto.
- Definición de la infraestructura necesaria para el despliegue de la plataforma (Dicom Routers y Redes).
- Recolección y construcción de la información necesaria para la parametrización del sistema (Macros).

Fundamentos TIC

El objetivo de este módulo es el de transmitir conocimientos básicos de tecnología de la información, sobre los cuales está fundamentado el mundo actual, conocer el origen de la expresión "digital" de estos conceptos y determinar su impacto en el sector salud. El módulo está dividido en tres Unidades. En la primera se inicia con la exploración de los sistemas numéricos binario, decimal y hexadecimal, se explora el porqué de la adopción de sistema binario como base de la computación y su implementación en los circuitos digitales, se hace un recorrido desde los computadores de primera generación con tubos al vacío, hasta tecnologías actuales como motores de bases de datos, contenedores, la inteligencia artificial y su impacto en la sociedad futura. En la segunda unidad abordaremos la programación de computadores con las principales sentencias o instrucciones en pseudocódigo aplicadas en el simulador Pseint, los diagramas de flujo y su interpretación, y realizaremos varios ejercicios para desarrollar el pensamiento secuencial necesario en el abordaje de problemas en informática. Por último en la unidad final, nos adentraremos en el mundo de las bases de datos relacionales, plantearemos problemas de mundo real, interpretados mediante diagramas entidad-relación, veremos la transformación de estos diagramas en modelos relacionales y por último programaremos consultas a la base de datos utilizando las principales sentencias SQL (Structured Query Language).

[VIDEOCONFERENCIA 1](#)

[VIDEOCONFERENCIA 2](#)

[Home](#)

Modelado de Procesos en Salud

El objetivo global, de este módulo es la de dar a conocer a los estudiante que es la interoperabilidad, cual es su valor, en qué niveles se puede implementar, cuales son los estándares que debe cumplir y cuál es su aporte a la eficiencia y seguridad en salud. Como definición general, podemos decir que la interoperabilidad es la capacidad de los sistemas de información de compartir datos y posibilitar el intercambio de información entre ellos. En el ámbito de la salud, las instituciones tienen el objetivo de adoptar sistemas de información que permitan una gestión eficiente y centrada en los pacientes, para lograr esta eficiencia uno de los objetivos es la interoperabilidad; una interoperabilidad que posibilite la comunicación transversal y longitudinal a lo largo de todas las estructuras de los servicios de salud,

Algoritmo

:

En matemáticas, lógica, ciencias de la computación y disciplinas relacionadas, un algoritmo (del griego y latín, dicit algorithmus y este del griego arithmos, que significa «número», quizá también con influencia del nombre del matemático persa Al-Juarismi) es un conjunto prescrito de instrucciones o reglas bien definidas, ordenadas y finitas que permite llevar a cabo una actividad mediante pasos sucesivos que no generen dudas a quien deba hacer dicha actividad.



UNIVERSIDAD CES
Un compromiso con la excelencia



Estructura de un Algoritmo

Todo algoritmo consta de tres secciones principales:



Entrada: Es la introducción de datos para ser transformados.

Proceso: Es el conjunto de operaciones a realizar para dar solución al problema.

Salida: Son los resultados obtenidos a través del proceso.



UNIVERSIDAD CES
Un compromiso con la excelencia

Metodología para la descomposición de un algoritmo

- **Definición del problema**

En esta etapa se deben establecer los resultados y objetivos que se desea para poder saber si los datos que se tienen son suficientes para lograr los fines propuestos.

- **Análisis**

Una vez definido el problema se deberán organizar los datos de tal manera que sean susceptibles de usar en los cálculos siguientes.

- **Diseño**

En esta etapa se proponen soluciones a los problemas a resolver, por lo que se realiza una toma de decisiones aplicando los conocimientos adquiridos y utilizando los datos existentes.

- **Verificación o prueba de escritorio**

Se consideran resultados previstos para datos conocidos a fin de que al probar cada una de sus partes podamos ir comprobando que el algoritmo sirve o requiere modificarse.



UNIVERSIDAD CES
Un compromiso con la excelencia

Tipos de datos :

Numéricos: Representan un valor entero y real. Ejemplo: *Entero:* 250, -5
Real: 3.1416, -27.5

Lógicos: Sólo pueden tener dos valores (verdadero o falso), y son el resultado de una comparación.

Alfanuméricos: Son una serie de caracteres que sirven para representar y manejar datos como nombres de personas, artículos, productos, direcciones, etc.



UNIVERSIDAD CES
Un compromiso con la excelencia

Variables y Constantes :

Variables: Permiten almacenar de forma temporal un valor y el cual puede cambiar durante la ejecución del algoritmo o programa.

Toda variable tiene un nombre que sirve para identificarla.

Ejemplo: $\text{prom} = (\text{calf1} + \text{calf2} + \text{calf3}) / 3$ Las variables son: prom, calf1, calf2, calf3.

Constantes: Son datos numéricos o alfanuméricos que contienen un valor y que no cambia durante la ejecución del algoritmo ó programa.

Ejemplos: $\text{prom} = (\text{calf1} + \text{calf2} + \text{calf3}) / 3$ $\text{PI} = 3.1416$ Las constantes son: 3, PI.



UNIVERSIDAD CES
Un compromiso con la excelencia

Operadores y Expresiones :

Expresiones: Es un conjunto de constantes, variables, operadores con lo que se realizan las operaciones y permite obtener un resultado.

Ejemplo: resultado = $a*(2*b+5)/c$ Cal_final = $(cali1+cali2)/2$

Operadores matemáticos

- \wedge $**$
- $*$ $/$ div mod
- $+$ $-$

Los operadores con igual nivel de prioridad se evalúan de izquierda a derecha.



UNIVERSIDAD CES
Un compromiso con la excelencia

Operadores y Expresiones :

Operador de asignación

- = ó <-

Sirve para recuperar o guardar los valores obtenidos al realizarse o ejecutarse una expresión.

Operadores de relación

- Mayor que >
- Menor que <
- Mayor igual que >=
- Menor igual que <=
- Igual =
- Diferencia < > o !=

- Son empleados para comparar dos ó más valores.
- Su resultado produce valores como verdadero y falso.
- Tienen el mismo nivel de prioridad



UNIVERSIDAD CES
Un compromiso con la excelencia

Operadores y Expresiones :

AND		
VAL1	VAL2	RESUL
Cierto	Cierto	Cierto
Cierto	Falso	Falso
Falso	Cierto	Falso
Falso	Falso	Falso

OR		
VAL1	VAL2	RESUL
Cierto	Cierto	Cierto
Cierto	Falso	Cierto
Falso	Cierto	Cierto
Falso	Falso	Falso

NOT	
VAL1	RESUL
Cierto	Falso
Falso	Cierto

- Son empleados para comparar dos valores (verdadero y falso)
- Su resultado produce valores como verdadero y falso.
- Los tres tienen el mismo nivel de prioridad.



Operadores y Expresiones :



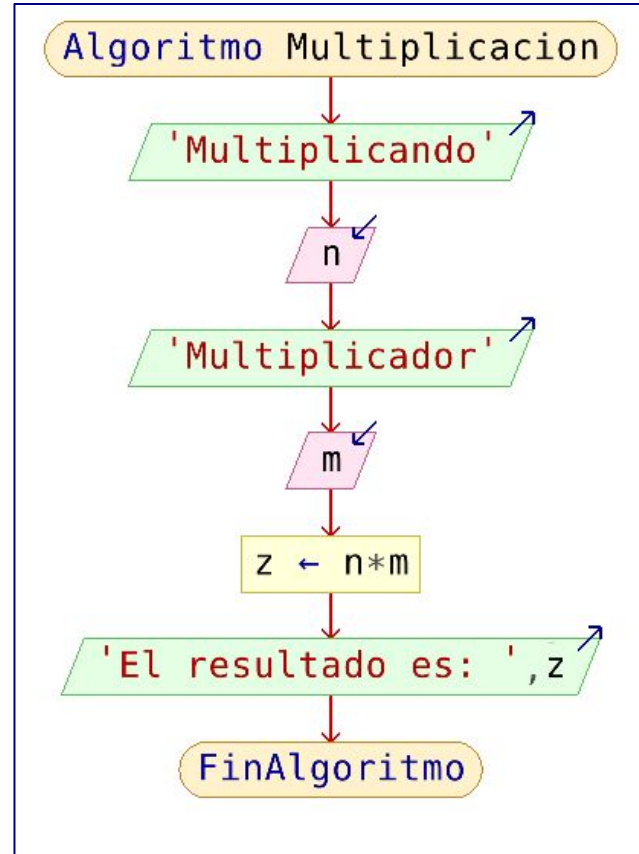
Siempre se ejecutan de izquierda a derecha en caso de haber dos ó más operadores con el mismo nivel de prioridad.



UNIVERSIDAD CES
Un compromiso con la excelencia

EJEMPLO : Pseudocode

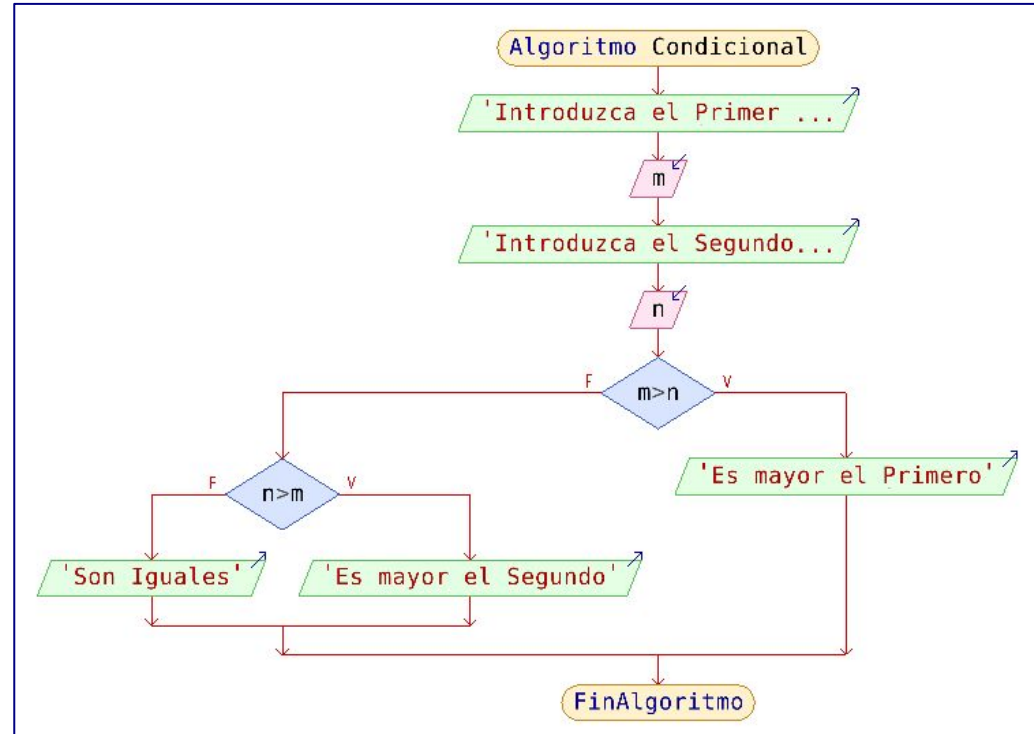
```
// Algoritmo Multiplicación de 2 Números  
output "Digite el Multiplicando"  
input n  
output "Digite el Multiplicador"  
input m  
x = n * m  
output "El Valor de la Multiplicación de  
",n," por ",m," es :",x
```



CONDICIONAL IF :

// Algoritmo de Comparación Entre 2 Números

```
output "Introduzca el Primer Número"
input n
output "Introduzca el Segundo Número"
input m
if n > m then
    output "El número ",n," es mayor que ",m
else
    if m > n then
        output "El número ",m," es mayor que ",n
    else
        output "Los números ",m," y ",n," son iguales"
    end if
end if
```



LOOP FOR :

// Algoritmo de Sumar los Primeros N Números Enteros con un For

output "Digite el Número de Enteros a Sumar"

input n

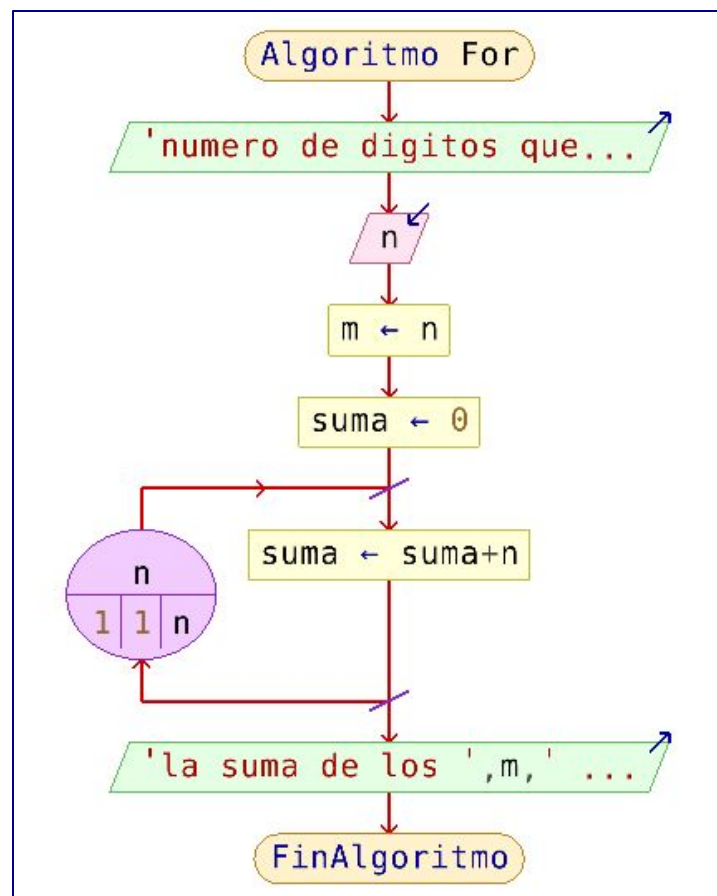
suma = 0

For i = 1 to n

 suma += i

end for

output "La suma de los ",n," primeros números enteros es :", suma



UNIVERSIDAD CES
Un compromiso con la excelencia

LOOP WHILE :

// Algoritmo de Sumar los Primeros N Números Enteros con un While

output "Digite el Número de Enteros a Sumar"

input n

suma = 0

i = 1

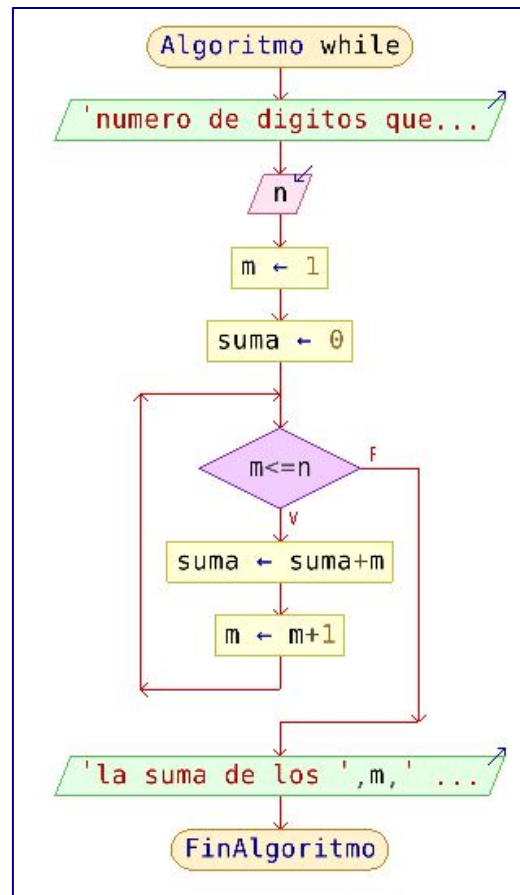
while i < n+1

 suma += i

 i += 1

end while

output "La suma de los ",n," primeros números enteros es : ", suma



UNIVERSIDAD CES
Un compromiso con la excelencia

EFICIENTE :

// Algoritmo de Sumar los Primeros N Números Enteros Eficiente

output "Digite el Número de Enteros a Sumar"

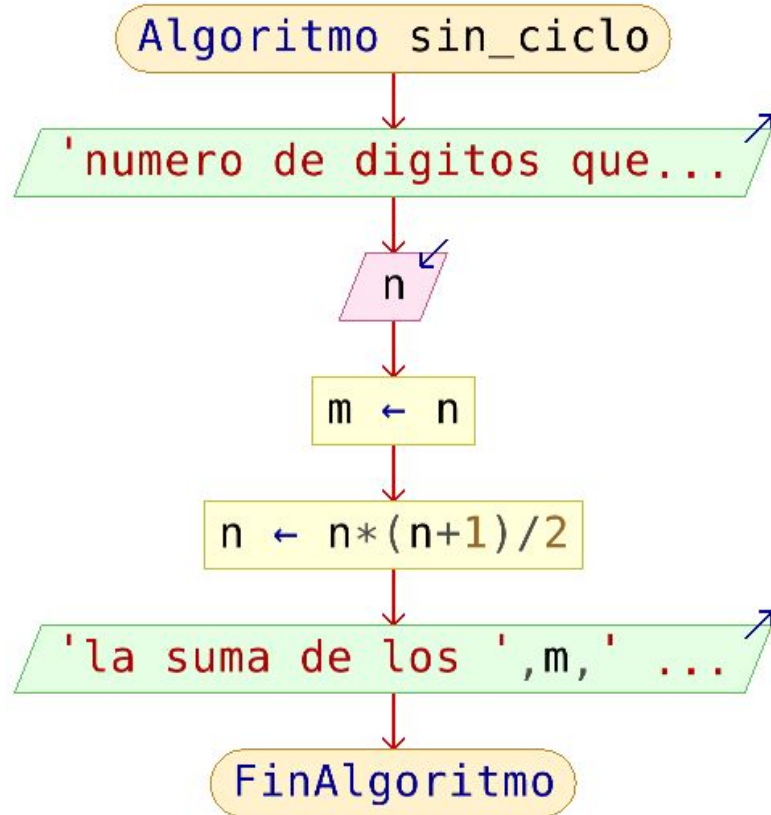
input n

suma = 0

suma = $n * (n + 1) / 2$

output "La suma de los ", n, " primeros números enteros es : ",

suma



UNIVERSIDAD CES
Un compromiso con la excelencia

SERIE ARMÓNICA :

$$\sum_{k=1}^{\infty} \frac{1}{k} = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} \dots$$

// Algoritmo de Serie Armónica

output "Digite el Número de Enteros a Sumar"

input n

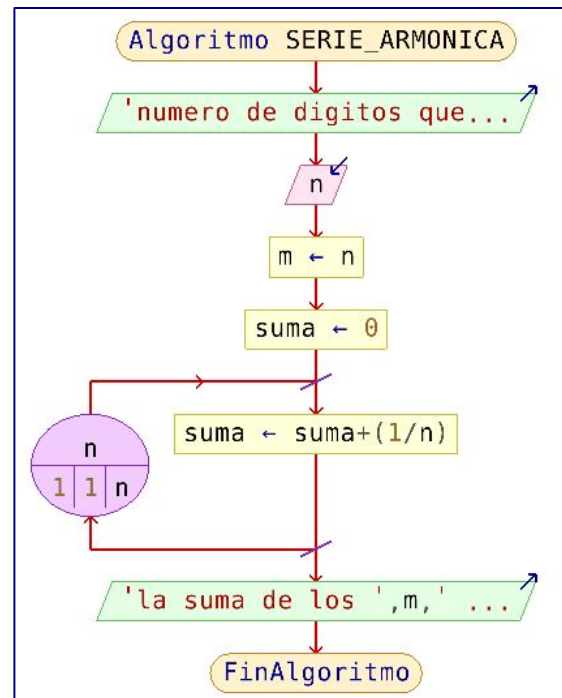
suma = 0

For i = 1 to n

 suma += 1 / i

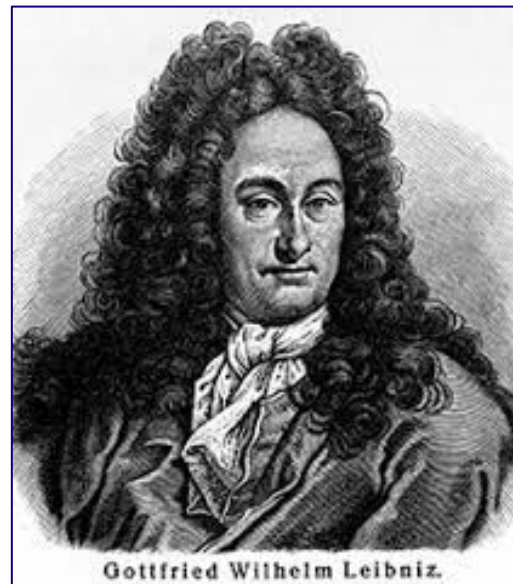
end for

output "La suma de los ",n," primeros términos de la Serie Armónica es :", suma



UNIVERSIDAD CES
Un compromiso con la excelencia

Gottfried Wilhelm Leibniz, a veces Gottfried Wilhelm von Leibniz (Leipzig, 1 de julio de 1646-Hannover, 14 de noviembre de 1716), fue un filósofo, matemático, lógico, teólogo, jurista, bibliotecario y político alemán.



UNIVERSIDAD CES
Un compromiso con la excelencia

SERIE Leibniz :

$$1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots = \frac{\pi}{4}$$

$$\sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} = \frac{\pi}{4}$$

// Algoritmo de Serie de Leibniz

output "Digite el Número de Enteros a Sumar"

input n

suma = 0

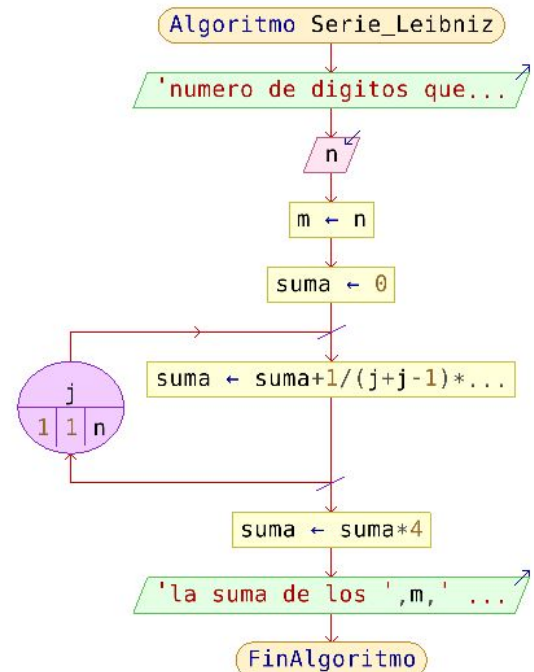
For i = 1 to n

 suma += (1/(i + i-1)) * (-1)^(i-1)

end for

suma1 = suma * 4

output "La suma de los ",n," primeros términos de la Serie de Leibniz es :", suma, "el Valor de π es :", suma1



UNIVERSIDAD CES
Un compromiso con la excelencia

Leonardo de Pisa (Pisa, c. 1170 - ib., post. 1240),¹ también llamado Leonardo Pisano, Leonardo Bigollo o simplemente **Fibonacci**, fue un matemático italiano. Difundió en Europa la utilidad práctica del sistema de numeración indo-arábigo frente a la numeración romana, y fue el primer europeo en describir la sucesión numérica que lleva su nombre.



UNIVERSIDAD CES
Un compromiso con la excelencia

SERIE Fibonacci :

// Algoritmo de Serie de Fibonacci

output "Digite el Número de Términos de la Serie de Fibonacci"

input n

if n = 1 then

output "Dígito 1 :",0

else if n = 2 then

output "Dígito 1 :",0

output "Dígito 2 :",1

else

output "Dígito 1 :",0

output "Dígito 2 :",1

ant = 0

sig = 1

For i = 3 to n

res = ant + sig

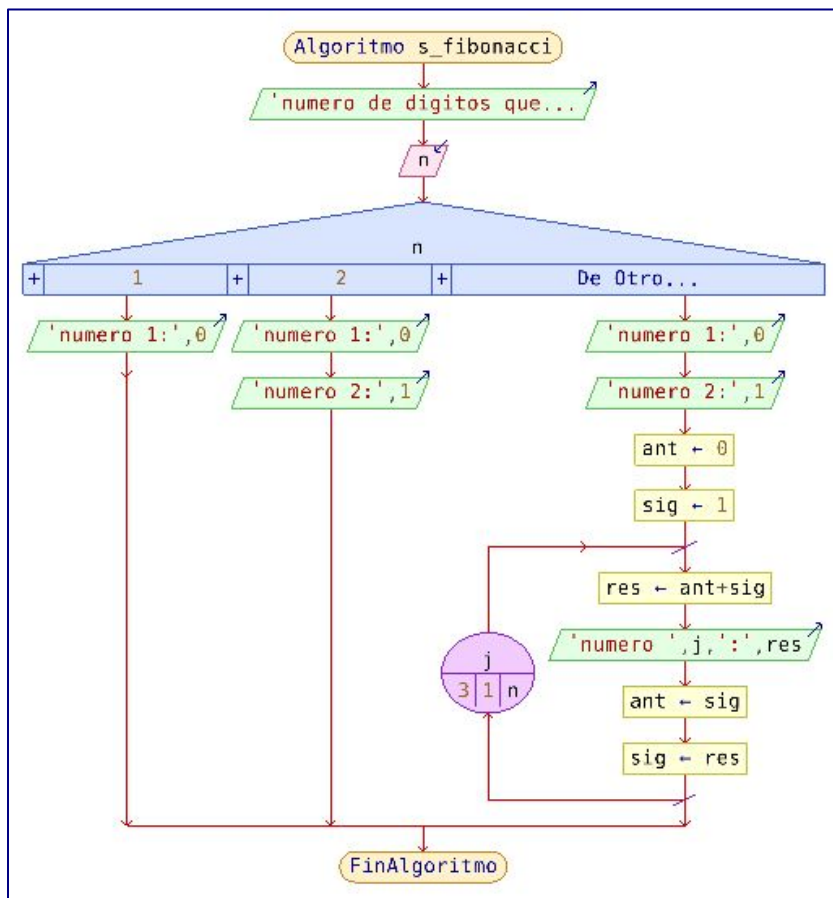
output "Digito ",i," :",res

ant = sig

sig = res

end for

end if



UNIVERSIDAD CES
Un compromiso con la excelencia

SERIE Fibonacci Recursivo :

// Algoritmo de Serie de Fibonacci Recursivo

```
function Fibonacci(n)
  if n = 0 or n = 1 then
    fi = n
    return fi
  else
    fi = Fibonacci(n-1) + Fibonacci(n-2)
    return fi
  end if
end function
```

output "Digite el Número de Términos de la Serie de Fibonacci"

input n

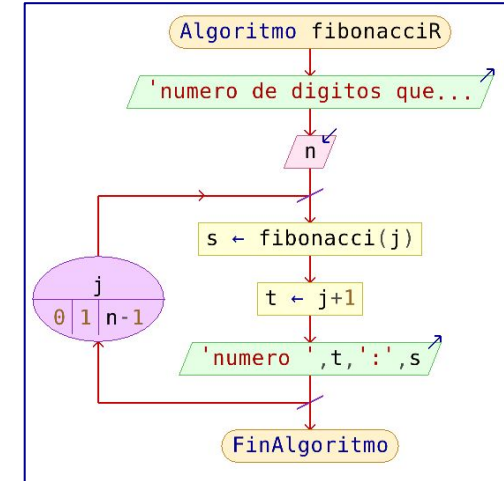
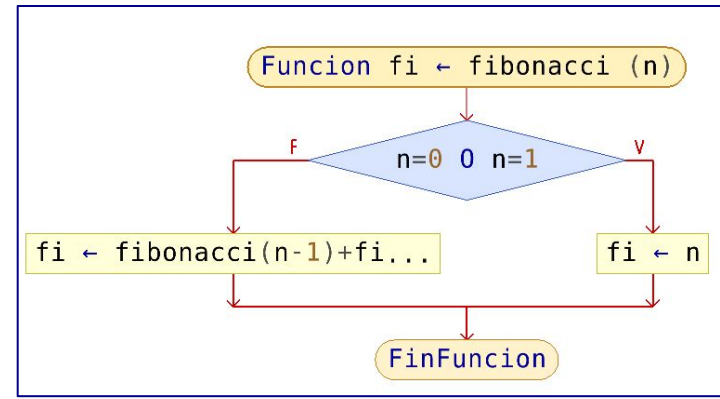
For i = 0 to n-1

s = Fibonacci(i)

t = i + 1

output "Dígito ",t," :",s

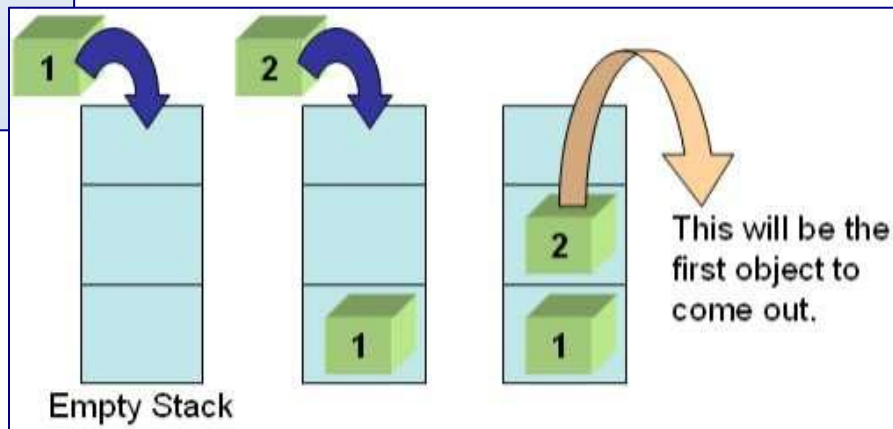
end for



UNIVERSIDAD CES
Un compromiso con la excelencia

RECURSIVIDAD:

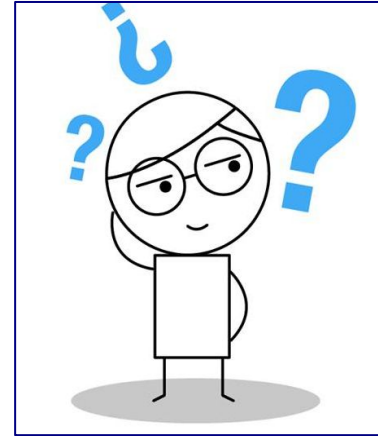
La recursividad es una técnica de programación que **nos permite la reducción de código, la programación de procesos y el manejo de la memoria principal**. Se utiliza para realizar una llamada a una función desde la misma función.



UNIVERSIDAD CES
Un compromiso con la excelencia

Problemas

- Calcular el área y el perímetro de cualquier circunferencia.
- Dados tres números determinar el mayor
- Determine la hipotenusa de un triángulo conocidos sus dos catetos
- Dado un número, determine si es par o impar
- Dado un número determinar si es primo



UNIVERSIDAD CES
Un compromiso con la excelencia



UNIVERSIDAD CES

Un compromiso con la excelencia

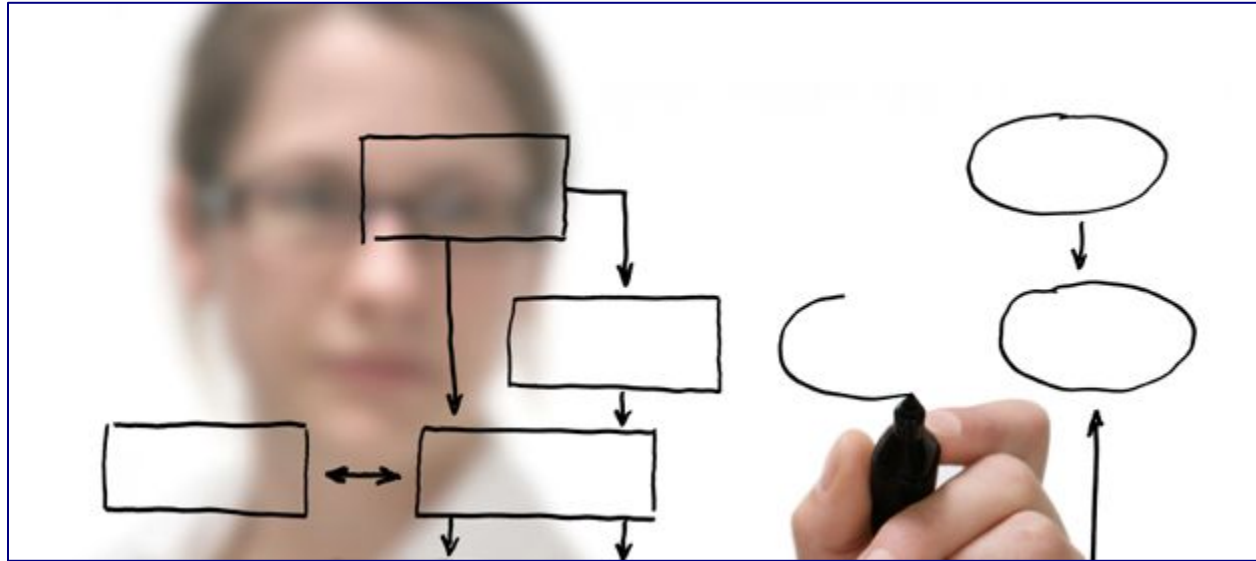
Maestría en TIC en Salud

Programación Avanzada de
Computadores

VIDEOCONFERENCIA 2

Ing. Pedro Ortiz Tamayo

La Universidad CES es la propietaria y titular de todos los derechos de propiedad intelectual asociados al presente contenido. La comunicación pública del mismo se realiza, única y exclusivamente, con fines de divulgación e información. Por lo tanto, el material no se podrá usar para propósitos diferentes a los indicados. La presente divulgación no implica licencia, cesión o autorización de uso o explotación de ningún tipo de derechos de propiedad intelectual diferentes sobre el mismo. La copia, reproducción total o parcial, modificación, adaptación, traducción o distribución, infringe los derechos de la Universidad y causa daños por los que se podrá ser objeto de las acciones civiles y penales correspondientes y de las medidas cautelares que se consideren pertinentes o necesarias. Las opiniones expresadas por los autores o partícipes no constituyen ni comprometen la posición oficial o institucional de la Universidad CES.



INTRODUCCIÓN A LA ARQUITECTURA DE SOFTWARE



UNIVERSIDAD CES
Un compromiso con la excelencia

ARQUITECTURA DE SOFTWARE

La arquitectura de software es el diseño de más alto nivel de la estructura de un sistema.

Una arquitectura de software, también denominada arquitectura lógica, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan un marco definido y claro para interactuar con el código fuente del software.

Una arquitectura de software se selecciona y diseña con base en objetivos (requisitos) y restricciones.

La arquitectura de software define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación entre ellos.

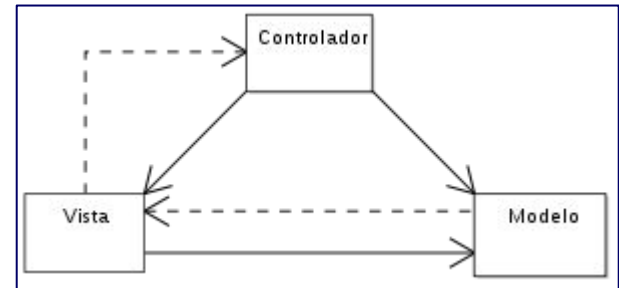


UNIVERSIDAD CES
Un compromiso con la excelencia

PATRÓN MVC

Modelo-vista-controlador (MVC) es un patrón de arquitectura de software, que separa los datos y la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones.

Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario



UNIVERSIDAD CES
Un compromiso con la excelencia

Arquitectura orientada a servicios SOA

La Arquitectura Orientada a Servicios (SOA, siglas del inglés Service Oriented Architecture) es un estilo de arquitectura de TI que se apoya en la orientación a servicios. La orientación a servicios es una forma de pensar en servicios, su construcción y sus resultados. Un servicio es una representación lógica de una actividad de negocio que tiene un resultado de negocio específico (ejemplo: comprobar el crédito de un cliente, obtener datos de clima, consolidar reportes de perforación).

SOA es una arquitectura de aplicación en la cual todas las funciones están definidas como servicios independientes con interfaces invocables que pueden ser llamados en secuencias bien definidas para formar los procesos de negocio.

En SOA la clave está en la interfaz, puesto que define los parámetros requeridos y la naturaleza del resultado. Esto significa que define la naturaleza del servicio y no la tecnología utilizada. Esta función permite realizar dos de los puntos críticos: los servicios son realmente independientes y pueden ser manejados.



UNIVERSIDAD CES
Un compromiso con la excelencia

Arquitectura de Microservicios MSA

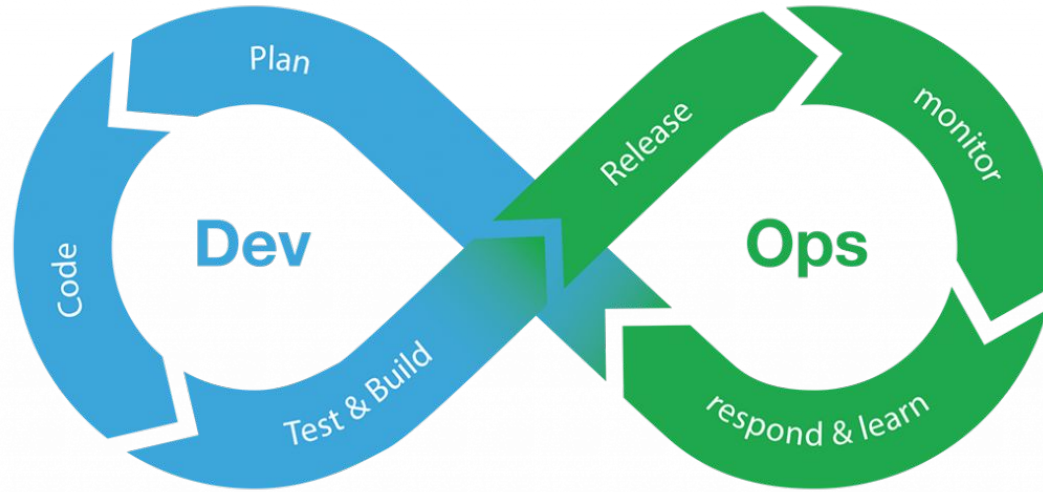
La Arquitectura de micro- servicios, conocido por las siglas MSA (del inglés Micro Services Architecture) es una aproximación para el desarrollo de software que consiste en construir una aplicación como un conjunto de pequeños servicios, los cuales se ejecutan en su propio proceso y se comunican con mecanismos ligeros (normalmente una API de recursos HTTP). Cada servicio se encarga de implementar una funcionalidad completa del negocio. Cada servicio es desplegado de forma independiente y puede estar programado en distintos lenguajes y usar diferentes tecnologías de almacenamiento de datos.

Se suele considerar la arquitectura de microservicios como una forma específica de realizar una arquitectura SOA.



UNIVERSIDAD CES
Un compromiso con la excelencia

SDLC - CICLO DE VIDA DE DESARROLLO DE SOFTWARE

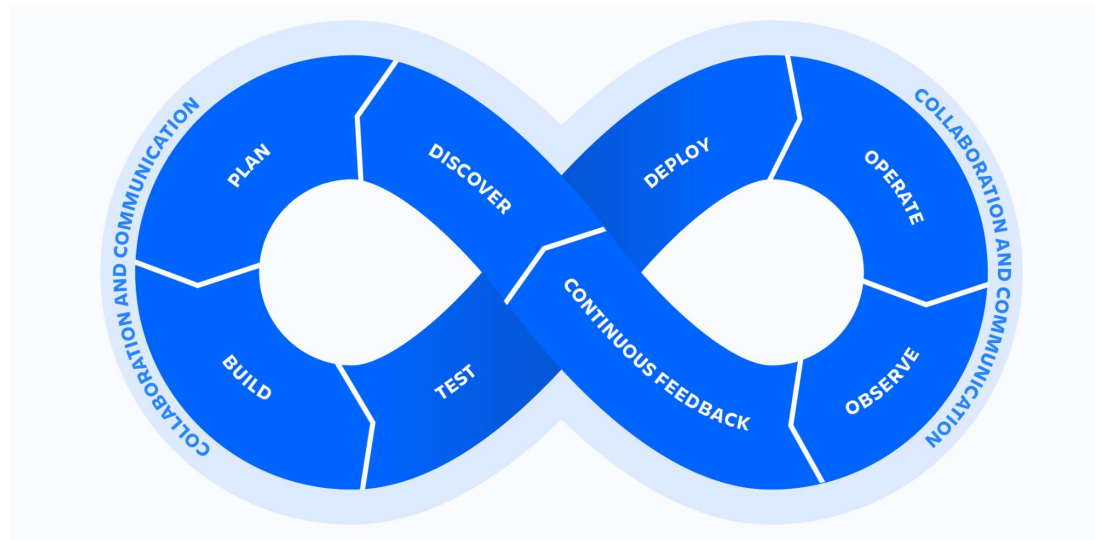


El ciclo de vida del desarrollo de software (SDLC) es un proceso rentable y eficiente en términos de tiempo empleado por los equipos de desarrollo para diseñar y crear software de alta calidad. El objetivo del SDLC es minimizar los riesgos del proyecto por medio de una planificación anticipada que permita que el software cumpla las expectativas del cliente durante la fase de producción y posteriormente. Esta metodología establece una serie de pasos que dividen el proceso de desarrollo de software en tareas que se pueden asignar, completar y medir.



UNIVERSIDAD CES
Un compromiso con la excelencia

DEVOPS

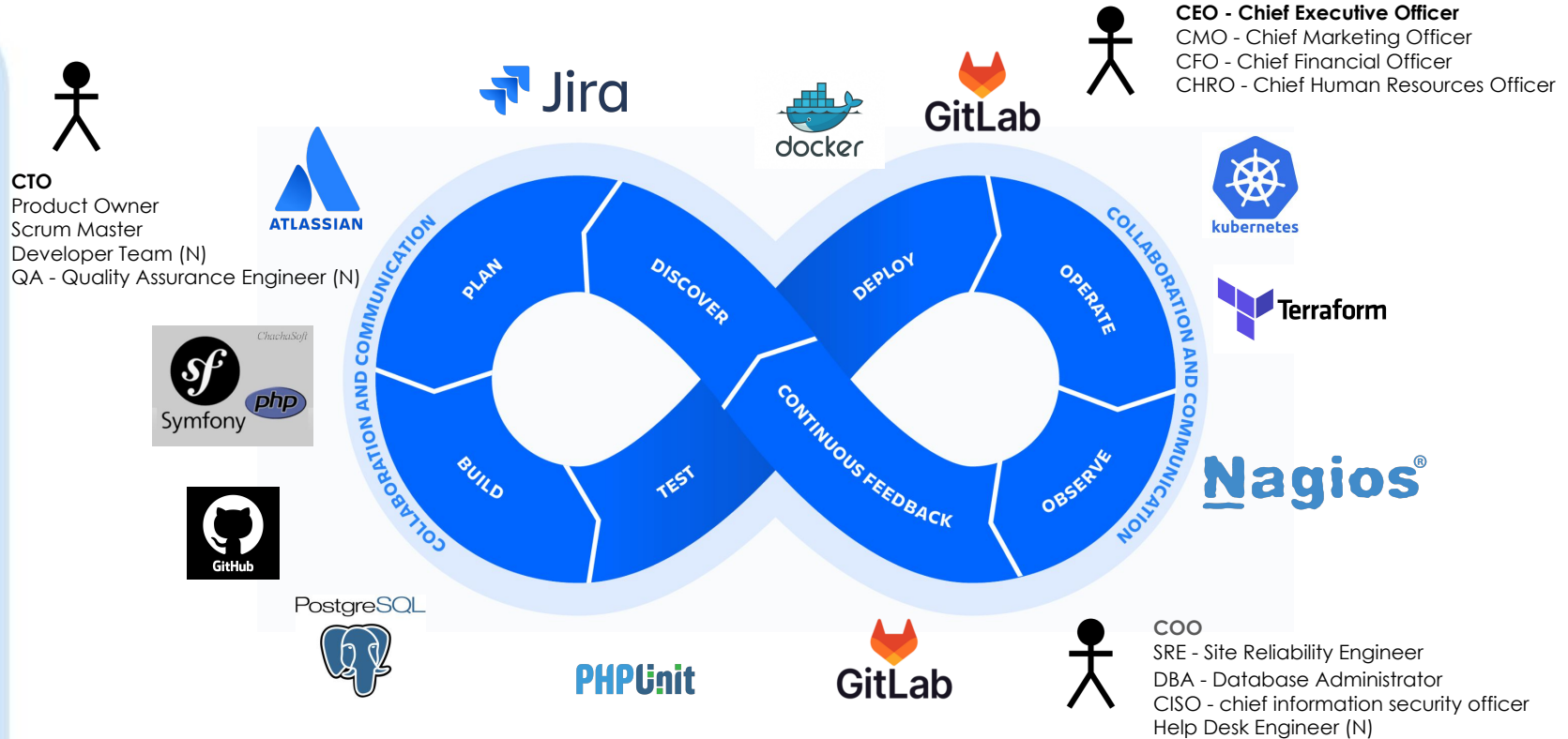


DevOps es un conjunto de prácticas, herramientas y filosofía cultural que sirve para automatizar e integrar los procesos que comparten el equipo de desarrollo de software y el de TI. Se centra en el empoderamiento de los equipos, la comunicación y colaboración entre ellos y la automatización de la tecnología.



UNIVERSIDAD CES
Un compromiso con la excelencia

ROLES Y HERRAMIENTAS



Estrategias:

- MSA Microservicios
- DEVOPS
- Estructura Multi Tenant
- CDN - content delivery network



UNIVERSIDAD CES
 Un compromiso con la excelencia

<https://htmledit.squarefree.com>



Tim Berners-Lee



UNIVERSIDAD CES
Un compromiso con la excelencia



Tim Berners-Lee

Tim Berners-Lee nació el 8 de junio de 1955, en Londres, Reino Unido.

Graduado en física en el Queen's College de la Universidad de Oxford en 1976, cuando tenía 21 años. Estando en la universidad, fabricó una computadora con un viejo televisor comprado en un taller de reparación.

World Wide Web

La idea de la World Wide Web era crear una red global de información a la que pudiera acceder cualquier persona con conexión a Internet. Berners-Lee imaginó un sistema que permitiera a los usuarios navegar por una vasta red de información utilizando hipervínculos.

Los hiperenlaces son enlaces en los que se puede hacer clic y que llevan a los usuarios de una página web a otra. Permiten a los usuarios navegar por la vasta red de información de la World Wide Web de forma rápida y sencilla.

El CERN era el nodo de Internet más importante de Europa y donde Berners-Lee consiguió unir Internet e hipertexto (HTTP y HTML), creándose así la World Wide Web el 12 de marzo de 1989, implementando la primera comunicación exitosa entre cliente y servidor de Protocolo de Transferencia de Hipertexto (HTTP) a través de Internet a mediados de noviembre de ese mismo año.



UNIVERSIDAD CES
Un compromiso con la excelencia



HTML es un lenguaje de marcado que se utiliza para el desarrollo de páginas de Internet. Se trata de la sigla que corresponde a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto, que podría ser traducido como Lenguaje de Formato de Documentos para Hipertexto.

Es un lenguaje muy simple y general que sirve para definir otros lenguajes que tienen que ver con el formato de los documentos. El texto en él se crea a partir de etiquetas, también llamadas tags, que permiten interconectar diversos conceptos y formatos.

Para la escritura de este lenguaje, pues, se crean las etiquetas que aparecen especificadas a través de corchetes o paréntesis angulares: < y >. Entre sus componentes, los elementos dan forma a la estructura esencial del lenguaje, ya que tienen dos propiedades (el contenido en sí mismo y sus atributos).



UNIVERSIDAD CES
Un compromiso con la excelencia

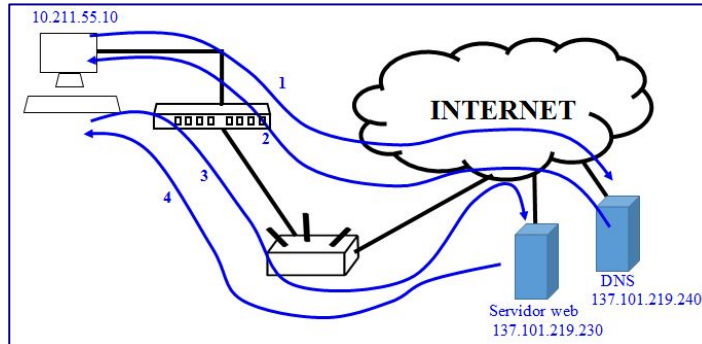


¿Qué es el protocolo http?

El http (del inglés **HyperText Transfer Protocol** o Protocolo de Transferencia de Hipertextos) es el protocolo de transmisión de información de la World Wide Web, es decir, el código que se establece para que el computador solicitante y el que contiene la información solicitada puedan “hablar” un mismo idioma a la hora de transmitir información por la red.

El funcionamiento del http se basa en un esquema de petición-respuesta entre el servidor web y el “agente usuario” (del inglés user agent) o cliente que realiza la solicitud de transmisión de datos. Un cliente puede ser un explorador determinado, cuando intentamos abrir una página web, o los rastreadores web (web crawlers o arañas web) que las inspeccionan.

A ellos el servidor brinda una respuesta estructurada de modo puntual y dotada de una serie de metadatos, que establecen las pautas para el inicio, desarrollo y cierre de la transmisión de la información. Estos son los “métodos de petición”, es decir, los comandos que disparan la ejecución de recursos determinados, cuyos archivos residen en el servidor.



UNIVERSIDAD CES
Un compromiso con la excelencia

¿Cómo escribir en HTML?

Una página en HTML es un archivo de texto simple, que puede ser programado desde en el Notepad de Windows o hasta en un editor de texto avanzado, y que contiene una serie de instrucciones identificadas con etiquetas para que el explorador pueda leer su contenido.

Los archivos de HTML tienen comúnmente las extensiones .html o .htm

Es importante que sea cual sea el medio por el cual se programa la página, se mantengan ciertas consideraciones de orden y limpieza del documento.



HTML Recomendaciones

1. Escribir todas las etiquetas en MAYÚSCULAS.
2. Una etiqueta puede contener atributos o propiedades que afectan la forma en que se muestra la información. Los atributos se escriben en minúsculas y deben sus valores de estar enmarcados por doble comillas (" ").
3. Para que una página esté bien programada, las etiquetas deben de estar propiamente acomodadas; se deben ordenar la primera y última etiqueta en la secuencia correcta. Esencialmente las etiquetas se colocan de afuera hacia dentro.

Veamos el ejemplo:

Incorrecto:

`<P>Esta es una frase connegritas</P>`

Correcto:

`<P>Esta es una frase connegritas</P>`



UNIVERSIDAD CES
Un compromiso con la excelencia

HTML index.html

`<HTML>` Indica el inicio del documento.
`<HEAD>` Define el inicio de la cabecera.
`<TITLE>` Inicia el título del documento.
`</TITLE>` Fin del título del documento.
`</HEAD>` Define el fin de la cabecera.
`<BODY>` Inicio del cuerpo del documento.
`</BODY>` Fin del cuerpo del documento.
`</HMTL>` Indica el fin del documento.

`<HTML>`
`<HEAD>`
`<TITLE>` Mi primer página de
HTML`</TITLE>`
`</HEAD>`
`<BODY>`
Este es el cuerpo de mi página
`</BODY>`
`</HTML>`



UNIVERSIDAD CES
Un compromiso con la excelencia

HTML <HEAD>

La cabecera está definida por las etiquetas <HEAD>...</HEAD>. Dentro de éstas se definen las características generales que afectarán a todo el documento.

<BASE> Define la liga de origen de todas las ligas situadas a lo largo del sitio completo.

<LINK> Define una referencia a otro documento.

<META> Describe las características generales de la programación que el autor desee incluir. Ejemplo: Si deseas incluir el estándar en que está programada aparecerá algo como:

<HEAD>

`<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">`

</HEAD>

<TITLE> Es el título de la página y se despliega en la parte superior del navegador. Ejemplo:

<HEAD>

`<title>Mi primer página de HTML</title>`

</HEAD>

<STYLE> Define características de formato sobre los elementos de la página. Ejemplo: El tipo de letra que será utilizado.

<SCRIPT> Ejecutar comandos o programación que complementan el HTML. Ejemplo: Incluir una línea de programación en Javascript para desplegar una cintilla de noticias.



UNIVERSIDAD CES
Un compromiso con la excelencia

HTML <BODY>

Cuerpo del documento (body)

El cuerpo es la segunda sección de la estructura del HTML y está definida por las etiquetas <BODY>...</BODY>.

Dentro de las etiquetas del cuerpo se localiza toda la información que será desplegada en los exploradores. Por ejemplo, podemos encontrar elementos tales como: encabezados, textos, tablas, imágenes, animaciones, etcétera, que contienen sus propios atributos y que veremos una a una en los siguientes temas.

```
<HTML>
<HEAD>
<TITLE> Mi primer página de HTML</TITLE>
</HEAD>
<BODY bgcolor="white" text="blue"
leftmargin="0" topmargin="0" marginwidth="0"
marginheight="0">
Este es el cuerpo de mi página
</BODY>
</HTML>
```



UNIVERSIDAD CES
Un compromiso con la excelencia

HTML <BODY> Atributos

- **bghcolor** Determina el color del fondo de la página.
- **background** Especifica una imagen determinada para ser usada como fondo de la página.
- **text** Color del texto de la página.
- **link** Color de las ligas o enlaces que contiene la página.
- **vlink** Color de las ligas o enlaces visitadas dentro de la página.
- **alink** Color que aparece al darle clic sobre una liga o enlace.
- **topmargin** Es la distancia en píxeles que existe entre la orilla superior del navegador y el contenido.
- **leftmargin** Es la distancia en píxeles que existe entre la orilla izquierda del navegador y el contenido.
- **marginheight** Tamaño del margen superior e inferior.
- **marginwidth** Tamaño del margen izquierdo y derecho.



HTML <P> Parágrafo

Los párrafos dentro del código sirven para designar líneas de texto que guardan la misma alineación y propiedades. Para definir un párrafo usamos la etiqueta <P>...</P>; la P significa Párrafo o Paragraph en inglés. Esta etiqueta puede ser abierta, aunque es preferible incluir las etiquetas de principio y fin de acción para evitar errores al momento de desplegar las páginas en los diferentes exploradores.

left center right

Para cambiar la alineación del texto incluye el atributo align (alinear) a la etiqueta de iniciar acción del párrafo <P>. El atributo de align acepta tres diferentes elementos: left (izquierda), center (centrado) o right (derecha). Veamos cómo se escribe la etiqueta:

```
<P align="center">Mi comunidad está llena de gente alegre y trabajadora</P>
```



UNIVERSIDAD CES
Un compromiso con la excelencia

Formato de texto

- **NEGRITAS** (BOLD) Existen dos etiquetas para asignar un texto en negritas ... Ó ...
- **ITÁLICAS** (ITALIC) Existen dos etiquetas para asignar un texto en Itálica <I>...</I> Ó ...
- **TAMAÑO DE LA FUENTE** (FONT SIZE) Se indica con el atributo size=* dentro de la etiqueta de FONT. Veamos un ejemplo: ...
- **COLOR DE LA FUENTE** (FONT COLOR) Se indica con el atributo color="#*****" dentro de la etiqueta font. Ejemplo: ...
- **SALTO DE LÍNEA** (LINE BREAK) Se indica utilizando la etiqueta abierta de

- **COMENTARIOS** (COMMENTS) Para escribir un comentario en el código sin que aparezca desplegado en el navegador indícalo con <!-- *** -->. Ejemplo: <!-- Este es mi comentario -->



UNIVERSIDAD CES
Un compromiso con la excelencia

Formato de texto : Ejemplo

```
<BODY bgcolor="#FFFFFF" text="#333333" link="#FF6600" vlink="#FF9900"
alink="#FF6600">
<!--Inicio de los ejemplos-->
<B>Este es un texto en negritas</B><BR>
<STRONG>Este es un texto en grueso</STRONG><BR>
<I>Este es un texto en itálicas</I><BR>
<EM>Este también es un texto en itálicas</EM><BR>
<B><I>Este es un texto en negritas e itálicas</I></B><BR>
<FONT SIZE="4">Este es un texto en tamaño 4</FONT><BR>
<FONT COLOR="#339900">Este es un texto verde</FONT>
<FONT face="verdana" size="4" color="#336699">Este es un texto mixto</FONT>
<!--Fin de los ejemplos-->
</BODY>
```



UNIVERSIDAD CES
Un compromiso con la excelencia

Títulos : Ejemplo

```
<BODY bgcolor="#FFFFCC">  
<FONT FACE="Arial" COLOR="#336699">  
<H2>Mi comunidad está llena de gente alegre y trabajadora.</H2><BR>  
<P ALIGN="left">Somos como una gran familia que trabaja en equipo  
por el bien de todos. Nos gusta que nuestros niños crezcan sanos y  
fuertes por eso todos nos esforzamos por dar lo mejor de nosotros  
mismos. Lo mejor que tenemos es la calidez de nuestra gente.</P>  
</FONT>  
</BODY>
```



UNIVERSIDAD CES
Un compromiso con la excelencia

Listas de elementos

Existen dos tipos de listas de elementos: listas ordenadas ("numbered lists" o listas numeradas) y listas no ordenadas ("bulleted lists" o listas con viñetas).

Las listas ordenadas se definen a través de la etiqueta `...` y las listas no ordenadas se definen con `...`. Cada elemento dentro de una lista tiene que estar identificado con la etiqueta `...` ("list item" o elemento de la lista).



Listas de elementos : Ejemplo

```
<H4>Esta es una lista ordenada</H4>  
<!--Esta es una lista ordenada-->  
<OL>  
<LI>Lunes</LI>  
<LI>Martes</LI>  
<LI>Miércoles</LI>  
</OL>  
<BR>  
<H4>Esta es una lista no ordenada</H4>  
<!--Esta es una lista no ordenada-->  
<UL>  
<LI>Lunes</LI>  
<LI>Martes</LI>  
<LI>Miércoles</LI>  
</UL>
```



Caracteres Especiales

Carácter Especial

- á
- Á
- é
- É
- í
- Í
- ó
- Ó
- ú
- Ú
- ñ
- Ñ

Etiqueta

á
Á
é
É
í
Í
ó
Ó
ú
Ú
ñ
Ñ



UNIVERSIDAD CES
Un compromiso con la excelencia



CONTINUACIÓN



UNIVERSIDAD CES
Un compromiso con la excelencia

1. Abre el editor.
2. Como primer punto deberás de crear la página principal de tu sitio, comúnmente conocida como Portada o Home.
3. Dale un nombre a tu página en el HEAD, en la etiqueta de Title, recuerda que ese título aparecerá en el navegador al visualizar tu documento.
4. Selecciona el color o imagen de fondo que usarás, colocando tu cursor en la etiqueta de BODY y modificando las propiedades en el Panel de Propiedades.
5. Determina el color del texto en la etiqueta de BODY.
6. Coloca un título en tu página, usando las opciones de Heading. Recuerda que deberás de mantener el diseño de tu sitio, así que haz varias pruebas hasta que quedes convencido de la mejor opción.
7. Redacta un pequeño texto que vaya de acuerdo al tema de tu sitio. Deberá ser un párrafo no menor de 8 líneas. El texto debe de llevar formato: tamaño, color y estilo (negrita, itálica) para resaltar algunas palabras; y deberá estar alineado a tu gusto.
8. Por último, incluye una lista de elementos dentro de tu texto.



HTML RECOMENDACIONES

Lograr un sitio interesante para los usuarios, no sólo depende de una buena programación técnica, es también importante saber organizar la información para que ésta sea clara y atractiva. El diseño visual o gráfico también forma parte de la programación de la página para lo cual debemos aprender ciertos conceptos básicos de la estructura de la información.

La sección superior de una página es la primera que un usuario ve, por lo que resulta recomendable incluir el logo o el título de la página, e incluso en algunas ocasiones el menú.

La sección del lado izquierdo es importante porque en nuestro idioma, leemos de izquierda a derecha, por lo que los usuarios verán inevitablemente esa parte. Es recomendable ubicar los elementos fijos de un sitio dentro de esta área, por ejemplo el menú.

En la sección central, que suele ser la más amplia de las páginas, se incluye el contenido del sitio. Dentro de esa área puedes organizar la información de la forma que creas más conveniente.



UNIVERSIDAD CES
Un compromiso con la excelencia

Hipervínculos

Una Hipervínculo es un enlace entre documentos (páginas) o sitios; puede ser una liga en textos (hipertexto) o una liga en imágenes o animaciones (hiper gráfico).

Precisamente el HTML está nombrado a partir de esta característica: Hyper Text Markup Language - Lenguaje de Marcas de HiperTexto.

La etiqueta general para definir las ligas es: `<A>...`
Cuando se trata de ligas hacia alguna página se usa el parámetro href (referencia a):
`...`

El elemento que se localice entre esas dos etiquetas será sensible, es decir que al tocar sobre ese elemento y dar un clic con el ratón, se creará un hiperenlace entre éste y la dirección hacia a donde se apunta.
Ejemplo:

`Da clic aquí para hacer una búsqueda en la página de Google`



UNIVERSIDAD CES
Un compromiso con la excelencia

Tipos de Hipervínculos

Vínculo externo

Todos los sitios en la Internet tienen una dirección o URL (Universal Resource Locator o en español, Localizador Universal de Recursos).

Los vínculos externos son enlaces entre sitios distintos y se indican colocando el URL de la página a dónde queremos enviar al usuario.

Los URL, generalmente son expresados de esta manera:

href="http://www.google.com"

Vínculo local

Un sitio es básicamente un conjunto de páginas ligadas entre sí. Sin embargo los sitios también contienen otros archivos, sonidos, imágenes, etc.

Los enlaces hechos entre estos elementos y páginas, se les conoce como locales, por ubicarse dentro de la misma agrupación.

Los vínculos locales se indican señalando directamente al archivo dentro de la organización del sitio.

href="archivo.html"



UNIVERSIDAD CES
Un compromiso con la excelencia

HTML EJEMPLO TIPO DE HIPERVÍNCULOS

```
<!--Este es el ejemplo de Vínculo local-->
<A href="dos.html">
<B>Ir a página dos &gt;</B>
</A>

<!--Aquí podemos ver un Vínculo externo-->
<A href="http://www.cca.org/" target="_blank">
Visitemos el Portal de los CCA<br>
a través de este vínculo externo
</A>
```



HTML ANCLAS

Anclas



Marcadores

Hemos visto los vínculos o ligas hacia otros documentos, de forma local o externa, sin embargo también existen ligas que apuntan hacia lugares precisos dentro del mismo documento que los contiene u otro. A esto se le conoce comúnmente como anclas (anchor), marcadores o apuntadores.

Las anclas se utilizan comúnmente cuando deseas hacer referencias rápidas hacia tu misma página y siempre constan de dos partes: el punto de anclaje o ancla y la liga que busca a esa ancla.

Por ejemplo, cuando tienes un texto muy largo y deseas volver al inicio sin tener que utilizar los botones de desplazamiento, puedes colocar un ancla en la parte superior de la página, y una liga en la parte final del documento que apunte al ancla.



UNIVERSIDAD CES
Un compromiso con la excelencia

HTML ANCLAS EJEMPLO

```
<BODY bgcolor="#FFFFFF" text="#333333"
link="#CC3300" vlink="#FF6600" alink="#CC3300">
<!--Este es el punto de anclaje o ancla-->
<A name="arriba"></A>
... Texto y contenido de la página... Antes de
finalizar aparece:
<!--Esta es la liga que va al punto de anclaje-->
<A href="#arriba">Regresar al t&iacute;tulo</A>
</BODY>
```



Imágenes

Para entender el proceso de insertar imágenes dentro del código de HTML, primero debemos de hablar de los formatos de imagen.

Formatos de imagen
Los formatos de imagen que se usan regularmente en los documentos en HTML, son el GIF, JPEG Y BITMAP. Al seleccionar el formato, hay que considerar particularmente el peso (kilobytes) y la calidad de una imagen (cantidad de colores).



UNIVERSIDAD CES
Un compromiso con la excelencia

HTML IMÁGENES EJEMPLO

```
<BODY bgcolor="#FFFFFF" text="#993300" link="#9933FF"
vlink="#CC99FF" alink="#9933FF">
<!--Inicio de los ejemplos-->
<h3 align="center">
<font face="Arial, Helvetica, sans-serif">Esta es la tiendita de
mi comunidad</font><br>
</h3>

...
</BODY>
```



UNIVERSIDAD CES
Un compromiso con la excelencia

`alt="***"` Texto alternativo de la imagen para navegadores que no tienen la Opción de Imagen Activada y/o también para que el visitante lea mientras se descarga la página.

`width="*"` Indica la anchura de la imagen en píxeles.

`height="*"` Indica el alto de la imagen, también en píxeles.

`border="*"` Indica el grosor del borde en píxeles. Especialmente cuando la imagen incluye un hiperenlace.

`align="top"` Alinear arriba, muy útiles cuando interactúan imágenes con texto o tablas, los atributos de alineación nos permiten jugar con el acomodo de los elementos de nuestra página.

`align="middle"` Alinear en medio.

`align="absmiddle"` Alinear perfectamente en medio.

`align="bottom"` Alinear abajo.

`align="left"` Alinear hacia la izquierda.

`align="right"` Alinear hacia la derecha.



Para agregar una liga en las imágenes deberás usar la etiqueta:
<IMG
src="nombreimagen.gif/jpeg">

```
<BODY bgcolor="#FFFFFF" text="#333333" link="#FF6600"  
vlink="#FF9900" alink="#FF6600">  
<!--Inicio del ejemplo-->  
<A href="http://www.cca.org.mx/mexico/oferta/homedoc.htm"  
target="_blank"><IMG src="../images/clipart/img_ejimglink.jpg"  
width="265" height="101" border="1"></A>  
<!--Fin del ejemplo-->  
</BODY>
```



UNIVERSIDAD CES
Un compromiso con la excelencia

Tablas

Las tablas son elementos de organización y sirven para estructurar el documento, situar otros elementos y mejorar el diseño. Están conformadas por filas (horizontales) y columnas (verticales) cuya unidad básica es la celda, de la cual hablaremos ampliamente en el siguiente tema.

Crear tablas puede parecer muy difícil en un comienzo, pero es cuestión de reflexionar previamente lo que necesitamos para luego construir el elemento.



UNIVERSIDAD CES
Un compromiso con la excelencia

Tablas

`<TABLE>`

`</TABLE>` Con esta etiqueta se define una tabla, definiendo dónde inicia y termina.

`<TR></TR>` Con estas etiquetas definimos las filas.

`<TD></TD>` Y finalmente con esta etiqueta definimos las celdas.

<code><TABLE></code>					
	<code><TR></code>	<code><TD></code> Celda 1 <code></TD></code>	<code><TD></code> Celda 3 <code></TD></code>	<code><TD></code> Celda 3 <code></TD></code>	<code></TR></code>
<code></TABLE></code>					



UNIVERSIDAD CES
Un compromiso con la excelencia

Tablas

Las tablas pueden configurarse como cualquier otro elemento dentro del HTML, a continuación encontrarás los atributos de la etiqueta TABLE:

border="*" Con este atributo podrás controlar el grosor en píxeles del borde de la tabla.

cellpadding="*" Define el espacio entre el borde de la celda y el contenido de la misma, se mide en píxeles.

cellspacing="*" Define el espacio entre celdas y está definido también en píxeles.

width="*" Determina la anchura de la tabla en píxeles o en porcentaje en relación al ancho del documento.

align="*"** Especifica la alineación de la tabla, izquierda, centrada o derecha. (left, center, right)



UNIVERSIDAD CES
Un compromiso con la excelencia

Las celdas

La celda es la unidad mínima y principal de una tabla. En ellas podemos colocar cualquier otro elemento que hayamos visto, títulos, textos, ligas, imágenes e incluso otras tablas.

width="*" Determina el ancho de celda, y se especifica en píxeles o porcentaje.

rowspan="*" Permite unir dos o más celdas en sentido horizontal (filas), su medida es la cantidad de filas a unir.

colspan="*" Permite unir dos o más celdas en sentido vertical (columnas), su valor es la cantidad de columnas a unir.

bgcolor="#*" Determina el color del fondo de la celda.

align="*" Especifica la alineación horizontal del contenido de la celda en posiciones: izquierda, centrada o derecha. (left, center, right)

valign="*" Especifica la alineación vertical del contenido de la celda en posiciones: arriba, centrada, abajo. (top, middle, bottom).



UNIVERSIDAD CES
Un compromiso con la excelencia



TALLER DE PROGRAMACIÓN



UNIVERSIDAD CES
Un compromiso con la excelencia

TALLER DE PROGRAMACIÓN

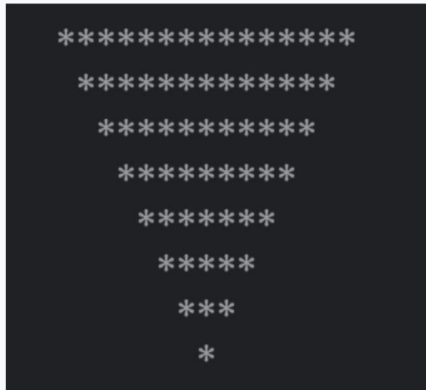
Apertura: viernes, 21 de febrero de 2020, 23:59

Cierre: domingo, 21 de abril de 2024, 23:59

1. El estudiante debe elaborar para cada problema planteado a continuación, un algoritmo que lo solucione y se pueda probar en el programa de pseudo-código **pseudocode**

Los problemas son:

- a) Se debe elaborar un algoritmo optimizado, el cual dado un número verifique si este es primo (Divisible por si mismo y por 1). Esta es una implementación mejorada de la Criba de Eratóstenes.
- b) Escribir un pequeño algoritmo, que pinte una pirámide invertida con símbolos asteriscos. El programa debe pedir la altura de la pirámide por pantalla. La imagen siguiente muestra una pirámide del altura 8.



UNIVERSIDAD CES
Un compromiso con la excelencia

CRIBA DE ERATÓSTENES

PRIMOS DEL 1 AL 100

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100



UNIVERSIDAD CES
Un compromiso con la excelencia

```

1 package main
2
3 import (
4     "fmt"
5 )
6
7 func main() {
8     var n int
9     fmt.Println("Digite la altura de la piramide ")
10    fmt.Scanln(&n)
11    for x := n; x > 0; x-- {
12        for z := 1; z <= n-x; z++ {
13            fmt.Print(" ")
14        }
15        for z := 1; z <= 2*x-1; z++ {
16            fmt.Print("*")
17        }
18        fmt.Println("")
19    }
20 }

```

```

PROBLEMS      OUTPUT      TERMINAL      DEBUG CONSOLE
● pedroortiz@MacBook-Air-de-PEDRO-3 Ipiramide % ls -l
total 3928
-rwxr-xr-x@ 1 pedroortiz  staff   2005744 Dec 27  2021 Ipiramide
-rw-r--r--@ 1 pedroortiz  staff      278 Dec 27  2021 Ipiramide.go
● pedroortiz@MacBook-Air-de-PEDRO-3 Ipiramide % go run Ipiramide.go
Digite la altura de la piramide
8
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
● pedroortiz@MacBook-Air-de-PEDRO-3 Ipiramide %

```

TALLER DE PROGRAMACIÓN

```
1 package main
2
3 import (
4     "fmt"
5     "strconv"
6 )
7
8 func main() {
9     var n, p, s, k int
10    fmt.Print("Digite el numero a verificar : ")
11    fmt.Scanln(&n)
12    s = 2
13    p = 0
14    k = int(n/2)
15    for s <= k && p == 0 {
16        if n%s == 0 {
17            p = 1
18        } else {
19            s++
20        }
21    }
22    if p == 0 {
23        fmt.Println("El termino", strconv.Itoa(n), " es Primo")
24    } else {
25        fmt.Println("El termino", strconv.Itoa(n), " NO es Primo")
26    }
27 }
28
```



```
EsPrimo --zsh -- 80x24
Last login: Fri Apr 12 14:51:20 on ttys000
[pedroortiz@MacBook-Air-de-PEDRO ~ % cd Desktop/UCES/GOLANG/EsPrimo
[pedroortiz@MacBook-Air-de-PEDRO EsPrimo % ls -l
total 8
-rw-----@ 1 pedroortiz  staff  390 Apr 12 15:45 EsPrimo.go
[pedroortiz@MacBook-Air-de-PEDRO EsPrimo % go run EsPrimo.go
Digite el numero a verificar : 1729
El termino 1729 NO es Primo
[pedroortiz@MacBook-Air-de-PEDRO EsPrimo %
```

Número Primo



UNIVERSIDAD CES
Un compromiso con la excelencia

FIN VIDEOCONFERENCIA #2

¡Gracias!



UNIVERSIDAD CES

Un compromiso con la excelencia