

ILERNA

Online

Videotutoría 1: Presentación e introducción

Módulo 03A: Programación



PRESENTACIÓN



¿QUÉ VAMOS A VER?

- ÍNDICE

- Unidades Formativas (UF)
- Plan de estudios (Guía didáctica)
- Evaluación
- Pruebas de Evaluación Continua (PAC)

UNIDADES FORMATIVAS

- **UF1. Programación estructurada**

- ¿Qué es un programa informático?
- ¿En qué consiste la programación estructurada? : Aprender a estructurar nuestro pensamiento
- Operadores y conversiones
- Comentarios de código

UNIDADES FORMATIVAS

- **UF2. Diseño modular**

- Métodos: Funciones y procedimientos
- Ámbito de las funciones
- Paso por valor y paso por referencia
- Errores en el código
- Recursividad

UNIDADES FORMATIVAS

- **UF3. Gestión de ficheros**

- Concepto de ficheros
- Operaciones con ficheros
- Ficheros de texto y ficheros binarios

PLAN DE ESTUDIO

GUÍA DIDÁCTICA. Información y planificación de la UF



¿Qué es la guía didáctica?

GUÍA DIDÁCTICA

- Documento colgado en **Plan de estudio**.
- Recoge los objetivos, contenidos, método de evaluación y calendario con las fechas de las PACs y videotutorías (Pruebas de evaluación continua).
- Fechas a tener en cuenta durante el semestre:
 - Publicación de la PACs.
 - Entrega de las PACs.
 - Publicación de Videotutorías

¿Qué son las PACs?

PRUEBAS DE EVALUACIÓN CONTINUA

- Colgadas en el apartado de **contenidos**.
- Organizadas de acuerdo a la guía didáctica del modulo.
- Objetivo: trabajar los contenidos importantes.
- Cada PAC incluye varias actividades sobre el temario (test, desarrollo, preguntas cortas..)
- 40% de la nota del módulo.
- La PAC tendrá que ser entregada en el formato pedido. En caso de entregarla en otro formato, el alumno se hace responsable de los problemas de incompatibilidad que puedan surgir al abrir los archivos, pudiendo esto afectar a su calificación.
- Entrega dentro del plazo establecido en la **Guía Didáctica** → Calificación máxima de 10.
- Las entregas fuera de plazo no se aceptarán. La calificación de esa PAC será 0, **suspenso**.

PRUEBAS DE EVALUACIÓN CONTÍNUA

PAC. Prueba de Evaluación Continua

- Tres tipos de PAC:
 - Actividad de autoevaluación
 - PAC Desarrollo
 - Test Evaluable

PRUEBAS DE EVALUACIÓN CONTÍNUA

PAC. Prueba de Evaluación Continua

- Actividad de autoevaluación:
 - Ejercicios interactivos.
 - Preguntas tipo test autocorregibles.

PRUEBAS DE EVALUACIÓN CONTÍNUA

PAC. Prueba de Evaluación Continua

- PACs de desarrollo:
 - Identificar bien la respuesta.
 - Citar las fuentes, aunque sea del material propio.
 - Si es enlace externo, pegarlo.
 - Se puede restar nota por presentación.
 - Enviar la PAC en el formato especificado: PDF, JPG, RAR, etc.

PRUEBAS DE EVALUACIÓN CONTÍNUA

PAC. Prueba de Evaluación Continua

- Test de evaluación:
 - Generalmente, preguntas tipo test autocorregibles.
 - Otro tipo de ejercicios.
 - Sólo 1 intento con un tiempo limitado
 - Las respuestas correctas se verán al finalizar la entrega de la PAC.

PRUEBAS DE EVALUACIÓN CONTÍNUA

PAC. Prueba de Evaluación Continua

¿Qué pasa si no entrego la PAC a tiempo?

- No se pueden entregar PAC fuera de plazo. Nota 0.
- Excepción: aviso con antelación de tiempo y con justificante.

¿Qué son las videotutorías?

- Contacto directo entre el alumno y el profesor.
- Una videotutoría semanal.
- Duración aproximada de 45-50 minutos.
- Calendario programado en la guía didáctica de cada módulo.
- Asistencia en directo o en diferido.
- Explicación de contenidos, resolución de dudas, comentarios sobre las actividades y preparación para el examen.
- Disponibles en el apartado de **contenidos**.

EJERCICIOS NO EVALUABLES

ÚNETE AL RETO

INTRODUCCIÓN

Tema 1: Introducción a la programación

INTRODUCCIÓN

“Everybody in this country should
learn how to program a computer...
because it teaches you how to think.”

- Steve Jobs

METODOLOGÍA

DIFERENCIA ENTRE SABER PROGRAMAR Y SABER CÓMO PENSAR COMO UN INFORMÁTICO

- Ejemplo: todo el mundo ha viajado en avión, pero obtener la licencia de conductor no forma parte de nuestros estudios.
- Pero los conocimientos que nos enseñan como funcionan los aviones sí lo son:
 - La física y las matemáticas ayudan a entender las fuerzas que intervienen (gravedad, presión atmosférica...)
 - La biología ayuda a entender qué le pasa al organismo humano (altura, oxígeno...)
 - La historia explica cómo fueron creados los aviones, cómo evolucionaron, y su papel en el transporte y el comercio.
- La asignaturas generales ayudan a entender una gran variedad de problemas, incluidos cómo vuelan los aviones. Es muy distinto de una asignatura que te enseña solo a manejar una de esas máquinas.

Yevgeniy Brikman.

<https://www.ybrikman.com/writing/2014/05/19/dont-learn-to-code-learn-to-think/>

METODOLOGÍA

¿CÓMO ENCAJA TODO EN NUESTRO CICLO?

- DAX 01: Sistemas informáticos: Manejo e instalación de sistemas operativos, redes y aplicaciones
- DAX 02: Bases de datos
- DAX 03: Programación
- DAX 04: Gestión de la información
- DAX 05: Ingeniería del software

IDEs

Visual Studio y Visual Studio Code

- <https://visualstudio.microsoft.com/es/vs/>
- [https://code.visualstudio.com/updates/v1 38](https://code.visualstudio.com/updates/v1_38)

INTRODUCCIÓN

¿¿¿QUÉ ORDEN DE ACCIONES DEBEMOS EJECUTAR PARA SABER
EL MAYOR DE DOS NÚMEROS???

INTRODUCCIÓN

¿¿¿QUÉ ORDEN DE ACCIONES DEBEMOS EJECUTAR PARA SABER EL MAYOR DE DOS NÚMEROS???

1. Pedimos al usuario los dos números
2. Leemos los valores
3. Creamos las variables para recoger esos valores
4. Condición -> Si la variable A es mayor que B, escribimos "A es mayor"
5. Condición - > Si la variable B es mayor que A, escribimos "B es mayor"
6. Sino, son iguales
7. Fin

DIAGRAMA DE FLUJO

¿CÓMO REPRESENTAMOS ESTAS SENTENCIAS?

- Diagrama de flujo de datos:

Referencia bibliográfica:

Fundamentos de programación
4ta Edición

Luis Joyanes Aguilar

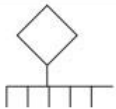
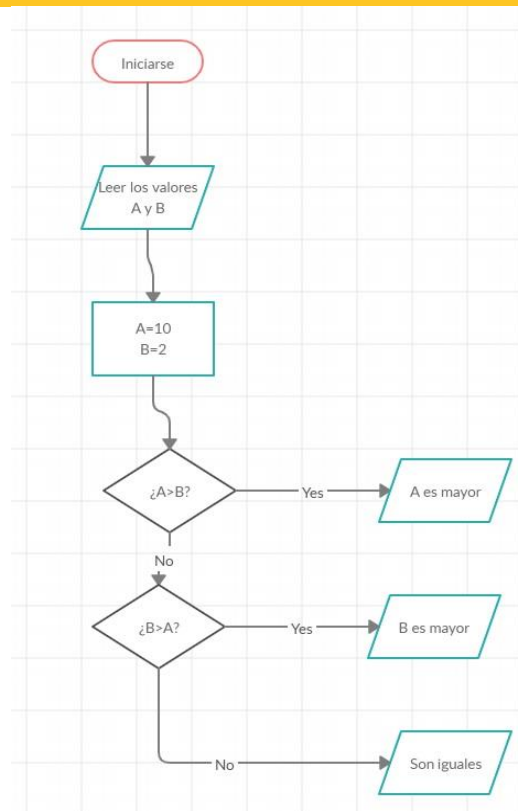
Símbolos principales	Función
	Terminal (representa el comienzo, "inicio", y el final, "fin" de un programa. Puede representar también una parada o interrupción programada que sea necesario realizar en un programa.
	Entrada/Salida (cualquier tipo de introducción de datos en la memoria desde los periféricos, "entrada", o registro de la información procesada en un periférico, "salida").
	Proceso (cualquier tipo de operación que pueda originar cambio de valor, formato o posición de la información almacenada en memoria, operaciones aritméticas, de transferencia, etc.).
	Decisión (indica operaciones lógicas o de comparación entre datos —normalmente dos— y en función del resultado de la misma determina cuál de los distintos caminos alternativos del programa se debe seguir; normalmente tiene dos salidas —respuestas SÍ o NO— pero puede tener tres o más, según los casos).
	Decisión múltiple (en función del resultado de la comparación se seguirá uno de los diferentes caminos de acuerdo con dicho resultado).
	Conector (sirve para enlazar dos partes cualesquiera de un ordinograma a través de un conector en la salida y otro conector en la entrada. Se refiere a la conexión en la misma página del diagrama.

DIAGRAMA DE FLUJO: EJEMPLO



VARIABLES EN C#

- Tipos de valor

- Tipos simples

- Entero con signo: `sbyte`, `short`, `int`, `long`
 - Entero sin signo: `byte`, `ushort`, `uint`, `ulong`
 - Caracteres Unicode: `char`
 - Punto flotante de IEEE: `float`, `double`
 - Decimal de alta precisión: `decimal`
 - Booleano: `bool`

- Tipos de enumeración

- Tipos definidos por el usuario con el formato `enum E {...}`

- Tipos de estructura

- Tipos definidos por el usuario con el formato `struct S {...}`

- Tipos de valor que aceptan valores NULL

- Extensiones de todos los demás tipos de valor con un valor `null`

- Entero con signo

- `sbyte` : 8 bits, de -128 a 127
 - `short` : 16 bits, de -32,768 a 32,767
 - `int` : 32 bits, de -2,147,483,648 a 2,147,483,647
 - `long` : 64 bits, de -9,223,372,036,854,775,808 a 9,223,372,036,854,775,807

- Entero sin signo

- `byte` : 8 bits, de 0 a 255
 - `ushort` : 16 bits, de 0 a 65,535
 - `uint` : 32 bits, de 0 a 4,294,967,295
 - `ulong` : 64 bits, de 0 a 18,446,744,073,709,551,615

- Punto flotante

- `float` : 32 bits, de 1.5×10^{-45} a 3.4×10^{38} , precisión de 7 dígitos
 - `double` : 64 bits, de 5.0×10^{-324} a 1.7×10^{308} , precisión de 15 dígitos

- Decimal

- `decimal` : 128 bits, al menos de -7.9×10^{-28} a 7.9×10^{28} , con una precisión mínima de 28 dígitos

Decimales

EJERCICIOS

Resuelve estos ejercicios utilizando un IDE en C#:

- Escribir un programa donde sume los primeros diez números naturales. Por definición convencional, se dirá que cualquier miembro del siguiente conjunto, $\mathbb{N} = \{1, 2, 3, 4, \dots\}$, es un número natural (+*)
- Escribir un programa donde se lea por teclado diez números y realizar la suma y la media de los mismos. (++*)
- Escribir un programa donde se muestre por pantalla los números impares desde el 1 al 99. (++*)
- Diseña un programa que calcule la suma, resta, multiplicación y división de dos números introducidos por teclado. (+++*)

*nivel de complejidad

