

Introducción a la programación

Lección 1

Objetivos

Habilidades/Conceptos	Objetivos del examen de MTA
Conceptos de la programación informática	Conceptos del almacenamiento y los tipos de datos de los equipos (1.1)
Conceptos de las estructuras de decisión	Conceptos de las estructuras de decisión informáticas (1.2)
Conceptos de las estructuras de repetición	Identificación de métodos apropiados para control de repetición (1.3)
Conceptos del control de excepciones	Conceptos del control de errores (1.4)

Algoritmos



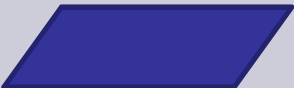


- Algoritmo se refiere a un método para resolver problemas.
- Técnicas comunes de representación de algoritmos:

Diagrama de flujo y tablas de decisión.	
Más preciso que el lenguaje natural	Menos formal y más fácil de usar que los lenguajes de programación

Diagrama de flujo

- Un diagrama de flujo es una representación gráfica de un algoritmo.

Símbolos comunes de los diagramas de flujo

	Inicio o fin de un algoritmo
	Proceso u operación computacional
	Operación de entrada o salida
	Operación de toma de decisiones
	Dirección del flujo de control

Ejemplo de diagrama de flujo

- Un diagrama de flujo que compara dos números:

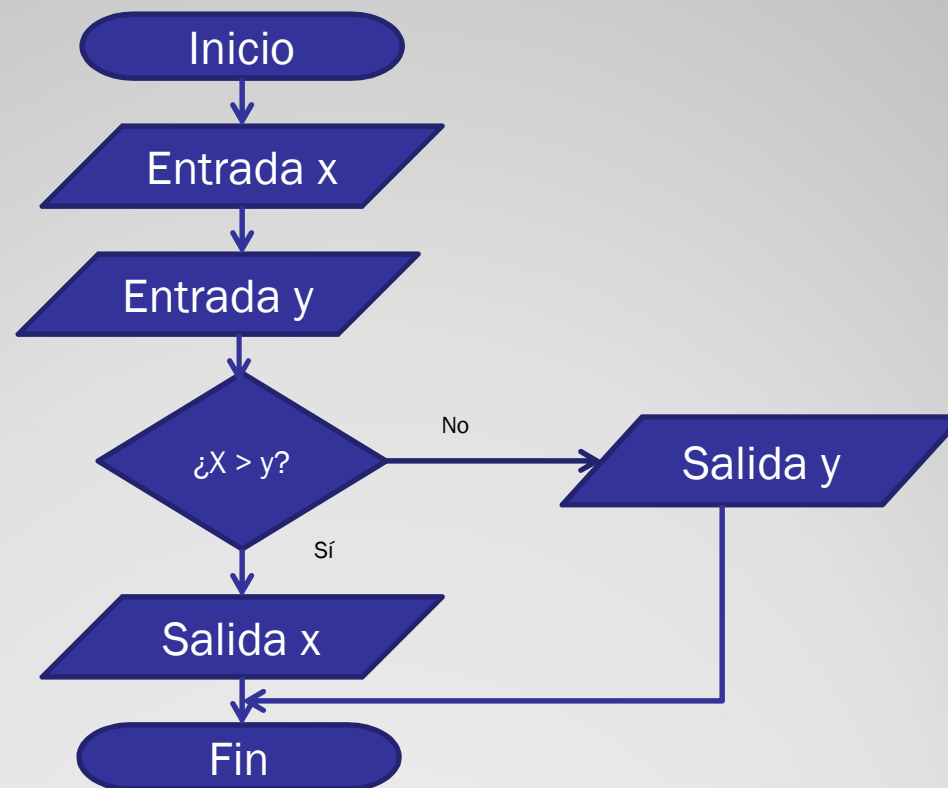


Tabla de decisión

- Útil para un gran número de condiciones
- Formato compacto y fáciles de leer
- Una tabla de decisión para calcular descuentos:

Cantidad < 10	S	N	N	N
Cantidad < 50	S	S	N	N
Cantidad < 100	S	S	S	N
Descuento	5%	10%	15%	20%

Introducción a C#

- Microsoft .NET Framework
 - Un entorno de ejecución
 - Bibliotecas de clases reutilizables
 - Compiladores de lenguaje
- El lenguaje de programación C#
 - Parte de .NET Framework
 - Lenguaje de alto nivel
 - El programa se debe compilar antes de ejecutarlo
 - Distingue mayúsculas y minúsculas

Estructura de un programa C#

```
using System;
namespace Lesson01
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("¡Hola, mundo!");
        }
    }
}
```


Elementos de un programa C#

- Seleccione elementos comunes de un programa C#:

Tipos de datos	Tipos de datos en un programa. Los tipos de datos comunes son int (entero), char (valor de carácter único), float (valores de punto flotante).
Variables	Proporciona almacenamiento temporal durante la ejecución del programa. <code>int number = 10;</code>
Constantes	Campos de datos cuyo valor no se puede modificar. <code>const int i = 10;</code>
Matrices	Una colección de elementos, a cada uno de los cuales se puede obtener acceso mediante un índice único. <code>int[] numbers = { 1, 2, 3, 4, 5 };</code>
Operadores	Símbolos que especifican la operación necesaria para realizar en los operandos antes de devolver un resultado.
Métodos	Los métodos son bloques de código que contienen una serie de instrucciones. Los métodos pueden recibir datos de entrada mediante argumentos y pueden devolver un valor al llamador.

Estructuras de decisión

Instrucción
if

Instrucción
if-else

Instrucción
switch

Instrucción if

- La instrucción if ejecutará una secuencia de instrucciones determinada únicamente si la expresión de tipo Boolean correspondiente se evalúa en true (verdadero).

```
int number1 = 10;  
int number2 = 20;  
if (number2 > number1)  
{  
    Console.WriteLine("number2 es mayor que number1");  
}
```

Instrucción if-else

- La instrucción **if-else** permite que un programa realice una acción si la expresión de tipo Boolean se evalúa en true y otra acción distinta si se evalúa en false.

```
public static void TestIfElse(int n)
{
    if (n < 10)
    {
        Console.WriteLine("n es menor que 10");
    }
    else if (n < 20)
    {
        Console.WriteLine("n es menor que 20");
    }
    else
    {
        Console.WriteLine("n es mayor o igual a 20");
    }
}
```

Instrucción switch

- La instrucción **switch** permite bifurcaciones multidireccionales. En muchos casos, usar instrucciones switch puede simplificar una combinación compleja de instrucciones if-else.

```
public static void TestSwitch(int op1, int op2, char opr)
{
    int result;
    switch (opr)
    {
        case '+':
            result = op1 + op2;
            break;
        case '-':
            result = op1 - op2;
            break;
        default:
            Console.WriteLine("Operador desconocido");
            return;
    }
    Console.WriteLine("Resultado: {0}", result);
    return;
}
```