Técnicas de Programación

Carrera Programador full-stack

Introducción (Repaso)

Algoritmos

Repaso

- Nos ayudan a resolver un problema
- Consisten de pasos lógicamente ordenados
- Dado un conjunto de datos de entrada da un resultado (solución al problema)
- Los algoritmos más simples son una lista de acciones que se ejecutan en orden (Secuencia)

Algoritmos

Repaso

```
console.log("Este");
console.log("algoritmo");
console.log("es");
console.log("secuencial");
```

Variables

Repaso

- Guarda información (números, letras, etc.)
- Tiene una dirección de memoria
- Tiene un nombre
- Todas las variables tienen un tipo (número, texto, lógico, etc.)
- Su contenido puede **variar** durante la ejecución del programa

Variables

Repaso

```
import * as rls from ‘readline-sync’;

let mensaje : string = rls.question("Ingrese el mensaje: ");
console.log("El mensaje es ", mensaje);
```

Operadores

Repaso

Operador	Significado	Ejemplo
=	Asignación	`nombre = "Juan"`
+	Suma	`total = cant1 + cant2`
-	Resta	`stock = disp - venta`
*	Multiplicación	`area = base * altura`
/	División	`porc = 100 * parte / total`
^	Potenciación	`sup = 3.41 * radio ^ 2`
% ó MOD	Resto de la división entera	`resto = num % div`

Operadores

Repaso

```
import * as rls from ‘readline-sync’;

let primerNumero : number = rls.questionInt("Ingrese el primer número: ");
console.log("el primer número es ", primerNumero);
let segundoNumero : number = rls.questionInt("Ingrese el segundo número: ");
console.log("el segundo número es ", segundoNumero);
let resultado : number = primerNumero + segundoNumero;
console.log("El resultado de la suma es ", resultado);
```

Técnicas de Programación

Carrera Programador full-stack

Ejercicios

Secuencia

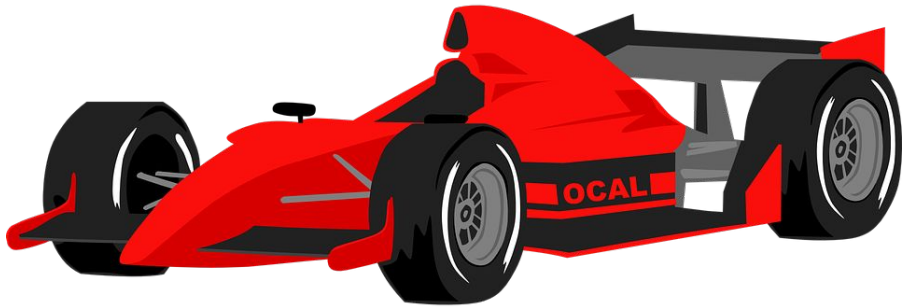
Ejercicio: Cálculo de Descuento

- Implemente un algoritmo que calcule y muestre por pantalla el precio final de un producto después de aplicarle un descuento
  - El precio inicial del producto es $450,50
  - El descuento a aplicar es del 10%. Recuerde que puede obtener el 10% de un valor multiplicado por 0,1
  - El precio y el descuento deben ser guardados en variables (no ingresados por teclado)

Estructuras de Control

Problema: Autos de Carrera

- En una prueba, un piloto tiene que completar 4 vueltas
- Se necesita un programa que permita ingresar por teclado el tiempo de cada vuelta
- El programa debe retornar el tiempo total y el promedio de vuelta



Técnicas de Programación

Carrera Programador full-stack

Selección (Conceptos)

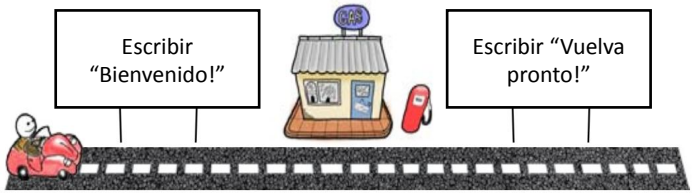
Estructuras de Control

Selección

Secuenciales

Selección o de Decisión

Repetitivas



Estructura de Control

Selección

- Un algoritmo puede ser más que una lista de comandos…

Estructura de Control

Selección

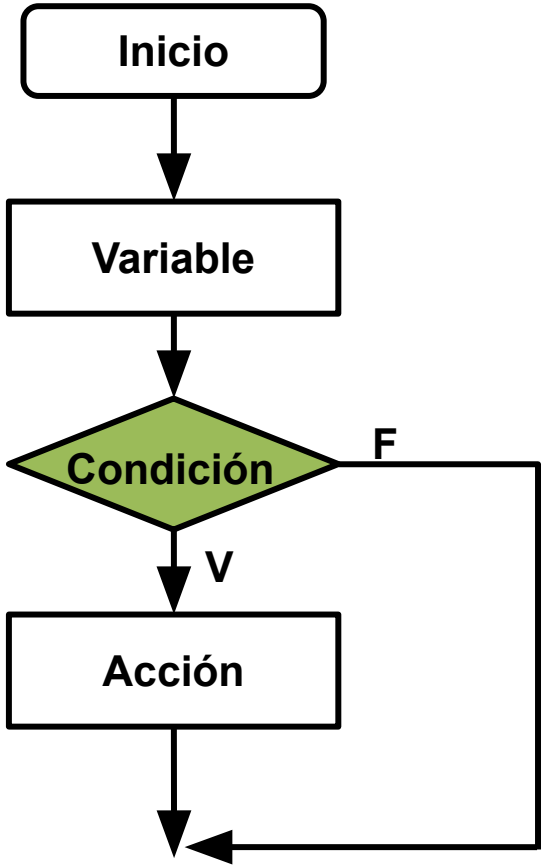
- ¿Qué camino tomo?

Extraído de: "Barry, P., & Griffiths, D. (2009). Head First Programming: A Learner's Guide to Programming Using the Python Language. " O'Reilly Media, Inc."."

Selección

Alternativa Simple

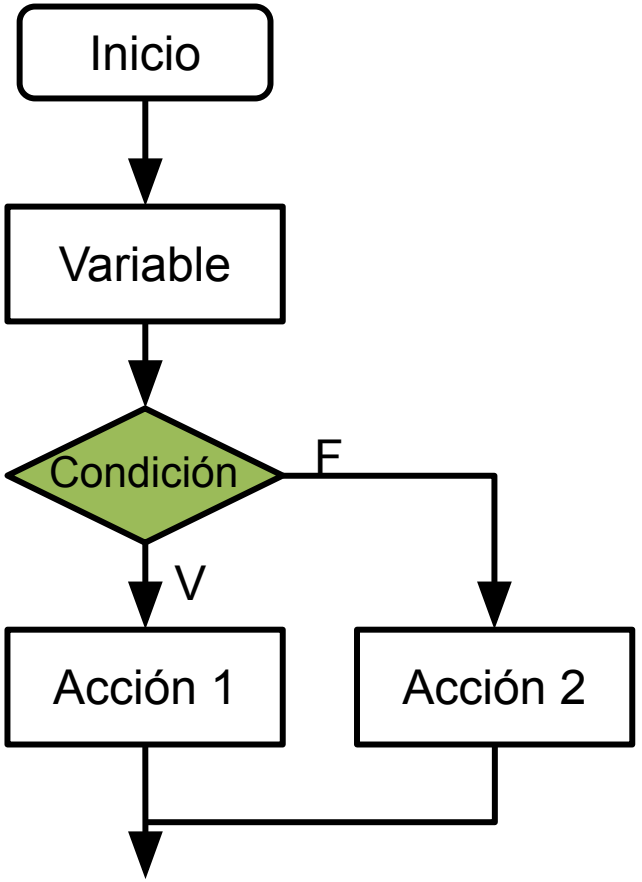
- Si la condición <Condición> es Verdadera <V> se realiza una acción <Acción>



Selección

Alternativa Doble

- Si la condición <Condición> es verdadera <V> se realiza una acción <Acción1>, pero si la condición es falsa <F> se realiza otra acción <Acción 2>



Selección

Sintaxis

```
if (condición) {
    <instrucciones si la condición se cumple>
} else {
    <instrucciones si la condición falla>
}
```

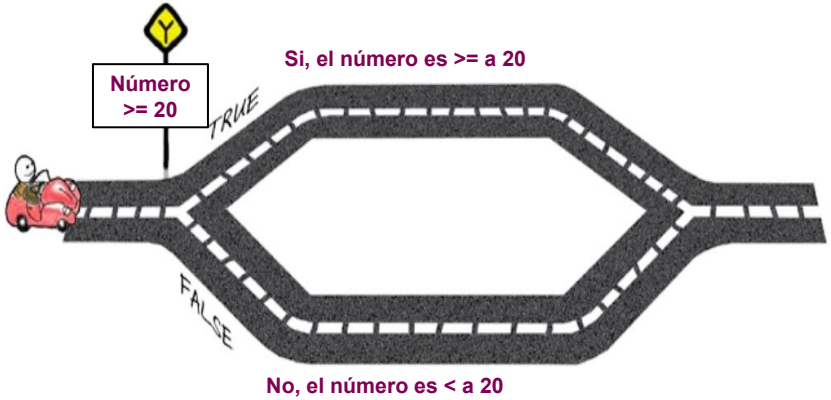
1. Se evalúa la condición
2. Se ejecutan las instrucciones que correspondan:
   a. el primer conjunto de instrucciones si la condición es verdadera
   b. o el juego de instrucciones luego del **else** si la condición es falsa.

La condición debe ser una expresión lógica, que al ser evaluada retorna Verdadero o Falso.

Comparativa code.org

Recuerden lo que haciamos en code.org al poner un bloque **“si”**. Lo que está dentro del bloque es lo que va entre llaves.

```
if (hayUnaPila) {
    console.log('Quito pila');
} else { //sino
    console.log('Lleno agujero');
}
```



Indentación

¿Qué es?

- Es una forma de escribir el código
- Sirve para que el código pueda *leerse más fácilmente*

¿Cómo se hace?

- Se usan *tabulaciones* (TAB)
- El código dentro de un bloque (llaves) se tabula hacia la *derecha*

```
let haceFrio : boolean = true;
let estaLloviendo : boolean = false;

if (haceFrio) {
if (estaLloviendo) {
console.log('Me quedo en casa');
} else {
console.log('Vengo al curso del CFL');
}
}
```

```
let haceFrio : boolean = true;
let estaLloviendo : boolean = false;

if (haceFrio) {
    if (estaLloviendo) {
        console.log('Me quedo en casa');
    } else {
        console.log('Vengo al curso del CFL');
    }
}
```

Operadores Condicionales

Operador	Significado	Ejemplo
>	Mayor que	3 > 1
<	Menor que	1 < 3
==	Igual que	1 == 2
>=	Mayor igual que	3 >= 2
<=	Menor igual que	4 <= 2
!=	Distinto que	1 != 2

Estructura de Control - Selección

Ejercicio - Mayor a 20 - Código

```
import * as rls from ‘readline-sync’;

let nroDeseado : number = rls.questionInt(”Escriba el número que desea verificar si es mayor o no a 20: “);

if (nroDeseado > 20) {
        console.log('El número es mayor a 20: ' +  nroDeseado);
} else {
        console.log('El número es menor o igual a 20: ' +  nroDeseado);
}
```

Operadores Relacionales

Los operadores relacionales definidos por JavaScript son idénticos a los que definen las matemáticas: mayor que **(>)**, menor que **(<)**, mayor o igual **(>=)**, menor o igual **(<=)**, igual que **(==)** y distinto de **(!=)**.

```
let numero1 : number = 3;
let numero2 : number = 5;
let resultado : boolean;
resultado = numero1 > numero2; // resultado=false
resultado = numero1 < numero2; // resultado=true

numero1 = 5;
numero2 = 5;
resultado = numero1 >= numero2; // resultado=true
resultado = numero1 <= numero2; // resultado=true
resultado = numero1 == numero2; // resultado=true
resultado = numero1 != numero2; // resultado=false
```

El resultado de todos estos operadores siempre es un valor booleano

Operadores Relacionales

El operador == se utiliza para comparar el valor de dos variables, por lo que es muy diferente del operador =, que se utiliza para asignar un valor a una variable

```
// El operador "=" asigna valores
let numero1 : number = 5;
let resultado : number = numero1 = 3;  // numero1=3 y resultado=3


// El operador "==" compara variables
let numero1 : number = 5;
let resultado : boolean = numero1 == 3; // numero1=5 y resultado=false
```

Operadores Relacionales

Los operadores relacionales también se pueden utilizar con variables de tipo cadena de texto

```
let texto1 : string = "hola";
let texto2 : string = "hola";
let texto3 : string = "adios";
let resultado : boolean;
resultado = texto1 == texto3; // resultado = false
resultado = texto1 != texto2; // resultado = false
resultado = texto3 >= texto2; // resultado = false
```

Cuando se utilizan cadenas de texto, los operadores "mayor que" (>) y "menor que" (<) siguen un razonamiento no intuitivo: se compara letra a letra comenzando desde la izquierda hasta que se encuentre una diferencia entre las dos cadenas de texto. Para determinar si una letra es mayor o menor que otra, las mayúsculas se consideran menores que las minúsculas y las primeras letras del alfabeto son menores que las últimas (a es menor que b, b es menor que c, A es menor que a, etc.)

Operadores Lógicos

- A veces no es necesario con una única condición
- Se pueden utilizar múltiples condiciones y unirlas con operadores lógicos

Operador	Significado	Descripción	Ejemplo
&&	Conjunción (Y)	Ambas son Verdaderas	(7>4) && (2==2)
||	Disyunción (O)	Al menos una es verdadera	(1==1 || 2==1)
!	Negación (No)	No es verdadero	!(2<5)

Ejemplo Condición Compuesta

- Se tienen dos condiciones evaluadas con una conjunción (condicion_a **Y** condición_b)
- Conjunción → *ambas* tienen que ser *verdaderas*

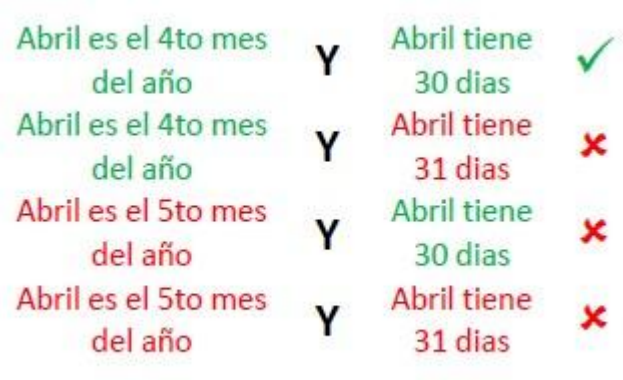
```
let lucesEncendidas : boolean = true;
let litrosNafta : number = 10;

if (lucesEncendidas && litrosNafta > 0) {
   console.log('Puedo manejar de noche');
}
```

Operadores Lógicos - AND

La operación lógica **AND** obtiene su resultado combinando dos valores booleanos. El operador se indica mediante el símbolo **&&** y su resultado es **true** si y sólo si, los dos operandos son **true**:

```
let valor1 : boolean = true;
let valor2 : boolean = false;
let resultado : boolean = (valor1 && valor2); // false
```

```
let valor1 : boolean = true;
let valor2 : boolean = true;
let resultado : boolean = (valor1 && valor2); // true
```

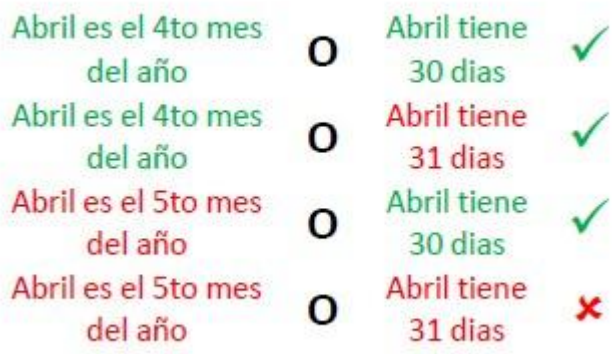


Operadores Lógicos - OR

La operación lógica **OR** también combina dos valores booleanos. El operador se indica mediante el símbolo || y su resultado es **true** si alguno de los dos operandos es **true**:

```
let valor1 : boolean = true;
let valor2 : boolean = false;
let resultado : boolean valor1 || valor2; // true
```

```
let valor1 : boolean = false;
let valor2 : boolean = false;
let resultado : boolean  = valor1 || valor2; // false
```



Operadores Lógicos - NOT

La operación lógica **NOT** se aplica a un único valor booleano.
El operador se indica mediante el símbolo **!** y su resultado es **true** si el operando es **false** y viceversa:

```
let valor : boolean = true;
let resultado : boolean = !(valor); // false
```

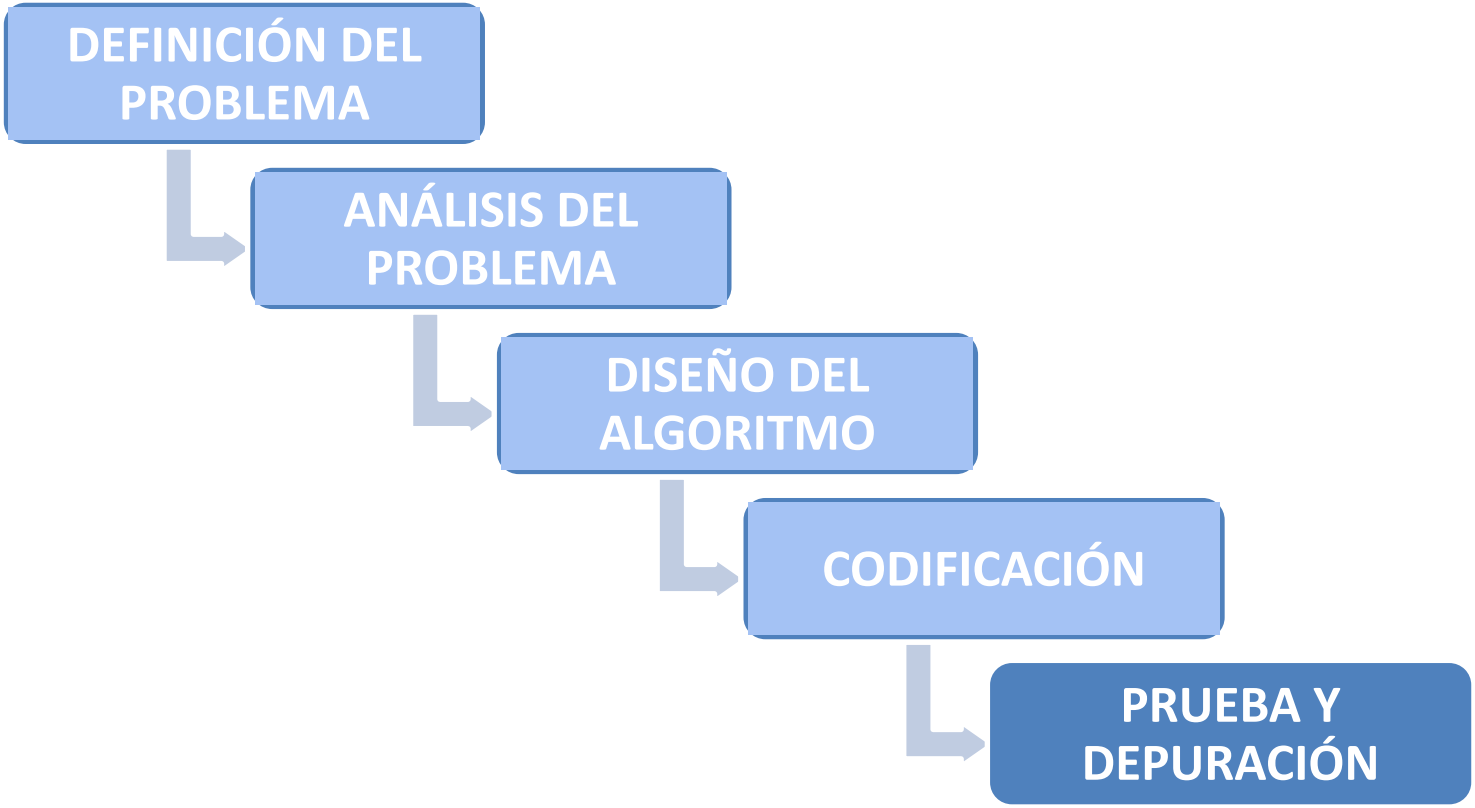
```
let valor : boolean = false;
let resultado : boolean = !valor; // true
```

Técnicas de Programación

Carrera Programador full-stack

Prueba de Escritorio (Conceptos)

Etapas del Ciclo de Vida



Etapas del Ciclo de Vida

- **Definición de Problema:** Determinar la información inicial para la elaboración del mismo
- **Análisis del Problema:** Datos de entrada, de salida, métodos y fórmulas
- **Diseño del Algoritmo:** Usar las herramientas de representación de algoritmos
- **Codificación:** Escribir la solución del problema, en instrucciones detalladas, en un lenguaje reconocible por la computadora (conocido como **Código Fuente**)
- **Prueba y Depuración**

Se toman escenarios posibles, válidos o inválidos y se corre la secuencia del algoritmo para ver si cumple con los resultados esperados

Pruebas de Escritorio

- Técnica utilizada para validar la resolución de problemas con algoritmos, de uso frecuente en el ámbito informático
- Sirve para validar utilizando datos reales como ejemplo, un algoritmo definido y así comprobar si se obtiene el resultado deseado
- Para poder realizar una prueba de escritorio, es necesario identificar cuáles son las variables de entrada, cuáles son las variables auxiliares y cuáles son las variables de salida
- El proceso consiste en seguir el algoritmo recorriendo sus líneas como lo haría la computadora
- A medida que se van recorriendo las líneas se anotan en una tabla auxiliar los valores que van tomando las variables

Cuántos Casos se Describen?

- La cantidad de casos necesarios depende de la complejidad del problema
- Es importante analizar los casos límites entre situaciones del algoritmo
- Ejemplo, recuerde el ejercicio de verificar si un número es mayor a 20. Se podría verificar con un número mayor a 20, un número igual a 20 y un número menor que 20

Estructura de Control – Selección Simple

Ejercicio - Mayor a 20 - Código

```
import * as rls from ‘readline-sync’;

let nroDeseado : number = rls.questionInt(”Escriba el número que desea verificar si es mayor o no a 20: “);

if (nroDeseado > 20) {
        console.log('El número es mayor a 20: ' +  nroDeseado);
} else {
        console.log('El número es menor o igual a 20: ' +  nroDeseado);
}
```

Estructura de Control – Selección Simple

Ejercicio - Mayor a 20 – Prueba de Escritorio

Código	Datos Entrada (nroDeseado)	Respuesta Deseada

`let nroDeseado : number = rls.questionInt("Escriba el número que desea verificar si es mayor o no a 20: ");`
 	
`if (nroDeseado > 20) {`
`  console.log('El número es mayor a 20: ' + nroDeseado);`
`} else {`
`  console.log('El número es menor o igual a 20: ' + nroDeseado);`
`}`
	20	El número es menor o igual a 20: 20

		El número es menor o igual a 20: 3
	200	El número es mayor a 20: 45

Estructura de Control – Selección Simple

Ejercicio – Aplicar Descuento

- Desarrolle un algoritmo que diga el precio de una compra
- La compra se compone del precio del producto y la cantidad
- Si el cliente gasta más de $1000 debemos aplicarle un descuento del 10%

Estructura de Control – Selección Simple

Ejercicio – Validar Altura

- Desarrolle un algoritmo que, de acuerdo a la altura de una persona, decida si puede entrar a un juego en un parque de diversiones
- Para poder subirse a la montaña rusa la persona debe medir 1.30 mts. o más

Estructura de Control – Selección Simple

Ejercicio - Login

- Desarrolle un algoritmo que permita loguearse (registrarse) a un sistema, ingresando un nombre de usuario y la contraseña adecuada.
- Considerar que tanto el usuario como la contraseña están formados sólo por letras.
- El sistema deberá validar que el usuario y la contraseña sean correctas, comparándolas con lo que es sistema tiene registrado para ese usuario. Tenga en cuenta que el sistema tiene registrado el usuario: Juan y la clave claveJuan

Recuerde plantear el Pseudocódigo y las Pruebas de Escritorio

Estructura de Control – Selección Simple

Ejercicio – Determinar Medalla

- Desarrolle un algoritmo que, dada una posición en una carrera se determine el tipo de medalla a entregar.
- Tenga en cuenta que para el primer puesto se entrega medalla de oro, segundo puesto medalla de plata y tercer puesto medalla de bronce. En caso que quede en otra posición se entrega certificado de participación

Recuerde plantear el Pseudocódigo y las Pruebas de Escritorio

Estructura de Control

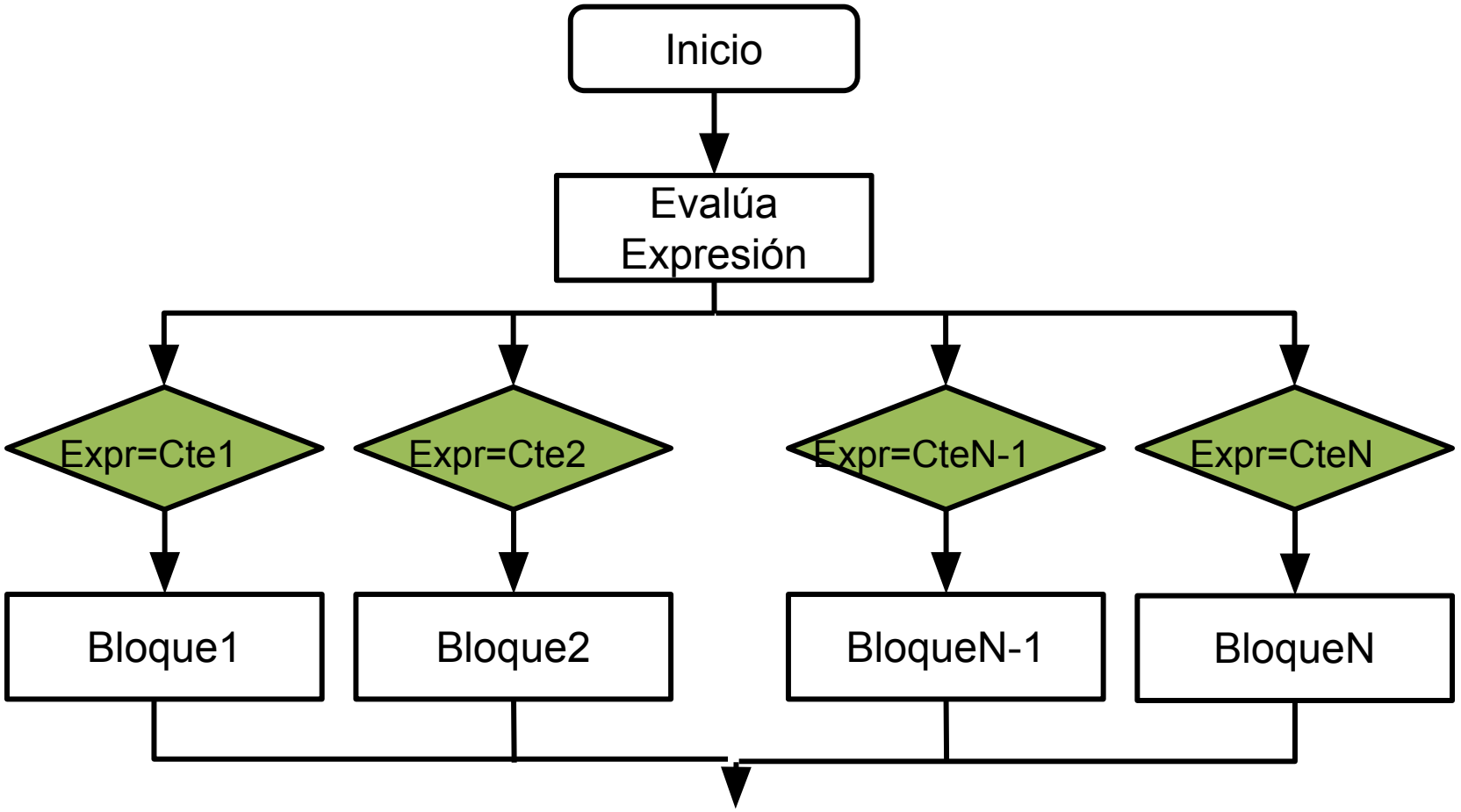
Selección Múltiple

```
import * as rls from 'readline-sync';

let posicionLlegada : number = rls.questionInt("Ingrese la posición de llegada del competidor: ");

if (posicionLlegada == 1) {
    console.log("Entregar medalla de oro");
} else {
    if (posicionLlegada == 2) {
        console.log("Entregar medalla de plata");
    } else {
        if (posicionLlegada == 3) {
            console.log("Entregar medalla de bronce");
        } else {
            console.log("Entregar mención de participación");
        }
    }
}
```

Por qué tantos anidamientos??

Estructura de Control

Selección Múltiple



Estructura de Control

Selección Múltiple

En el **switch** la secuencia de instrucciones a ejecutar depende de una variable numérica:

```
switch <variable>{

    case <número1>:
            <instrucciones>;
            break;

     case <número2>:
            <instrucciones>;
            break;
           <...>

     case:
      default: <instrucciones>
}
```

Al ejecutarse, se evalúa el contenido de la variable y se ejecuta la

Estructura de Control - Selección Múltiple

Determinar Medalla - Código

```
import * as rls from 'readline-sync';

let posicionLlegada : number = rls.questionInt("Ingrese la posición de llegada del competidor: ");

switch (posicionLlegada) {
        case 1:
                console.log("Entregar medalla de oro");
                break;
        case 2:
                console.log("Entregar medalla de plata");
                break;
        case 3:
                console.log("Entregar medalla de bronce");
                break;
        default:
                console.log("Entregar mención de participación");
}
```

Estructura de Control - Selección Múltiple

Determinar Medalla – Prueba de Escritorio

Código	Datos de Entrada	Salida deseada

`let posicionLlegada : number = rls.questionInt("Ingrese la posición de llegada del competidor: ");`
`switch (posicionLlegada) {`
`  case 1:`
`  console.log("Entregar medalla de oro");`
`  break;`
`  case 2:`
`  console.log("Entregar medalla de plata");`
`  break;`
`  case 3:`
`  console.log("Entregar medalla de bronce");`
`  break;`
`  default:`
`  console.log("Entregar mención de participación");`
`}`
	posicionDeLlegada = 1	Entregar medalla de oro

	posicionDeLlegada = 2	Entregar medalla de plata
	posicionDeLlegada = 3	Entregar medalla de bronce
	posicionDeLlegada = 6	Entregar mención de participación

Estructura de Control - Selección

Determine qué Hace el Siguiente Código

```
import * as rls from 'readline-sync';

let no1 : number = rls.questionInt("Ingrese el primer número: ");
let no2 : number = rls.questionInt("Ingrese el segundo número: ");
let no3 : number = rls.questionInt("Ingrese el tercer número: ");
let result : number = 0;
if (no1 < 0) {
        result = no1*no2*no3;
} else {
        result = no1+no2+no3;
}
console.log(result);
```

Estructura de Control - Selección

Determine qué Hace el Siguiente Código

Dados tres números, ingresados por el usuario, el algoritmo se fija si el primer número es negativo en cuyo caso muestra el producto de los tres números, sino muestra la suma de estos.

```
import * as rls from 'readline-sync';


let no1 : number = rls.questionInt("Ingrese el primer número: ");
let no2 : number = rls.questionInt("Ingrese el segundo número: ");
let no3 : number = rls.questionInt("Ingrese el tercer número: ");
let result : number = 0;
if (no1 < 0) {
        result = no1*no2*no3;
} else {
        result = no1+no2+no3;
}
console.log(result);
```

Estructura de Control - Selección

Determine qué Hace el Siguiente Código

```
import * as rls from 'readline-sync';


let e : number = rls.questionInt("Introduce: ");
if (e >= 18) {
        console.log("Es " + e);
} else {
        console.log("No es " + e);
}
```

Estructura de Control - Selección

Determine qué Hace el Siguiente Código

- Dada la edad de una persona informa si es mayor de 18 o no

```
import * as rls from 'readline-sync';

let edad : number = rls.questionInt("Introduce: ");
if (edad >= 18) {
        console.log("Es mayor de edad");
} else {
        console.log("No es mayor de edad");
}
```

Técnicas de Programación

Carrera Programador full-stack

Selección (Ejercicios)

Estructura de Control - Selección

Ejercicio – Mayor de Tres

- Desarrolle un algoritmo que dados tres números determine cuál es el mayor de los tres

Recuerde plantear el Pseudocódigo y las Pruebas de Escritorio

Estructura de Control - Selección

Ejercicio – Par/Impar

- Desarrollar un algoritmo que dado un número, ingresado por el usuario determine si el número es par o impar y le informe al usuario
- En el caso de ser 0 (cero) el algoritmo deberá informarlo

Recuerde plantear el Pseudocódigo y las Pruebas de Escritorio

Estructura de Control - Selección

Ejercicio – Descuento Octubre

- Una tienda al cumplir años en Octubre ofrece un descuento del 15% a sus clientes en todas sus compras
- Desarrolle un algoritmo que dada una compra: precio unitario, cantidad y el mes, indicados por el usuario, determine si el cliente tiene descuento o no

Recuerde plantear el Pseudocódigo y las Pruebas de Escritorio

Estructura de Control - Selección

Ejercicio – Aumento de Sueldo

- Una empresa desea premiar a sus empleados con un aumento de sueldo. Este aumento se ajusta a la siguiente tabla:

Sueldo Actual	Sueldo con Aumento
0 - 15.000 $	20%
15.001 - 20.000 $	10%
20.001 - 25.000 $	5%
Más de 25.000 $	No hay aumento

- Desarrolle un algoritmo dado el salario actual de un empleado determine el aumento de sueldo a aplicar y se lo muestre

Recuerde plantear el Pseudocódigo y las Pruebas de Escritorio