

Técnicas de Programación

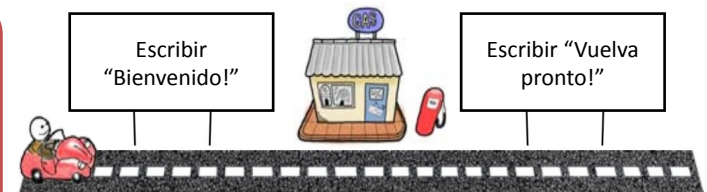
Programador full-stack

Selección (Repaso)

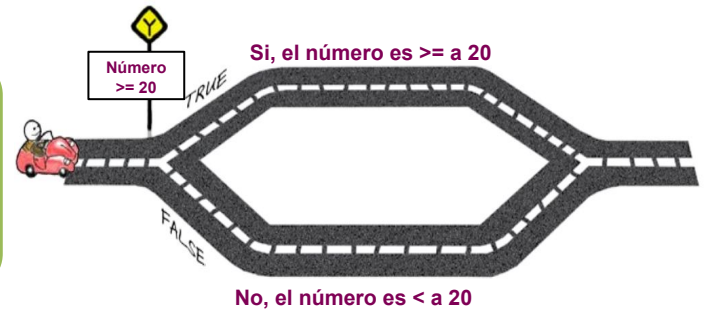
Estructura de Control

Selección

Secuenciales



Selección o de Decisión



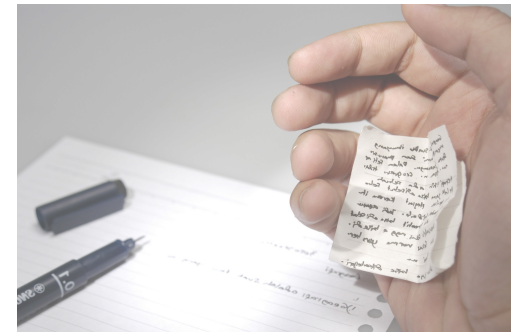
Repetitivas

Estructura de Control

Selección Simple y Múltiple

Selección Simple

```
if (condición) {  
    <instrucciones>  
} else {  
    <instrucciones>  
}
```



Alternativa Múltiple

```
switch <variable>{  
    case <número1>:  
        <instrucciones>;  
        break;  
    case <número2>:  
        <...>  
    default: <instrucciones>  
}
```

Prueba 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
- 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

Prueba de Escritorio

Ejercicio - Mayor a 20

Código	Datos Entrada	Respuesta Deseada
<pre>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); }</pre>	nroDeseado = 20	El número es menor o igual a 20: 20
	nroDeseado = 3	El número es menor o igual a 20: 3
	nroDeseado = 45	El número es mayor a 20: 45

Técnicas de Programación

Programador full-stack

Selección (Ejercicios)

Estructura de Control – Selección

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

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 las Pruebas de Escritorio



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

Recuerde plantear las Pruebas de Escritorio

Técnicas de Programación

Programador full-stack

Selección (Resolución)

Estructura de Control – Selección

Ejercicio – Validar Altura - Código

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

let alturaPermitida : number = 1.3;
let alturaPersona : number = rls.questionFloat("Indique la altura de la persona: ");
if (alturaPersona <= alturaPermitida) {
    console.log("La persona no puede subir al juego");
} else {
    console.log("La persona puede subir al juego");
}
```



Estructura de Control – Selección

Ejercicio – Login - Código

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

```
let usuario : string = "Juan";
```

```
let clave : string = "claveJuan";
```

```
let userIngresado : string = rls.question("Por favor ingrese el usuario: ");
```

```
let claveIngresada : string = rls.question("Por favor ingrese la clave: ",  
    { hideEchoBack : true } ); /* ver https://www.npmjs.com/search?q=readline-sync */
```

```
if (usuario == userIngresado && clave == claveIngresada) {
```

```
    console.log("Bienvenido");
```

```
} else {
```

```
    console.log("El usuario o la clave son incorrectos");
```

```
}
```



Estructura de Control – Selección

Ejercicio - Mayor de Tres – Código

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

let numero1 : number = rls.questionInt("Ingrese el primer número: ");
let numero2 : number = rls.questionInt("Ingrese el segundo número: ");
let numero3 : number = rls.questionInt("Ingrese el tercer número: ");
if (numero1 > numero2 && numero1 > numero3) {
    console.log("El número " + numero1 + " es el mayor");
} else if (numero2 > numero1 && numero2 > numero3) {
    console.log("El número " + numero2 + " es el mayor");
} else if (numero3 > numero1 && numero3 > numero2) {
    console.log("El número " + numero3 + " es el mayor");
}
```



Estructura de Control – Selección

Ejercicio – Par/Impar - Código

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

let num : number = rls.questionInt("Ingrese un número: ");

if (num == 0) {
    console.log("El " + num + " no es par ni impar");
} else if ( num % 2 == 0) {
    console.log("El " + num + " es par");
} else {
    console.log("El " + num + " es impar");
}
```



Estructura de Control – Selección

Ejercicio – Descuento Octubre - Código

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

let monto : number = rls.questionInt("Ingresar monto: ");
let montoConDescuento : number = 0;
let descuento : number = 0;
let cantidad : number = rls.questionInt("Ingresar cantidad: ");
let precioTotal : number = monto * cantidad;
let mes : string = rls.question("Ingresar el nombre del mes en minúsculas : ");
if (mes == "octubre") {
    descuento = (precioTotal*15)/100
    montoConDescuento = precioTotal - descuento;
    console.log("Por comprar en el mes de Octubre Ud tiene un 15% de descuento");
    console.log("El monto a pagar es: " + montoConDescuento);
} else {
    console.log("Ud. no tiene descuento, el monto a pagar es: " + precioTotal);
}
```

15%
DE DESCUENTO
EN TODAS TUS COMPRAS

Estructura de Control – Selección

Ejercicio – Aumento de Sueldo - Código

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

let sueldoActual : number = rls.questionInt("Ingrese el salario actual del empleado: ");
let nuevoSueldo : number = 0;
if (sueldoActual <= 15000 ) {
    nuevoSueldo = sueldoActual*1.2;
    console.log("Ud. tiene un aumento de 20%, su nuevo sueldo es: ",nuevoSueldo);
} else if (sueldoActual <= 20000 ) {
    nuevoSueldo = sueldoActual*1.1;
    console.log("Ud. tiene un aumento de 10%, su nuevo sueldo es: ",nuevoSueldo);
} else if (sueldoActual <= 25000 ) {
    nuevoSueldo = sueldoActual * 1.05;
    console.log("Ud. tiene un aumento de 5%, su nuevo sueldo es: ",nuevoSueldo);
} else {
    nuevoSueldo = sueldoActual;
    console.log("Ud. no tiene aumento");
}
```

Técnicas de Programación

Programador full-stack

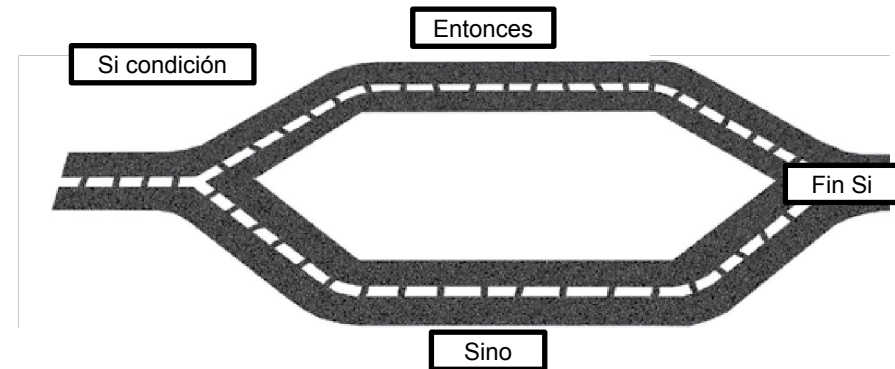
Repetición (Conceptos)

Estructuras de Control

Secuenciales

Selectivas o De
Decisión

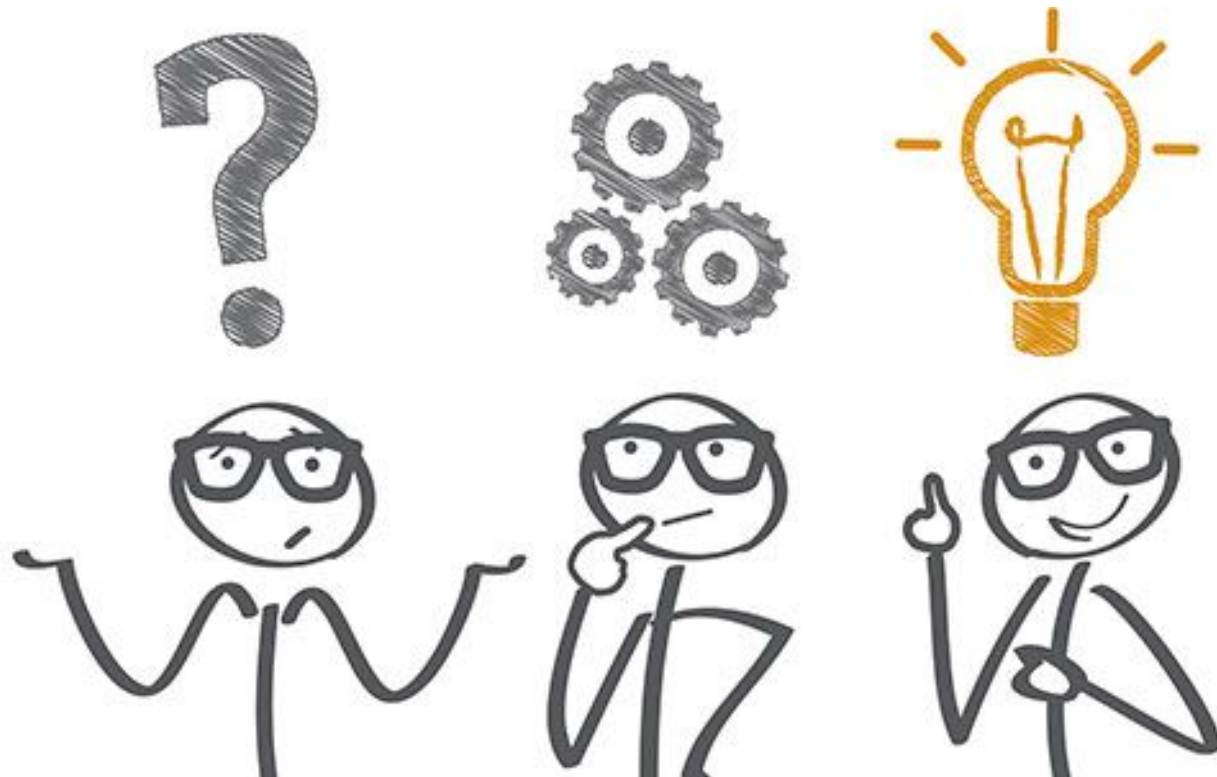
Repetitivas



Estructuras de Control

Decir 5 Veces “Hola”

- Escriba un programa que salude 5 veces por pantalla de forma secuencial



Estructuras de Control

Decir 5 Veces "Hola"

```
console.log("Hola");  
console.log("Hola");  
console.log("Hola");  
console.log("Hola");  
console.log("Hola");
```

¿Es práctico repetir
las instrucciones
explícitamente?

Estructuras de Control

Calcular el Promedio de 10 Notas

- Escriba un programa que solicite 10 números al usuario y calcule el promedio de las mismas. Luego, muestre el resultado por pantalla.

EVALUACIÓN ASIGNATURA	EVALUACIÓN				CALIFICACIÓN FINAL
	SEPT.	OCT.	NOV.	DIC.	
GEOM. ANALÍTICA Y FUN.	10	10	10	10	10
TALLER DE LECT. Y RED. III	9	10	10	10	10
HISTORIA DE MÉXICO I	10	10	10	10	10
ORI. VOCACIONAL	10	10	10	10	10
FÍSICA I	10	10	10	10	10
INFORMÁTICA	10	10	10	10	10
INGLÉS III	9	10	10	10	10
CONTABILIDAD BÁSICA	10	10	10	10	10
ADMINISTRACIÓN I	10	10	10	10	10
QUÍMICA	10	10	10	10	10
EDUCACIÓN FÍSICA	10	10	10	10	10
FORMACIÓN EN VALORES	10	10	10	10	10
EDUCACIÓN EN LA FE	9	9	9	9	9
HÁBITOS PERSONALES	10	10	10	10	10
INASISTENCIAS HORA/CLASE					
FIRMA DEL PADRE O TUTOR	OCT. - NOV.				
SEPT.				DIC.	

$$\bar{x} = \frac{\sum x}{n}$$

Estructuras de Control

Calcular el Promedio de 10 Notas

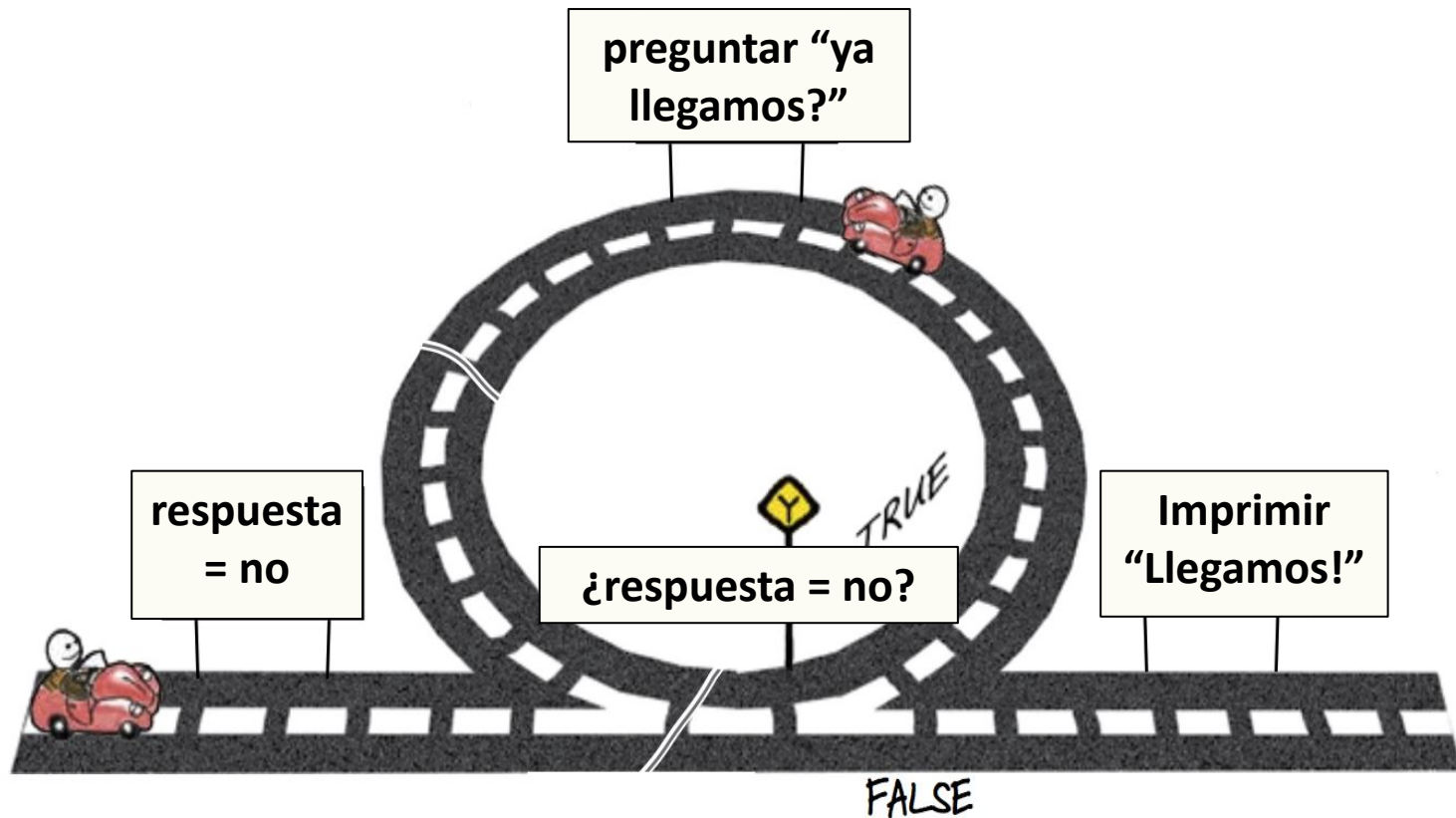
```
import * as rls from 'readline-sync';
```

```
let nota1 : number = rls.questionInt("Ingrese la nota 1: ");  
let nota2 : number = rls.questionInt("Ingrese la nota 2: ");  
let nota3 : number = rls.questionInt("Ingrese la nota 3: ");  
let nota4 : number = rls.questionInt("Ingrese la nota 4: ");  
let nota5 : number = rls.questionInt("Ingrese la nota 5: ");  
let nota6 : number = rls.questionInt("Ingrese la nota 6: ");  
let nota7 : number = rls.questionInt("Ingrese la nota 7: ");  
let nota8 : number = rls.questionInt("Ingrese la nota 8: ");  
let nota9 : number = rls.questionInt("Ingrese la nota 9: ");  
let nota10 : number = rls.questionInt("Ingrese la nota 10: ");  
let total : number = nota1+nota2+nota3+nota4+nota5+nota6+nota7+nota8+nota9+nota10;  
let promedio : number = total/10;  
console.log("El promedio de las notas es: " + promedio);
```

¿Puede
hacerse más
corto este
código?
Es bastante
tedioso...

Estructuras de Control

Repetición



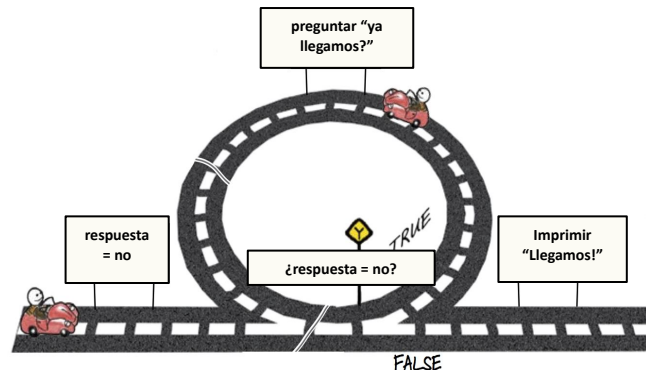
Extraído de: "Barry, P., & Griffiths, D. (2009). Head First Programming: A Learner's Guide to Programming. " O'Reilly Media, Inc.."

Comparativa code.org



Recuerden lo que hacíamos en code.org al poner un bloque **mientras**. Lo que está dentro del bloque es lo que va entre llaves.

```
while (hayCaminoAdelante) {  
    console.log("Avanzar");  
}
```



Estructuras de Control

*Instrucción **While***

- La instrucción **while** ejecuta una secuencia de instrucciones mientras una condición sea verdadera
 - También llamados iteraciones o “loops” en Inglés
 - Sirven para ejecutar código varias veces
 - La condición se verifica al principio
 - La cantidad de veces ejecutado depende de una condición (puede que no se ejecute ninguna vez)



Estructuras de Control

*Instrucción **While***

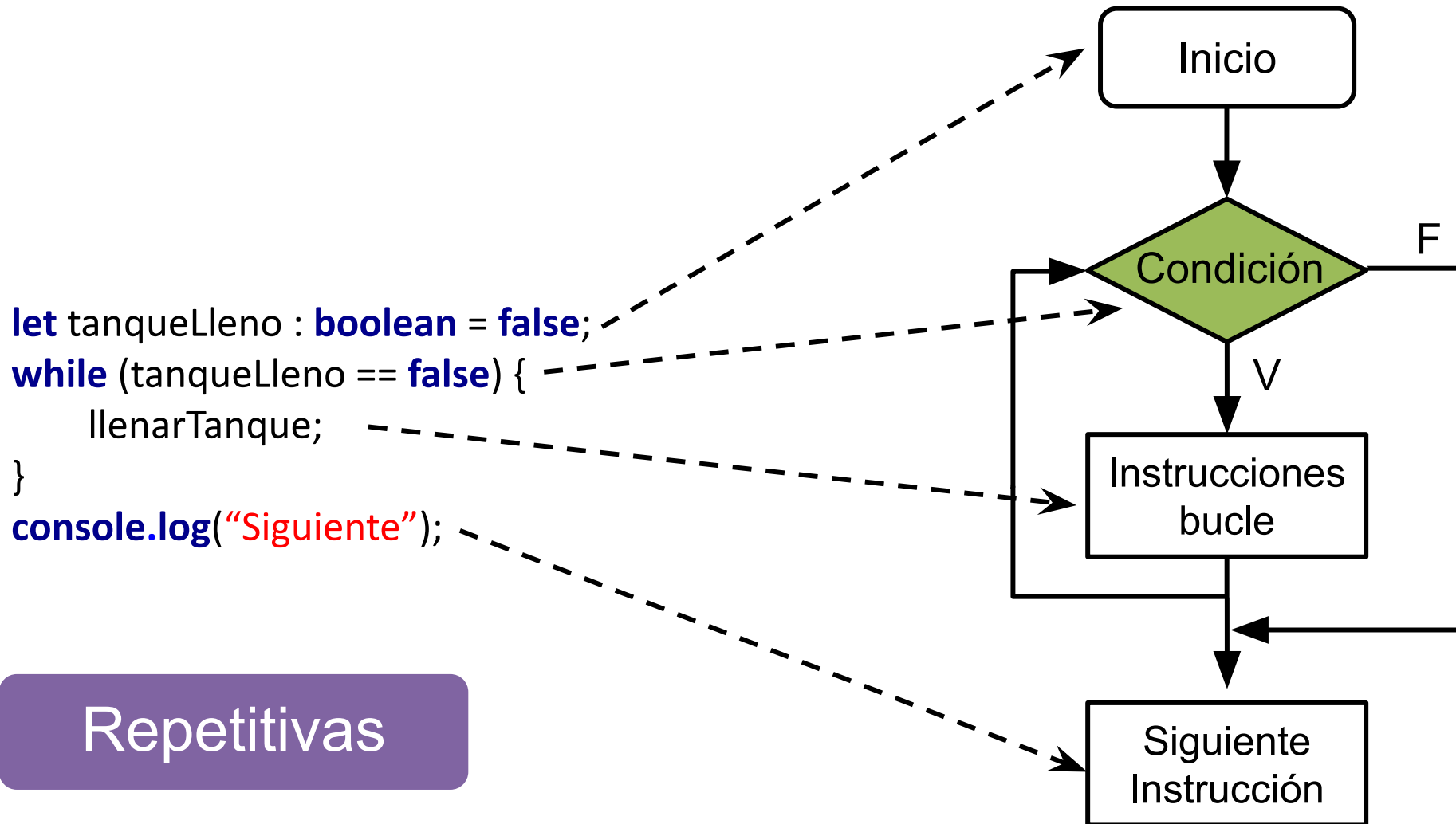
- La instrucción **while** ejecuta una secuencia de instrucciones mientras una condición sea verdadera



```
while (<condición>) {  
    <instrucciones>  
}
```

Estructuras de Control

Instrucción Mientras



Estructuras de Control

Decir 5 Veces "Hola"

```
let cantHolas : number = 1;  
while (cantHolas <= 5) {  
    console.log("Hola");  
    cantHolas = cantHolas + 1;  
}
```

Usamos la variable
cantHolas como
contador

Se repite hasta que
cantHolas sea
mayor que 5

En cada iteración se
incrementa en 1

Estructuras de Control

Calcular el Promedio de 10 Notas

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

let nota, suma, promedio, contador : number;
contador = 1;
suma = 0;
while (contador <= 10) {
    nota = rls.questionInt("Ingrese una nota: ");
    suma = suma + nota;
    contador = contador + 1;
}
promedio = suma / 10;
console.log("El promedio de las notas es: " + promedio);
```

Usamos la variable *suma* como acumulador de las notas

La variable *suma* se inicializa en cero

En cada iteración se suma la nota ingresada por el usuario en la variable

Estructuras de Control

Simular la Espera de un Colectivo

- Cuando llegamos a la parada, miramos si el colectivo arribó a la parada
- Siempre tenemos que esperar antes de que llegue



Estructuras de Control

Simular la Espera de un Colectivo



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

let llegadaColectivo : string;
console.log("Esperando el colectivo");
llegadaColectivo = rls.question("Llegó el colectivo? (S/N): ");
while (llegadaColectivo=="N") {
    console.log("Esperando el colectivo");
    llegadaColectivo = rls.question("Llegó el colectivo? (S/N): ");
}
console.log("Llegó el colectivo");
```

Estructuras de Control

Instrucción For


- La instrucción **for** ejecuta una secuencia de instrucciones utilizando contadores con principio, incrementos y final
- Son útiles cuando hay que hacer un conteo (fijo)
- El valor inicial del conteo, el valor final de corte y los se definen en una sola instrucción
- La declaración de la variable debe realizarse antes



Estructuras de Control

Instrucción For

- La instrucción **for** ejecuta una secuencia de instrucciones utilizando contadores con principio, incrementos y final




```
for (num=inicial; <condición> ; incremento) {  
    <instrucciones>  
}
```

Estructuras de Control

Instrucción For

- **num**: la variable que se va a usar de contador
- **inicial**: el valor de *num* desde el cual se comenzará a iterar
- **condición**: La sentencia que define si se debe seguir iterando o si ya se llegó al final del recorrido.
- **incremento**: la modificación que se debe realizar a *num* en cada iteración



```
for (num=inicial; <condición> ; incremento) {  
    <instrucciones>  
}
```

Comparativa code.org



Recuerden lo que hacemos en code.org al poner un bloque **contar**. Lo que está dentro del bloque es lo que va entre llaves.

```
for (let contador : number = 50; contador <= 150; contador+=50 ) {  
    console.log("Dibujar una casa. Longitud:",contador);  
}
```



Estructuras de Control

Instrucción For y While Intercambiables

```
for (let rueda : number = 1; rueda <= 4; rueda+=1) {  
    console.log("Inflar Rueda: ", rueda);  
}
```

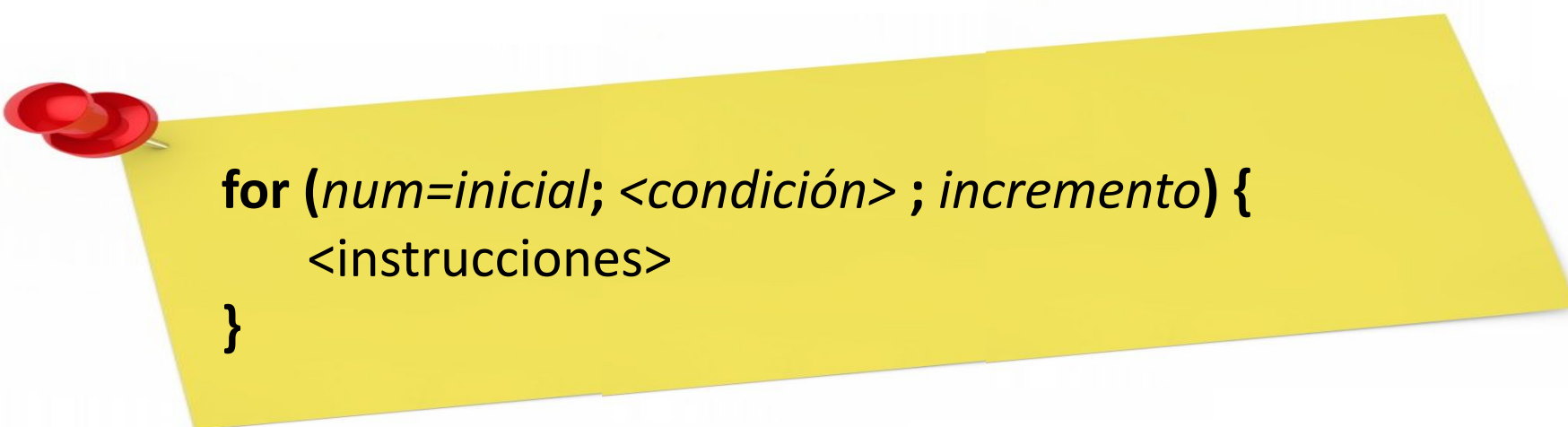
```
let rueda : number = 1;  
while (rueda <= 4) {  
    console.log("Inflar Rueda: ", rueda);  
    rueda+=1; //rueda = rueda + 1 // rueda ++  
}
```

Repetitivas

Estructuras de Control

Calcular el Promedio de 10 Notas

Retomar el ejercicio del promedio de 10 notas y resolverlo utilizando la estructura de control **for**



```
for (num=inicial; <condición> ; incremento) {  
    <instrucciones>  
}
```


Estructuras de Control

Calcular el Promedio de 10 Notas

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

let nota, suma, promedio, contador : number;

suma=0;

for (contador=1; contador<11; contador++) {
    nota = rls.questionInt("Ingrese la nota " + contador + ": ");
    suma +=nota;
}

promedio = suma/10;

console.log("El promedio de las notas es: " + promedio);
```

Si necesitamos repetir un conjunto de instrucciones por un número predeterminado de veces, la instrucción **for** es muy útil

Nos evitamos inicializar e incrementar el contador, ya que es parte del Para

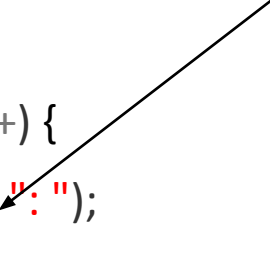
Estructuras de Control

Calcular el Promedio de 10 Notas

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

let nota, suma, promedio, contador : number;
suma=0;
for (let contador : number =1; contador<11; contador++) {
    nota = rls.questionInt("Ingrese la nota " + contador + ": ");
    suma +=nota;
}
promedio = suma/10;
console.log("El promedio de las notas es: " + promedio);
```

Tener en cuenta
que el contador va
desde 1 (inclusive)
hasta 10 (inclusive)



Técnicas de Programación

Programador full-stack

Repetición (Ejercicios)

Estructuras de Control

Eureka

- Escribir un algoritmo que nos pida una clave y verifique que sea la correcta
- Tenga en cuenta que la clave es la palabra “eureka”
- Solo tenemos 3 intentos para acertar, si fallamos los 3 intentos el sistema mostrará un mensaje indicándonos que hemos agotado todas las oportunidades
- Si acertamos la clave, saldremos directamente del programa



Estructuras de Control

Múltiplos

- Cree un algoritmo que liste los números múltiplos de 2 o de 3 que hay entre 1 y 100
- Tener en cuenta que hay números que son múltiplos de 2 y de 3 al mismo tiempo
- En dichos casos, solamente indique el número una vez



Estructuras de Control

Par/Impar

- Realizar un algoritmo que dado un número entero ingresado por el usuario, visualice en pantalla si es par o impar
- En el caso de ingresar un cero, se debe volver a pedir el número por teclado (hasta que se ingrese un número mayor que cero)



Estructuras de Control

Suma entre Números

- Escriba un programa que pida al usuario dos números enteros, y luego retorne la suma de todos los números que están entre ellos
- Por ejemplo, si los números son 2 y 7, debe entregar como resultado $2 + 3 + 4 + 5 + 6 + 7 = 27$

Ingrese num: 2
Ingrese num: 7
La suma es 27



Estructuras de Control

Tablas de Multiplicación

- Diseñar un algoritmo que muestre por pantalla la tabla de multiplicación del número ingresado por el usuario
- Para definir hasta qué número desea que muestre la tabla de multiplicación, el usuario también deberá ingresar un valor

Ingrese el número: 9
 Ingrese hasta qué número: 4
 $9 \times 1 = 9$
 $9 \times 2 = 18$
 $9 \times 3 = 27$
 $9 \times 4 = 36$

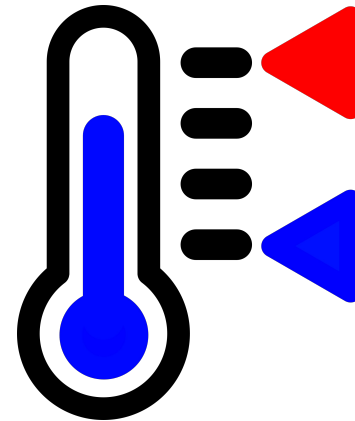
X	1	2	3	4	5	6	7	8	9	10	11	12
1	1	2	3	4	5	6	7	8	9	10	11	12
2	2	4	6	8	10	12	14	16	18	20	22	24
3	3	6	9	12	15	18	21	24	27	30	33	36
4	4	8	12	16	20	24	28	32	36	40	44	48
5	5	10	15	20	25	30	35	40	45	50	55	60
6	6	12	18	24	30	36	42	48	54	60	66	72
7	7	14	21	28	35	42	49	56	63	70	77	84
8	8	16	24	32	40	48	56	64	72	80	88	96
9	9	18	27	36	45	54	63	72	81	90	99	108
10	10	20	30	40	50	60	70	80	90	100	110	120
11	11	22	33	44	55	66	77	88	99	110	121	132
12	12	24	36	48	60	72	84	96	108	120	132	144

Estructuras de Control

Encontrar el Número Máximo

- Leer valores hasta que se introduzca un cero (0)
- El usuario puede introducir valores positivos y negativos
- Encontrar el máximo de los elementos que se introdujeron
- Analizar cómo cambia el programa para hallar el mínimo

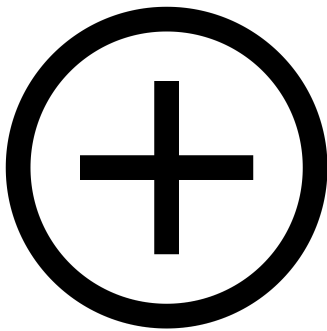
Ingrese número: 9
Ingrese número: 7
Ingrese número: -1
Ingrese número: 1
Ingrese número: 0
El máximo es 9



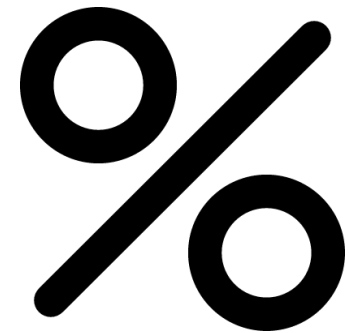
Estructuras de Control

Cantidad y Distribución de Positivos

- Leer valores del usuario hasta que introduzca un 0
- El usuario puede introducir valores numéricos, tanto positivos como negativos
- Contar la cantidad de valores introducidos que sean mayores a 0 y el porcentaje de positivos respecto del total



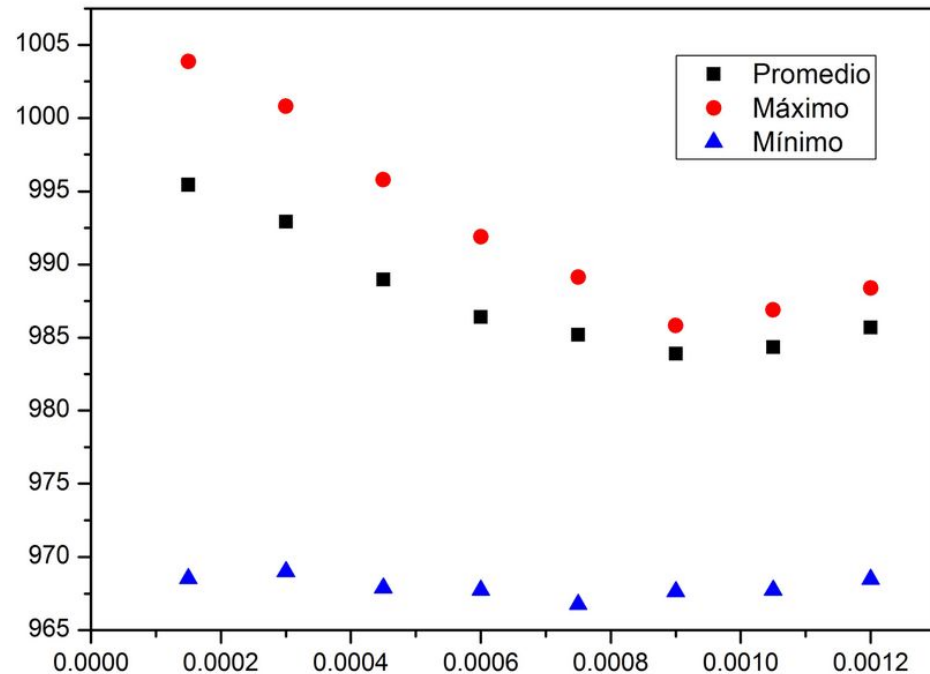
Ingrese número: 9
Ingrese número: 7
Ingrese número: -1
Ingrese número: 1
Ingrese número: 0
3 positivos, 75% del total



Estructuras de Control

Promedio-Máximo-Mínimo

- Diseñar un algoritmo que lea números enteros hasta teclear 0
- Determinar y mostrar el máximo, el mínimo y la media (promedio) de todos los números ingresados
- Pensar cuidadosamente cómo debemos inicializar las variables



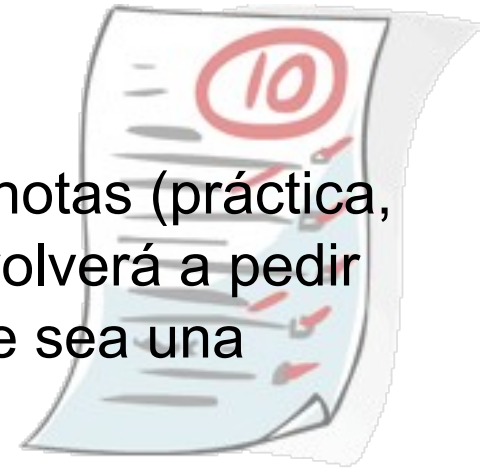
Estructuras de Control

Calificaciones

Calcular las calificaciones de un grupo de alumnos, donde la nota final de cada alumno se calcula según el siguiente criterio:

- la parte práctica vale el 10%
- la parte de ejercicios vale el 50%
- la parte teórica el 40%

Se debe ingresar el nombre del alumno y sus tres notas (práctica, ejercicios y teórica), se escribirá el resultado y se volverá a pedir los datos del siguiente alumno hasta que el nombre sea una cadena vacía



Las notas deben estar entre 0 y 10 (si no lo están, no imprimirá las notas, mostrará un mensaje de error y continuará con otro alumno)

Estructuras de Control

Dados

- Al tirar un dado tenemos $1/6$ de probabilidades de sacar 6
- Si tiramos dos dados tenemos $1/36$ probabilidades de sacar doble 6
- Al aumentar el número de dados la probabilidad de sacar todos 6 es cada vez menor
- Escriba un programa que calcule la probabilidad de sacar todos los dados 6 siendo que tiramos N dados (N es un dato ingresado por el usuario)

