

FUNDAMENTOS DE PROGRAMACIÓN

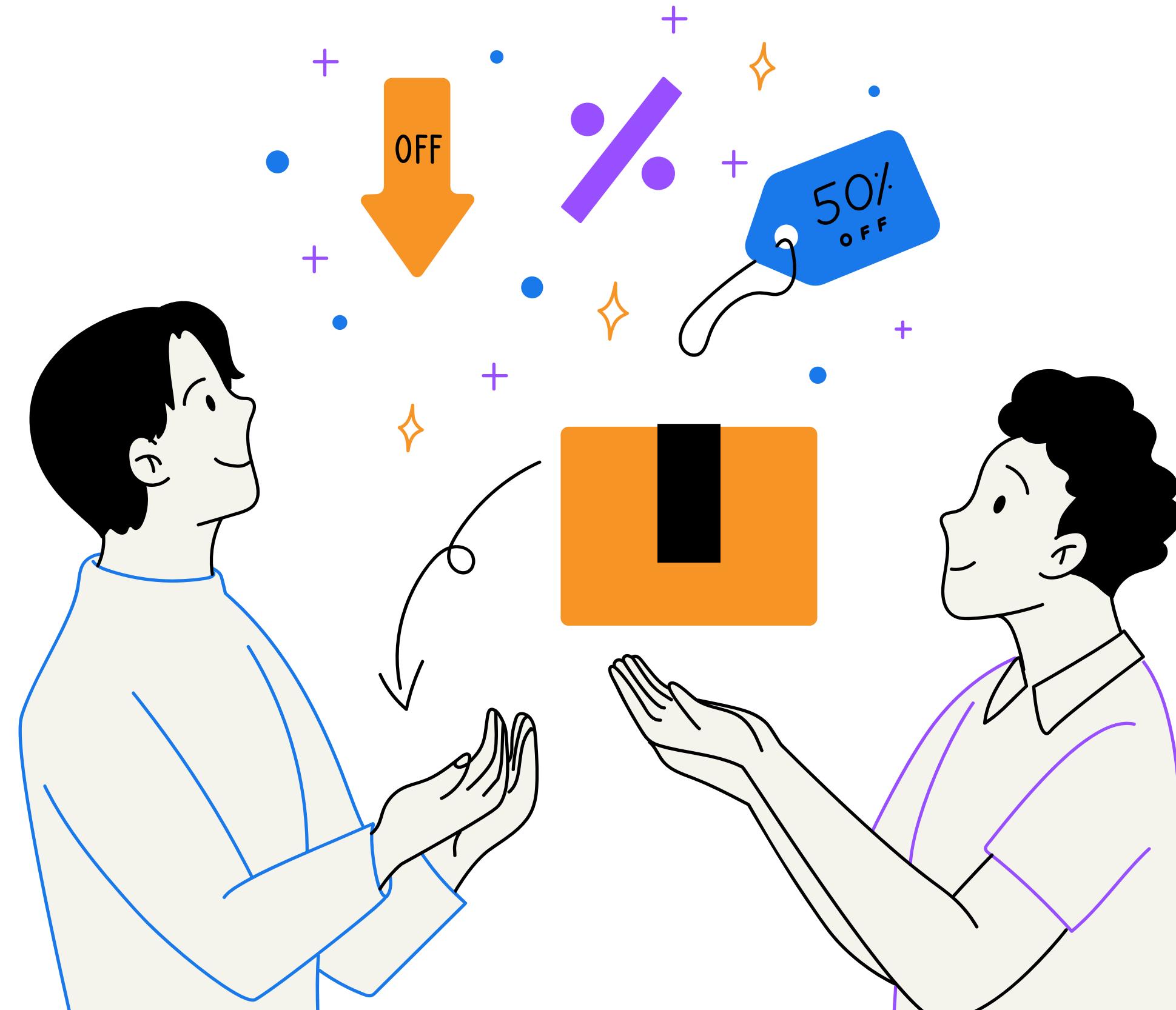


FUNDAMENTOS DE PROGRAMACIÓN



¿QUÉ VAMOS A VER?

- Funciones.
- Recursividad.





FUNCIONES

- Hablamos de un conjunto de líneas de código que realiza una tarea en específico y nos puede retornar un valor.
- La podremos invocar o llamar desde cualquier parte de nuestro programa tantas veces como nos sea necesaria.
- También tendremos la opción de poder pasarle parámetros que van a influir directamente en la respuesta que nos puede entregar.
- En esencia, una función es creada para simplificar el uso del código y, sobre todo, evitar reescribirlo innecesariamente.





SINTAXIS DE UNA FUNCIÓN

- **variable_de_retorno:** Es el nombre de la variable que almacenará el dato que nos va a devolver la función. En caso de que tengamos una función no retorne nada, podemos borrar este componente.
- **Nombre:** Es el nombre que va a recibir nuestra función, con el que mismo que la vamos a invocar dentro de nuestro código principal.
- **Argumentos:** Son variables que requiere la función y utilizamos coma para separarlas. También existe la posibilidad de dejar este espacio en blanco en caso de que no tengamos argumentos.

```
1 Funcion variable_de_retorno <- Nombre ( Argumentos )  
2 // Bloque de código de la función.  
3 FinFuncion  
4  
5 Proceso ClaseFunciones  
6 // Bloque de código principal.  
7 FinProceso
```

```
1 Funcion Nombre()  
2 // Bloque de código de la función.  
3 FinFuncion  
4  
5 Proceso ClaseFunciones  
6 // Bloque de código principal.  
7 FinProceso
```



EJEMPLO

- Crearemos una función que permita obtener el promedio de cinco calificaciones.

```
1 Funcion promedio ← calcularPromedio
2
3 Definir notas, i Como Entero;
4 Definir promedio Como Real;
5 Dimension notas[5];
6 promedio = 0;
7
8 Para i = 0 Hasta 4 Con Paso 1 Hacer
9   Escribir "Ingrese la nota ", i + 1 ,": ";
10  Leer notas[i];
11  promedio = promedio + notas[i];
12 FinPara
13
14 promedio = promedio / 5;
15
16 FinFuncion
17
18 Proceso ClaseFunciones
19
20 Definir promFinal Como Real;
21
22 promFinal ← calcularPromedio;
23
24 Escribir "El promedio obtenido es: ", promFinal;
25
26 FinProceso
```



EJEMPLO

- Crearemos una función que permita obtener el promedio de cinco calificaciones.

```
1  Funcion promedio ← calcularPromedio(notas)
2
3      Definir promedio Como Real;
4      Definir i Como Real;
5      promedio = 0;
6
7      Para i = 0 Hasta 4 Con Paso 1 Hacer
8          promedio = promedio + notas[i];
9      FinPara
10
11     promedio = promedio / 5;
12
13 FinFuncion
14
15 Proceso ClaseFunciones
16
17     Definir notas, i Como Entero;
18     Definir promedio Como Real;
19     Dimension notas[5];
20
21     Para i = 0 Hasta 4 Con Paso 1 Hacer
22         Escribir "Ingrese la nota ", i + 1, ": ";
23         Leer notas[i];
24     FinPara
25
26     promedio ← calcularPromedio(notas);
27     Escribir "El promedio obtenido es: ", promedio;
28
29 FinProceso
```



EJEMPLO

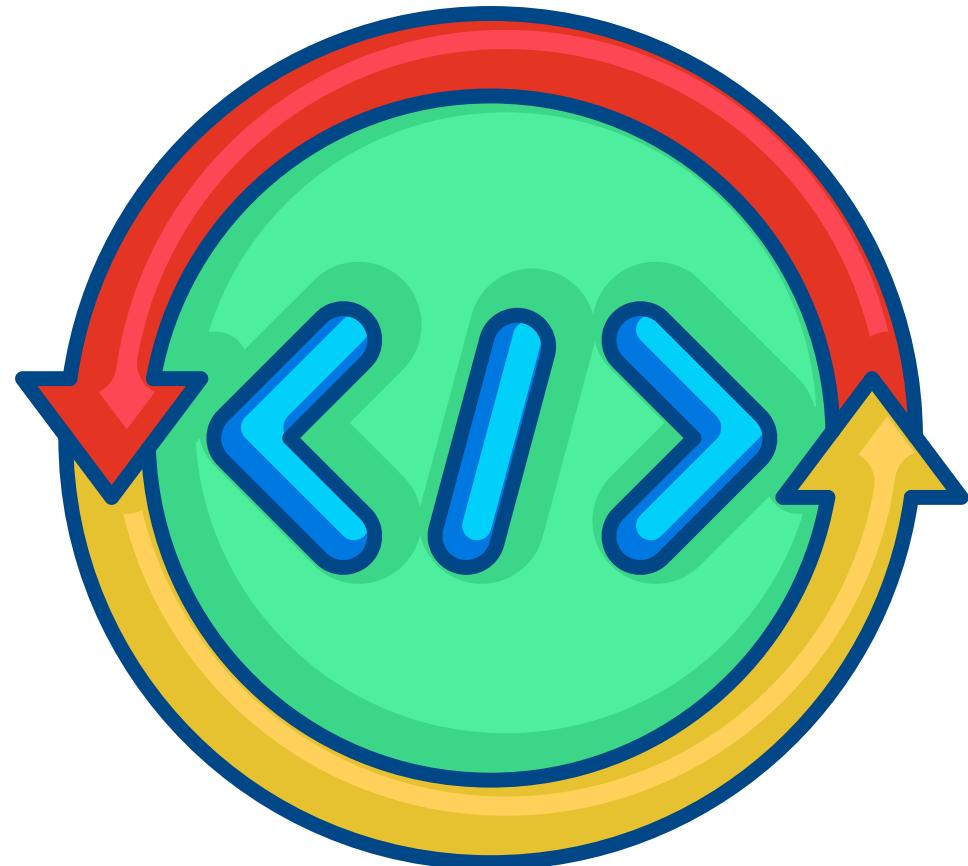
- Desarrolle un algoritmo en pseudocódigo que retorne todos los números primos entre 1 y el 30.

```
1  Funcion resultado ← primo(num)
2
3      Definir divisores, contador Como Entero;
4      Definir resultado Como Logico;
5      divisores = 0;
6
7      Para contador = 1 Hasta num Hacer
8          Si num % contador = 0 Entonces
9              divisores = divisores + 1;
10         FinSi
11
12         Si divisores ≤ 2 Entonces
13             resultado = verdadero;
14         SiNo
15             resultado = falso;
16         FinSi
17     FinPara
18
19 FinFuncion
20
21 Proceso ClaseFunciones
22
23     Definir n Como Entero;
24     Para n = 1 Hasta 30 Hacer
25         Si primo(n) Entonces
26             Escribir n;
27         FinSi
28     FinPara
29
30 FinProceso
```



RECUSIVIDAD

- Las funciones recursivas son aquellas que se llaman a sí mismas durante su propia ejecución.
- Funciona de forma similar a los ciclos, pero es necesario saber en qué momento debemos dejar de llamarla, o tendremos una función recursiva infinita.
- Este tipo de funciones se utilizan para dividir una tarea en sub-tareas más simples.





EJEMPLO

- Desarrolla una función recursiva que calcule la factorial de un número.

```
1 Funcion resultado ← factorial(n)
2     Definir resultado Como Entero;
3     Si (n = 0) O (n = 1) Entonces
4         resultado = 1;
5     SiNo
6         resultado = n * factorial(n - 1);
7     FinSi
8 FinFuncion
9
10 Proceso ClaseFunciones
11     Definir num Como Entero;
12     Escribir "Ingrese un número: ";
13     Leer num;
14     Escribir "La factorial de ", num, " es ", factorial(num);
15 FinProceso
```



EJEMPLO

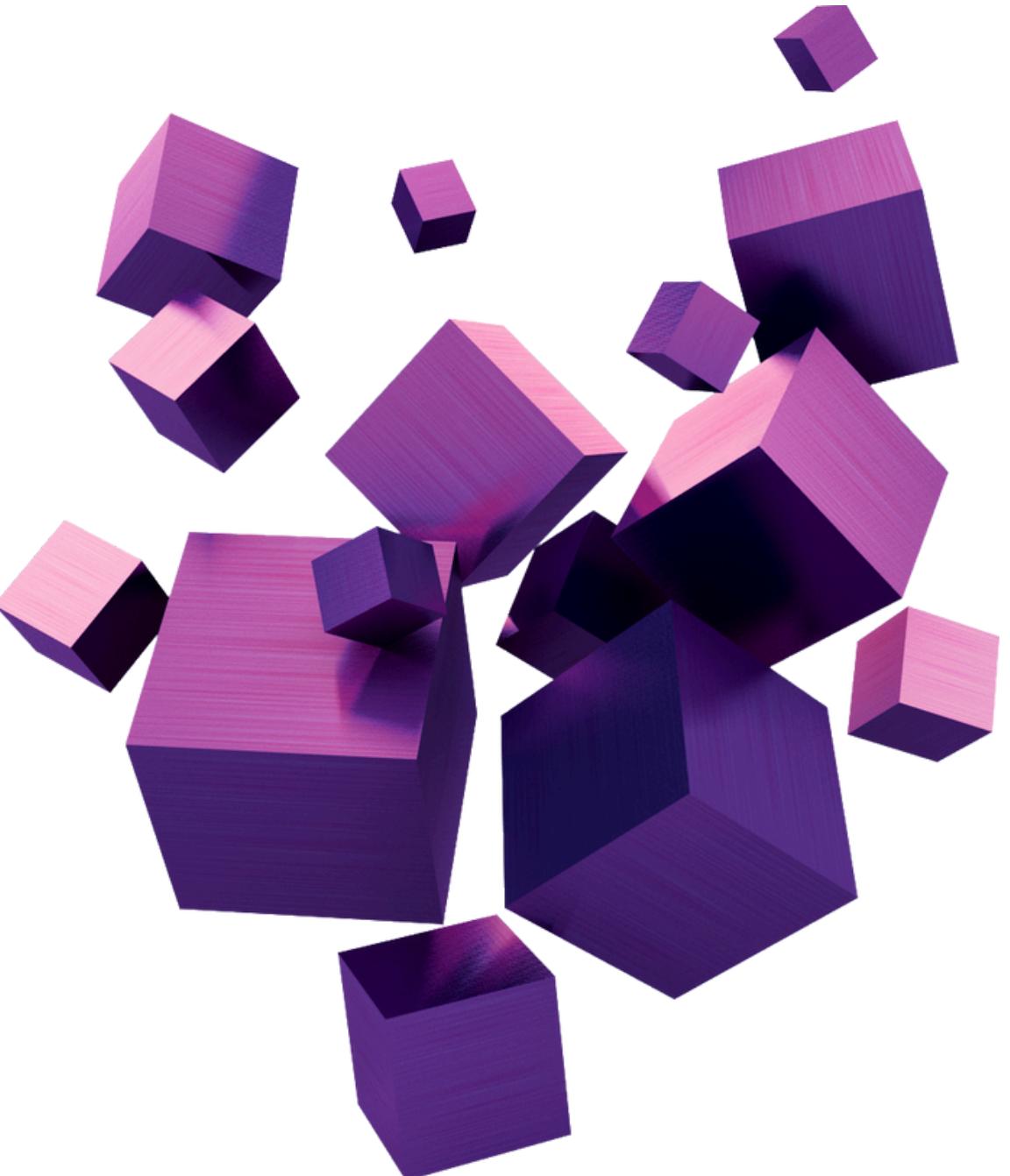
- Desarrolla una función recursiva que simule una cuenta regresiva.

```
1 Funcion cuentaRegresiva(contador)
2   Si contador > 0 Entonces
3     Escribir contador;
4     cuentaRegresiva(contador - 1);
5   SiNo
6     Escribir "Lanzamiento!";
7   FinSi
8 FinFuncion
9
10 Proceso ClaseFunciones
11   cuentaRegresiva(10);
12 FinProceso
```



EJERCICIOS

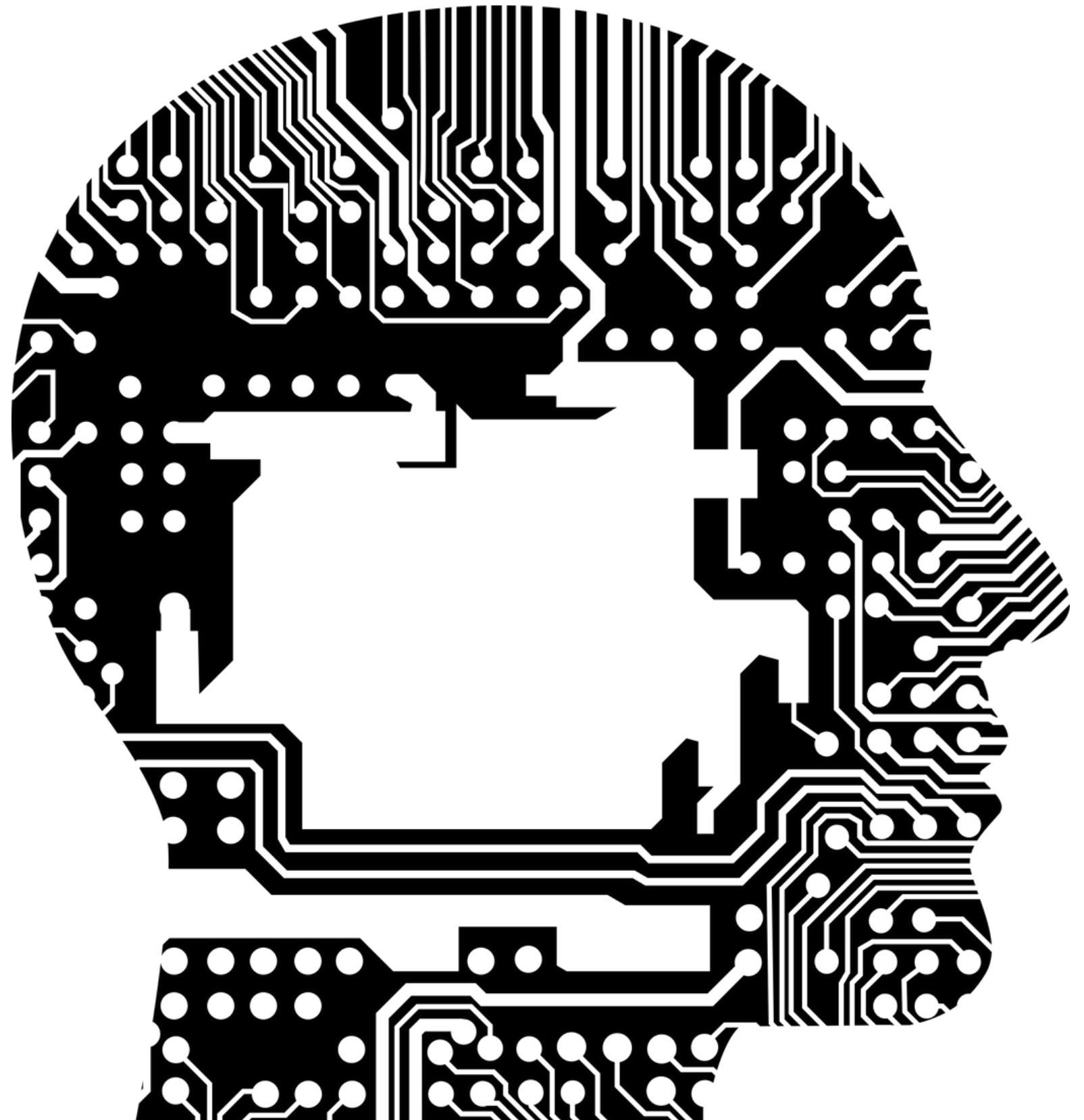
- Define una función para calcular el área y otra el perímetro de un cuadrado.
- Solicita al usuario la longitud de un lado del cuadrado.
- Y no olvides mostrarle el resultado una vez lo obtengas.
- **Tips:**
 - Área: Lado * Lado.
 - Perímetro: Suma de sus lados.

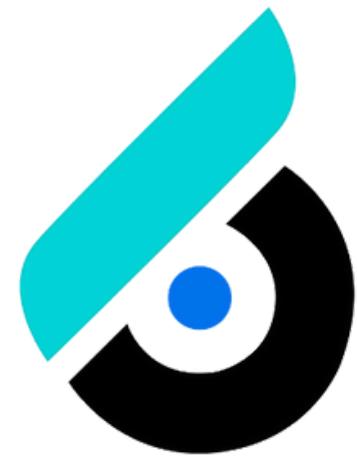




EJERCICIOS

- Crea una función que rellene un arreglo con números aleatorios y otra que los muestre por consola.
- Haz lo mismo con diferentes funciones para crear una matriz y mostrarla.
- Prueba crear un menú para organizar las opciones en la consola.





AWAKELAB

#programmingbootcamp

nodovirtual.awakelab.cl

 jELOU futurejob by  Chile