

Pensamiento Lógico: Funciones y Estructuras de Control

Ana Belisa Martínez
Platzi Julio 2022

Tipos de datos: ¿Qué es un dato?

¿Qué es un dato?

Es una porción de información muy concreta y específica que nos permite, a través de diferentes operaciones y estructuras, tomar decisiones.

Hacen parte de un algoritmo en una estructura más grande

Son la base sobre la cual partimos y sobre la cual llegamos

Componen todas las estructuras para que puedan funcionar y sean operables.

Tipos de datos: Texto



Tipos de datos: Números

Tipo de dato: numéricos

Pueden ser Int y/o Float

Los datos **Int** (Enteros) contiene solo números enteros: 7 , 10, 30, etc

Los datos **Float / Double** contiene número con decimales: 6.5 , 8.6, 3.141623 etc.

Los operadores aritméticos nos sirven para hacer operaciones matemáticas con los datos numéricos

No es posible concatenar cadenas de diferente tipo.

Para hacer una concatenación de números deberían ser convertidos a texto

Para convertir un número a texto ingresamos el dígito en comillas dobles: "22"

Sumar: $19 + 92 = 111$
Concatenar: "19" + "92" = 1992

Tipos de datos: Booleanos

Tipo de dato:
Booleanos

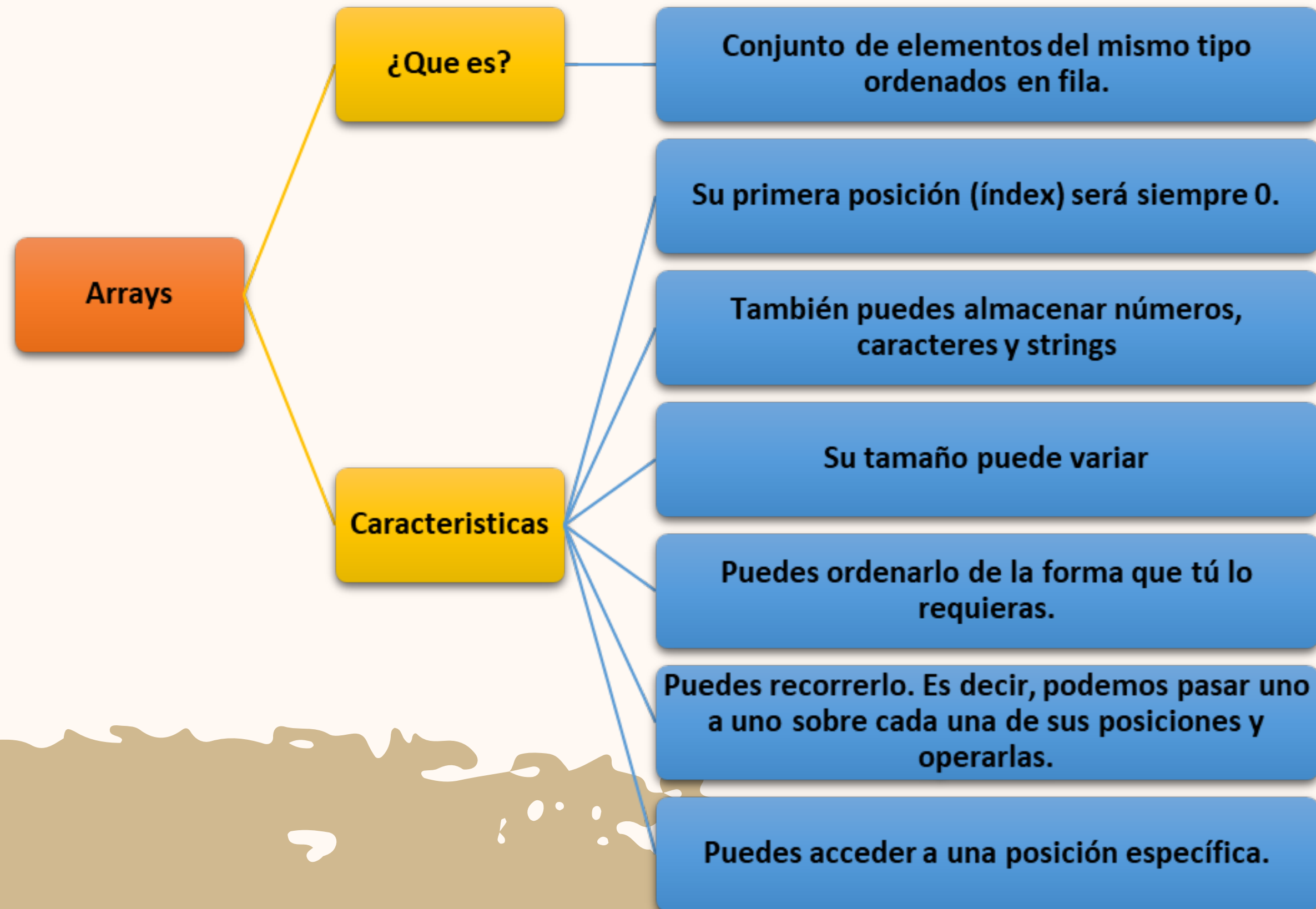


Este tipo de
datos
representa una
cantidad lógica
que es una
respuesta a una
pregunta de
verdadero o
falso.



**Su
representación**
True = 1
False = 0

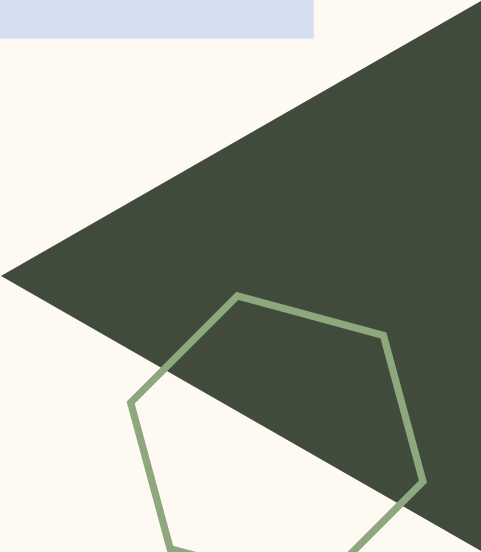
Tipos de datos: Arrays



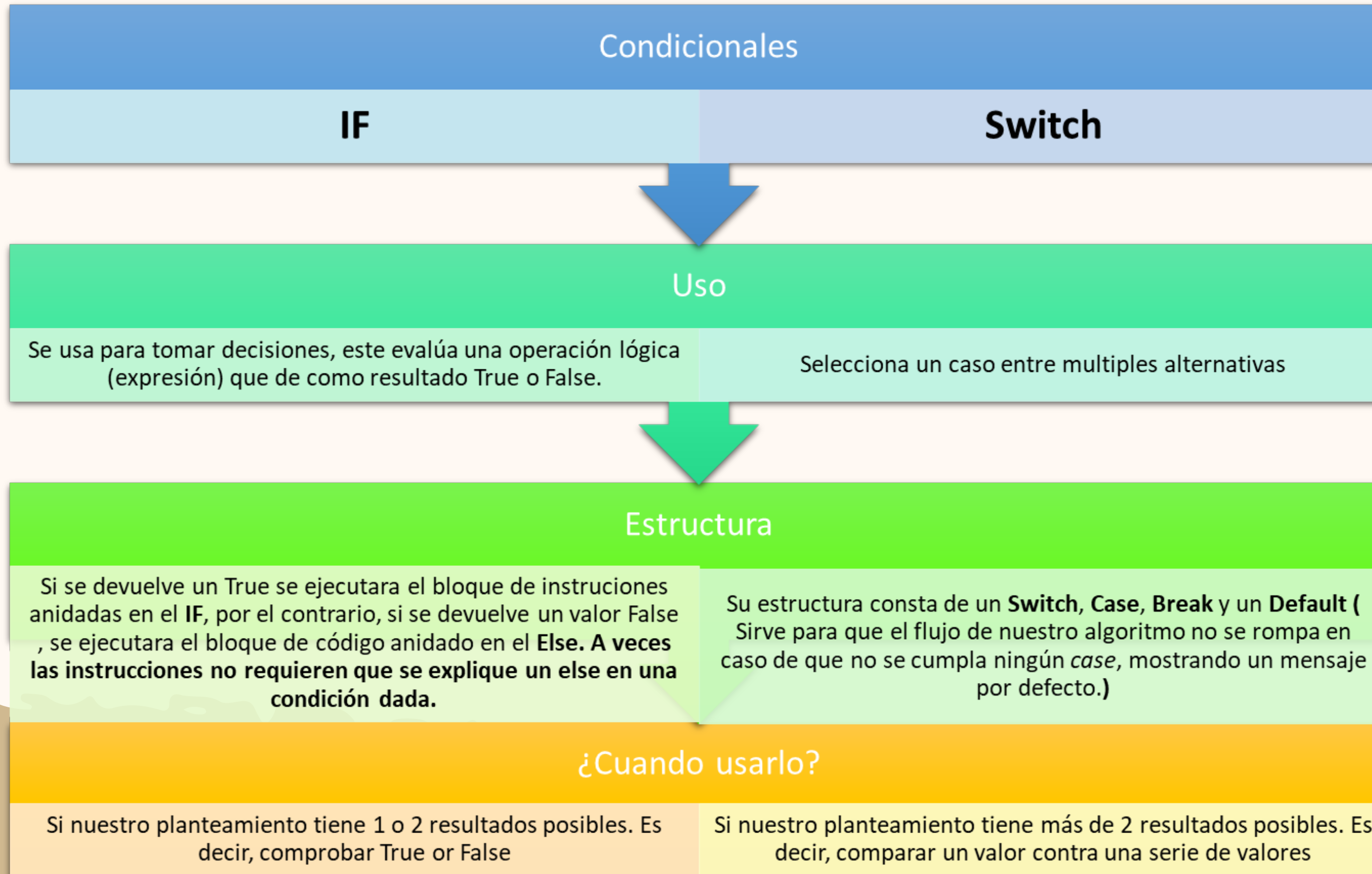


¿Qué son las estructuras de control?

Estructuras	Condicionales	Ciclos	IF/Else	Switch	While	Do While	For	Try/Catch
<ul style="list-style-type: none">• Herramientas sobre las cuales se construye y controla el flujo de un algoritmo.• Permiten tomar decisiones y realizar un proceso repetidas veces.	<ul style="list-style-type: none">• Permiten tomar decisiones a partir de una comparación u operación<ul style="list-style-type: none">• If/else , Switch, Try/Catch.	<ul style="list-style-type: none">• Realizan diferentes iteraciones de acuerdo a condiciones o situaciones que pide el algoritmo• For, While, Do-While.	<ul style="list-style-type: none">• Si se cumple determinada condición se ejecuta acción/código anidada en el if, si no es así se ejecutará la anidada en el else.• Esta es muy usada para validar información.	<ul style="list-style-type: none">• Nos permite evaluar múltiples casos que puede llegar a cumplir una variable y realizar una acción en esa situación.• Su estructura consta de un Switch, Case, Break y un Default.	<ul style="list-style-type: none">• While (ciclo indefinido): Cuando no sabes las veces que se repetirá un ciclo.	<ul style="list-style-type: none">• Do While (ciclo indefinido): Cuando no sabes las veces que se repetirá un ciclo pero necesitas que se realice por lo menos una vez.	<ul style="list-style-type: none">• For (ciclo definido): Cuando sabes (o puedes saber) las veces (iteraciones) que repetirás el ciclo	<ul style="list-style-type: none">• Intenta hacer X, pero, si falla haz Y.



Condicionales



Ciclos

¿Qué es un ciclo/bucle?

Es una estructura de control que ejecuta un bloque de instrucciones de manera repetida.

For (Ciclo Definido)

Cuando sabes o puedes saber las veces (**iteraciones**) que repetirás el ciclo.

Su estructura consta de: Punto de partida/inicialización (Que puede ser 0 o cualquier valor que tu requieras), Cantidad de Iteraciones e Incremento

Ejemplo: Suministrar 100 litros de gasolina for (x=0; x<=100; x++){ Suministrar un litro }

While (Ciclo Indefinido)

Cuando no sabes las veces (**iteraciones**) que se repetirá un ciclo.

Primero evalua las iteraciones despues ejecuta el código

El ciclo puede terminar en la primera **iteración** o puede terminar después de 1000 iteraciones

Ejemplo: Escribe un número! Si quieres salir escribe 0: While (x!=0) {Has escrito el número x}

Do While (Ciclo Indefinido)

Cuando no sabes las veces que se repetirá un ciclo y necesitas que se realice por lo menos una vez.

Podemos generar números de series aleatorios únicos en nuestra base de datos que no se repitan.

Do {Genera un número de serie, busca si ya existe en la base de datos } While (Existe en la base de datos)

Primero ejecuta el bloque de código luego evalua la cantidad de iteraciones a realizar

Try Catch: Funcionamiento

Primero, se ejecuta el código en try {...}.



Si no hubo errores, se ignora catch (err): la ejecución llega al final de try y continúa, omitiendo catch.



Si se produce un error, la ejecución de try se detiene y el control fluye al comienzo de catch (e).



La variable e (podemos usar cualquier nombre para ella) contendrá un objeto de error con detalles sobre lo que sucedió.

¿Qué es una función?

Función

- Una función es un bloque de instrucciones que hace una actividad específica de forma independiente al resto del algoritmo.

Objetivo

- Con una función puedes organizar y segmentar partes de tu algoritmo, las cuales puedes reutilizar.

Componente

- Parametros
- Instrucciones de la función
- Resultado

Parametros

- Son los valores que recibe la función como entrada (puede llevarlos o no, pero en la mayoría de casos si lleva).

Instrucciones

- Son las operaciones que hace la función.

Resultado

- O valor de retorno, es el valor final que entrega la función.

Scope p alcance en funciones

Función Pública

El alcance de la función es global

Puedes mandar a llamar las funciones de un algoritmo a otro.

Función Privada

El alcance de la función es local.

No pueden ser llamadas en otros algoritmos

Recomendaciones

Es importante determinar el alcance de la función.



Gracias