ACÁMICA

Agenda

Daily

Funciones

Break

Cierre

TEMA DEL DÍA

Funciones

Hoy veremos variables. Con ellas podremos almacenar información para utilizarla en cualquier momento de nuestra aplicación.

Daily

Daily

Sincronizando... 20 min.

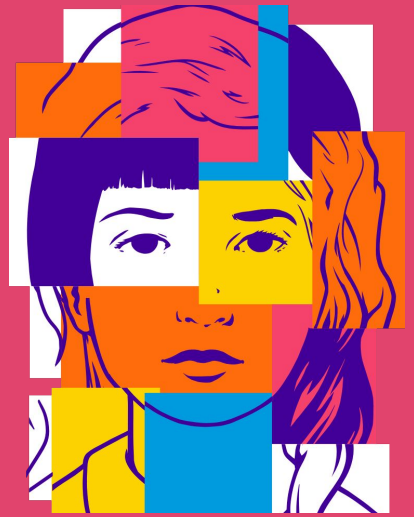
Toolbox

¿Cómo te ha ido?
¿Obstáculos?
¿Cómo seguimos?

Challenge

¿Cómo te ha ido?
¿Obstáculos?
¿Cómo seguimos?

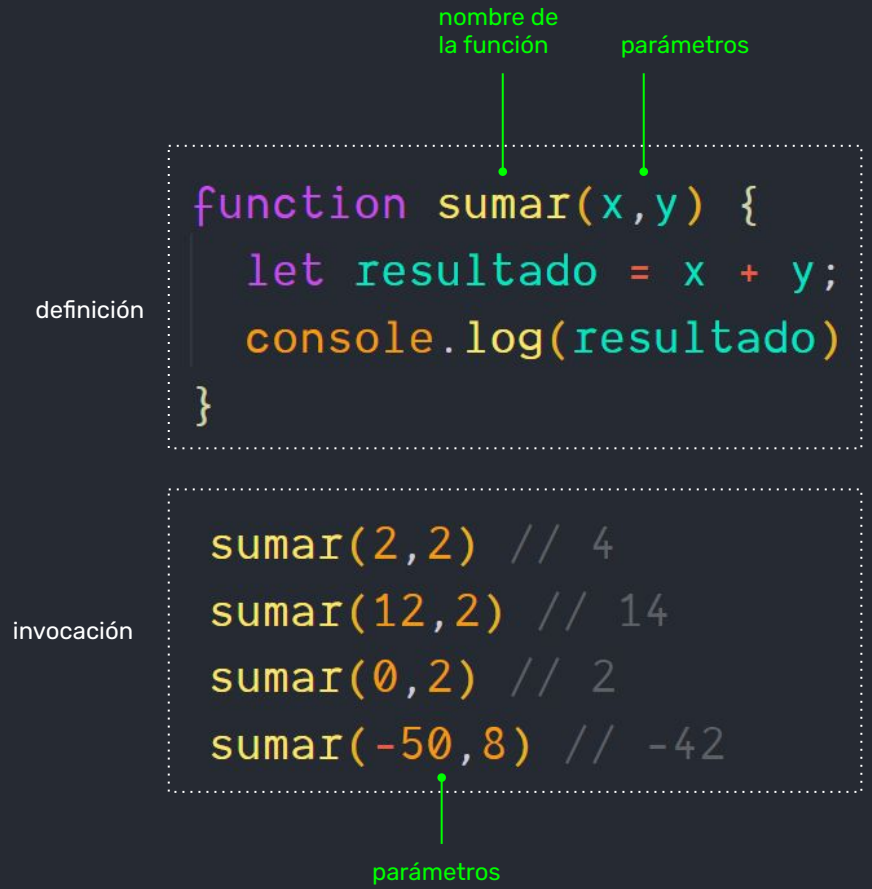
Funciones



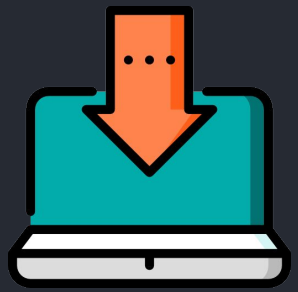
Funciones

Las funciones proveen una forma de definir una serie de pasos, los cuales podemos invocar en cualquier lugar de la aplicación, sin necesidad de volverlos a escribir.

Se pudiera decir que las funciones guardan algoritmos.



ENTRADA



PROCESO

SALIDA

¿¿¿ SALIDA ???

Más adelante hablamos de la salida de la función.

¡Programamos!

Ya tu sabes...

Justo como lo hicimos en el encuentro anterior, estaremos construyendo un ejercicio mientras practicamos el concepto de funciones.

Instrucciones para el **squad lead**:

- Compartir esta presentación como PDF por el canal de slack. Todxs lxs trainees deberán ir siguiendo la presentación, paso a paso.
- Abrir un Code Sandbox nuevo con el template “Vanilla JS”, ir al archivo index.js y borrar todo el contenido. Dejar el archivo en blanco.
- Escoger una persona voluntaria al azar, para que ejecute el paso 1, listado en las próximas láminas.

Paso 1, Escribir variables.

Las funciones nos permiten ejecutar una serie de pasos definidos, considerando datos de entrada. Antes de escribir funciones, vamos a escribir una aplicación que nos permita calcular el promedio de 3 números. Para esto, declara 3 variables, y completa la fórmula para calcular el promedio, en la variable correspondiente.

```
let num1 = 43;
let num2 = 23.12;
let num3 = 84;

let promedio = ???
```

Ahora, ¡escoge a otra persona para que haga el paso 2!

Paso 2, Imprimir el promedio por consola

Una vez que tengamos el promedio calculado, utilicemos el console.log() para mostrar por consola el resultado.

```
let promedio = num1 + num2 + num3 / 3;
console.log("el promedio es" + promedio);
```

Está aplicación funciona como esperamos. Pero, ¿qué podemos hacer si queremos calcular el promedio de otros 3 números?

Ahora, ¡escoge a otra persona para que haga el paso 3!

Paso 3, creamos una función.

Crea la función `calcularPromedio`, la cual toma 3 parámetros, y muestra por consola el promedio de los mismos.

```
function calcularPromedio(n1, n2, n3) {
  let promedio = num1 + num2 + num3 / 3;
  console.log("el promedio es" + promedio);
}
```

Lo anterior se conoce como la definición de la función. Si queremos invocarla, tendremos que escribir esto:

```
let num1 = 43;
let num2 = 23.12;
let num3 = 84;

calcularPromedio(num1, num2, num3);
```

Escoge a otra persona para que haga el paso 4

Paso 4, reutilizar funciones

La ventaja de las funciones es que podemos reutilizarlas, pero pasando un conjunto de datos de entrada nuevos. Por ejemplo, ahora si queremos calcular el promedio de otros 3 números, podemos hacer lo siguiente:

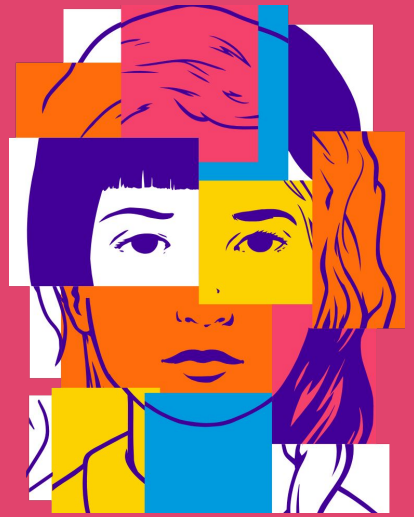
```
calcularPromedio(24, 12, 10);
calcularPromedio(23, 231, 34);
calcularPromedio(6753, 23442, 83453);
```

¿Que vimos por consola, al ejecutar este código?

¡BREAK!

Ph. Credit: Drew Coffmann

return

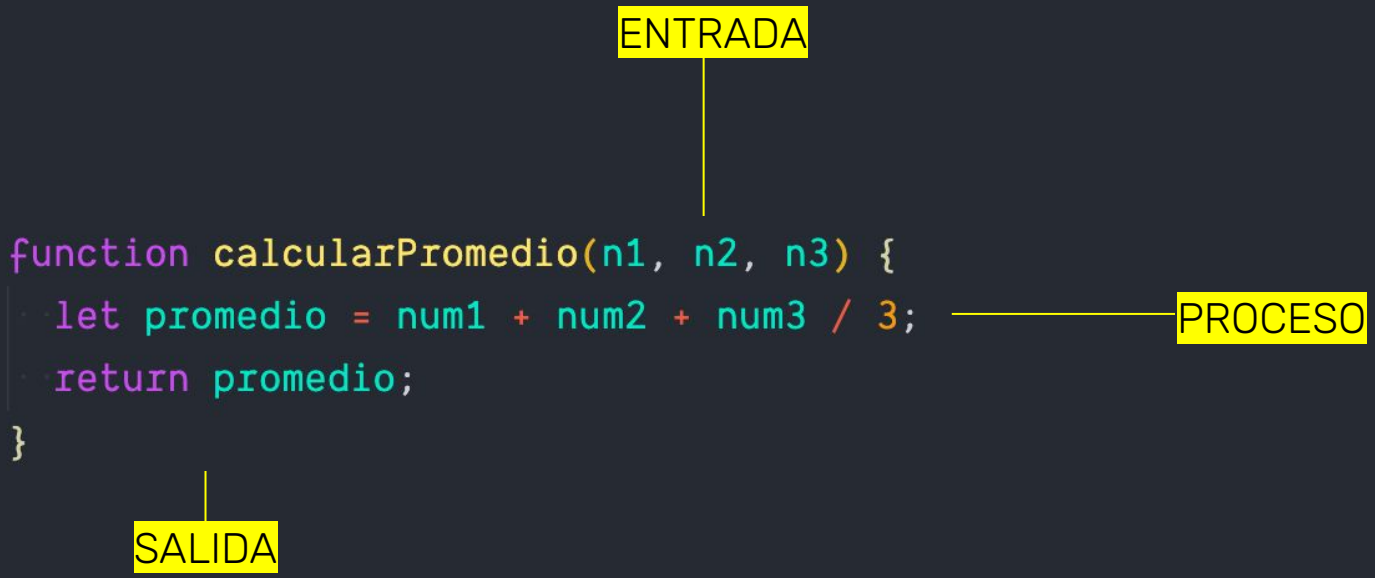


return

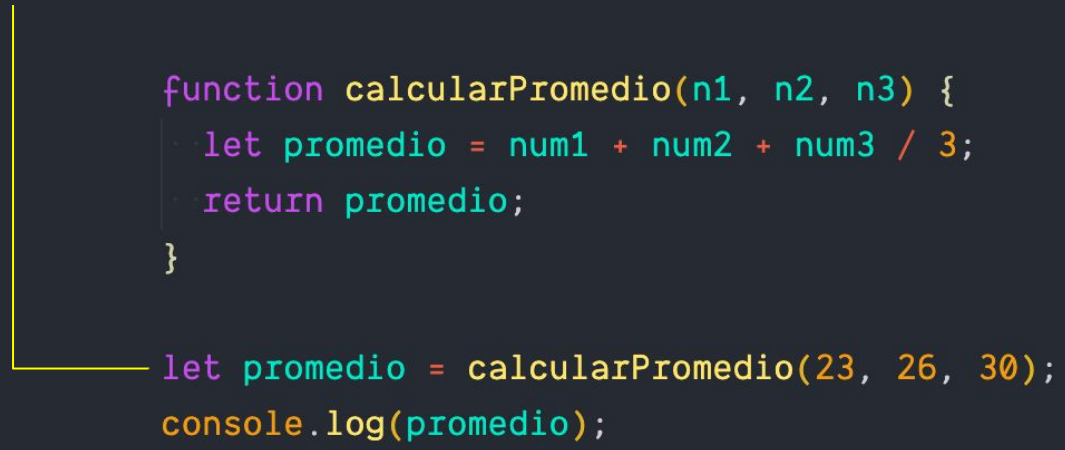
Las funciones opcionalmente pueden retornar un valor.

¿Recuerdan las palabras reservadas de Javascript? `return` es una de ellas.

```
function calcularPromedio(n1, n2, n3) {
  let promedio = num1 + num2 + num3 / 3;
  return promedio;
}
```


ENTRADA
function calcularPromedio(n1, n2, n3) {
let promedio = num1 + num2 + num3 / 3;
PROCESO
return promedio;
}
SALIDA

Cuando las funciones retornan valores, debemos declarar variables que actúan como receptoras de los mismos.



```
function calcularPromedio(n1, n2, n3) {
  let promedio = num1 + num2 + num3 / 3;
  return promedio;
}

let promedio = calcularPromedio(23, 26, 30);
console.log(promedio);
```

¡Programamos!

Paso 1, modificar la función

Modifiquemos la función de calcular promedio, para que nos retorne el promedio de 3 números.

```
function calcularPromedio(n1, n2, n3) {
  let promedio = num1 + num2 + num3 / 3;
  return promedio;
}
```

Luego, invoca la función con 3 números, pero recuerda que deberás declarar una variable receptora del resultado. Al finalizar, imprime por consola el promedio.

```
let promedio = calcularPromedio(23, 26, 30);
console.log(promedio);
```

DWFE

Nos tomamos unos minutos para completar esta encuesta.

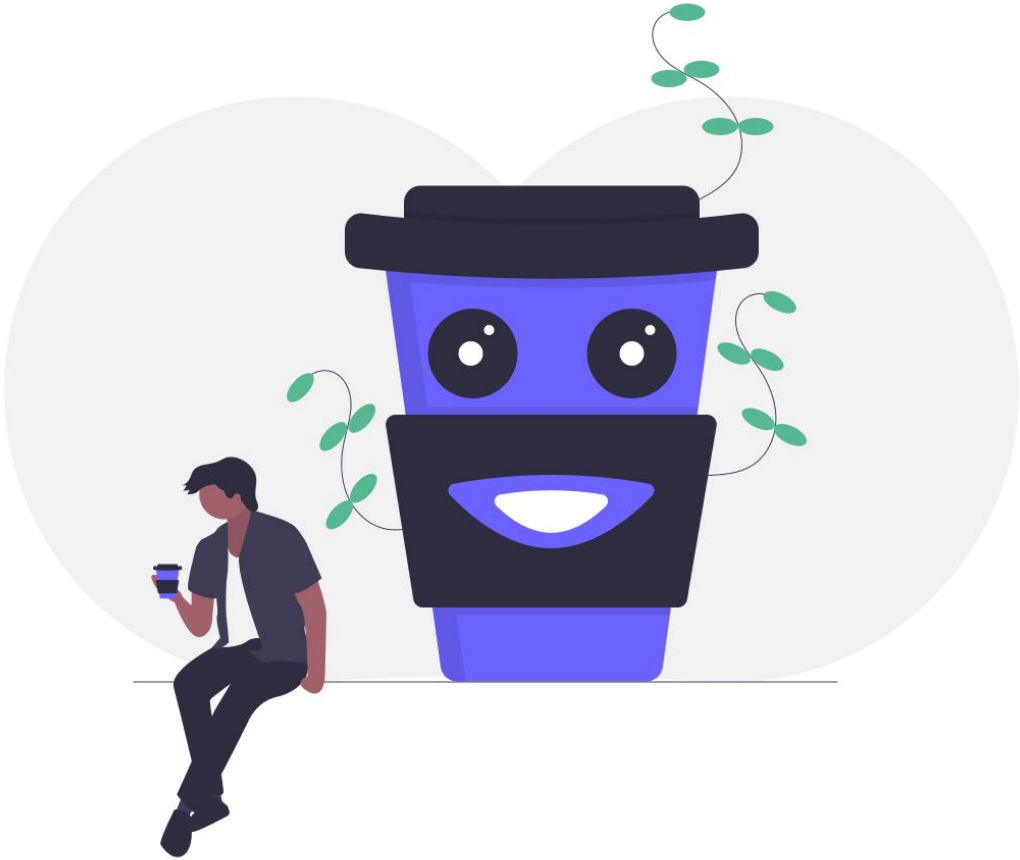
Queremos saber cómo valoran mi tarea hasta acá. ¡Les va a llevar solo un minuto!

Actividad

Pensemos en un algoritmo para servir una taza de cafe.

Partamos de la idea que queremos preparar un cafe negro, con azucar.

¿Cuáles serían los datos de entrada?



Agua	Cafe	Azúcar	Taza
			

¿Cuál será la unidad de medida para cada una de estas variables?

Litros	*Gramos*	*Gramos/cubo/si/no*	*Litros/Capacidad*

¿Cuál será el tipo de variable para cada una?

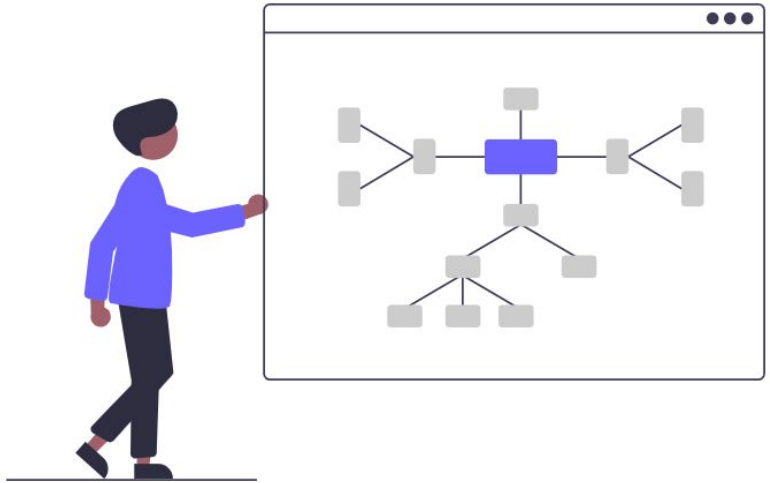
número	*número*	*número* (¿o boolean?)	*Litros*

Café-**O**-matic

Desarrolla un algoritmo que muestre mensajes sobre el proceso de preparación de un cafe.

Considera las siguientes variables de entrada:

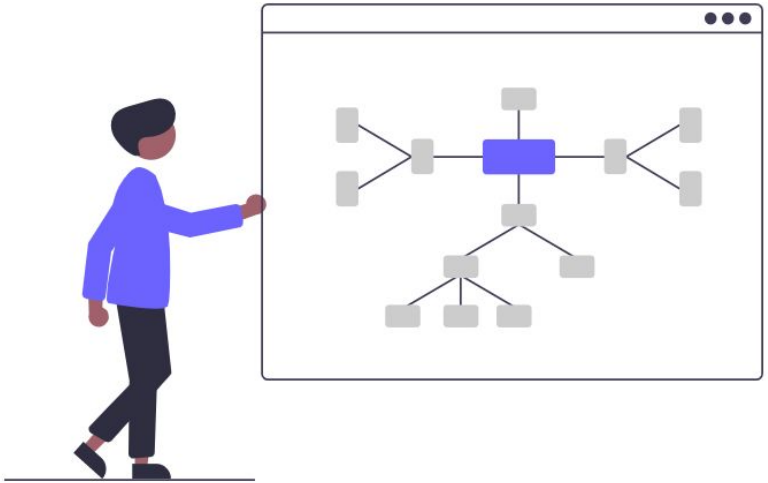
- capacidad de la taza, en litros ☕
- cantidad de agua, en litros 💧
- cantidad de café, en gramos 🌰
- azúcar, como una preferencia (si / no) 🍬



Café-**O**-matic

En el algoritmo, deberás verificar las siguientes condiciones:

- si la cantidad de agua es mayor que la capacidad de la taza, finalizar el proceso y mostrar un mensaje indicativo.
- si la cantidad de gramos de café es mayor a 50, avisar que el café está quedando intenso.
- mostrar un mensaje si el café lleva azúcar, y si no, también.
- mostrar un mensaje que el proceso ha terminado.



Café-**O**-matic

Por ejemplo, si le damos al algoritmo estos datos de entrada:

- capacidad de taza = 300;
- gramos de café = 30;
- azúcar = si;
- cantidad de agua = 280;

Deberíamos de ver los siguientes mensajes:

- “Preparando el café”
- “Tendrá un sabor suave”
- “Con un poco de azúcar”
- “Aquí tienes el café preparado”

Pero, si le damos estos datos de entrada:

- capacidad de taza = 300;
- gramos de café = 30;
- azúcar = si;
- cantidad de agua = 320;

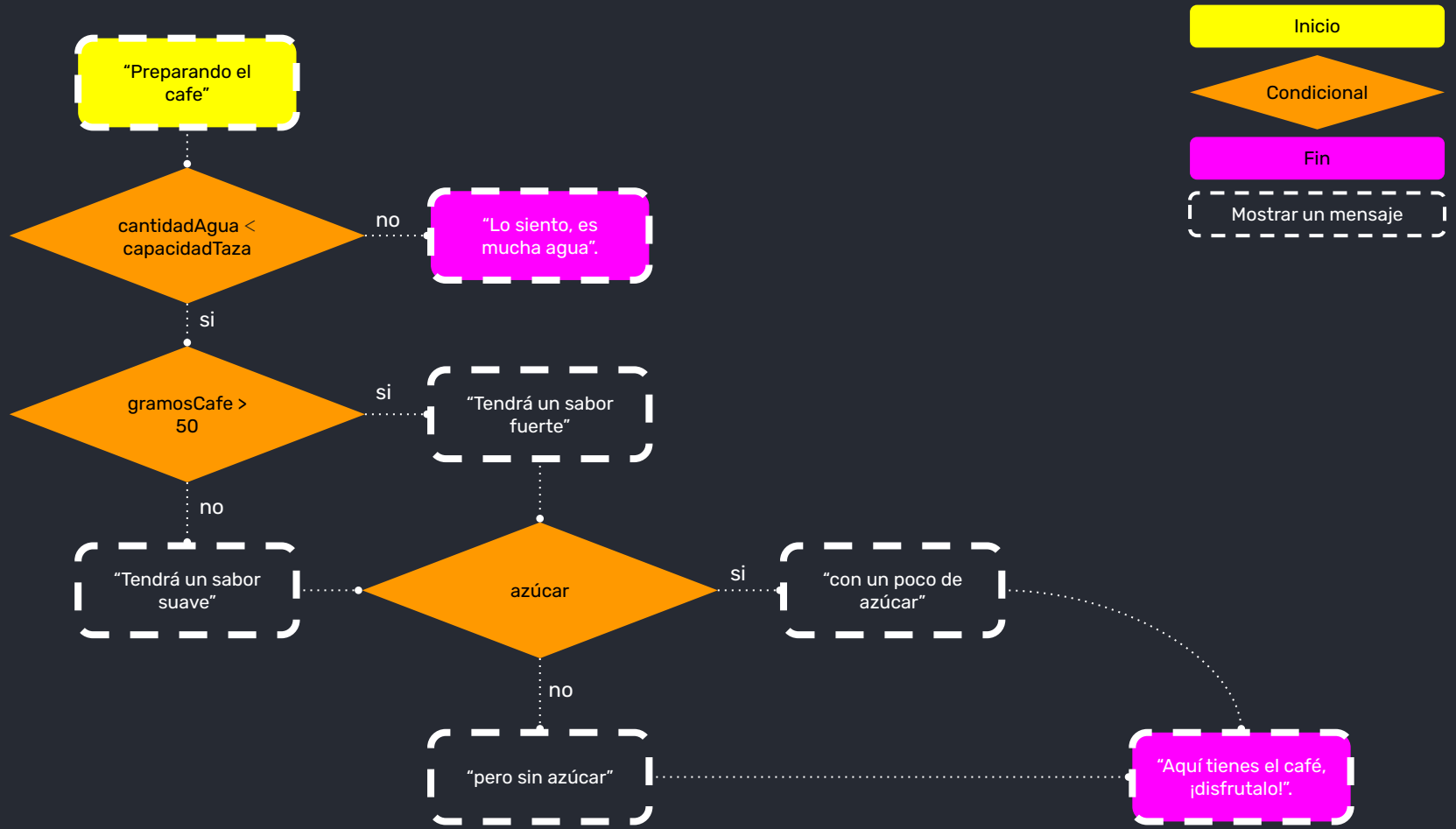
Deberíamos de ver los siguientes mensajes:

- “Preparando el café”
- “Lo siento, es mucha agua”.

Café-**O**-matic

En grupo, diseñen una propuesta de algoritmo considerando los puntos anteriores.

Tomaremos unos 10min para esto.


Inicio
Condicional
Fin
Mostrar un mensaje
“Preparando el cafe”
cantidadAgua < capacidadTaza
no
“Lo siento, es mucha agua”.
si
gramosCafe > 50
si
“Tendrá un sabor fuerte”
no
“Tendrá un sabor suave”
azúcar
si
“con un poco de azúcar”
no
“pero sin azúcar”
“Aquí tienes el café, ¡disfrutalo!”.

Café-**O**-matic **BETA**

Encapsula el algoritmo previamente hecho, en una función de Javascript.

Adicionalmente, define los parámetros de entrada que toma la función.

Una vez hecho lo anterior, invoca la función con los siguientes ejemplos:

```
cafeomatic(200,220,30,true)
cafeomatic(200,100,80,false)
cafeomatic(100,220,90,true)
```

agua taza café azúcar

Para la casa

Escándalo-tron **v0.1**

Desarrollemos la función **`escandalotron();`** la cual toma como parámetros 3 valores, representando los niveles de decibelios.

La función debe retornar un string indicando el resultado según el algoritmo creado en el meeting anterior. Considera los siguientes casos de uso.

```
let resultado2 = escandalotron(50,20,40);
console.log(resultado2) // "ESCÁNDALO"

let resultado2 = escandalotron(10,10,4);
console.log(resultado2) // "SUSURRO"

let resultado31 = escandalotron(10,20,40);
console.log(resultado3) // "NORMAL"
```

Sumatoria

Desarrolla la función “sumatoria”, la cual debe tomar 5 parámetros (números) y retorna la suma de los mismos.

Por ejemplo, considera este caso de uso:

```
let suma = sumatoria(2,3,4,5,6);
console.log(suma) // debe ser 20
```

Sumatoria multiplicada

Desarrolla la función “sumatoriaMultiplicada”, la cual debe tomar 5 parámetros (números). La función debe retornar la suma de los 4 primeros números, multiplicada por el último.

Por ejemplo, considera este caso de uso:

```
let sumaMultiplicada = sumatoriaMultiplicada(2,3,4,5,6);
console.log(suma) // debe ser 84 ((2 + 3 + 4 + 5) * 6)
```

“Hola, _____“

Desarrolla la función “saludarA”, la cual toma un string, representando un nombre. La función debe imprimir por consola un saludo, con un “Hola” concatenado.

Por ejemplo, considera este caso de uso:

```
saludarA("carolina");
// debe mostrar por consola "Hola, carolina."
```

Es mayor a 10?

Desarrolla la función “esMayorAdiez”, la cual toma 1 número como elemento de entrada. La función debe retornar “true” si el número es mayor a 10, o de lo contrario, false.

Por ejemplo, considera estos casos de uso:

```
let veinteEsMayorAdiez = esMayorAdiez(20);
console.log(veinteEsMayorAdiez) // true

let cincoEsMayorAdiez = esMayorAdiez(5);
console.log(cincoEsMayorAdiez) // false
```

Para la próxima

1) Termina el ejercicio de la meeting de hoy.
2) Lee la toolbox 21.
3) Resuelve el challenge.

ACÁMICA