

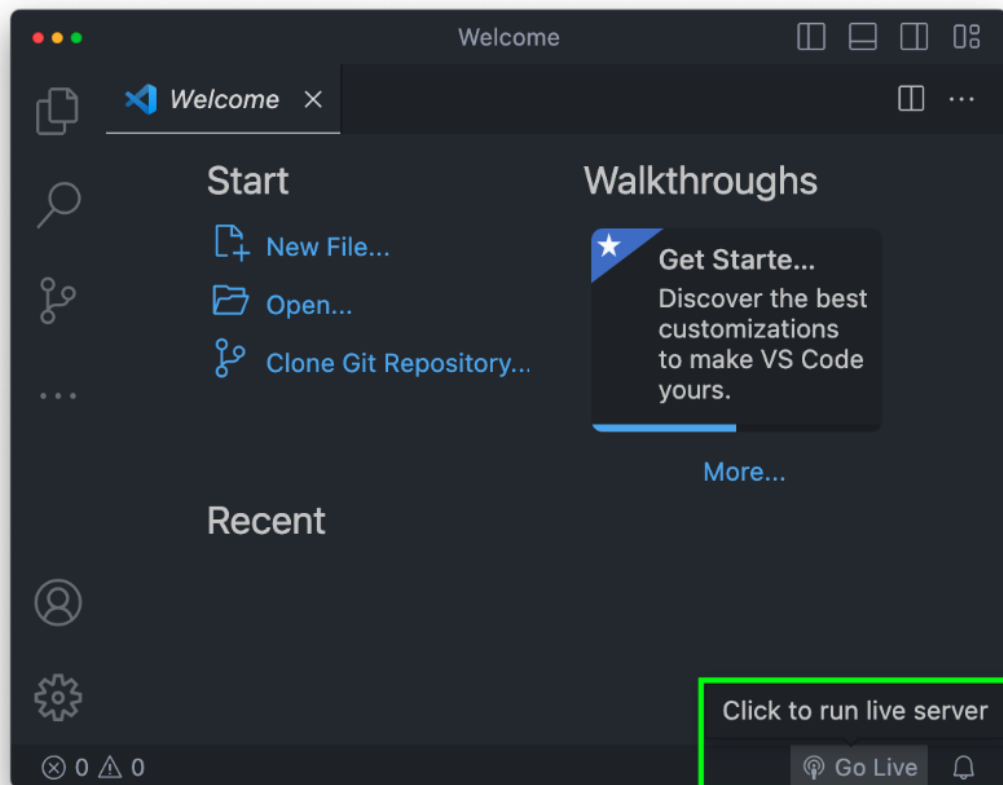


## Guía de resolución

### Resolución de microactividades

#### Actividad 1

Finalizada la instalación, deberás ver la interfaz de VS CODE similar a la de esta imagen un botón con la leyenda Go Live, en el extremo inferior derecho de la ventana de VS CODE.





## Actividad 2

Archivo index.html con la referencia al archivo index.js en la última línea del apartado </head>.

```
4 <meta charset="UTF-8">
5 <title>Venta de Frutas</title>
6 <link rel="icon" href="images/icon-fruits-64.png" type="image/png">
7 <meta name="viewport" content="width=device-width, initial-scale=1.0">
8 <link rel="stylesheet" href="css/style.css">
9 <script src="js/index.js"></script>
```

Declaración de dos variables (string y number), con valores asignados al azar.

Declaración de arrayFrutas. Se incluyeron al menos 5 elementos del tipo fruta.

```
index.html  js index.js  X
js > js index.js > ...
1 let instituto = 'TECLAB'
2 let numero = 1975
3 const arrayFrutas = ['Pomelo', 'Naranja', 'Limón', 'Frambuesas', 'Cerezas']
4
5 console.log(instituto)
6 console.log(numero)
7
8 console.table(arrayFrutas)
```

## Actividad 3

Estructura del código de ejemplo de iteración de arrayFrutas.

```
10
11 function iterarArrayFrutas() {
12     for (let i = 0; i < arrayFrutas.length; i++) {
13         if (i === 1 || i === 3) {
14             continue
15         }
16         console.log(arrayFrutas[i])
17     }
18 }
19 iterarArrayFrutas()
```



Visualización del resultado en la Consola JS, de acuerdo a los elementos del array definidos. Sobre un array de 5 elementos, son tres los que nos retorna la iteración.

► Array(5)	
Pomelo	<a href="#">index.js:16</a>
Limón	<a href="#">index.js:16</a>
Cerezas	<a href="#">index.js:16</a>
>	

#### Actividad 4

Modificación de la función de la microactividad anterior, y conversión de la misma en Arrow function que recibe un array como parámetro. Ahora podrá iterar no solo el array frutas, sino también cualquier otro array que le enviemos por parámetro.

```
22  const iterarArrayFrutas = (array)=> {  
23    for (let i = 0; i < array.length; i++) {  
24      if (i === 1 || i === 3) {  
25        continue  
26      }  
27      console.log(array[i])  
28    }  
29  }
```



## Actividad 5

La clase Producto con su método descontar stock, y las validaciones previas sobre el parámetro unidades recibido.

```
class Producto {  
  constructor(id, precio, stock) {  
    this.id = id  
    this.precio = precio  
    this.stock = stock  
  }  
  descontarDeStock(unidades) {  
    if (typeof unidades !== 'number') {  
      console.error("Se esperaba un parámetro numérico.")  
      return  
    }  
    if (this.stock - unidades < 0) {  
      console.error("No puede haber stock en negativo.")  
      return  
    }  
    return this.stock -= unidades  
  }  
}
```

## Actividad 6

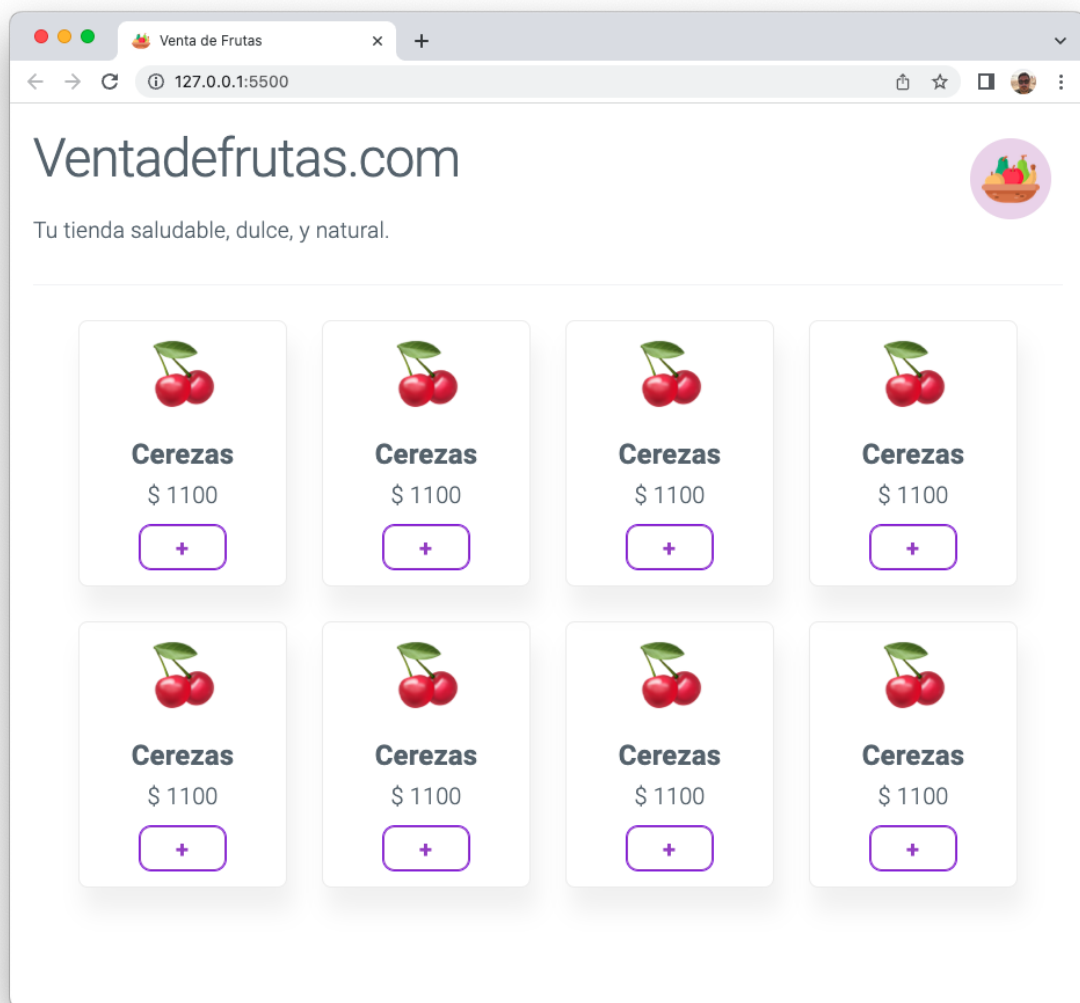
Función que permite agregar elementos a un array, validando previamente que estos no existan.

```
const paisesDelSur = ['Argentina', 'Uruguay', 'Brasil', 'Venezuela', 'Chile']  
  
function agregarElemento(pais) {  
  const resultado = paisesDelSur.indexOf(pais.toLowerCase())  
  if (resultado === -1) {  
    console.warn("El país ya existe en el array.")  
    return  
  }  
  paisesDelSur.push(pais)  
}
```



## Actividad 7

La Card HTML de frutas es replicada al menos 8 veces en el mismo documento HTML.



Declaración de la constante botones, la cual nos conecta a todos los botones de las Cards HTML.

```
const botones = document.querySelectorAll("button.button.button-outline.button-add")
console.log(botones)
```



Resultado de `console.log(botones)` en la Consola JS. Este nos retorna un array del tipo `HTMLElement`, con la lista de nodos a los cuales estamos conectados desde botones.

```
NodeList(8) [button.button.button-outline.button-add,  
button.button.button-outline.button-add,  
button.button.button-outline.button-add,  
button.button.button-outline.button-add,  
button.button.button-outline.button-add,  
button.button.button-outline.button-add,  
button.button.button-outline.button-add,  
button.button.button-outline.button-add]
```

The screenshot shows the Chrome DevTools Console with the 'Consola' tab selected. The console displays the result of a `console.log` statement, which is a `NodeList(8)` containing eight `button.button.button-outline.button-add` elements. The console interface includes a search bar, a filter dropdown, and a 'Sin problemas' status indicator.

## Actividad 8

Resultado de la última microactividad. `CardHTML` que retorna un `Template String + Literals`, armando una tarjeta HTML por cada producto del array. En la función `cargarProductos()` podemos darle un valor agregado mayor, preguntando si el array posee al menos un elemento previo a iterarlo.



```
const container = document.querySelector("div.container")

const retornarCardHTML = (producto) => {
  return `<div class="card ">
    <div class="card-image">${producto.imagen}</div>
    <div class="card-name">${producto.nombre}</div>
    <div class="card-price">$ ${producto.precio}</div>
    <div class="card-button">
      <button class="button button-outline button-add" id=""
        title="Clic para agregar al carrito">+</button>
    </div>
  </div>`
}

const cargarProductos = (array) => {
  container.innerHTML = ""
  array.forEach(producto => {
    container.innerHTML += retornarCardHTML(producto)
  })
}
```