

**Universidad Nacional de San Agustín
Facultad de Ingeniería
Escuela Profesional de Ingeniería de
Sistemas**



**Curso: Introducción al Desarrollo
Web**

Laboratorio

Grupo: “D”

**Docente: Carlo Jose Luis Corrales
Delgado**

**Título de la Actividad: Laboratorio 16
– Javascript: Eventos y JSON**

**Estudiante: Alessandro Josue Justo
Vilca**

2025

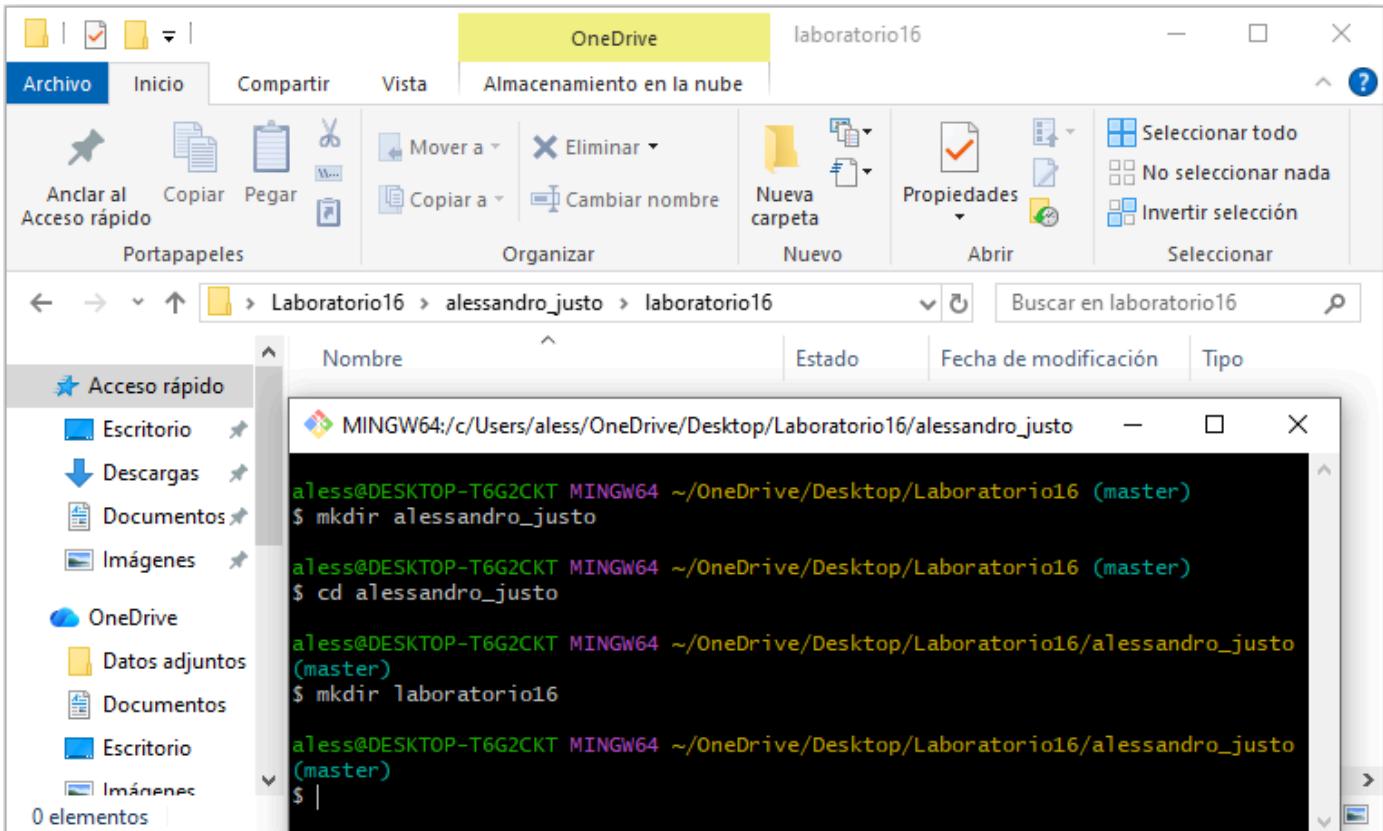
1. Crear un directorio que tenga su nombre y un subdirectorio laboratorio16. Recomendación: usar minúsculas, sin espacios, sin tildes ni “ñ” y con guiones medios o bajos

// Laboratorio Nro 16 - Ejercicio1

// Autor: Alessandro Josue Justo Vilca

// Colaboró : sin colaboradores

// Tiempo : 1min



2. Utilizar los atajos de teclado o combinación de teclas para agilizar su trabajo

// Laboratorio Nro 16 - Ejercicio2

// Autor: Alessandro Josue Justo Vilca

// Colaboró : sin colaboradores

// Tiempo : 1min

Atajos de teclado que use y tuve en cuenta en VS Code:

Ctrl + Ñ: abre la terminal integrada.

Ctrl + /: comentar línea seleccionada.

Alt + ↑: mover línea hacia arriba.

Alt + ↓: mover línea hacia abajo.

Ctrl + Alt + ↓: seleccionar con el tabulador varias líneas para modificar simultáneamente las líneas.

Shift + Alt + ↓: duplicar línea.

Atajos de teclado útiles (recomendados)

Ctrl + S → guardar

Ctrl + C / Ctrl + V → copiar / pegar

Ctrl + Z / Ctrl + Y → deshacer / rehacer

Ctrl + F → buscar texto en archivo

Ctrl + Shift + F → buscar en todo el proyecto

Ctrl + P → abrir archivo por nombre (VSCode)

Ctrl + ` → abrir terminal integrado (VSCode)

Ctrl + Shift + T → reabrir pestaña cerrada (navegador)

Alt + Tab → cambiar de aplicación

Crear el html y js que manipule el DOM para cada ejercicio:

3. Cambiar contenido dinámicamente: crear un botón que al presionarlo cambie el texto de un párrafo, de “Texto original” a “Texto cambiado”

HTML Básico para resolver todos los ejercicios

// Laboratorio Nro 16 - Ejercicio3

// Autor: Alessandro Josue Justo Vilca

// Colaboró : sin colaboradores

// Tiempo : 25min

The screenshot shows a code editor with the file `index.html` open. The code is a combination of HTML, CSS, and JavaScript. The CSS part defines several classes for styling elements like backgrounds, colors, and transitions. The JavaScript part uses functions to change the text content of paragraphs based on button clicks.

```
index.html X JS ejercicio03.js JS ejercicio04.js JS ejercicio05.js JS ejercicio06.js JS eje
index.html > html > body > section > h2
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <title>Laboratorio16</title>
6      <style>
7          .oscuro {
8              background: #222;
9              color: white;
10         }
11
12         #cuadro12 {
13             transition: transform 0.8s;
14         }
15         .mover {
16             transform: translateX(200px);
17         }
18
19         .completada {
20             color: green;
21         }
22
23         .error {
24             color: red;
25             font-size: 14px;
26         }
27     </style>
28 </head>
29 <body>
30     <h1>Laboratorio DOM 03</h1>
31
32     <section>
33         <h2>Ejercicio 3</h2>
34         <p id="p3">Texto original</p>
35         <button onclick="ejercicio03()">Cambiar texto</button>
36     </section>
37
38     <section>
39         <h2>Ejercicio 4</h2>
40         <p id="p4">Texto original</p>
41         <button onclick="ejercicio04()">Alternar texto</button>
42     </section>
```

```
43      <section>
44          <h2>Ejercicio 5</h2>
45          <button onclick="ejercicio05()">Modo oscuro</button>
46      </section>
47
48      <section>
49          <h2>Ejercicio 6</h2>
50          <button onclick="ejercicio06_menos()">-</button>
51          <span id="contador6">0</span>
52          <button onclick="ejercicio06_mas()">+</button>
53          <p id="msg6"></p>
54      </section>
55
56      <section>
57          <h2>Ejercicio 7</h2>
58          <input id="input7">
59          <button onclick="ejercicio07_agregar()">Agregar</button>
60          <button onclick="ejercicio07_borrar()">Borrar último</button>
61          <ul id="lista7"></ul>
62      </section>
63
64      <section>
65          <h2>Ejercicio 8</h2>
66          <input id="nom8" placeholder="Nombre"><br>
67          <input id="cor8" placeholder="Correo"><br>
68          <button onclick="ejercicio08()">Validar</button>
69          <div id="err8"></div>
70      </section>
71
72      <section>
73          <h2>Ejercicio 9</h2>
74          <div>
75              
76              
77          </div>
78          
79      </section>
80  
```

```
81
82     <section>
83         <h2>Ejercicio 10</h2>
84         <button onclick="ejercicio10()">Generar tabla</button>
85         <div id="tabla10"></div>
86     </section>
87
88     <section>
89         <h2>Ejercicio 11</h2>
90         <ul id="ul11">
91             <li>Item A</li>
92             <li>Item B</li>
93             <li>Item C</li>
94         </ul>
95     </section>
96
97     <section>
98         <h2>Ejercicio 12</h2>
99         <div id="cuadro12" style="width:50px;height:50px;background: red;"></div>
100        <button onclick="ejercicio12()">Mover</button>
101        <button onclick="ejercicio12_reset()">Reiniciar</button>
102    </section>
103
104    <section>
105        <h2>Ejercicio 13</h2>
106        Nombre: <input id="n13">
107        Edad: <input id="e13">
108        <button onclick="ejercicio13_add()">Agregar</button>
109        <table border="1" id="t13">
110            <tr>
111                <th>Nombre</th>
112                <th>Edad</th>
113                <th>Acciones</th>
114            </tr>
115        </table>
116    </section>
```

```
117
118     <section>
119         <h2>Ejercicio 14</h2>
120         <button onclick="ejercicio14()">Mostrar JSON</button>
121     </section>
122
123     <section>
124         <h2>Ejercicio 15</h2>
125         <p id="p15"></p>
126         <button onclick="ejercicio15()">Cargar JSON</button>
127     </section>
128
129     <section>
130         <h2>Ejercicio 16</h2>
131         <ul id="ul16"></ul>
132         <button onclick="ejercicio16()">Generar lista JSON</button>
133     </section>
134
135     <section>
136         <h2>Ejercicio 17</h2>
137         <button onclick="ejercicio17()">Guardar usuario</button>
138         <p id="p17"></p>
139     </section>
140
141     <section>
142         <h2>Ejercicio 18</h2>
143         <button onclick="ejercicio18()">Generar tabla libros</button>
144         <div id="div18"></div>
145     </section>
146
147     <section>
148         <h2>Ejercicio 19</h2>
149         <button onclick="ejercicio19()">Modificar JSON</button>
150     </section>
151
152     <section>
153         <h2>Ejercicio 20</h2>
154         <input id="nom20">
155         <button onclick="ejercicio20()">Guardar</button>
156     </section>
```

```

157
158     <section>
159         <h2>Ejercicio 21</h2>
160         <ul id="ul21"></ul>
161         <button onclick="ejercicio21()">Generar tareas</button>
162     </section>
163
164     <section>
165         <h2>Ejercicio 22</h2>
166         <input id="n22" placeholder="Nombre">
167         <input id="e22" placeholder="Edad">
168         <input id="p22" placeholder="País">
169         <button onclick="ejercicio22_guardar()">Guardar perfil</button>
170         <p id="perfil22"></p>
171     </section>
172
173     <script src="ejercicios/ejercicio03.js"></script>
174     <script src="ejercicios/ejercicio04.js"></script>
175     <script src="ejercicios/ejercicio05.js"></script>
176     <script src="ejercicios/ejercicio06.js"></script>
177     <script src="ejercicios/ejercicio07.js"></script>
178     <script src="ejercicios/ejercicio08.js"></script>
179     <script src="ejercicios/ejercicio09.js"></script>
180     <script src="ejercicios/ejercicio10.js"></script>
181     <script src="ejercicios/ejercicio11.js"></script>
182     <script src="ejercicios/ejercicio12.js"></script>
183     <script src="ejercicios/ejercicio13.js"></script>
184     <script src="ejercicios/ejercicio14.js"></script>
185     <script src="ejercicios/ejercicio15.js"></script>
186     <script src="ejercicios/ejercicio16.js"></script>
187     <script src="ejercicios/ejercicio17.js"></script>
188     <script src="ejercicios/ejercicio18.js"></script>
189     <script src="ejercicios/ejercicio19.js"></script>
190     <script src="ejercicios/ejercicio20.js"></script>
191     <script src="ejercicios/ejercicio21.js"></script>
192     <script src="ejercicios/ejercicio22.js"></script>
193 </body>
194 </html>

```

Ejercicio03

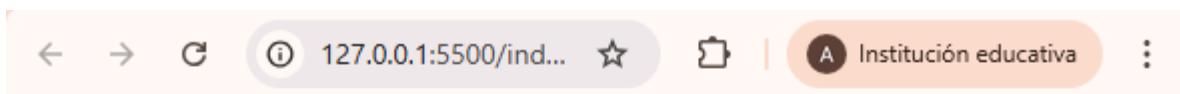
index.html JS ejercicio03.js X JS ejercicio04.js JS ejercicio05.js JS ejercicio06.js JS ejercicio07.js JS ejercicio ...

ejercicios > JS ejercicio03.js > ...

```

1  function ejercicio03(){
2      document.getElementById('p3').textContent='Texto cambiado';
3  }

```



Laboratorio16

Ejercicio 3

Texto original

[Cambiar texto](#)

Ejecución:



Laboratorio16

Ejercicio 3

Texto cambiado

[Cambiar texto](#)

4. Anterior ejercicio, pero que al volver a hacer click vuelva al texto anterior (alternadamente)

// Laboratorio Nro 16 - Ejercicio4

// Autor: Alessandro Josue Justo Vilca

// Colaboró : sin colaboradores

// Tiempo : 16min

```
index.html      JS ejercicio03.js      JS ejercicio04.js X      JS ejercicio05.js      JS ejercicio06.js      JS ejercicio07.js      JS ejercicio □ ...  
ejercicios > JS ejercicio04.js > ...  
1  let t4=false;  
2  function ejercicio04(){  
3    t4=!t4;  
4    document.getElementById('p4').textContent=t4?'Texto cambiado':'Texto original';  
5  }
```

Ejercicio 4

Texto original

[Alternar texto](#)

Ejecución:

Ejercicio 4

Texto cambiado

[Alternar texto](#)

Ejercicio 4

Texto original

Alternar texto

5. Manipular clases CSS: que un botón active/desactive una clase “oscuro” en body para simular modo oscuro

// Laboratorio Nro 16 - Ejercicio5

// Autor: Alessandro Josue Justo Vilca

// Colaboró : sin colaboradores

// Tiempo : 16min

```
index.html      JS ejercicio03.js      JS ejercicio04.js      JS ejercicio05.js X      JS ejercicio06.js      JS ejercicio07.js      JS ejercicio  ⌂ ...  
ejercicios > JS ejercicio05.js > ...  
1  function ejercicio05(){  
2  |    document.body.classList.toggle('oscuro');  
3  }
```

Ejercicio 5

Modo oscuro

Ejecución:

Ejercicio 5

Modo oscuro

6. Contador interactivo: crear botones “+”, “-” y un span que muestre el valor actual (empezar en 0) Evita que el contador baje de cero y mostrar mensaje

// Laboratorio Nro 16 - Ejercicio6

// Autor: Alessandro Josue Justo Vilca

// Colaboró : sin colaboradores

// Tiempo : 16min

```
index.html      JS ejercicio03.js      JS ejercicio04.js      JS ejercicio05.js      JS ejercicio06.js X      JS ejercicio07.js      JS ejercicio  ⌂ ...  
ejercicios > JS ejercicio06.js > ...  
1  let c6=0;  
2  function ejercicio06_mas(){  
3  |    c6++;document.getElementById('contador6').textContent=c6;  
4  }  
5  function ejercicio06_menos(){  
6  |    if(c6>0){  
7  |    |    c6--;  
8  |    }  
9  |    document.getElementById('contador6').textContent=c6;  
10 }
```

Ejercicio 6

0

Ejecución:

Ejercicio 6

- 6 +

7. Agregar y eliminar elementos de una lista: permitir al usuario ingresar texto en un <input> y añadirlo como dentro de una lista. Incluir un botón para borrar el último elemento

// Laboratorio Nro 16 - Ejercicio7

// Autor: Alessandro Josue Justo Vilca

// Colaboró : sin colaboradores

// Tiempo : 17min

The screenshot shows a browser developer tools interface with several tabs at the top: index.html, ejercicio03.js, ejercicio04.js, ejercicio05.js, ejercicio06.js, ejercicio07.js (selected), ejercicio08.js, and ejercicio09.js. Below the tabs, the code for ejercicio07.js is displayed:

```
ejercicios > JS ejercicio07.js > ...
1 function ejercicio07_agregar(){
2     let v=document.getElementById('input7').value;
3     if(v){
4         let li=document.createElement('li');
5         li.textContent=v;
6         document.getElementById('lista7').appendChild(li);
7     }
8 }function ejercicio07_borrar(){
9     let l=document.getElementById('lista7');if(l.lastChild)l.removeChild(l.lastChild);
10 }
```

Ejercicio 7

Ejecución:

Ejercicio 7

- Frutas
- Frutas
- Verduras

8. Validar formulario con DOM: validar que los campos “nombre” y “correo” no estén vacíos. Si hay error, mostrar mensajes debajo del input usando createElement("span") con estilo rojo

// Laboratorio Nro 16 - Ejercicio8

// Autor: Alessandro Josue Justo Vilca

// Colaboró : sin colaboradores

// Tiempo : 18min

The screenshot shows a browser developer tools interface with several tabs at the top: ex.html, ejercicio03.js, ejercicio04.js, ejercicio05.js, ejercicio06.js, ejercicio07.js, ejercicio08.js (selected), ejercicio09.js, and ejercicio10.js. Below the tabs, the code for ejercicio08.js is displayed:

```
ejercicios > JS ejercicio08.js > ...
1 function ejercicio08(){
2     let n=document.getElementById('nom8').value;
3     let c=document.getElementById('cor8').value;
4     let e=document.getElementById('err8');
5     e.innerHTML='';
6     if(!n)e.innerHTML+='
```

Ejercicio 8

Nombre
Correo
<input type="button" value="Validar"/>

Ejecución:

Ejercicio 8

Juan
juan@gmail.com
<input type="button" value="Validar"/>

9. Galería con miniaturas: crear una galería donde al hacer clic en una miniatura (img) se muestre la imagen grande en un <div> principal

// Laboratorio Nro 16 - Ejercicio9

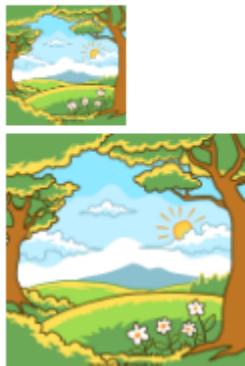
// Autor: Alessandro Josue Justo Vilca

// Colaboró : sin colaboradores

// Tiempo : 15min

```
s JS ejercicio05.js JS ejercicio06.js JS ejercicio07.js JS ejercicio08.js JS ejercicio09.js X JS ejercicio10.js JS ...  
ejercicios > JS ejercicio09.js > ...  
1 function ejercicio09(img){  
2 |   document.getElementById('img9').src=img.src;  
3 }
```

Ejercicio 9



Ejecución:

Ejercicio 9



10. Tabla dinámica desde JavaScript: generar una tabla HTML al presionar un botón, a partir de un arreglo de objetos
const productos = [

```

{ nombre: "Laptop", precio: 3500 },
{ nombre: "Mouse", precio: 80 },
];
// Laboratorio Nro 16 - Ejercicio10
// Autor: Alessandro Josue Justo Vilca
// Colaboró : sin colaboradores
// Tiempo : 19min

```

```

ejercicios > JS ejercicio10.js > ...
1  function ejercicio10(){
2      const productos=[{nombre:'Laptop',precio:3500},{nombre:'Mouse',precio:80}];
3      let h='<table border=1>';
4      productos.forEach(p=>h+=`<tr><td>${p.nombre}</td><td>${p.precio}</td></tr>`);
5      h+='</table>';
6      document.getElementById('tabla10').innerHTML=h;
7  }

```

Ejercicio 10

[Generar tabla](#)

Ejecución:

Ejercicio 10

Generar tabla	
Laptop	3500
Mouse	80

11. Delegación de eventos: crear una lista que permita eliminar cualquier al hacer clic sobre él, sin asignar eventos individualmente. Usa event.target dentro del listener del

```

// Laboratorio Nro 16 - Ejercicio11
// Autor: Alessandro Josue Justo Vilca
// Colaboró : sin colaboradores
// Tiempo : 18min

```

```

ejercicios > JS ejercicio11.js > ...
1  document.getElementById('ul11').addEventListener('click',e=>{
2      if(e.target.tagName==='LI')e.target.remove();
3  });
4
5

```

Ejercicio 11

- Item A
- Item B
- Item C

Ejecución:

Ejercicio 11

Ejercicio 12

12. Animación con DOM y CSS: hacer que al presionar un botón, un cuadrado (`<div>`) se mueva horizontalmente usando `classList.add("mover")` y una animación CSS. Agrega un botón "Reiniciar" que quite la clase

// Laboratorio Nro 16 - Ejercicio12
// Autor: Alessandro Josue Justo Vilca
// Colaboró : sin colaboradores
// Tiempo : 20min

```
$ ejercicio08.js X  JS ejercicio09.js  JS ejercicio10.js  JS ejercicio11.js  JS ejercicio12.js X  JS ejercicio13.js  JS ejercicio...  ...
ejercicios > JS ejercicio12.js > ...
1 function ejercicio12(){
2     document.getElementById('cuadro12').classList.add('mover')
3     ;
4 }
5 function ejercicio12_reset(){
6     document.getElementById('cuadro12').classList.remove('mover');
7 }
8 }
```

Ejercicio 12



Ejecución:

Ejercicio 12



13. Construir una pequeña interfaz CRUD (Crear, Leer, Actualizar y Eliminar)

- Formulario para añadir usuarios (nombre, edad)
- Tabla que muestra los usuarios agregados
- Botones de editar y eliminar por fila
- Usa `querySelector`, `appendChild`, `dataset` y `event delegation`

// Laboratorio Nro 16 - Ejercicio13
// Autor: Alessandro Josue Justo Vilca
// Colaboró : sin colaboradores
// Tiempo : 18min

ejercicios > JS ejercicio13.js > ...

```

1  function ejercicio13_add(){
2      let n=document.getElementById('n13').value;
3      let e=document.getElementById('e13').value;
4      let tr=document.createElement('tr');
5      tr.innerHTML=`<td>${n}</td><td>${e}</td><td><button class='edit'>E</button><button class='del'>X</button></td>`;
6      document.getElementById('t13').appendChild(tr);
7  }
8  document.getElementById('t13').addEventListener('click',ev=>{
9      let tr=ev.target.closest('tr');if(ev.target.className=='del')tr.remove();
10 });

```

Ejercicio 13

Nombre: Edad: Agregar

Nombre	Edad	Acciones

Ejecución:

Ejercicio 13

Nombre: Lucas Edad: 14 Agregar

Nombre	Edad	Acciones
Juan	12	E X
Lucas	14	E X

14. Declara un objeto persona con nombre, edad y ciudad. Convierte el objeto a JSON con JSON.stringify() y muéstralos en consola

// Laboratorio Nro 16 - Ejercicio14

// Autor: Alessandro Josue Justo Vilca

// Colaboró : sin colaboradores

// Tiempo : 15min

ejercicios > JS ejercicio14.js > ...

```

1  function ejercicio14(){
2      let persona={nombre:'Ana',edad:20,ciudad:'Arequipa'};
3      console.log(JSON.stringify(persona));
4  }

```

Ejercicio 14

Ejecución:

The screenshot shows a browser developer tools console with the following error message:

```

Content      content-scripts.js:1
Script Bridge: Sending response
back to page context:  ▶ Object
13 {"nombre":"Ana", ejercicio14.js:3
  "edad":20,"ciudad":"Arequipa"}
>

```

15. Declara una cadena JSON y conviértela en objeto con JSON.parse(). Muestra uno de sus valores en el DOM (por ejemplo, dentro de un <p>)

// Laboratorio Nro 16 - Ejercicio15

// Autor: Alessandro Josue Justo Vilca

// Colaboró : sin colaboradores

// Tiempo : 15min

```

ejercicios > JS ejercicio15.js > ...
1  function ejercicio15(){
2    let cad='{"curso":"DOM","anio":2025}';
3    let o=JSON.parse(cad);
4    document.getElementById('p15').textContent=o.curso;
5  }

```

Ejercicio 15

[Cargar JSON](#)

Ejecución:

Ejercicio 15

DOM

[Cargar JSON](#)

16. Crea un arreglo de objetos productos (nombre, precio). Convierte a JSON y luego vuelve a objeto.

Muestra los nombres en una lista generada dinámicamente

// Laboratorio Nro 16 - Ejercicio16

// Autor: Alessandro Josue Justo Vilca

// Colaboró : sin colaboradores

// Tiempo : 18min

```

ejercicios > JS ejercicio16.js > ...
1  function ejercicio16(){
2    let arr=[{nombre:'A',precio:1},{nombre:'B',precio:2}];
3    let json=JSON.stringify(arr);let obj=JSON.parse(json);
4    let ul=document.getElementById('ul16');
5    ul.innerHTML='';
6    obj.forEach(p=>{
7      let li=document.createElement('li');
8      li.textContent=p.nombre;ul.appendChild(li);
9    });
10 }

```

Ejercicio 16

[Generar lista JSON](#)

Ejecución:

Ejercicio 16

- A
- B

[Generar lista JSON](#)

17. Guarda un objeto JSON con datos de usuario (nombre, correo, rol) en localStorage. Luego recupéralo y muéstralos en pantalla

// Laboratorio Nro 16 - Ejercicio17

// Autor: Alessandro Josue Justo Vilca

// Colaboró : sin colaboradores

// Tiempo : 17min

The screenshot shows a browser's developer tools console with multiple tabs open. The active tab is 'ejercicio17.js'. The code within the tab is as follows:

```
1 function ejercicio17(){
2     let u={nombre:'Luis',correo:'a@b.com',rol:'admin'};
3     localStorage.setItem('u17',JSON.stringify(u));
4     document.getElementById('p17').textContent=JSON.stringify(u);
5 }
```

Ejercicio 17

[Guardar usuario](#)

Ejecución:

Ejercicio 17

[Guardar usuario](#)

{"nombre":"Luis","correo":"a@b.com","rol":"admin"}

18. Define una variable con un JSON que simule una lista de libros. Parsea el JSON y genera una tabla con sus títulos y autores usando el DOM

// Laboratorio Nro 16 - Ejercicio18

// Autor: Alessandro Josue Justo Vilca

// Colaboró : sin colaboradores

// Tiempo : 19min

```
io16.js      JS ejercicio17.js      JS ejercicio18.js X      JS ejercicio19.js      JS ejercicio20.js      JS ejercicio21.js      JS ejercicio22.js
ejercicios > JS ejercicio18.js > ...
1  function ejercicio18(){
2    let json='[{"titulo":"A","autor":"X"}, {"titulo":"B","autor":"Y"}]';
3    let arr=JSON.parse(json);
4    let h='<table border=1>';
5    arr.forEach(l=>h+=`<tr><td>${l.titulo}</td><td>${l.autor}</td></tr>`);
6    h+='</table>';
7    document.getElementById('div18').innerHTML=h;
8  }
9
10 |
```

Ejercicio 18

Generar tabla libros

Ejecución:

Ejercicio 18

Generar tabla libros

A	X
B	Y

19. Convierte un objeto JSON en objeto JS, modifica un valor (por ejemplo: edad o precio), y vuelve a convertirlo a JSON actualizado

// Laboratorio Nro 16 - Ejercicio19

// Autor: Alessandro Josue Justo Vilca

// Colaboró : sin colaboradores

// Tiempo : 18min

```
io16.js      JS ejercicio17.js      JS ejercicio18.js X      JS ejercicio19.js      JS ejercicio20.js      JS ejercicio21.js      JS ejercicio22.js
ejercicios > JS ejercicio19.js > ...
1  function ejercicio19(){
2    let o={nombre:'Ana',edad:20};o.edad=21;console.log(JSON.stringify(o));
3  }
4
5 |
```

Ejercicio 19

Modificar JSON

Ejecución:

The screenshot shows a browser's developer tools console. At the top, there are tabs for 'Consola' (selected), 'top', and 'Filtrar'. Below the tabs, there are buttons for 'Niveles predeterminados' and 'Sin problemas'. A gear icon indicates settings. The main area displays the following JSON object:

```
3 {"nombre": "Ana", "edad": 21}
```

20. Cuando el usuario escriba su nombre en un <input> y haga clic en un botón, guarda los datos en un objeto, conviértelo a JSON y muéstralos en consola

// Laboratorio Nro 16 - Ejercicio20

// Autor: Alessandro Josue Justo Vilca

// Colaboró : sin colaboradores

// Tiempo : 18min

The screenshot shows a browser's developer tools console with several tabs at the top: 'ejercicio16.js', 'ejercicio17.js', 'ejercicio18.js', 'ejercicio19.js', 'ejercicio20.js' (selected), 'ejercicio21.js', 'ejercicio22.js', and a '...' tab. The main area contains the following JavaScript code:

```
function ejercicio20(){
    let n=document.getElementById('nom20').value;
    let o={nombre:n};
    console.log(JSON.stringify(o));
}
```

Ejercicio 20

Ejecución:

The screenshot shows a browser's developer tools console. At the top, there are tabs for 'Consola' (selected), 'top', and 'Filtrar'. Below the tabs, there are buttons for 'Niveles predeterminados' and 'Sin problemas'. A gear icon indicates settings. The main area displays the following JSON objects:

```
{"nombre": "Pepe"} ejercicio20.js:4
{"nombre": "Albert"} ejercicio20.js:4
{"nombre": "Juan"} ejercicio20.js:4
```

21. Define un JSON con una lista de tareas (título, completada). Crea dinámicamente una lista en el DOM que muestre el estado de cada tarea (por ejemplo, con color verde si está completada)

// Laboratorio Nro 16 - Ejercicio21

// Autor: Alessandro Josue Justo Vilca

// Colaboró : sin colaboradores

// Tiempo : 20min

```

io16.js      JS ejercicio17.js    JS ejercicio18.js    JS ejercicio19.js    JS ejercicio20.js    JS ejercicio21.js X  JS ejercicio22.js    ...
ejercicios > JS ejercicio21.js > ...
1  function ejercicio21(){
2      let t=[{t:'A',c:true},{t:'B',c:false}];
3      let ul=document.getElementById('ul21');
4      ul.innerHTML='';
5      t.forEach(x=>{let li=document.createElement('li');
6          li.textContent=x.t;if(x.c)li.classList.add('completada');
7          ul.appendChild(li);
8      });
9  }

```

Ejercicio 21

[Generar tareas](#)

Ejecución:

Ejercicio 21

- A
- B

[Generar tareas](#)

22. Simulador de perfil

- Crea un formulario con nombre, edad y país
- Al enviar, guarda los datos como JSON en localStorage
- Si el usuario recarga la página, muestra el perfil guardado en pantalla

// Laboratorio Nro 16 - Ejercicio22

// Autor: Alessandro Josue Justo Vilca

// Colaboró : sin colaboradores

// Tiempo : 18min

```

io16.js      JS ejercicio17.js    JS ejercicio18.js    JS ejercicio19.js    JS ejercicio20.js    JS ejercicio21.js    JS ejercicio22.js X  ...
ejercicios > JS ejercicio22.js > ...
1  function ejercicio22_guardar(){
2      let p={n:document.getElementById('n22').value,e:document.getElementById('e22').value,p:document.
3          getElementById('p22').value};
4      localStorage.setItem('p22',JSON.stringify(p));
5      document.getElementById('perfil22').textContent=JSON.stringify(p);
6  }
7  window.onload=()=>{let x=localStorage.getItem('p22');
8      if(x)document.getElementById('perfil22').textContent=x;
9  }

```

Ejercicio 22

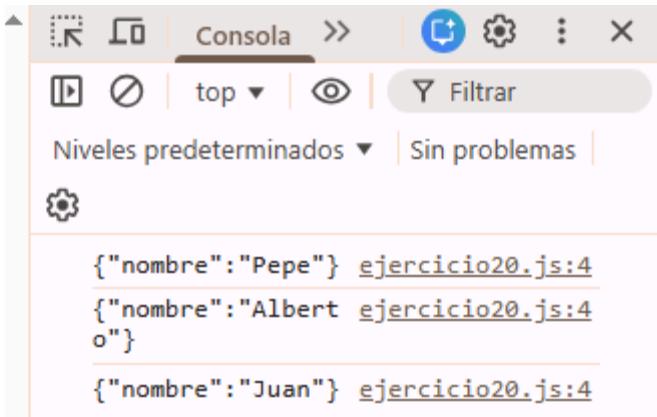
Nombre	Edad	País
Guardar perfil		

Ejecución:

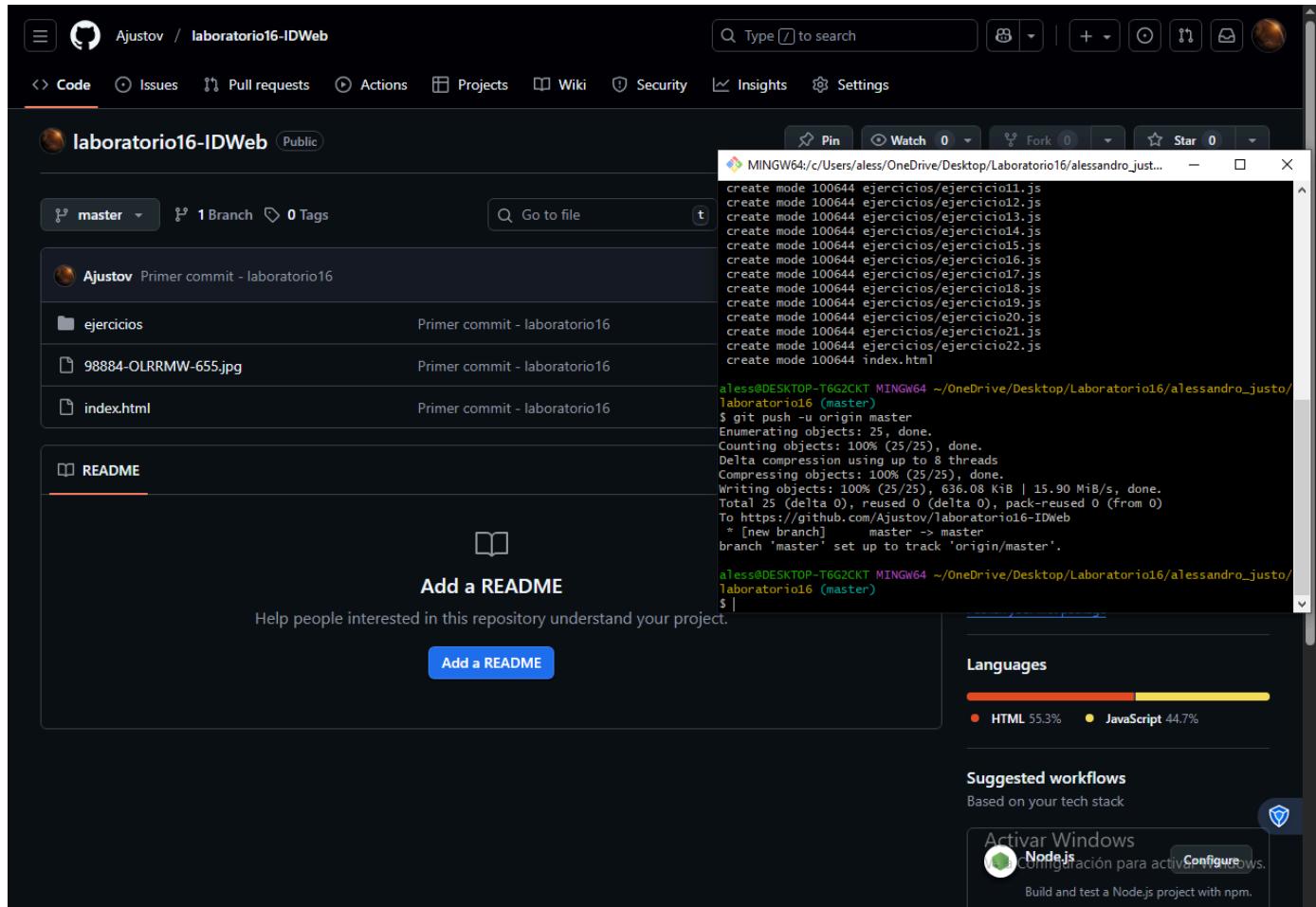
Ejercicio 22

Lucas	
18	
Argentina	Guardar perfil

{"n": "Lucas", "e": "18", "p": "Argentina"}



23. Crear un repositorio remoto en GitHub y subir tu repositorio local. Compartir URL y pdf con captura de pantalla del código de los archivos js y de la ejecución



The screenshot shows a GitHub repository page for 'laboratorio16-IDWeb'. The repository has 1 branch and 0 tags. The commit history shows a single commit from 'Ajustov' titled 'Primer commit - laboratorio16'. The commit message is 'Primer commit - laboratorio16'. The commit details show the creation of files like 'ejercicios', '98884-OLRRMW-655.jpg', and 'index.html'. A terminal window is overlaid on the right side of the page, showing the command line activity:

```
alessandro@DESKTOP-T6G2CKT MINGW64 ~/OneDrive/Desktop/Laboratorio16/alessandro_justo/laboratorio16 (master)
$ git push -u origin master
Enumerating objects: 25, done.
Counting objects: 100% (25/25), done.
Delta compression using up to 8 threads
Compressing objects: 100% (25/25), done.
Writing objects: 100% (25/25), 636.08 KiB | 15.90 MiB/s, done.
Total 25 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Ajustov/laboratorio16-IDWeb
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
alessandro@DESKTOP-T6G2CKT MINGW64 ~/OneDrive/Desktop/Laboratorio16/alessandro_justo/laboratorio16 (master)
$ |
```

Below the terminal, there are sections for 'Languages' (HTML 55.3%, JavaScript 44.7%) and 'Suggested workflows' (Activar Windows, Node.js).

URL: <https://github.com/Ajustov/laboratorio16-IDWeb>