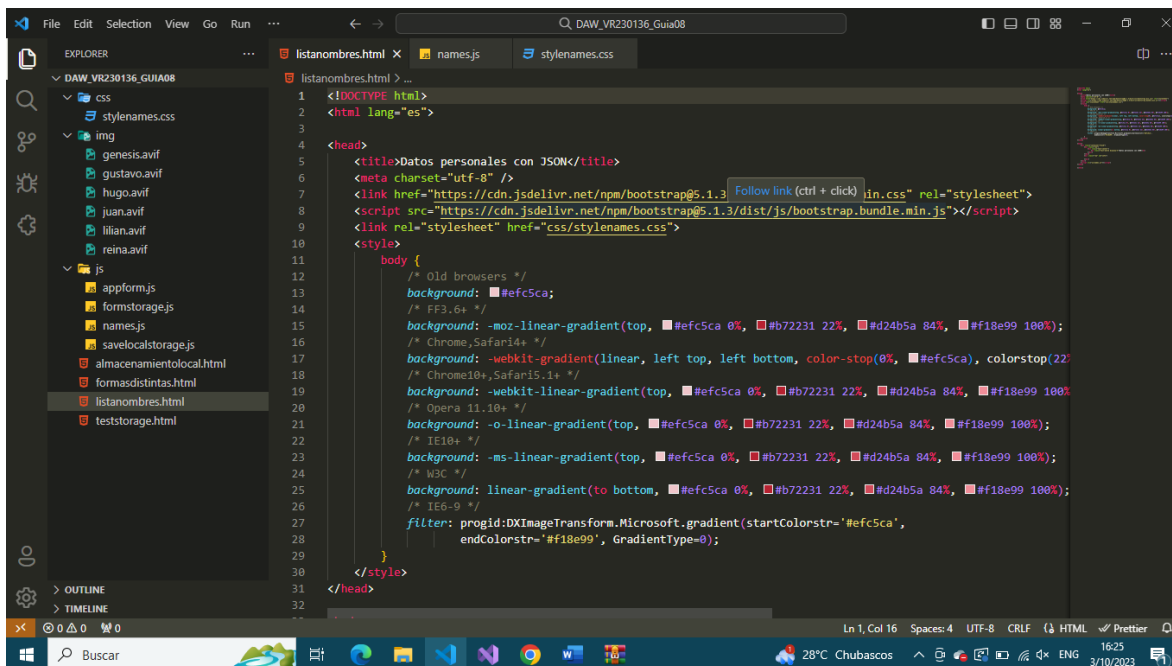


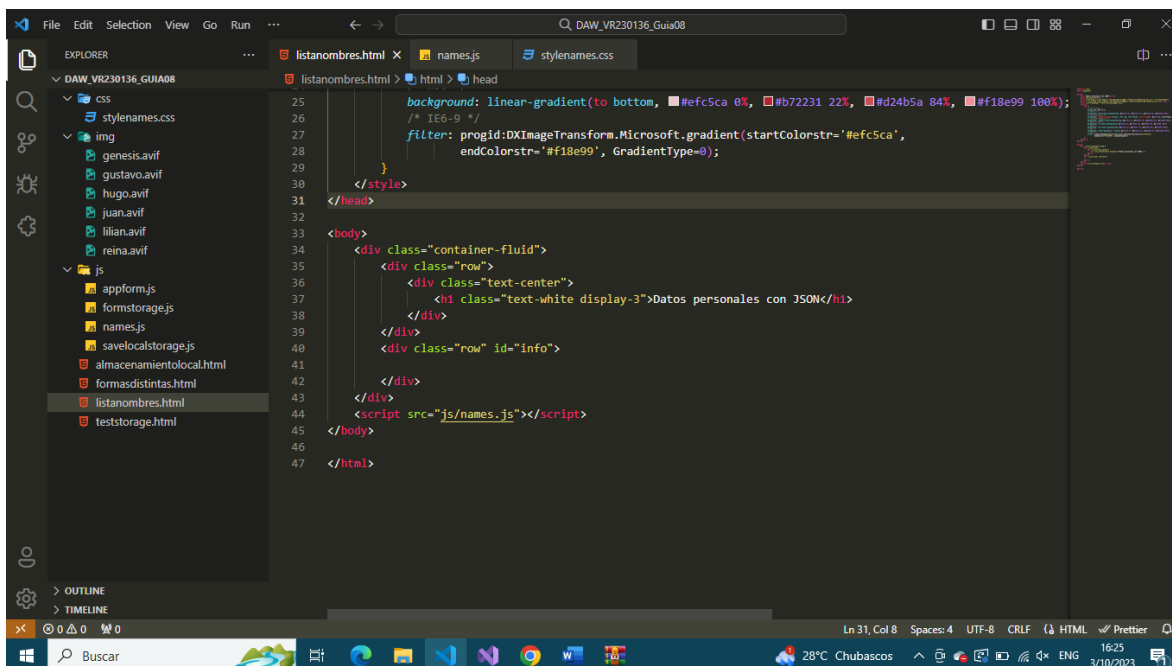
JENNIFER PATRICIA ZALDAÑA BELTRAN
ZB212784

EJ1 HTML



```
1 <!DOCTYPE html>
2 <html lang="es">
3
4
5 <head>
6   <title>Datos personales con JSON</title>
7   <meta charset="utf-8" />
8   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
9   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
10  <link rel="stylesheet" href="css/stylenames.css">
11  <style>
12    /* Old browsers */
13    background: #efc5ca;
14    /* FF3.6+ */
15    background: -moz-linear-gradient(top, #efc5ca 0%, #b72231 22%, #d24b5a 84%, #f18e99 100%);
16    /* Chrome,Safari4+ */
17    background: -webkit-linear-gradient(linear, left top, left bottom, color-stop(0%, #efc5ca), color-stop(22%
18    /* Chrome10+,Safari5.1+ */
19    background: -webkit-linear-gradient(top, #efc5ca 0%, #b72231 22%, #d24b5a 84%, #f18e99 100%
20    /* Opera 11.10+ */
21    background: -o-linear-gradient(top, #efc5ca 0%, #b72231 22%, #d24b5a 84%, #f18e99 100%);
22    /* IE10+ */
23    background: -ms-linear-gradient(top, #efc5ca 0%, #b72231 22%, #d24b5a 84%, #f18e99 100%);
24    /* W3C */
25    background: linear-gradient(to bottom, #efc5ca 0%, #b72231 22%, #d24b5a 84%, #f18e99 100%);
26    /* IE6-9 */
27    filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#efc5ca',
28              endColorstr='#f18e99', GradientType=0);
29  </style>
30 </head>
31
32
```

EJ1 HTML

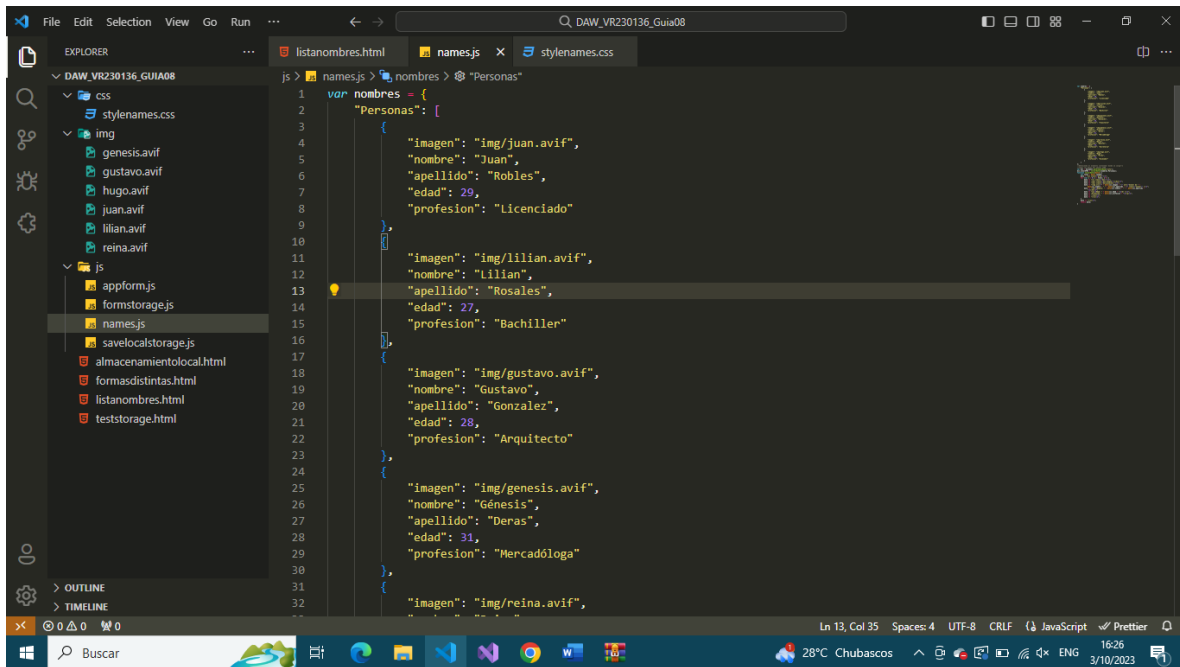


```
25 background: linear-gradient(to bottom, #efc5ca 0%, #b72231 22%, #d24b5a 84%, #f18e99 100%);
26 /* IE6-9 */
27 filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#efc5ca',
28              endColorstr='#f18e99', GradientType=0);
29
30 </style>
31 </head>
32
33 <body>
34   <div class="container-fluid">
35     <div class="row">
36       <div class="text-center">
37         <h1 class="text-white display-3">Datos personales con JSON</h1>
38       </div>
39     </div>
40     <div class="row" id="info">
41
42     </div>
43   </div>
44   <script src="js/names.js"></script>
45 </body>
46
47 </html>
```

DAW06L
GUIA 8

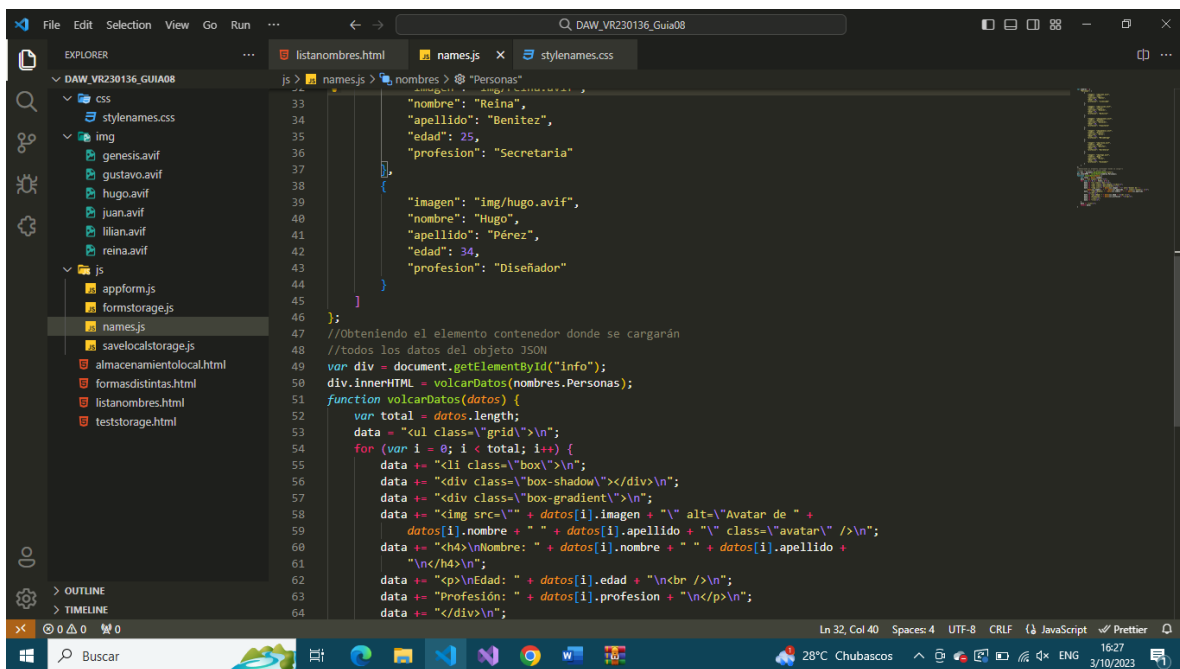
JENNIFER PATRICIA ZALDAÑA BELTRAN
ZB212784

EJ1 JS



```
1 var nombres = {
2   "Personas": [
3     {
4       "imagen": "img/juan.avif",
5       "nombre": "Juan",
6       "apellido": "Robles",
7       "edad": 29,
8       "profesion": "Licenciado"
9     },
10    {
11      "imagen": "img/lilian.avif",
12      "nombre": "Lilian",
13      "apellido": "Rosales",
14      "edad": 27,
15      "profesion": "Bachiller"
16    },
17    {
18      "imagen": "img/gustavo.avif",
19      "nombre": "Gustavo",
20      "apellido": "Gonzalez",
21      "edad": 28,
22      "profesion": "Arquitecto"
23    },
24    {
25      "imagen": "img/genesis.avif",
26      "nombre": "Génesis",
27      "apellido": "Deras",
28      "edad": 31,
29      "profesion": "Mercadóloga"
30    },
31    {
32      "imagen": "img/reina.avif",
```

EJ1 JS

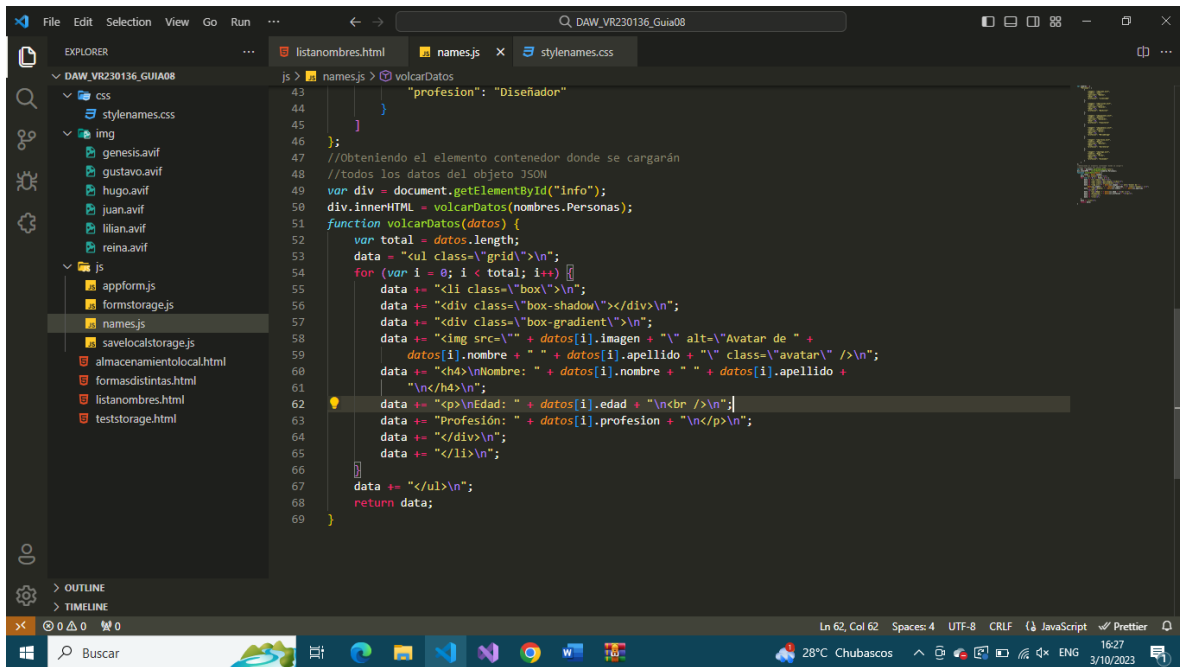


```
33 "nombre": "Reina",
34 "apellido": "Benitez",
35 "edad": 25,
36 "profesion": "Secretaria"
37 },
38 {
39   "imagen": "img/hugo.avif",
40   "nombre": "Hugo",
41   "apellido": "Pérez",
42   "edad": 34,
43   "profesion": "Diseñador"
44 }
45 ]
46 };
47 //Obteniendo el elemento contenedor donde se cargarán
48 //todos los datos del objeto JSON.
49 var div = document.getElementById("Info");
50 div.innerHTML = volcarDatos(nombres.Personas);
51 function volcarDatos(datos) {
52   var total = datos.length;
53   data = "<ul class='grid'>\n";
54   for (var i = 0; i < total; i++) {
55     data += "<li class='box'>\n";
56     data += "<div class='box-shadow'></div>\n";
57     data += "<div class='box-gradient'>\n";
58     data += "<img src='\" + datos[i].imagen + \"' alt='Avatar de \" + "
59     datos[i].nombre + " + datos[i].apellido + "\" class='avatar' />\n";
60     data += "<h4>Nombre: \" + datos[i].nombre + \" + datos[i].apellido + "
61     "\"</h4>\n";
62     data += "<p>Edad: \" + datos[i].edad + \"<br />\n";
63     data += "Profesión: \" + datos[i].profesion + \"</p>\n";
64     data += "</div>\n";
```

DAW06L
GUIA 8

JENNIFER PATRICIA ZALDAÑA BELTRAN
ZB212784

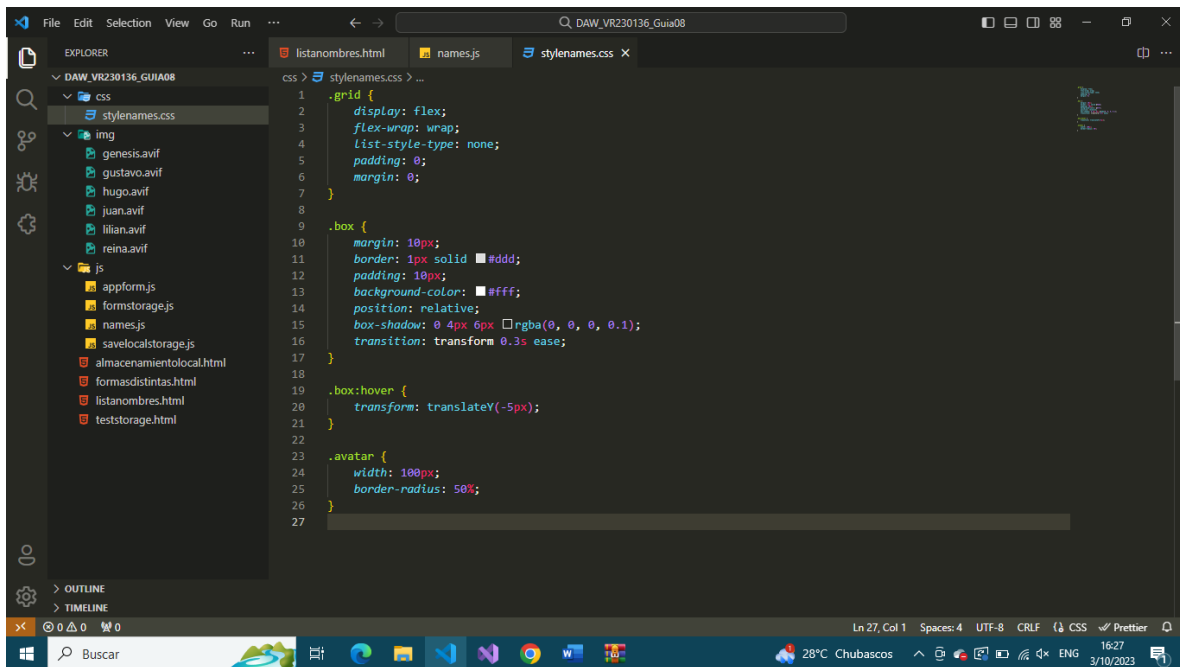
EJ1 JS



The screenshot shows a Visual Studio Code editor with a dark theme. The Explorer sidebar on the left shows a project named 'DAW_VR230136_GUIA08' with folders for 'css', 'img', and 'js'. The 'js' folder is expanded, showing files like 'appform.js', 'formstorage.js', 'names.js', 'saveLocalStorage.js', 'almacenamientoLocal.html', 'formasdistintas.html', 'listanombres.html', and 'teststorage.html'. The 'listanombres.html' file is open in the editor, displaying JavaScript code. The code defines a 'volcarDatos' function that takes an array of objects and returns a string of HTML. The function iterates over the array, creating a list of boxes. Each box contains an image, a name, an age, and a profession. The code uses template literals and DOM manipulation to insert the generated HTML into a container with the ID 'info'. The status bar at the bottom indicates the cursor is at line 62, column 62, with 4 spaces, UTF-8 encoding, CRLF line endings, and JavaScript language mode.

```
43         "profesion": "Diseñador"
44     }
45 }
46
47 //Obteniendo el elemento contenedor donde se cargarán
48 //todos los datos del objeto JSON.
49 var div = document.getElementById("info");
50 div.innerHTML = volcarDatos(nombres.Personas);
51 function volcarDatos(datos) {
52     var total = datos.length;
53     data = "<ul class='grid'>\n";
54     for (var i = 0; i < total; i++) {
55         data += "<li class='box'>\n";
56         data += "<div class='box-shadow'></div>\n";
57         data += "<div class='box-gradient'>\n";
58         data += "<img src='\" + datos[i].imagen + \"' alt='Avatar de \" +
59             datos[i].nombre + \" \" + datos[i].apellido + \"' class='avatar' />\n";
60         data += "<h4>\nNombre: \" + datos[i].nombre + \" \" + datos[i].apellido +
61             "\n</h4>\n";
62         data += "<p>\nEdad: \" + datos[i].edad + \"\n<br />\n";
63         data += "Profesión: \" + datos[i].profesion + "\n<p>\n";
64         data += "</div>\n";
65         data += "</li>\n";
66     }
67     data += "</ul>\n";
68     return data;
69 }
```

CSS EJ1



The screenshot shows the same Visual Studio Code editor with the 'stylesnames.css' file open. The CSS code defines three classes: '.grid', '.box', and '.box:hover'. The '.grid' class is a flex container with a wrap. The '.box' class has a margin, a solid border, padding, a white background, relative positioning, a box shadow, and a transition. The '.box:hover' class adds a translateY transform. The '.avatar' class has a width of 100px and a border radius of 50%. The status bar at the bottom indicates the cursor is at line 27, column 1, with 4 spaces, UTF-8 encoding, CRLF line endings, and CSS language mode.

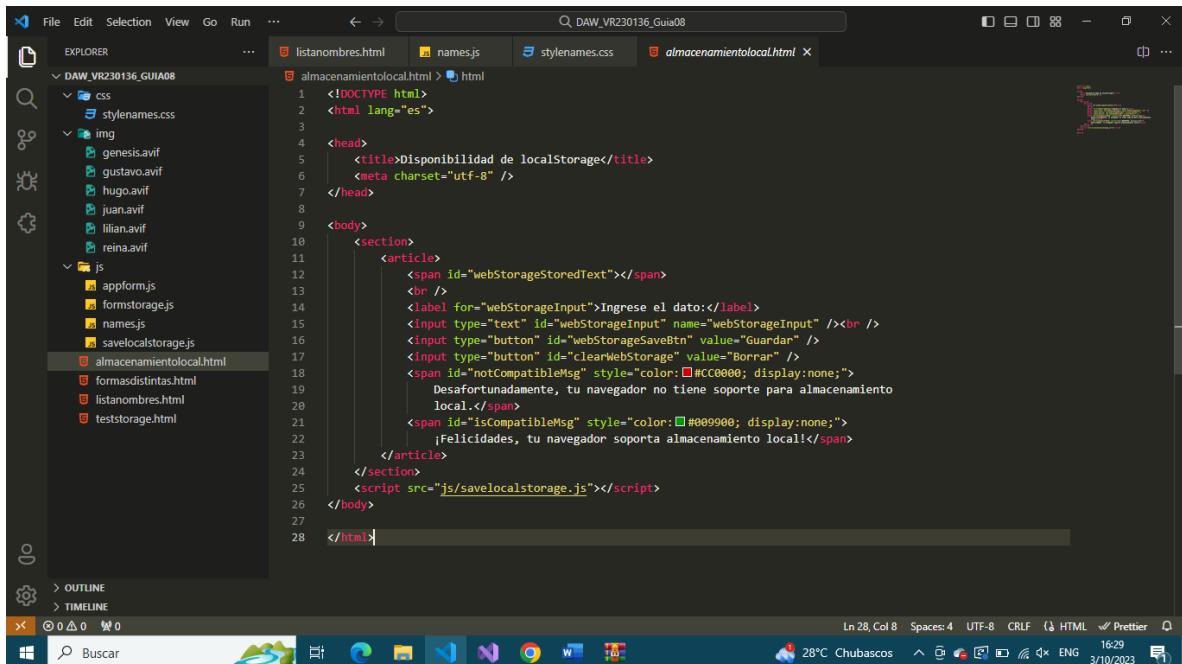
```
1 .grid {
2     display: flex;
3     flex-wrap: wrap;
4     list-style-type: none;
5     padding: 0;
6     margin: 0;
7 }
8
9 .box {
10     margin: 10px;
11     border: 1px solid #ddd;
12     padding: 10px;
13     background-color: #fff;
14     position: relative;
15     box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
16     transition: transform 0.3s ease;
17 }
18
19 .box:hover {
20     transform: translateY(-5px);
21 }
22
23 .avatar {
24     width: 100px;
25     border-radius: 50%;
26 }
27
```

DAW06L
GUIA 8

JENNIFER PATRICIA ZALDAÑA BELTRAN
ZB212784



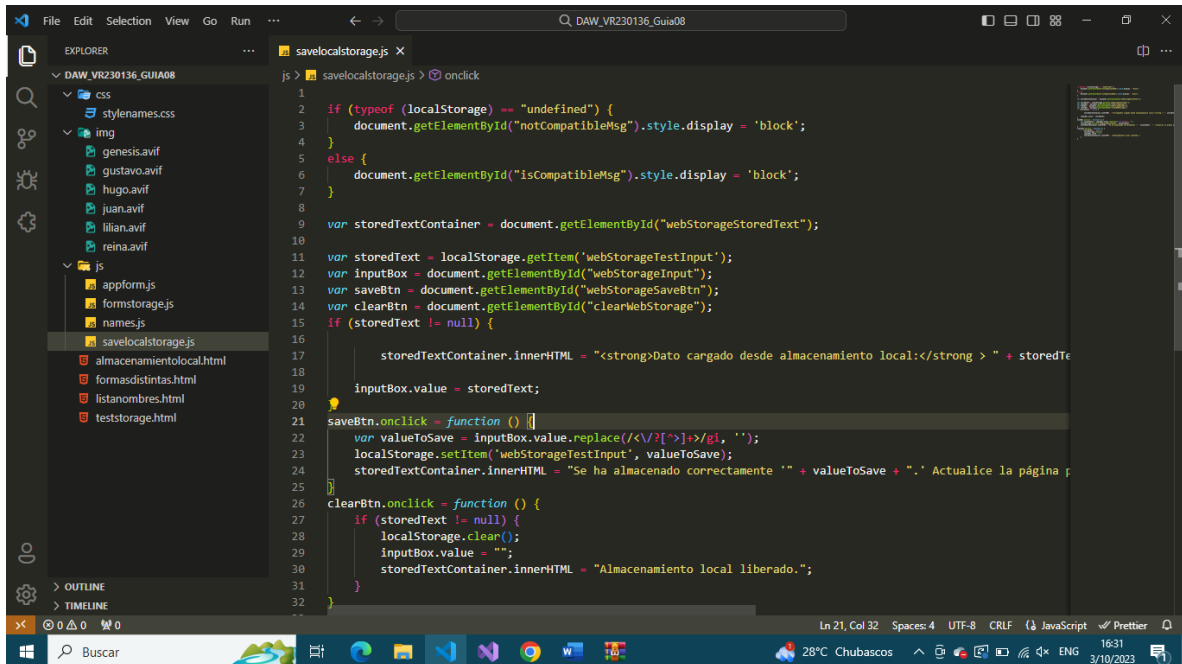
EJ2 HTML



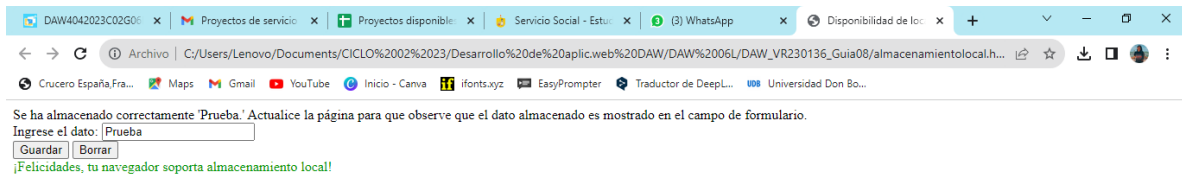
DAW06L
GUIA 8

JENNIFER PATRICIA ZALDAÑA BELTRAN
ZB212784

EJ2 JS

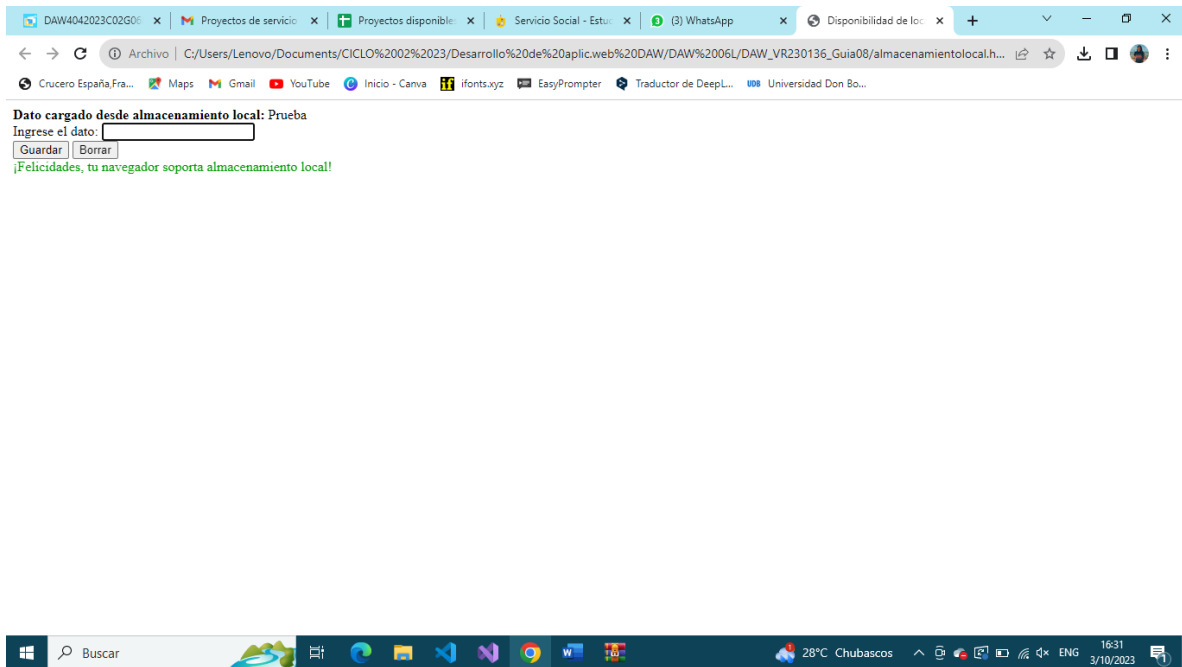


```
1  if (typeof (localStorage) == "undefined") {
2      document.getElementById("notCompatibleMsg").style.display = 'block';
3  }
4  else {
5      document.getElementById("isCompatibleMsg").style.display = 'block';
6  }
7
8  var storedTextContainer = document.getElementById("webStorageStoredText");
9
10
11  var storedText = localStorage.getItem('webStorageTestInput');
12  var inputBox = document.getElementById("webStorageInput");
13  var saveBtn = document.getElementById("webStorageSaveBtn");
14  var clearBtn = document.getElementById("ClearWebStorage");
15  if (storedText != null) {
16
17      storedTextContainer.innerHTML = "<strong>Dato cargado desde almacenamiento local:</strong> " + storedText;
18
19      inputBox.value = storedText;
20  }
21  saveBtn.onclick = function () {
22      var valueToSave = inputBox.value.replace(/<\/?[\>]+>/g, '');
23      localStorage.setItem('webStorageTestInput', valueToSave);
24      storedTextContainer.innerHTML = "Se ha almacenado correctamente '" + valueToSave + "', Actualice la página para que observe el dato almacenado.";
25  }
26  clearBtn.onclick = function () {
27      if (storedText != null) {
28          localStorage.clear();
29          inputBox.value = "";
30          storedTextContainer.innerHTML = "Almacenamiento local liberado.";
31      }
32  }
```

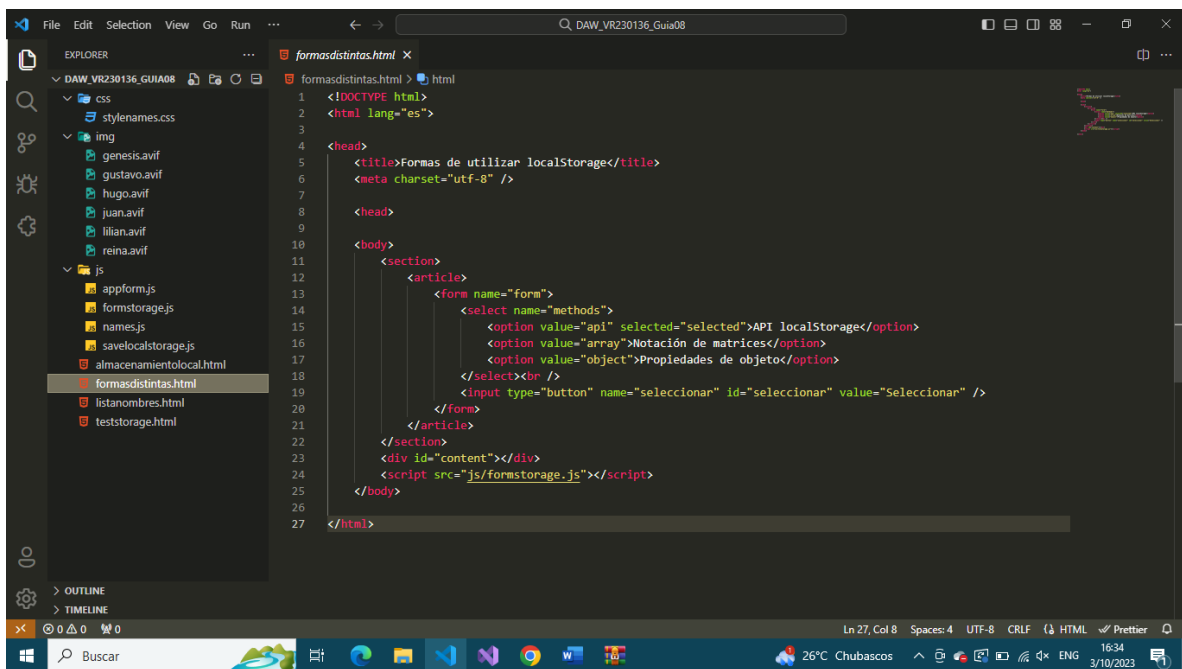


DAW06L
GUIA 8

JENNIFER PATRICIA ZALDAÑA BELTRAN
ZB212784



EJ3 HTML



DAW06L
GUIA 8

JENNIFER PATRICIA ZALDAÑA BELTRAN
ZB212784

EJ3 JS

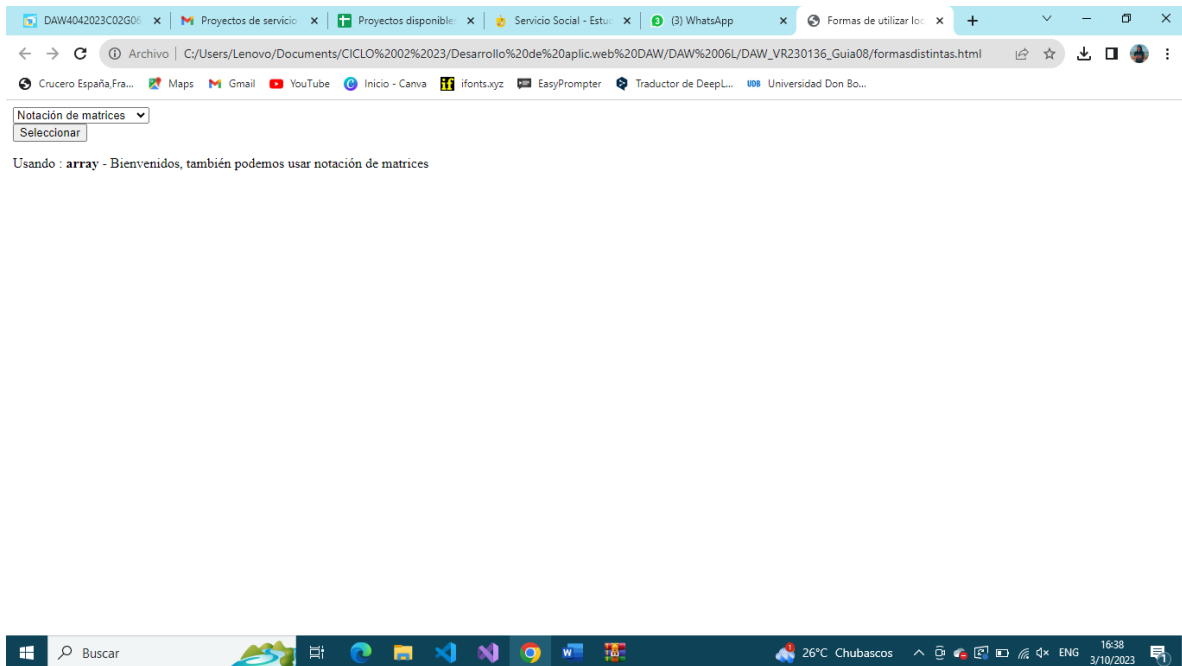
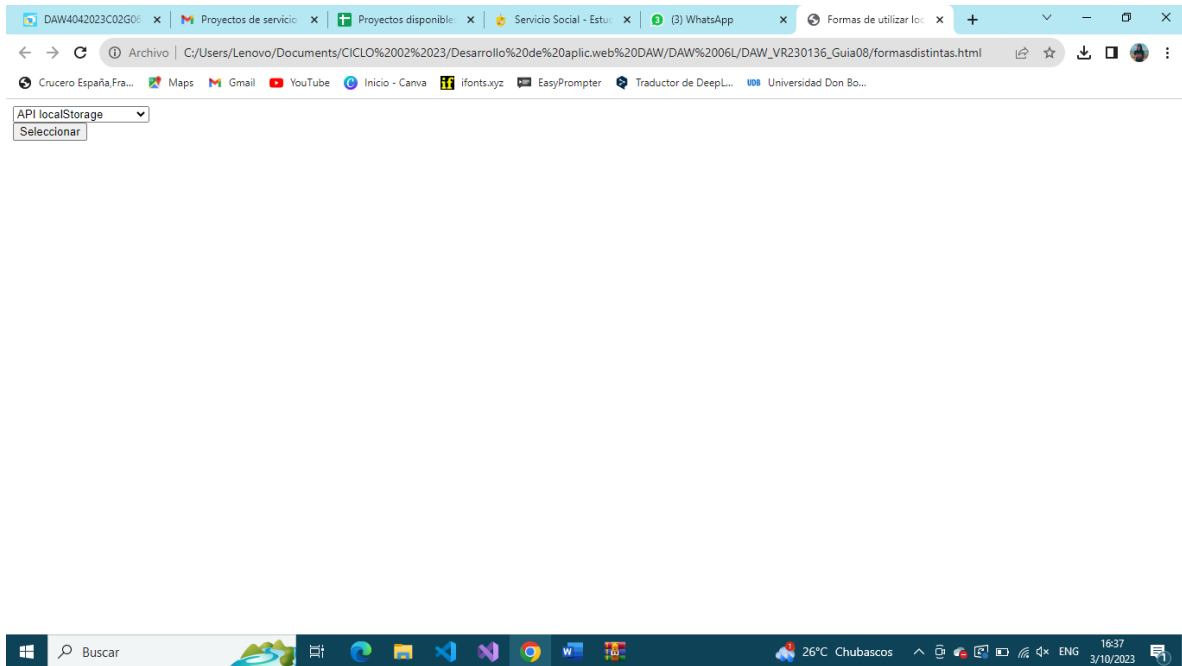
```
1 var valor;  
2 function init(){  
3   var seleccion = document.getElementById("seleccionar");  
4   if(seleccion.addEventListener){  
5     seleccion.addEventListener("click", mostrar, false);  
6   }  
7   else if(seleccion.attachEvent){  
8     seleccion.attachEvent("onclick", mostrar);  
9   }  
10 }  
11 function mostrar(){  
12   //Determinando la opción seleccionada  
13   var opcion =  
14   document.form.methods.options[document.form.methods.selectedIndex].value;  
15   switch(opcion){  
16     case "api":  
17     // Utilizando la Api de Local Storage, podemos guardar y luego recuperar un valor  
18     localStorage.setItem('bienvenida', 'Bienvenidos a localStorage con API de JavaScript');  
19     valor = localStorage.getItem('bienvenida');  
20     break;  
21     case "array":  
22     // Utilizar notación de matrices o arreglos  
23     localStorage['bienvenida'] = 'Bienvenidos, también podemos usar notación de matrices';  
24     valor = localStorage['bienvenida'];  
25     break;  
26     case "object":  
27     // Y claro, también es válido utilizar notación de propiedades de objetos */  
28     localStorage.bienvenida = 'Bienvenidos, igual podemos usarlo como propiedades de objeto';  
29     valor = localStorage.bienvenida;  
30     break;  
31     default:  
32     alert("Esta opción no existe");  
33   }  
34 }  
35 var contenido = document.getElementById("content");  
36 contenido.innerHTML = "<p>\nUsando : <strong>" + opcion + "</strong> - " +  
37 valor + "\n\t</p>\n";  
38 }  
39 if(window.addEventListener){  
40   window.addEventListener("load", init, false);  
41 }  
42 else if(window.attachEvent){  
43   window.attachEvent("onload", init);  
44 }
```

```
14 document.form.methods.options[document.form.methods.selectedIndex].value;  
15 switch(opcion){  
16   case "api":  
17   // Utilizando la Api de Local Storage, podemos guardar y luego recuperar un valor  
18   localStorage.setItem('bienvenida', 'Bienvenidos a localStorage con API de JavaScript');  
19   valor = localStorage.getItem('bienvenida');  
20   break;  
21   case "array":  
22   // Utilizar notación de matrices o arreglos  
23   localStorage['bienvenida'] = 'Bienvenidos, también podemos usar notación de matrices';  
24   valor = localStorage['bienvenida'];  
25   break;  
26   case "object":  
27   // Y claro, también es válido utilizar notación de propiedades de objetos */  
28   localStorage.bienvenida = 'Bienvenidos, igual podemos usarlo como propiedades de objeto';  
29   valor = localStorage.bienvenida;  
30   break;  
31   default:  
32   alert("Esta opción no existe");  
33   break;  
34 }  
35 var contenido = document.getElementById("content");  
36 contenido.innerHTML = "<p>\nUsando : <strong>" + opcion + "</strong> - " +  
37 valor + "\n\t</p>\n";  
38 }  
39 if(window.addEventListener){  
40   window.addEventListener("load", init, false);  
41 }  
42 else if(window.attachEvent){  
43   window.attachEvent("onload", init);  
44 }
```

DAW06L
GUIA 8

JENNIFER PATRICIA ZALDAÑA BELTRAN
ZB212784

EJ3

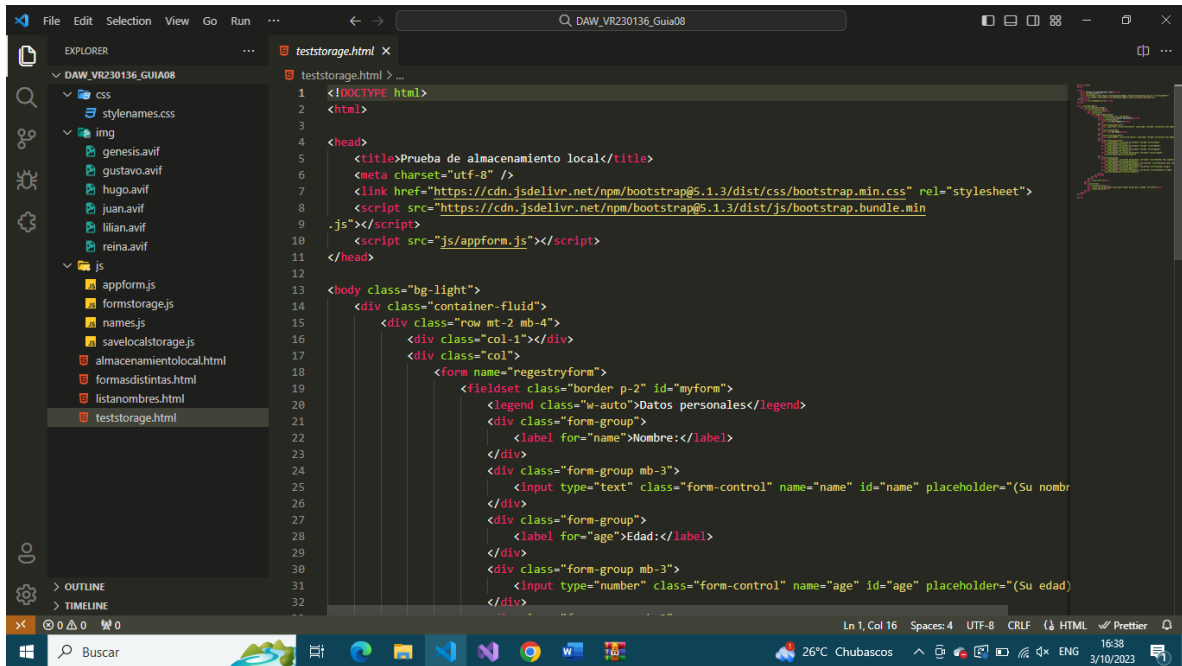


DAW06L
GUIA 8

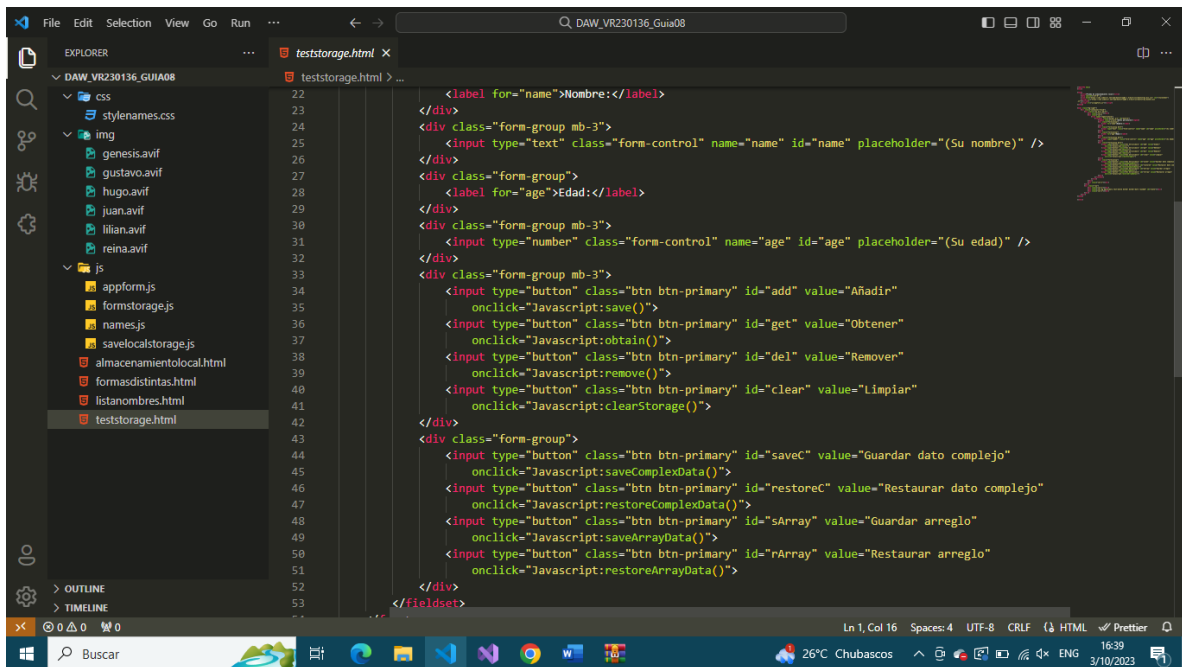
JENNIFER PATRICIA ZALDAÑA BELTRAN

ZB212784

EJ4 HTML



```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <title>Prueba de almacenamiento local</title>
6   <meta charset="utf-8" />
7   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
8   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
9   <script src="js/appform.js"></script>
10 </head>
11
12 <body class="bg-light">
13   <div class="container-fluid">
14     <div class="row mt-2 mb-4">
15       <div class="col-1"></div>
16       <div class="col">
17         <form name="registryform">
18           <fieldset class="border p-2" id="myform">
19             <legend class="w-auto">Datos personales</legend>
20             <div class="form-group">
21               <label for="name">Nombre:</label>
22             </div>
23             <div class="form-group mb-3">
24               <input type="text" class="form-control" name="name" id="name" placeholder="(Su nombre)" />
25             </div>
26             <div class="form-group">
27               <label for="age">Edad:</label>
28             </div>
29             <div class="form-group mb-3">
30               <input type="number" class="form-control" name="age" id="age" placeholder="(Su edad)" />
31             </div>
32           </fieldset>
33         </form>
34       </div>
35     </div>
36   </div>
37 </body>
38 </html>
```

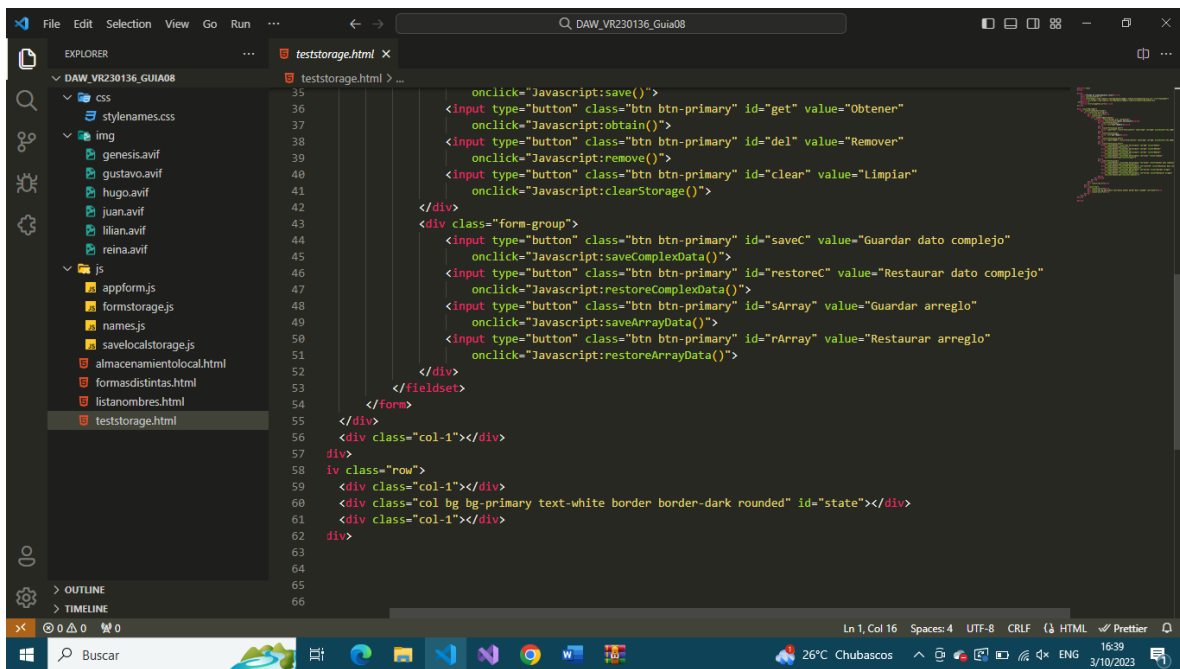


```
22 <label for="name">Nombre:</label>
23 </div>
24 <div class="form-group mb-3">
25   <input type="text" class="form-control" name="name" id="name" placeholder="(Su nombre)" />
26 </div>
27 <div class="form-group">
28   <label for="age">Edad:</label>
29 </div>
30 <div class="form-group mb-3">
31   <input type="number" class="form-control" name="age" id="age" placeholder="(Su edad)" />
32 </div>
33 <div class="form-group mb-3">
34   <input type="button" class="btn btn-primary" id="add" value="Añadir"
35     onclick="javascript:save()" />
36   <input type="button" class="btn btn-primary" id="get" value="Obtener"
37     onclick="javascript:obtain()" />
38   <input type="button" class="btn btn-primary" id="del" value="Remover"
39     onclick="javascript:remove()" />
40   <input type="button" class="btn btn-primary" id="clear" value="Limpiar"
41     onclick="javascript:clearStorage()" />
42 </div>
43 <div class="form-group">
44   <input type="button" class="btn btn-primary" id="saveC" value="Guardar dato complejo"
45     onclick="javascript:saveComplexData()" />
46   <input type="button" class="btn btn-primary" id="restoreC" value="Restaurar dato complejo"
47     onclick="javascript:restoreComplexData()" />
48   <input type="button" class="btn btn-primary" id="sArray" value="Guardar arreglo"
49     onclick="javascript:saveArrayData()" />
50   <input type="button" class="btn btn-primary" id="rArray" value="Restaurar arreglo"
51     onclick="javascript:restoreArrayData()" />
52 </div>
53 </fieldset>
54 </form>
55 </div>
56 </div>
57 </body>
58 </html>
```

DAW06L

GUIA 8

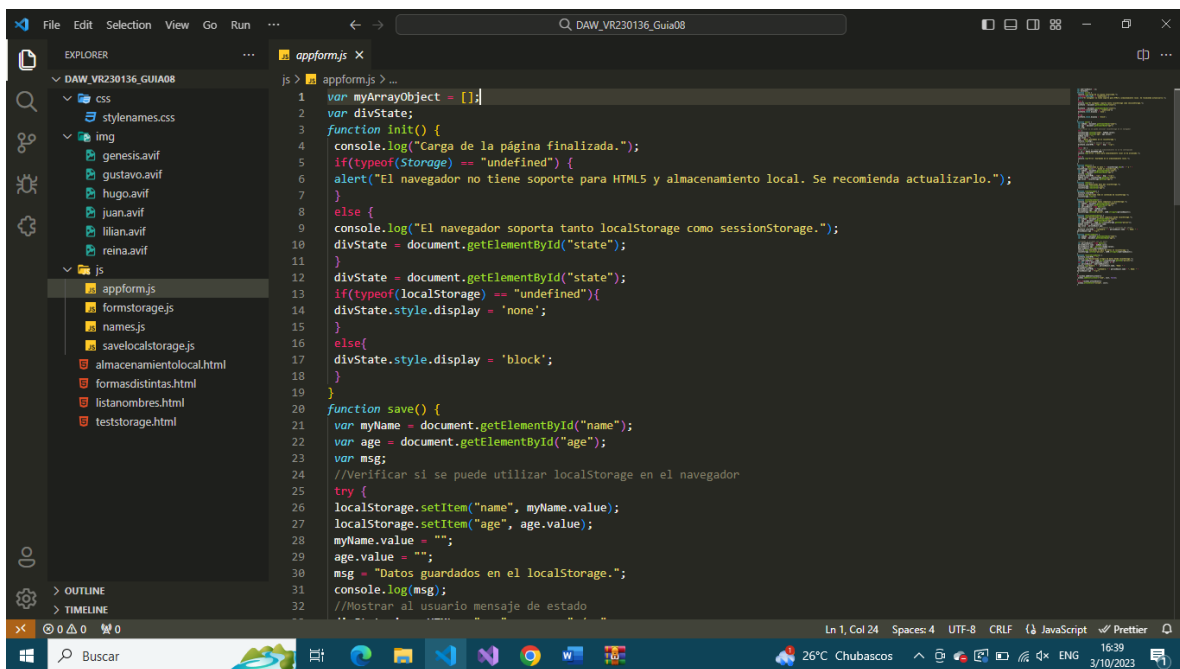
JENNIFER PATRICIA ZALDAÑA BELTRAN
ZB212784



The screenshot shows a Visual Studio Code editor window with the file explorer on the left displaying a project named 'DAW_VR230136_GUIA08'. The file 'teststorage.html' is selected. The main editor area shows the HTML code for 'teststorage.html', which includes a form with several buttons for saving, restoring, and clearing data. The code is as follows:

```
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
onclick="Javascript:save()"
<input type="button" class="btn btn-primary" id="get" value="Obtener"
onclick="Javascript:obtain()"
<input type="button" class="btn btn-primary" id="del" value="Remover"
onclick="Javascript:remove()"
<input type="button" class="btn btn-primary" id="clear" value="Limpiar"
onclick="Javascript:clearStorage()"
</div>
<div class="form-group">
<input type="button" class="btn btn-primary" id="saveC" value="Guardar dato complejo"
onclick="Javascript:saveComplexData()"
<input type="button" class="btn btn-primary" id="restoreC" value="Restaurar dato complejo"
onclick="Javascript:restoreComplexData()"
<input type="button" class="btn btn-primary" id="sArray" value="Guardar arreglo"
onclick="Javascript:saveArrayData()"
<input type="button" class="btn btn-primary" id="rArray" value="Restaurar arreglo"
onclick="Javascript:restoreArrayData()"
</div>
</fieldset>
</form>
</div>
<div class="col-1"></div>
<div>
<div class="row">
<div class="col-1"></div>
<div class="col bg bg-primary text-white border border-dark rounded id="state"></div>
<div class="col-1"></div>
</div>
```

EJ4 JS



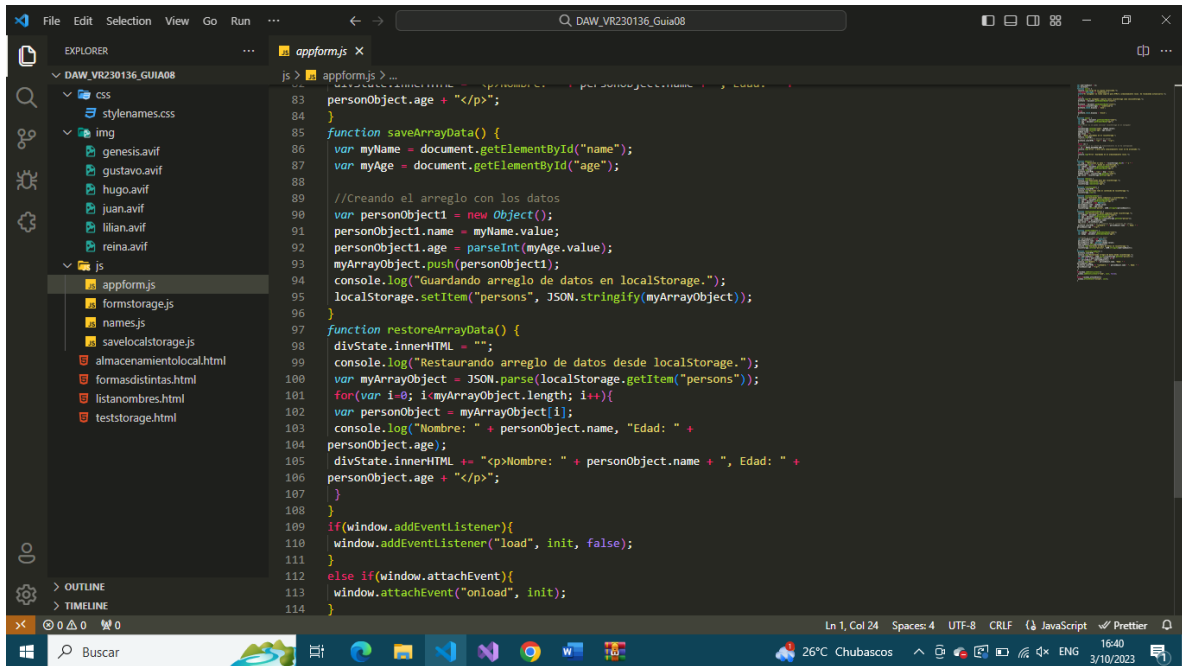
The screenshot shows a Visual Studio Code editor window with the file explorer on the left displaying the same project. The file 'appform.js' is selected. The main editor area shows the JavaScript code for 'appform.js', which includes functions for initializing the page, saving data to localStorage, and restoring data. The code is as follows:

```
js > appform.js > ...
1 var myArrayObject = [];
2 var divState;
3 function init() {
4   console.log("Carga de la página finalizada.");
5   if(typeof(Storage) == "undefined") {
6     alert("El navegador no tiene soporte para HTML5 y almacenamiento local. Se recomienda actualizarlo.");
7   }
8   else {
9     console.log("El navegador soporta tanto localStorage como sessionStorage.");
10    divState = document.getElementById("state");
11  }
12  divState = document.getElementById("state");
13  if(typeof(localStorage) == "undefined"){
14    divState.style.display = 'none';
15  }
16  else{
17    divState.style.display = 'block';
18  }
19 }
20 function save() {
21   var myName = document.getElementById("name");
22   var age = document.getElementById("age");
23   var msg;
24   //Verificar si se puede utilizar localStorage en el navegador
25   try {
26     localStorage.setItem("name", myName.value);
27     localStorage.setItem("age", age.value);
28     myName.value = "";
29     age.value = "";
30     msg = "Datos guardados en el localStorage.";
31     console.log(msg);
32   } catch (e) {
33     //Mostrar al usuario mensaje de estado
```

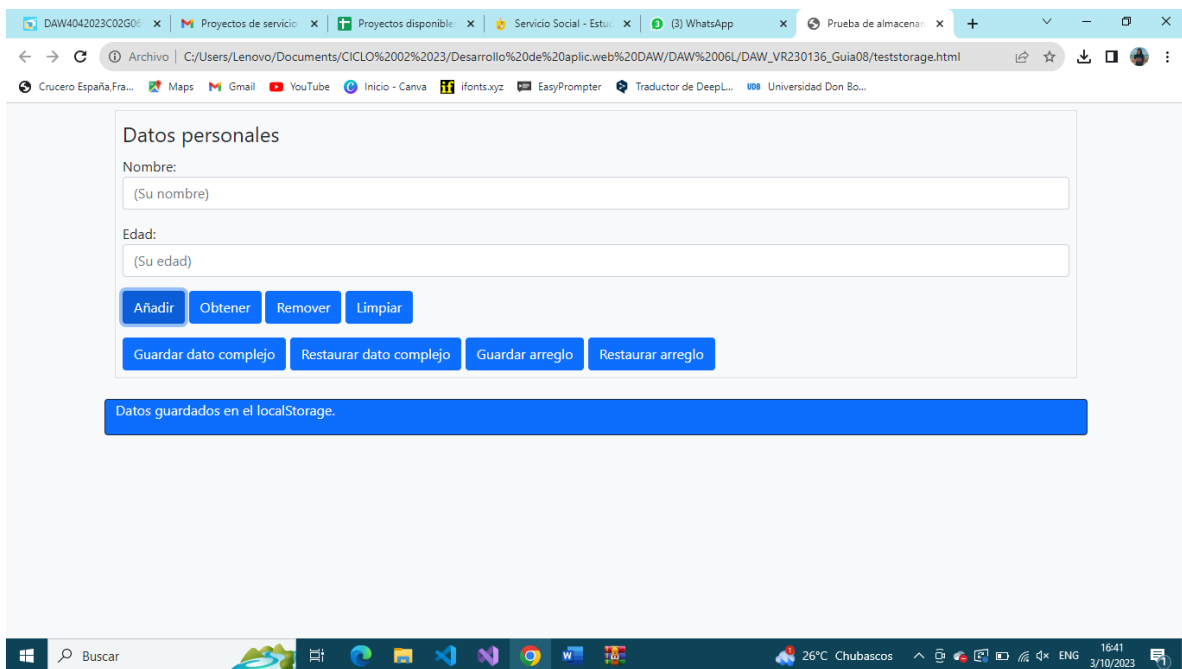
DAW06L
GUIA 8

```
25 try {
26   localStorage.setItem("name", myName.value);
27   localStorage.setItem("age", age.value);
28   myName.value = "";
29   age.value = "";
30   msg = "Datos guardados en el localStorage.";
31   console.log(msg);
32   //Mostrar al usuario mensaje de estado
33   divState.innerHTML = "<p>" + msg + "</p>";
34 }
35 catch (e) {
36   //Verificar si el límite de almacenamiento no se ha sobrepasado
37   if (e >= QUOTA_EXCEEDED_ERR) {
38     console.log("Error: Límite para almacenamiento local se ha alcanzado.");
39   }
40   else {
41     console.log("Error: Guardando en el almacenamiento local.");
42   }
43 }
44 }
45 function obtain() {
46   var msg = "Obteniendo el dato " + localStorage.key(0) + " y " +
47   localStorage.key(1) + " desde el localStorage.";
48   var myName = document.getElementById("name");
49   var age = document.getElementById("age");
50   console.log(msg);
51   divState.innerHTML = "<p>" + msg + "</p>";
52   myName.value = localStorage.getItem("name");
53   age.value = localStorage.getItem("age");
54 }
55 function remove() {
56   console.log("Removiendo dato del localStorage.");
```

```
54 }
55 function remove() {
56   console.log("Removiendo dato del localStorage.");
57   localStorage.removeItem("name");
58   localStorage.removeItem("age");
59 }
60 function clearStorage() {
61   divState.innerHTML = "";
62   console.log("Borrando todo el contenido de localStorage.");
63   localStorage.clear();
64 }
65 function saveComplexData() {
66   console.log("Guardando datos complejos a localStorage.");
67   var myName = document.getElementById("name");
68   var age = document.getElementById("age");
69   var personObject = new Object();
70   personObject.name = myName.value;
71   personObject.age = age.value;
72   localStorage.setItem("person", JSON.stringify(personObject));
73 }
74 function restoreComplexData() {
75   console.log("Restaurando datos complejos desde localStorage.");
76   var myName = document.getElementById("name");
77   var age = document.getElementById("age");
78   var personObject = JSON.parse(localStorage.getItem("person"));
79   myName.value = personObject.name;
80   age.value = personObject.age;
81   //Mostrar al usuario en el elemento DIV el contenido del objeto
82   divState.innerHTML = "<p>Nombre: " + personObject.name + ", Edad: " +
83   personObject.age + "</p>";
84 }
85 function saveArrayData() {
```



```
js > appform.js
83 personObject.age + "</p>";
84 }
85 function saveArrayData() {
86   var myName = document.getElementById("name");
87   var myAge = document.getElementById("age");
88
89   //Creando el arreglo con los datos
90   var personObject1 = new Object();
91   personObject1.name = myName.value;
92   personObject1.age = parseInt(myAge.value);
93   myArrayObject.push(personObject1);
94   console.log("Guardando arreglo de datos en localStorage.");
95   localStorage.setItem("persons", JSON.stringify(myArrayObject));
96 }
97 function restoreArrayData() {
98   divState.innerHTML += "<br>";
99   console.log("Restaurando arreglo de datos desde localStorage.");
100   var myArrayObject = JSON.parse(localStorage.getItem("persons"));
101   for(var i=0; i<myArrayObject.length; i++){
102     var personObject = myArrayObject[i];
103     console.log("Nombre: " + personObject.name, "Edad: " +
104     personObject.age);
105     divState.innerHTML += "<p>Nombre: " + personObject.name + ", Edad: " +
106     personObject.age + "</p>";
107   }
108 }
109 if(window.addEventListener){
110   window.addEventListener("load", init, false);
111 }
112 else if(window.attachEvent){
113   window.attachEvent("onload", init);
114 }
```



JENNIFER PATRICIA ZALDAÑA BELTRAN
ZB212784

