


Evaluación: Primera (parte práctica)	Fecha: 03/12/2021		
Módulo/Materia: Desarrollo Web en Entorno Cliente	Curso: 2º		
Departamento: INFORMÁTICA			
Nombre y apellidos:			

## Parte práctica (7 puntos)

No se permite el uso de foros, chats, o cualquier web o programa de comunicación. Por supuesto tampoco se puede usar el teléfono móvil.

No se permite el uso de **REPOSITORIOS**.

Para cualquier duda o pregunta los alumnos levantarán la mano y esperarán a que el profesor acuda a su puesto.

Cualquier infracción supondrá tener una calificación de 0 puntos.

**Además de subir el código a la tarea, completa este documento con una captura de cada ejercicio con la salida del programa.**

Para cada ejercicio se seguirá esta rúbrica:

	Peso	3	2	1	0
<b>Funcionalidad</b>	50%	La web tiene toda la funcionalidad solicitada	Alguna funcionalidad no está implementada correctamente	Dos o tres funcionalidades no están implementadas correctamente	Más de 3 funcionalidades no se implementan correctamente
<b>Seguridad</b>	10%	Se contemplan bastantes comportamientos inesperados en las interacciones con el usuario.	Se contemplan algunos comportamientos inesperados.		No se contempla ningún comportamiento inesperado.
<b>Organización y legibilidad</b>	10%	Se utilizan archivos independientes y el código está estructurado en secciones. Los comentarios estructuran el script en secciones y aclaran la funcionalidad de funciones y determinado código.	No se utilizan archivos independientes o el código no está estructurado en secciones. Faltan algunos de los comentarios citados.		No se utilizan archivos independientes ni el código está estructurado en secciones. El script no contiene comentarios.
<b>Uso del lenguaje</b>	20%	Se hace un uso adecuado de las funciones.	Se hace un uso excesivo o faltan funciones.		
<b>Originalidad y usabilidad</b>	10%	La web es amigable, de uso sencillo. Incluye funcionalidad no solicitada.	Resulta complicado usar la web.		Es necesario ser un usuario experto para poder utilizar la página.

## Ejercicio 1 (2 puntos)

Sobre el siguiente código, realiza las siguientes modificaciones:

- Cada vez que se pulse el botón se añada un nuevo elemento a la lista mediante el uso de métodos: `addEventListener`.
- Crea el evento para añadir un nuevo usuario en la lista `ul` (elemento `li`), con formato:  
"Soy xxxxx. Tengo xxxxx años y mi telefono es el xxxxxxxx".  
En las pruebas inserta al menos tu nombre de usuario en un elemento de la lista.
- No se puede usar `innerHTML` ni tocar el código `html`. Hay que usar métodos de Javascript para terminar el programa, como por ejemplo `createElement`.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Ejercicio 1 - lista</title>
    <script>

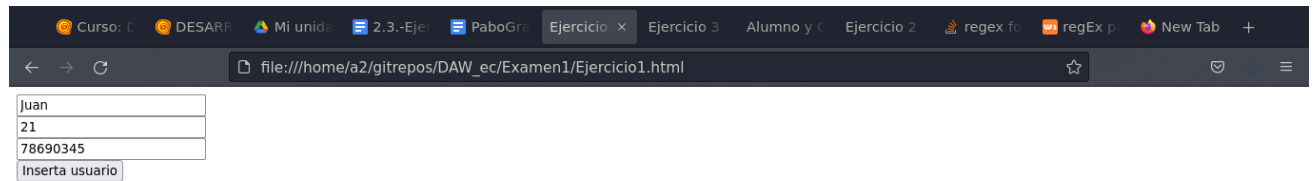
      </script>
    </head>

    <body >
      <form id="form1" name="form1">
        <input id="nombre" name="nombre"><br>
        <input id="edad" name="edad"><br>
        <input id="telefono" name="telefono"><br>
        <input type="button" id="botonInsertar" value="Inserta usuario"><br>
      </form>
      <h1>LISTA DE CLIENTES</h1>
      <ul>

      </ul>

      <script>

      </script>
    </body>
  </html>
```



A screenshot of a web browser window. The address bar shows a file path: file:///home/a2/gitrepos/DAW\_ec/Examen1/Ejercicio1.html. The page content includes a form with three input fields containing the text 'Juan', '21', and '78690345'. Below these fields is a button labeled 'Inserta usuario'.

## LISTA DE CLIENTES

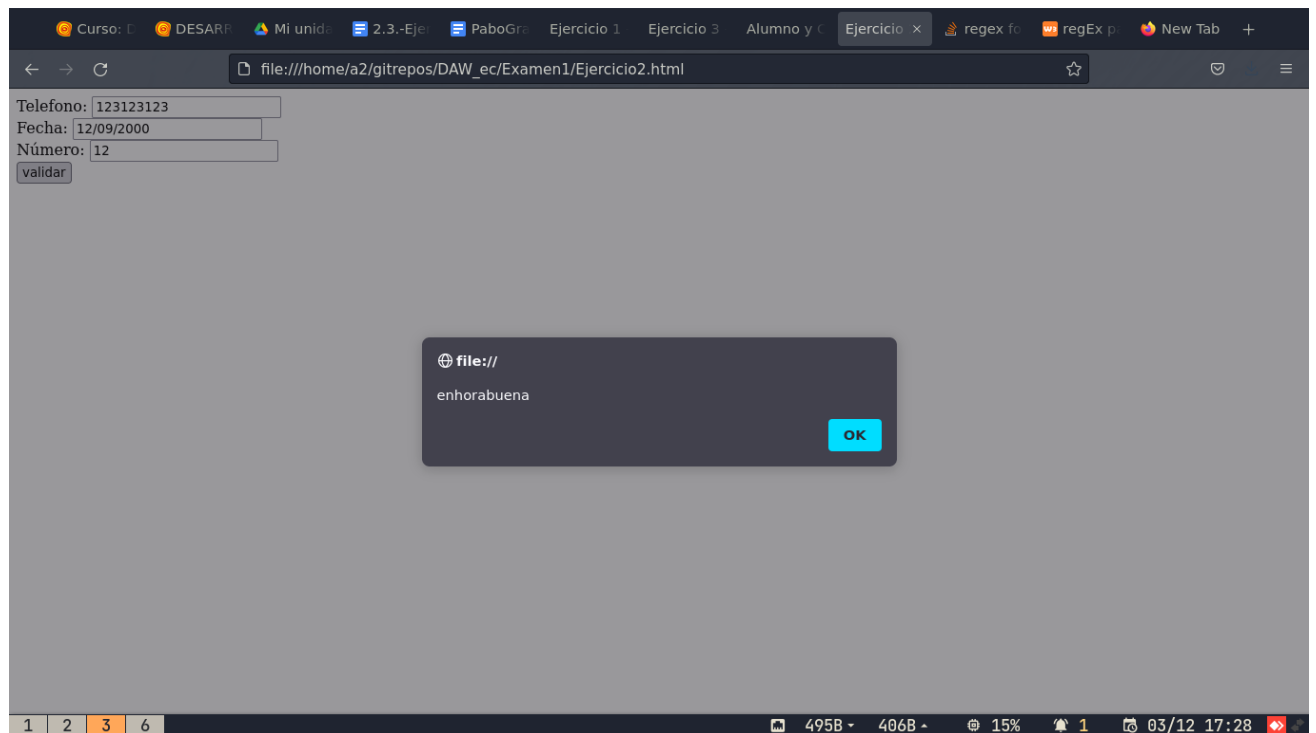
- Soy Pablo. Tengo 21 años y mi telefono es el 78690345
- Soy Juan. Tengo 21 años y mi telefono es el 78690345

## Ejercicio 2 (1 puntos)

Utiliza las Expresiones Regulares de JavaScript para validar campos.

**Requisitos:**

- Usa una función para cada input a comprobar.
  - Puedes añadir código html pero no quitar el que tienes de inicio.
- **Formatos:**
- a) Número de teléfono con 9 dígitos (sin espacios).
  - b) Fecha con formato dd/mm/yyyy (no validar mayor de 31 días ni mayor de 12 meses)
  - c) Número entre 0 y 23.



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ejercicio 2 - Expresiones regulares</title>
    <script>

    </script>
  </head>
  <body>
    <form id="form1">
      <label>Telefono: <input type="text" id="telefono"></label><br>
      <label>Fecha: <input type="text" id="fecha"></label><br>
      <label>Número: <input type="text" id="numero"></label><br>
      <input type="button" id="boton" value="validar">
    </form>
    <script>

    </script>
  </body>
</html>
```

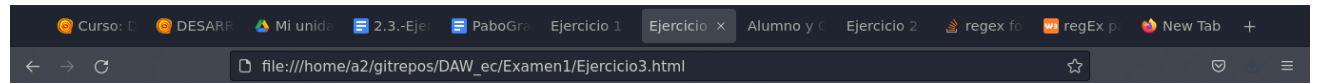
## Ejercicio 3 (1 puntos)

Dado el código siguiente:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ejercicio 3 - Enlaces</title>
    <script>
      function prepararEnlaces(){
        var paginas=['google','hotmail','msn','unizar'];
        var c = document.getElementById('container');
        var cadena = "<ul>";
        for (i=0;i<paginas.length;i++){
          cadena += "<li>http://www."+ paginas[i] + ".com</li>";
        }
        cadena += "</ul>";
        c.innerHTML=cadena;
        crearEnlaces();
      }
      function crearEnlaces(){
        // ...
      }
    </script>
  </head>

  <body onload="prepararEnlaces()">
    <h1>Ejercicio de enlaces</h1>
    <div id="container"></div>
  </body>
</html>
```

Programa SIN MODIFICAR el código original, la función crearEnlaces. Dicha función modificará el contenido de la lista transformando el listado en hipervínculos que el usuario podrá utilizar para conectarse a dichas páginas web.



## Ejercicio de enlaces

- <http://www.google.com>
- <http://www.hotmail.com>
- <http://www.msn.com>
- <http://www.unizar.com>



## Ejercicio 4 (3 puntos)

Una tienda on-line necesita almacenar en un programa todos los artículos a la venta.

### En Javascript ES6 (con class):

Crea un objeto Articulo que almacene la siguiente información:

- Código.
- Nombre del artículo.
- Localización: almacenará un número de estantería.
- Precio.

Además, tendrá el siguiente método:

- Un método que muestre toda la información de un artículo.

Crea una clase ProductoFresco que herede de Articulo y tenga dos atributos propios:

- FechaRecepción: donde se almacena la fecha de llegada al almacén (con Date());
- Duración: donde se almacena el número de días hasta su fecha de caducidad.

Cuando el usuario cargue la página, se le mostrará el listado de productos frescos en una tabla y se le permitirá añadir un producto fresco mediante un formulario (con un botón del formulario).

El usuario también tendrá disponible un campo con el código que quiera y mostrará con un botón la información de ese producto en pantalla con un alert (usar el método de la clase Articulo);



Curso: Desar... DESARROLLO Mi unidad - Examen\_DW Ejercicio 1 - list: Ejercicio 3 - Enl Alumno y Col... file:///home/a2/... pgutrilla/DAV +

file:///home/a2/gitrepos/DAW\_ec/Examen1/Ejercicio4.html

Codigo  
LEX  
Nombre  
pepino  
Localizacion  
Zaragoza  
Precio  
1234  
Duracion  
2134  
Crear

Codigo	Nombre	Localizacion	Precio	Fecha Recepcion	Duracion
LIX	pepino	Zaragoza	1234	Fri Dec 03 2021	34
LUX	pepino	Zaragoza	1234	Fri Dec 03 2021	34
LEX	pepino	Zaragoza	1234	Fri Dec 03 2021	34

Codigo  
LUX  
Busqueda

file://  
LUX  
pepino  
Zaragoza  
1234  
OK

1 2 3

583B 268B 01% 1 03/12 17:03