**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:

Tabla

Descripción generada automáticamente

# 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 listaul (elemento li), con formato:

"Soy xxxxx. Tengo xxxxx años y mi telefono es el xxxxxxx".

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.

<!DOCTYPEhtml>

<html>

    <head>

        <metacharset="UTF-8">

        <title>Ejercicio 1 - lista</title>

        <script>

        </script>

    </head>

    <body>

        <formid="form1"name="form1">

            <inputid="nombre"name="nombre"><br>

            <inputid="edad"name="edad"><br>

            <inputid="telefono"name="telefono"><br>

            <inputtype="button"id="botonInsertar"value="Inserta usuario"><br>

        </form>

        <h1>LISTA DE CLIENTES</h1>

        <ul>

        </ul>

        <script>

        </script>

    </body>

</html>

****

# 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:**

1. Número de teléfono con 9 digitos (sin espacios).
2. Fecha con formato dd/mm/yyyy (no validar mayor de 31 días ni mayor de 12 meses)
3. 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">

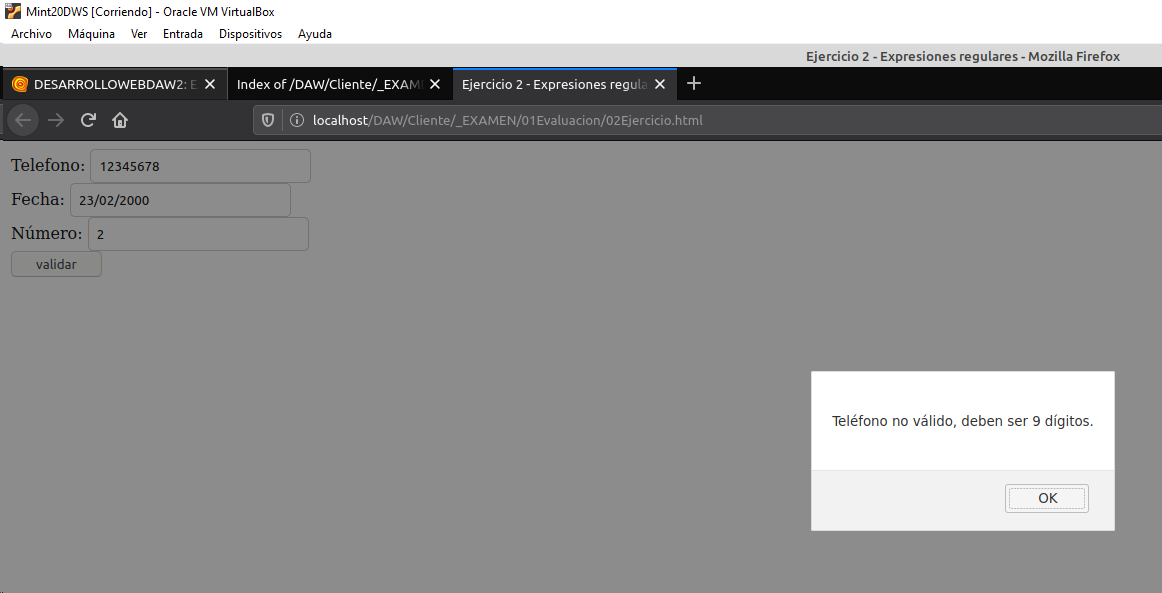
</form>

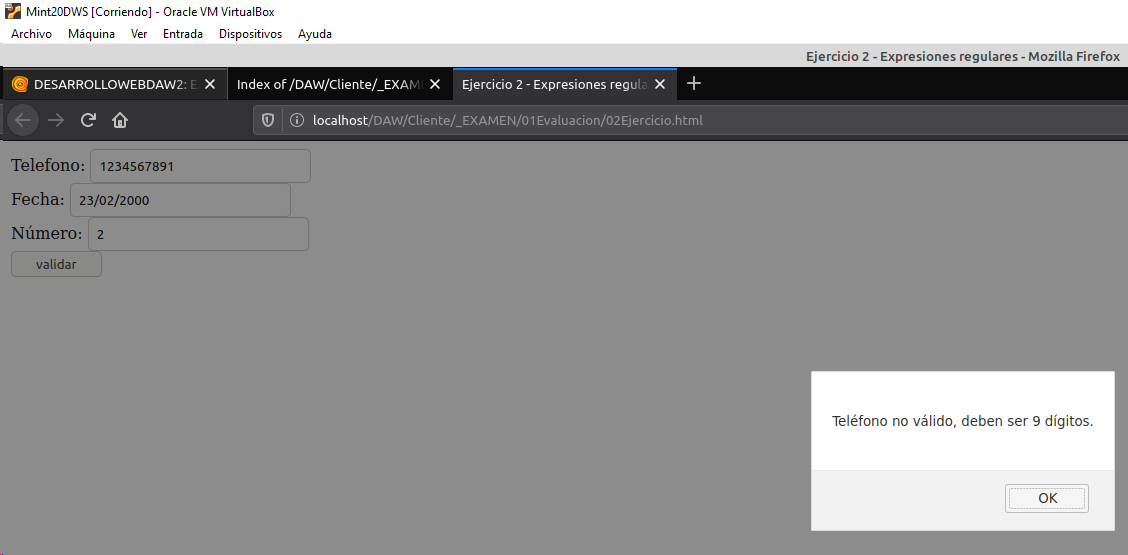
<script>

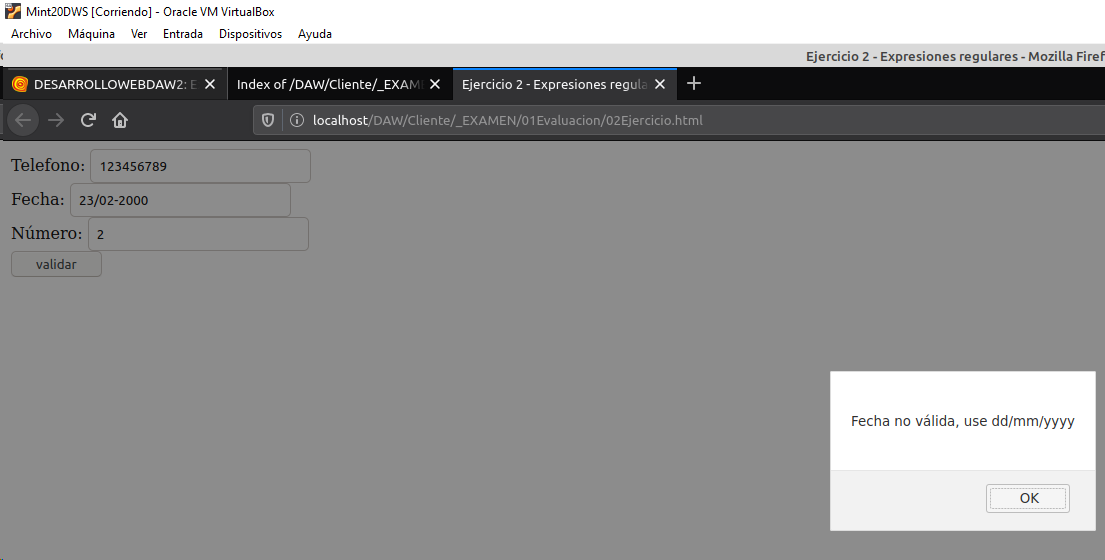
</script>

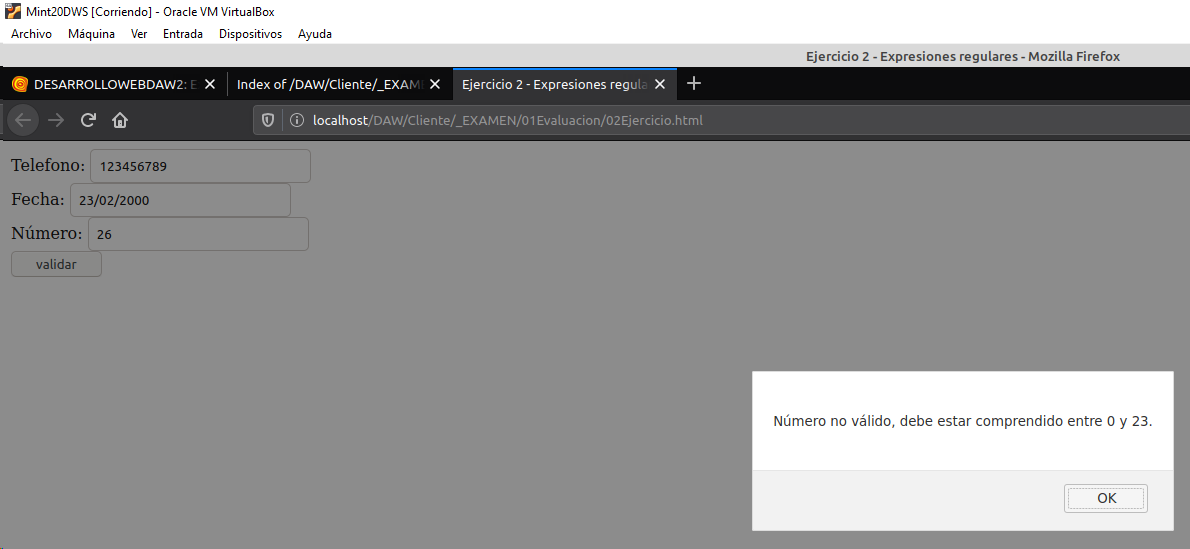
</body>

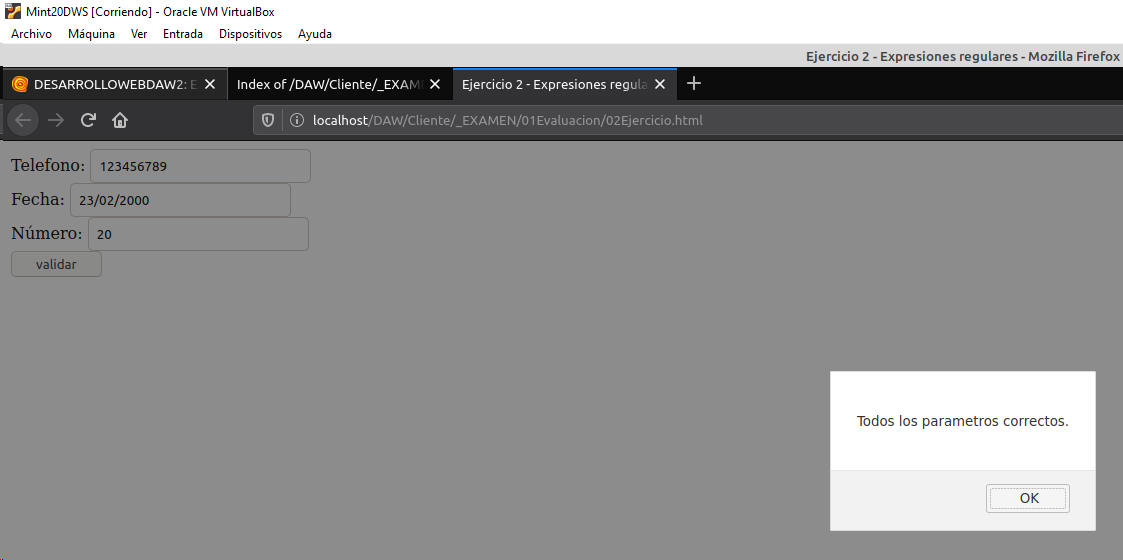
</html>

****

****

****

****

****

# Ejercicio 3 (1 puntos)

Dado el código siguiente:

<!DOCTYPEhtml>

<html>

    <head>

        <metacharset="utf-8">

        <title>Ejercicio 3 - Enlaces</title>

    <script>

        functionprepararEnlaces(){

            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();

        }

        functioncrearEnlaces(){

            // …

        }

    </script>

</head>

    <bodyonload="prepararEnlaces()">

        <h1>Ejercicio de enlaces</h1>

        <divid="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 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 productofresco 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);

