



Nombre: _____

Importante: Compruebe las instrucciones que se dan para realizar la entrega, siga las mismas, márquelas cuando las haya comprobado y firme en la casilla correspondiente. Cualquier entrega que no cumpla las reglas de nombrado, el formato de compresión del archivo o el contenido de los archivos del mismo, será penalizada con 2 puntos sobre 10 por cada incumplimiento. Pasado el límite de entrega se aceptará el envío del trabajo, con una penalización de un 2 puntos sobre 10 de la calificación por cada minuto o fracción de retraso a partir del tercer minuto.

<ul style="list-style-type: none"><input type="checkbox"/> Restricciones: Las rutas de los ficheros empleados serán relativas, a fin de que su solución pueda ser examinada en cualquier equipo.<input type="checkbox"/> Material a entregar: Los archivos <u>solicitarAsistencia.html</u>, <u>funciones.js</u> e <u>implementacionjQuery.js</u>.<input type="checkbox"/> Formato de entrega: Archivo comprimido ZIP.<input type="checkbox"/> Nombre del archivo de entrega: El nombre del fichero tendrá un formato específico dictado por el nombre de cada alumno. Por ejemplo, para un alumno llamado "José María Núñez Pérez" el fichero se nombrará como NunyezPerezJM.zip. Obsérvese que las tildes son ignoradas y las eñes sustituidas.<input type="checkbox"/> Vía entrega: La entrega del trabajo se hará a través de la tarea correspondiente en el Campus Virtual y se completará cuando se reciba por correo electrónico resguardo de la entrega. Conserve dicho correo como prueba del envío.<input type="checkbox"/> Comprobación de la entrega: Los alumnos deben comprobar que el archivo depositado en la plataforma es correcto descargándolo de la misma, descomprimiéndolo y comprobando su contenido.	He realizado todas las comprobaciones indicadas. Firma del alumno:
---	--

Objetivos

- Demostrar los conocimientos adquiridos en el desarrollo de páginas web interactivas con JavaScript y jQuery.

Descripción del ejercicio propuesto

IMPORTANTE: Recuerde que el uso del atributo innerHTML no se tolera y su uso invalidará el ejercicio entregado.

Se va a implementar un sistema para dar de alta solicitudes de asistencia en las que se expondrá el coste objetivo de la posible incidencia (a 24 horas) así como las observaciones que el usuario considere oportunas.

Para ello, haciendo uso exclusivamente de tecnologías del lado del cliente, implemente la lógica necesaria para seleccionar un determinado cliente (o darlo de alta si así se requiere) para, a continuación, indicar los parámetros requeridos para la solicitud (coste por hora, frecuencia y observaciones) y, por último, mostrar toda la información por pantalla, a modo resumen. Tenga en cuenta que este sistema solo funciona de forma local (sin refrescar la página). La comunicación de los datos con el sistema externo quedará fuera del alcance de la presente prueba práctica.

Para lograr la funcionalidad requerida, implemente todas las funciones JavaScript descritas en el fichero *funciones.js* y todos los comportamientos y plugins jQuery en el fichero *implementacionjQuery.js*. Ambos ficheros son importados en la página *solicitarAsistencia.html*.

Dispone de la clase CSS *oculto*. Haga uso de la misma cuando se le solicite mostrar u ocultar un elemento. Observe que la página está organizada en secciones (algunas de ellas ocultas) a las que nos referiremos por el *id* del elemento *div* que las contiene.



EJERCICIO 1 [7 puntos]. Usando exclusivamente **JavaScript** y partiendo de la página *solicitarAsistencia.html* suministrada como material adicional, incluya en un archivo llamado *funciones.js* todo el código necesario para dotar a la página de las siguientes funcionalidades interactivas, para lo que se recomienda que se muestren mensajes de *log* en cada una de ellas para poder ver la traza de ejecución de cada punto en la consola del navegador:

1. Inicializar sistema **[3 puntos]**:

- a) Inicialización de variables globales y constantes **(0.5 puntos)** Defina de forma global:
 1. Una constante *CLIENTE_NO_SELECCIONADO* con el valor -1.
 2. Una variable *infoClientes*, que se tratará de un vector en el que por cada posición tendrá a su vez un vector definido con 3 posiciones que contendrán el nombre, el email y el teléfono de cada uno de los clientes. Inicialícelo con la siguiente información:
 - Posición 1: Carlos, cdbargon@upo.es, 654112233
 - Posición 2: Jose Antonio, jalagrod@upo.es, 645445566
 - Posición 3: Alberto, agonriv@upo.es, 643778899
 3. Una variable *clienteSeleccionado*, inicializada a nulo. Esta variable será usada para almacenar el cliente seleccionado en cada momento. Se tratará por tanto de un vector de 3 posiciones (nombre, email y teléfono).
- b) Función *inicializaVistaAcceso()* **(1.5 puntos)**. En primer lugar, limpie la lista de opciones del selector de clientes. Tras esto, añada a este mismo campo, estableciéndola como opción por defecto, una opción con texto "Seleccione un cliente..." cuyo valor será el asociado en la constante *CLIENTE_NO_SELECCIONADO*. Del mismo modo, añada una opción por cliente cargado en la variable global *infoClientes*, indicando como texto el nombre del cliente y como valor la posición que éste ocupa en el vector. Por último, modifique la página *solicitarAsistencia.html* de forma que esta función sea llamada al cargar la misma.
- c) Función *acceder()* **(1 punto)**. Obtenga el valor de la opción seleccionada en el selector de clientes y
 1. En caso de que no se haya seleccionado cliente (La primera opción), muestre un mensaje por la consola de *log* indicando que no hay cliente seleccionado.
 2. En cualquier otro caso:
 - Oculte la sección *bloqueAcceso*.
 - Cargue en la variable *clienteSeleccionado* los datos del cliente seleccionado en selector de clientes.
 - Rellene los datos del cliente seleccionado empleando los *spans* dentro de la sección *bloqueSolicitarAsistencia*.
 - Elimine el valor de los campos de la solicitud (coste, frecuencia y observaciones del cliente).
 - Muestre la sección *bloqueSolicitarAsistencia*.
 3. Por último, establezca lo necesario para que esta función sea llamada cada vez que se seleccione un cliente en el selector.

2. Alta nuevo cliente **[2 puntos]**:

- a) Función *altaNuevoCliente()* **(0.75 puntos)**. Al llamar esta función, oculte la sección *bloqueAcceso*, para, a continuación, eliminar el contenido de todos los campos de texto de esta sección y muestre la sección *bloqueAltaCliente*. Por último, establezca el mecanismo necesario para llamar a esta función al hacer clic sobre el botón para dar de alta un nuevo cliente.
- b) Función *guardarCliente()* **(1 punto)**. Al ser ejecutada esta función, debe obtener los valores de todos los campos de texto, referentes al nombre, email y teléfono del cliente a dar de alta en el sistema, añadiendo estos como un nuevo cliente al vector *infoClientes*. A continuación, oculte la sección *bloqueAltaCliente* y muestre la sección *bloqueAcceso*, tras inicializarla ejecutando la función *inicializaVistaAcceso*. Finalmente, haga que esta nueva función se ejecute al hacer clic sobre el botón para guardar un nuevo cliente.
- c) Función *cancelarAltaCliente()* **(0.25 puntos)**. En la que, tras su ejecución, la sección *bloqueAltaCliente* quede oculta, quedando visible la sección *bloqueAcceso*. Establezca el mecanismo necesario para llamar a esta función al hacer clic sobre el botón para cancelar el alta de un cliente.

3. Procesar solicitud de asistencia **[2 puntos]**:



- a) Función *procesarSolicitudAsistencia()* **(1.5 puntos)**. Al iniciar la ejecución de esta función oculte el bloque *bloqueSolicitarAsistencia*. para a continuación obtener los datos que se hayan introducido para crear la solicitud de asistencia (observaciones, coste y frecuencia hora). Lo siguiente que debe hacer es obtener el coste diario. Para ello, multiplique el valor del *coste* por la *frecuencia Hora*, y todo ello por las 24 horas que conforman un día.

Con la información que ha recopilado, cree el texto resumen de la solicitud de forma similar a la captura que se muestra en el material adjunto y colóquelo en el *span* correspondiente de la sección *bloqueResumen*, haciendo que ésta sea visible.

- b) Función *cancelarSolicitud()* **(0.5 puntos)**. En la que debe ocultar la sección *bloqueSolicitarAsistencia*, reinicializar el contenido de la sección *bloqueAcceso* (haciendo uso de la función *inicializaVistaAcceso*) y finalizar mostrando la sección *bloqueAcceso*. Establezca el mecanismo necesario para llamar a esta función al hacer clic sobre el botón para cancelar la solicitud.

EJERCICIO 2 [3 puntos]. Usando **exclusivamente jQuery** defina en el fichero *implementacionjQuery.js*:

- a. **Plugin jQuery [1.5 puntos]**: Validación de campos de texto. Cree un *plugin* para validar que todos los campos de tipo texto tengan contenido donde, en caso de detectar algún campo vacío, establezca su color de fondo a rojo y en caso contrario establezca el color de fondo del campo a blanco. Por último, si todos los campos de texto tienen contenido, el *plugin* debe devolver el valor *true*, y en caso contrario muestre un mensaje de *log* por consola indicando que se han detectado errores y devolver valor *false*. Haga uso de este *plugin* en la función *guardarCliente()*, de forma que solo se ejecute la funcionalidad de dicha función en caso de que el *plugin* devuelva el valor *true*.
- b. **Comportamiento jQuery [1 punto]**: Valores numéricos informados y mayor o igual a cero. Defina el comportamiento jQuery necesario para ejecutar la función *procesarSolicitudAsistencia()*, al hacer clic sobre el botón con *id solicitarAsistencia*, sólo en caso de haber validado que todos los campos numéricos no son vacíos ni menores que cero. En caso de que los valores de dichos campos no sean informados o sean menor que cero, debe mostrar dichos campos con fondo rojo. Tenga en cuenta que una vez solucionado cualquier problema el fondo debe volver a ser establecido a blanco.
- c. **Comportamiento jQuery [0.5 puntos]**: Volver a inicio. Defina el comportamiento jQuery necesario para que, al hacer clic sobre el botón con *id volverAlInicio*, se oculte el bloque *bloqueResumen*, se ejecute la función *inicializaVistaAcceso* y, por ultimo, se muestre la sección *bloqueAcceso*.

Material suministrado

Como material adicional dispondrá del fichero "Material Adicional.7zip" que contendrá:

- Página HTML base para crear el sistema descrito (*solicitarAsistencia.html*).
- Librería jQuery (*jquery-3.5.1.js*).
- Descripción función para mostrar mensajes de *log* en consola del navegador (*Console.log()* - Referencia de la API Web _ MDN.pdf)
- Capturas de pantalla del estado de los distintos bloques (carpeta *capturas*).