

Apellidos: .....  
Nombre: ..... UVUS: .....



Bases de datos (GE/DGME)  
Segundo parcial oficial

23/06/2021  
Primera parte

**Ejercicio 1 (1 punto)** Responda a las siguientes cuestiones acerca del desarrollo web:

- a) (0,5) Indique lenguajes, extensiones, bibliotecas y herramientas relacionadas con estos aspectos del desarrollo web: a) contenido, b) estilo, c) dinámica en cliente y d) dinámica en servidor. Describa brevemente cada elemento que haya indicado.
- b) (0,5) Ilustre gráficamente y explique el proceso desde que escribimos una dirección web en un navegador hasta que visualizamos la respuesta, incluyendo resolución de nombre, y asumiendo que se pasa por servidor y base de datos.

**Ejercicio 2 (2,5 puntos)** Desarrolle el código HTML correspondiente a:

- a) (0,25) Un texto en una cabecera de tercer nivel, y otra de cuarto nivel. Incorpore en el primero un elemento de línea sin ninguna semántica asociada por defecto.
- b) (0,25) Una lista ordenada de elementos con sus grupos o solistas favoritos, incluyendo para cada uno de ellos una lista no ordenada de canciones favoritas.
- c) (0,25) Un enlace a alguna parte de su misma página, visitado al pinchar sobre una imagen. Garantice que si la imagen no está disponible, nos salga un texto.
- d) (0,5) Una tabla con sus bloques principales y al menos tres filas, incluyendo tanto celdas simples como celdas resaltadas en los lugares adecuados.
- e) (0,5) Un formulario con controles básicos de distintos tipos, algunos propios de HTML 5 (no versiones previas) y con las etiquetas de texto asociadas fuera de los controles. Incorpore en ellos atributos interesantes, además de los obligatorios.
- f) (0,5) Un elemento de selección desplegable que muestre varias opciones posibles. Resuélvalo mediante los dos mecanismos vistos en clase.
- g) (0,25) Un fragmento de página con elementos semánticos, incorporando en alguno un párrafo con texto, con un elemento importante y otro destacado.

**Ejercicio 3 (1 punto)** Indique la expresión Emmet necesaria para generar:

- a) Una tabla con un identificador “miTabla”, que contenga tres filas, cada una de dos campos, conteniendo cada uno de ellos 4 palabras generadas automáticamente.
- b) Un contenedor de clase “miContenedor” con 5 párrafos, cada uno con un id (“miParrafo1” a “miParrafo5”), y conteniendo un enlace a una página (“pag1” a “pag5”), con el texto del enlace componiéndose de 3 palabras generadas aleatoriamente.

**Ejercicio 4 (2,5 puntos)** Desarrolle el código CSS necesario para lo siguiente:

- a) (0,25) Asignar a un elemento con clase “destacada” un fondo de color azul, un color de texto blanco y un borde verde continuo de 1 píxel.
- b) (0,25) Aplicar a los párrafos hijos directos de un `div` con id “mi\_div” un espaciado interno de 20 píxels. Asignarles la mitad del ancho, centrados horizontalmente.
- c) (0,25) Dejar a los 3 bloques de una tabla (en la misma regla CSS, selector múltiple) un espaciado exterior con una medida que dependa del tamaño de la fuente.

- d) (0,25) Poner en *cursiva* el texto de los elementos de lista que sean hermanos de un elemento cuyo id sea “capataz”. Poner en verde el texto de los enlaces cuyo destino contenga “comp”, y cambiar el ratón a una mano, como hemos visto en clase.
- e) (0,25) Ocultar un párrafo de clase “esconder” al pasar el ratón sobre él. Hágalo empleando los dos mecanismos distintos para ocultar vistos en clase. ¿Qué diferencia hay en el resultado obtenido, según el mecanismo empleado?
- f) (0,25) Aplique un estilo al div con id “barra-fija”, de modo que quede posicionado en la parte superior de la pantalla, separada 2 píxels de la parte superior y 3 del lado izquierdo, independientemente de dónde lo hayamos incluido en nuestro código html, pero de forma que al hacer scroll hacia abajo suficientemente dejemos de verlo.
- g) (0,5) Aplicar a todo elemento con la clase “divide” un grid que tenga 3 columnas, siendo la columna del medio de la mitad del tamaño de las otras dos. Aplicar una pequeña separación entre las filas y columnas del grid.
- h) (0,5) Emplear a todo elemento que tenga la clase “relaja” un diseño flexible a nivel de fila invertida, y de modo que los elementos incluidos queden dispuestos al final de la misma. Además, aplique estilo a uno de esos elementos hijos de modo que, independientemente de su posición en el código, este elemento aparezca el último.

**Ejercicio 5 (2 puntos)** Haga uso de Bootstrap para crear la estructura necesaria para:

- a) (0,75) Crear un elemento de navegación (pestañas, pills o menús) con 2 opciones, cada una de ellas habilitando una sección. Cree los elementos origen de navegación y código mínimo del destino, marcando adecuadamente la opción por defecto.
- b) (0,75) Generar un formulario Bootstrap con dos campos, y dos botones de distinto tipo, usando las clases adecuadas. Para dispositivos de tamaño medio o superior debemos garantizar que aparezca un campo por fila, incluyendo etiqueta y control, debidamente alineados, quedando el texto de las etiquetas a la misma distancia del comienzo del control de formulario para todas las filas.
- c) (0,25) Aplicar a una tabla las clases adecuadas de Bootstrap, apareciendo además las filas con colores alternos y tal que al pasar sobre cada fila ésta se destaque.
- d) (0,25) Incorporar otros elementos que haya visto de Bootstrap (colores de texto, fondo, tarjetas, etc.), aportando el código mínimo indispensable.

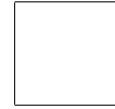
**Ejercicio 6 (1 punto)** Emplee código SCSS visto en clase para lo siguiente:

- a) (0,25) Declarar variables SASS para establecer algunos datos por defecto, y emplearlos en alguna regla de estilo básica.
- b) (0,375) Usando el mecanismo de anidamiento visto, proporcione estilo a ciertos elementos jerárquicamente organizados con varios niveles de profundidad. Puede usar ejemplos de html de ejercicios anteriores. Debe incluir selectores para elementos hijos, descendientes, y el operador para referirnos al propio elemento dentro del anidamiento, añadiéndole además una pseudo-clase.
- c) (0,375) Definir un `mixin` con parámetros, valores declarados por defecto, y asignando varias propiedades de estilo. Utilice este `mixin` en un par de reglas de estilo, en una pasando datos específicos y en la otra dejando valores por defecto.

**Notas:**

- Escriba su nombre, apellidos y UVUS en la hoja del examen.
- Cumplimente los ejercicios en los folios proporcionados en el aula.

Apellidos: .....  
Nombre: ..... UVUS: .....



Bases de datos (GE/DGME)  
Segundo parcial oficial

23/06/2021  
Segunda parte

**Ejercicio 1 (0,75 puntos)** Responda a las siguientes cuestiones:

- a) (0,5 pts) Indique y describa brevemente los programas que hemos empleado para:  
a) editar código; b) servir archivos; c) lanzar peticiones a servidor; d) ejecutar programas en servidor; e) servir bases de datos; f) actuar como cliente de estas; g) visualizar sitios; h) testear aspectos del funcionamiento en web.
- b) (0,25 pts) ¿Qué protocolo hemos visto para hacer peticiones web y recibir resultados? ¿De qué dos formas principales podemos transmitir en este protocolo los datos entre cliente y servidor? ¿Qué método suele emplearse para cada forma?

**Ejercicio 2 (3 puntos)** Desarrolle el código HTML y JavaScript necesario para:

- a) (1 pts) Terminada de cargar la página, deberá pedir en mensaje emergente al usuario un texto a repetir, y un segundo mensaje donde indique el número de veces a repetir. A continuación, deberá imprimir el mensaje por consola tantas veces como se haya indicado. Modifique el código para mostrar el mensaje en un elemento de bloque del DOM con un cierto id, generando un párrafo por cada repetición.
- b) (1 pts) Crear una función que reciba un array, con objetos del tipo `[{nombre: "Luis", ape: "Pérez"}, {nombre: "Elena", ape: "López"}]`, y devuelva otro array con la suma de caracteres (ej: `[9, 10]`). Asociar a un botón un manejador de evento en cuyo cuerpo se llame a esta función, pasándole el array del ejemplo, y lo asocie a una variable/constante. Mostrar un mensaje emergente con un texto que indique cuál es la suma de todos los números del array devuelto.
- c) (1 pts) Crear una función que reciba un array `a` (del tipo del anterior) y un entero `ncar`, y devuelva un array con los objetos de `a` cuyo nombre tenga menos de `ncar` caracteres. Definir fuera de la función un array de ejemplo. Suponiendo que existe un elemento `<select>` con números, crear un manejador de evento tal que al cambiar la selección, se llame a la función anterior, pasándole el array definido y el valor del elemento seleccionado.

**Ejercicio 3 (3,25 puntos)** Desarrolle el código PHP necesario para:

- a) (1,5 pts) Crear una biblioteca con las siguientes funciones:
  - Una función que reciba como parámetros un nombre de base de datos, un usuario y un password, construya la cadena de conexión para acceder a la misma y se conecte como hemos visto en clase, pasándole url, usuario y password. Devuelva la conexión creada.
  - Una función que reciba dos parámetros (una conexión a BD y un valor numérico), y que lance una consulta preparada que, haciendo uso de la conexión pasada, emplee el valor recibido para filtrar los datos del SELECT sobre una tabla de su base de datos, obteniendo los registros cuyo valor para un cierto campo superen el valor numérico dado por el parámetro. La función debe devolver a quien la llame las filas devueltas por la consulta, sin más procesamiento.

- Una función que reciba tres parámetros (conexión a BD, id y nombre), y que lance una inserción preparada, que inserte en los dos campos correspondientes de una tabla un nuevo registro con los datos recibidos. La función deberá devolver un array asociativo con un campo “mensaje” con un valor “Ok”.
- b) (1,75 pts) Crear una página que reciba peticiones GET/POST del cliente y responda según lo recibido, usando las funciones de la biblioteca anterior, y devolviendo una salida JSON en todos los casos:
- En primer lugar, intentar recibir un dato “tipoAccion”, con posibles valores *consulta* o *inserta*. Si el tipo recibido no es ninguno de los dos, debe responder con un objeto con un campo “mensaje”, indicando que no es válido. En otro caso, comprobar en la petición si se recibieron **bd**, **user** y **pass**. Si no es así, se deben asignar los valores por defecto “prueba”, “root” y “”, respectivamente. Con estos datos, establecer la conexión con la BD, usando la función creada.
  - Si la acción es *consulta*, comprobar si se ha recibido **num**, y en su caso emplearlo para lanzar la consulta desarrollada del ejercicio anterior. Además, enviar a la salida como JSON un array asociativo con un campo *mensaje* que indique “Ok” y otro *resultado* que incluya la respuesta de la función anterior.
  - Si la acción es *inserta*, debe comprobarse que se han proporcionado en la petición los datos **id** y **nombre**, y a continuación llamarse a la función desarrollada. Se deberá finalmente emitir a la salida un objeto JSON con el contenido que se haya recibido de la llamada a la función.

**Ejercicio 4 (3 puntos)** Desarrolle el código HTML y Javascript necesario para responder a los siguientes apartados:

- a) (0,5 pts) Crear un formulario con un select con dos opciones con valores **bd1** y **bd2** y nombres inventados, con dos campos de texto para usuario y password, un control de número “val”, otro “id” y un control para proporcionar en “nombre”.
- b) (1,5 pts) Definir una función que lance con Javascript una petición GET al programa de servidor creado, solicitando la acción de tipo *consulta*, pasándole los datos que se necesiten para la misma. Al recibir el resultado del servidor, deberá comprobarse si el resultado fue “Ok”, y en ese caso deberá generar en la misma página un elemento de tipo lista que visualice un elemento por cada registro recibido del servidor, mostrando el valor de algunos campos. Asociar dicha función como manejador de evento tanto al cambiar el select como al modificar el valor del control “val”.
- c) (1 pts) Incorporar al formulario anterior un botón tal que, al hacer click sobre el mismo, lance desde Javascript una petición de tipo POST al servidor, solicitando la acción de tipo *inserta*, pasándole los valores de los elementos necesarios. Al recibir la respuesta del servidor, bastará con emitir un mensaje emergente que diga “¡Excelente!” en caso de que el mensaje contenido en el resultado de la operación haya sido “Ok”, y otro mensaje “¡Algo ha ido mal!” en caso contrario.

## Notas:

- Escriba su nombre, apellidos y UVUS en la hoja del examen.
- Cumplimente los ejercicios en los folios proporcionados en el aula.
- Se valorará la variedad de técnicas empleadas, como se planteó en clase.