

Instrucciones para la entrega del trabajo – Segundo cuatrimestre

El trabajo a entregar en el segundo cuatrimestre consiste en un proyecto de desarrollo de una aplicación web que permita el manejo de la base de datos (MariaDB/MySQL y/o MongoDB) desarrollada en el primer cuatrimestre, empleando las tecnologías vistas a lo largo del segundo. Los requisitos incluirán al menos los siguientes aspectos:

- **Memoria del trabajo**, describiendo brevemente el trabajo desarrollado (lo principal es el código y la defensa, este documento va a ser un mero guion). Se recomienda elaborar esta parte al final, a modo de resumen de qué se ha querido plasmar y cómo se ha ido realizando, destacando las principales decisiones funcionales y tecnológicas. La extensión será libre, pero será suficiente con un pequeño documento de unas pocas páginas (nunca más de 10 si incluye imágenes, no más de 5 si solamente consta de texto), ya que se entiende que el esfuerzo principal se va a volcar en el desarrollo con las tecnologías vistas en clase.
- **Aspectos funcionales**: la aplicación deberá acceder a la base de datos y proporcionar información útil para el usuario potencial de la misma, incluyendo algunas funcionalidades pensando en un rol de administrador de la base de datos (con una visión más cercana a la de poder insertar, actualizar, eliminar y consultar tablas individuales de la base de datos) y otras más orientadas a una persona interesada en la temática pero no en los detalles de cómo se encuentra almacenada cada tabla, sino tener una visión más general, con consultas o funcionalidades que vayan obteniendo información de diversas tablas en una sola vista en función del uso práctico de la aplicación o planteando acciones relacionadas. Nota: no es necesario hacer gestión de usuarios, pues la seguridad queda fuera del alcance de la asignatura, pero sí tener esos dos tipos de funcionalidades diferenciadas.
- **Aspectos tecnológicos**:
 - Desarrollo en cliente:
 - **Contenido (HTML)**: debe incorporar los aspectos esenciales vistos en clase, incluyendo naturalmente elementos clásicos de bloque y de línea, elementos de texto, tablas, listas, enlaces, imágenes, aspectos semánticos de HTML5, formularios con controles diversos (clásicos y de HTML5), crucial para poder proporcionar datos para su base de datos. Por lo demás, se valorará cualquier aportación adicional que se haga, siempre que se demuestre en la defensa del trabajo la adecuada asimilación de los conceptos y la destreza en su uso (es decir, solo se va a valorar aquello que se demuestre dominar en la defensa, no código ajeno que podamos adoptar y adaptar pero que no se comprenda y domine adecuadamente.)
 - **Estructura**: se deberá seguir el tipo de estructura que hemos ido promoviendo en clase (SPA – Single Page Application), con una página principal que contenga toda la estructura de navegación y vaya mostrando unos contenidos u otros en función de las interacciones del usuario.
 - **Estilo**:
 - CSS básico: deben hacerse uso de selectores tanto básicos como avanzados, y demostrar su manejo en la defensa del trabajo. Además, se deberán manejar propiedades relacionados con modelo de cajas (espaciado interno y externo, bordes, etc.), texto, color, y se valorará la variedad en el uso de otras propiedades y la variedad y complejidad de los selectores incluidos.

- Layout: se debe hacer uso de los mecanismos fundamentales para el manejo del layout, incluyendo el manejo de atributos como display, position, visibility, etc. Se requerirá el uso y dominio de los mecanismos esenciales de CSS Grid y Flexbox (de modo que se demuestre poder estructurar los contenidos adecuadamente, naturalmente no es necesario para ello conocer toda la variedad de propiedades de estos elementos, pero sí ser capaz de emplear sus mecanismos básicos de forma efectiva.) Se valorará además el empleo efectivo de mecanismos como la declaración de variables y su uso adecuado, así como el del anidamiento.
- Bibliotecas de estilo: se deberán emplear los elementos básicos de Bootstrap 5 vistos en clase, **demostrando dominar los aspectos de navegación** (menús, *tabs* o *pills*, pudiendo incorporar *dropdowns*), *grid* (pudiendo adaptar el número de columnas a distintos tamaños de dispositivos), formularios, tablas y algunos elementos de utilidad (m, p, gx, colores, textos, etc.). Además, se valorará cualquier otra aportación de otros componentes de la biblioteca, como *cards* u otros que se propongan en el curso, o la incorporación de componentes de otras bibliotecas externas.
- **Dinámica en cliente:**
 - JavaScript: deberán emplearse y dominarse las estructuras básicas del lenguaje de programación, incluyendo estructuras de datos e instrucciones de asignación, selectivas, repetitivas, etc., así como los operadores básicos. Se valorará la variedad de tipos de estructuras empleadas (ej: programación imperativa y de orden superior, funciones nombradas y anónimas, etc.) Además, se deberá poder manejar el acceso a los elementos del DOM empleando técnicas de JavaScript moderno, así como la interpolación de cadenas y las llamadas al servidor a través de los mecanismos vistos en este tema (API Fetch).
 - Se deberá demostrar el conocimiento del flujo de las comunicaciones entre cliente y servidor, y refrescar por alguno de los métodos anteriores los elementos del DOM pertinentes en función de la interacción del usuario y de los datos recibidos del servidor.
 - Se deberá incluir la definición e instanciación de algunas clases que capturan elementos relacionados con la temática de su trabajo.
 - Se deberá demostrar en el trabajo y la correspondiente defensa el conocimiento y destreza a la hora de entender los principales eventos que hemos visto en el curso y la asociación de los manejadores correspondientes, preponderando las técnicas que hemos promovido, sobre aquellas que hemos recalado que van quedando obsoletas.
 - Se valorará el buen uso de scripts y la reusabilidad del código, eligiendo los casos en los que será adecuado incluir el código en los archivos html y aquellos en los que será mejor dejarlo en archivos externos, demostrando el conocimiento a la hora de importar y utilizar las funciones desarrolladas.
- Desarrollo en servidor:
 - NodeJS: se deberá demostrar en el trabajo y su defensa el dominio de los mecanismos de recepción de parámetros en Node enviados por el cliente, tanto para peticiones get como post. Naturalmente, tendrá especial peso el acceso adecuado a la base de datos, a través de los mecanismos vistos en clase, y

empleando consultas adecuadas. Será fundamental disponer de funcionalidades de inserción, actualización y eliminación para poder gestionar parte de la información de su base de datos (una serie de tablas de interés, no es necesario naturalmente que sean todas). Además, deberá dotarse de algunas funcionalidades interesantes para un usuario interesado en la temática del trabajo, de forma que se le muestren determinadas consultas sobre la base de datos, accediendo a una serie de tablas que proporcionen una vista significativa de aquellos aspectos que considere más relevantes dentro de su trabajo. Puede tomar como referencia algunas de las consultas desarrolladas durante el trabajo del primer cuatrimestre, de forma que el usuario final pueda seleccionar criterios de filtrado y que a partir de los mismos se lancen consultas avanzadas y se le muestre de forma atractiva el resultado obtenido.

- Se deberá dominar el uso de algunas estructuras de datos (arrays asociativos y no asociativos, cadenas, números, etc.), así como las estructuras selectivas y repetitivas vistas en clase, interpolación de cadenas, etc.
 - Las salidas del servidor deberán realizarse fundamentalmente en formato JSON, como hemos visto en clase, de forma que sea el cliente quien tenga la responsabilidad de procesar los datos recibidos y refrescar el DOM.
 - Se valorará el buen uso del código, su reusabilidad y la división coherente de la funcionalidad en distintos ficheros.
- Base de datos:
 - Base de datos MySQL (y/o MongoDB, si finalmente también lo vemos en clase) a la que accede su aplicación. Bastará con aportar la exportación del archivo .sql con todo el diseño y contenido de su base de datos, listo para poder cargarse en un servidor MySQL (y/o su equivalente en MongoDB).

En general, recuerde que el trabajo viene a demostrar que su capacidad desarrollar una buena aplicación web cliente-servidor que dota de algunas funcionalidades para un usuario encargado de mantener la información de la base de datos y de otras destinadas al usuario final, y que se valorará toda aportación relacionada con los contenidos y destrezas que se han promovido a lo largo del curso, y siempre que el dominio se demuestre tanto en los archivos entregados como en la defensa oral del trabajo, y que cualquier contenido que se entregue y no se demuestre dominar en la defensa no cumplirá con los requisitos esenciales para ser valorados debidamente.

Deberá entregarse en una carpeta comprimida (ZIP, RAR, 7z o similar) que contenga tanto la memoria como todo el código anterior, de modo que se pueda importar la base de datos y poner en marcha la aplicación desarrollada. La fecha límite de entrega será el 15 de junio, aunque se dará la opción de entrega anticipada (el 1 de junio, tras el segundo examen parcial oficial, de cara a ser evaluado lo antes posible). En cualquiera de las opciones, una vez corregido el trabajo, se programará la fecha y hora para la defensa oral de la misma, donde se analice extensamente el trabajo desarrollado, y se contraste el dominio real de los conceptos por parte del alumnado.