Proyecto
Diseña e implementa una aplicación web

Cambios v2:

- Corregidas las fechas de entrega de las prácticas.

Cambios v3:

- Añadidas clarificaciones de la Práctica 2.

Cambios v4:

- Añadidas clarificaciones de la Práctica 3.

Cambios v5:

- Actualizada rúbrica de evaluación de la Práctica 3.

NOTA: El contenido de este enunciado puede actualizarse y completarse a medida que avanza el curso. Cualquier actualización será notificada en los foros de la asignatura en el aula virtual.

Objetivo

Que el alumno implemente una aplicación web de la forma más parecida a como lo haría a nivel profesional. Algunas de las características de la práctica que simulan un entorno real son las siguientes:

- La práctica se desarrollará en un equipo formado por 3 o 4 alumnos. De esa forma el equipo será similar al que se puede encontrar en un desarrollo web profesional.
- Se utilizarán herramientas profesionales, tanto para el desarrollo en sí como para compartir el código entre los miembros del equipo (Visual Studio Code, GitHub...)
- La temática de la web podrá ser elegida libremente por los alumnos de cada equipo. De esta forma, los alumnos estarán más motivados y podrán incluir en la aplicación aquellas funcionalidades que deseen (dentro de unos límites que se definirán más adelante en este enunciado)

Temática de la aplicación web

La temática de la aplicación web que hay que diseñar e implementar será elegida libremente por los miembros de cada equipo. Cada equipo puede decidir qué funcionalidades ofrece al usuario, el aspecto gráfico, el esquema de navegación, etc.

A modo de ejemplo, se presentan algunos tipos de aplicaciones web que se podrían implementar, pero el equipo podrá elegir cualquier temática (aunque no esté aquí listada):

- Web del menú de un restaurante
- Web de tienda online
- Web de un foro online
- Web de liga de fútbol

Metodología de desarrollo

El proyecto deberá realizarse siguiendo la siguiente metodología de desarrollo.

Desarrollo colaborativo

La aplicación web se desarrollará usando un repositorio de la plataforma GitHub[1]. El código fuente de la aplicación tendrá la licencia Apache 2. El repositorio será creado por el profesor de la asignatura y se le darán permisos de edición a todos los miembros del mismo.

Durante el desarrollo de la aplicación se irá subiendo el código al repositorio a medida que se vaya desarrollando. Esto permitirá al profesor saber qué partes de la aplicación han sido desarrolladas por cada miembro del grupo.

Para gestionar el repositorio de github se puede usar cualquier cliente de git.

Desarrollo

El proyecto será desarrollado en diferentes prácticas (partes de la misma aplicación). De esa forma, los alumnos realizarán un trabajo continuo a lo largo del curso y el profesor podrá hacer un mejor seguimiento del mismo.

Las prácticas en las que se divide el desarrollo de la aplicación web son:

- **Preparación:** Definición de las funcionalidades de la web.
- **Práctica 1:** Maquetación de páginas web con HTML y CSS.
- **Práctica 2:** Aplicación web en el lado del servidor con Node.js y MongoDB.
- **Práctica 3:** Interactividad en el cliente con JavaScript.

El contenido de cada una de estas prácticas se describe en detalle más adelante en el enunciado del proyecto.

[1] https://github.com

El código de cada una de las prácticas deberá estar en el repositorio de código en las siguientes fechas antes de que comience la clase:

- **Preparación:** 23 de septiembre de 2025 antes de las 15:00
- **Práctica 1:** 15 de octubre de 2025 antes de las 17:00
- **Práctica 2:** 19 de noviembre de 2025 antes de las 17:00
- **Práctica 3:** 17 de diciembre de 2025 antes de las 17:00

Cuando el código del repositorio esté listo para la entrega de una práctica, será necesario hacer un tag en el repositorio con el nombre de la práctica que se vaya a entregar. Por ejemplo, al entregar la Práctica 1 se deberá crear un tag llamado “parctica1”, que será el que será evaluado. Los tags se pueden hacer desde la interfaz web de GitHub al crear una “release”.

Documentación

La documentación del proyecto se incluirá en el fichero README.md en la raíz del repositorio de código. Este fichero se editará usando adecuadamente el formato Markdown (títulos, negritas/cursivas, tablas, texto en formato código, etc.). Se debe verificar que el documento README.md se visualiza correctamente desde la web de GitHub.

En la descripción de cada una de las partes se indica de forma detallada el contenido de la documentación solicitada.

Evaluación

Cada una de las prácticas serán evaluadas mediante defensa presencial en horario de clase. Todos los alumnos del equipo deben estar presentes en la defensa al tratarse de una actividad de evaluación, aunque tengan dispensa académica.

Preparación (no evaluable)

La preparación no lleva asociada calificación porque consiste únicamente en definir la funcionalidad de la web y la creación del equipo.

El profesor verificará que la funcionalidad de la web definida por los alumnos cumple con los requisitos exigidos en el enunciado. Si se identifican carencias serán indicadas a los alumnos para que las solventen antes de comenzar con práctica.

Evaluación de las prácticas 1, 2 y 3

La evaluación de las prácticas 1, 2 y 3 llevará asociada una calificación. La evaluación de cada práctica se realizará en dos partes:

- **Demostración:** El profesor pedirá que se le haga una demostración de la aplicación usándola como si fuera un usuario. Los miembros del equipo realizarán un recorrido por la

aplicación mostrando sus funcionalidades más importantes con los diferentes tipos de usuarios: anónimo, registrado y administrador.

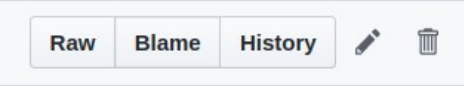
IMPORTANTE: La demostración deberá estar preparada de antemano, de forma que los alumnos la puedan realizar de forma rápida. Por ejemplo, si hay que subir una imagen, el alumno que haga la demostración deberá tener una imagen adecuada preparada en su disco.

- **Revisión de la plantilla de corrección:** El profesor revisará cada uno de los puntos de la plantilla de corrección de la práctica correspondiente disponible en el aula virtual durante la defensa. Para cada pregunta, pedirá que los miembros del equipo realicen una demostración del funcionamiento de la aplicación web o pedirá que le muestren la documentación o el código fuente.

Participación de los miembros del equipo

Todos los miembros del equipo tienen que participar activamente en el desarrollo de cada una de las prácticas. Para poder justificar su participación, cada uno de los miembros del equipo debe incluir en el README.md:

- Un párrafo describiendo de forma textual las tareas realizadas en esa práctica.
- Listado de los 5 commits más significativos durante la práctica. Cada uno de esos commits deberá ser un link que lleve a este commit en la interfaz web de GitHub.
- Listado de los 5 ficheros en los que más haya participado el miembro. Se podrá demostrar su participación al usar la opción “blame” de GitHub al visualizar ese fichero.



Si el profesor detecta una baja participación de un alumno concreto, podrá reducirse la nota de ese alumno y podría llegar a **suspender** dicha práctica.

Aunque el trabajo es colaborativo y es importante el reparto de tareas, todos los miembros del equipo deberán conocer el código completo de la aplicación (aunque algunas partes hayan sido desarrolladas por otro compañero).

Todos los miembros del equipo deberán estar presentes en la defensa porque el profesor podrá preguntar individualmente a cualquier alumno si lo considera necesario. La ausencia no justificada en la defensa llevará aparejado el **suspenso** de la práctica y el alumno tendrá que recuperar en la convocatoria ordinaria o extraordinaria.

Calidad del código fuente

Se exigirá una mínima calidad del código fuente. Al menos deberán tenerse en cuenta los siguientes aspectos:

- El código deberá estar formateado correctamente y usar la misma reglas de estilo en todos los ficheros (se recomienda configurar el entorno de desarrollo para formatear al guardar y así evitar problemas de este tipo).

- El código y los comentarios del mismo deberán estar escritos completamente en inglés. Sólo podrán existir términos en castellano en el código cuando vayan a mostrarse en el interfaz de usuario. La existencias de texto en castellano en el código llevará aparejada una penalización en la calificación.
- Las variables, parámetros, atributos, clases e interfaces deberán tener nombres adecuados que describan su objetivo.
- Se evitará el código duplicado. Cuando dos fragmentos de código sean similares, se utilizarán las técnicas adecuadas para fomentar la reutilización (herencia, composición, subprogramación…)
- Los métodos no serán muy largos y no tendrán mucha complejidad ciclomática.
- El código deberá ser razonablemente eficiente. Por ejemplo, no se considerará válido hacer una consulta a la base de datos para obtener una lista de elementos y luego filtrar esa lista en memoria usando código JavaScript.

Si se incumple alguna de estos buenas prácticas la calificación podrá verse penalizada hasta en **2 puntos** dependiendo de su grado.

Calificación

Al evaluar cada una de las prácticas se tendrán en cuenta los siguientes aspectos:

- Las prácticas 1, 2 y 3 tendrán una calificación entre 0 y 10.
- Si la práctica no cumple con los mínimos exigidos (falta de funcionalidad, incompleta) o no funciona correctamente (tiene errores) se considerará suspensa. El equipo que suspenda una práctica podrá realizar las entregas de las sucesivas prácticas sólo si el profesor verificar que han subsanando los errores de la entrega suspensa. Posteriormente podrán recuperar la práctica suspensa en la convocatoria ordinaria o extraordinaria.
- La nota de cada alumno del equipo es independiente. Un alumno puede obtener más o menos nota que el resto de sus compañeros. La nota individual de cada alumno se determina en base a:
  - El conocimiento de la aplicación durante la defensa, ya que todos los alumnos tienen que conocer el código de toda la aplicación aunque ellos no hayan desarrollado esa parte.
  - La contribución de cada miembro en esa práctica. Se puede llegar al extremo de suspender la práctica a un miembro del equipo si no ha colaborado de forma mínima en la práctica.

Convocatoria extraordinaria

Aquellas prácticas aprobadas en la convocatoria ordinaria se guardan para la convocatoria extraordinaria. Es decir, en la convocatoria extraordinaria los alumnos sólo tendrán que realizar aquellas prácticas suspensas o no entregadas.

La nota final de la práctica se calculará con los mismos porcentajes que en la convocatoria ordinaria.

En la convocatoria extraordinaria si no se cumplen con las funcionalidades o la aplicación tiene fallos, la práctica (y por tanto la asignatura) se considerará suspensa.

Modificación de miembros del equipo

Aunque se debería evitar en la medida de lo posible, los equipos pueden tener modificaciones en sus miembros debido a las siguientes causas:

- **División del equipo en dos debido a conflictos entre los miembros:** En ese caso, el equipo se puede dividir en 2 y cada nuevo equipo podrá seguir el desarrollo de la práctica partiendo del código en el momento de la división. Hay que evitar esta situación por el incremento de trabajo que supondría para los dos nuevos equipos.
- **Alumno abandona un equipo por conflictos:** Se podrá unir a otro equipo siempre y cuando tenga menos de 4 alumnos y tenga aprobadas las mismas prácticas que el alumno que quiere incorporarse. Es decir, un alumno no se puede ir a un equipo que tenga aprobadas las prácticas 2 y 3 si él no tiene aprobada la práctica 2 en su equipo original.
- **Alumno deja la asignatura:** Se considera que el alumno ya no forma parte del equipo.

Formación del equipo

Los alumnos registrarán su equipo en el siguiente documento compartido antes del **18 de Septiembre a las 17:00**.

https://urjc-my.sharepoint.com/:x:/g/personal/micael_gallego_urjc_es/EV0ORRW1MdZCpUVwVjSTgCoBUe9mwI-mUKilW_h0IvR05w?e=CMe0td

Los equipos serán de 3 o 4 integrantes. Aquellos alumnos que no hayan podido formar equipo en la fecha prevista serán asignados a un equipo por el profesor.

Aquellos equipos de 3 alumnos podrán incorporar un nuevo alumno no asignado si el profesor lo considera oportuno.

Fase de Preparación: Definición de las funcionalidades de la web

En la fase de preparación se deben definir las funcionalidades de la aplicación web.

Esta fase no es evaluable.

Requisitos de la aplicación web

Para que la complejidad sea similar para todos los equipos, la aplicación web se deberán diseñar de forma que cumpla con los siguientes requisitos:

- **Entidades:** Una entidad representa un concepto que la aplicación web guarda en la base de datos. Son los documentos de la base de datos. Estas entidades corresponden a las clases del dominio de la aplicación, las cuales modelan la lógica del negocio y definen los atributos y comportamientos de los objetos que el sistema manejará. La aplicación web deberá gestionar una entidad principal y una entidad secundaria. La entidad secundaria estará relacionada con la entidad principal y dependerá de ella en algún aspecto. Su función es extender o complementar la información de la entidad principal, proporcionando detalles adicionales. Las entidades deberán contener atributos. La entidad principal deberá contener una o varias imágenes asociadas. Opcionalmente, las

entidades secundarias podrán tener también una o varias imágenes asociadas.

Por ejemplo:

- **Web del menú de un restaurante:** La entidad principal es el plato (por ejemplo “ensalada césar”, “espaguetis a la carbonara”, etc). Esta entidad podría tener como atributos el nombre, la descripción, el tipo de plato (entrante, principal, postre), precio, etc. Y la entidad secundaria podría ser cada uno de sus ingredientes (por ejemplo “tomate”, “lechuga”...). Un ingrediente podría tener como atributos su nombre corto, una descripción del mismo, origen, si lo pueden comer los veganos o vegetarianos, sus alérgenos. La relación existe porque cada plato tiene asociado la lista de ingredientes para prepararlo.
- **Web de catálogo online:** Entidad principal es el producto (que podría tener un nombre, descripción, precio, fotografías, etc.). La entidad secundaria podría ser una reseña de ese producto (con su texto, su puntuación, autor, fecha, etc). La relación se tiene porque cada producto puede contener reseñas.
- **Web de un foro online:** Entidad principal tema (con nombre, descripción, fecha de creación…). La entidad secundaria un mensaje dentro de ese tema (con texto, autor, fecha, etc) La relación se tiene porque cada tema contiene mensajes.
- **Web de equipos deportivos:** Entidad principal es el equipo (con su nombre, descripción, fecha de fundación, deporte, etc). La entidad secundaria es cada uno de los jugadores (con su nombre, fecha de nacimiento, posición de juego, etc.). La relación se tiene porque cada equipo esta formado por jugadores.
- **Web de una plataforma de cursos online:** Entidad principal es el curso (con nombre, fechas de impartición, descripción, temática…). La entidad secundaria son los temas en los que se divide ese curso (nombre, descripción, fechas de impartición…).
- **Web de seguimiento de series:** Entidad principal es la serie (títulos, sinopsis, temática, calificación de edad, % de visionado…). La entidad secundaria son los capítulos de la serie (nombre, sinopsis, duración, % visionado…)

- **Imágenes:** La web tiene que permitir la subida de imágenes desde el navegador web para que sean vincualadas a la entidad principal. Por ejemplo fotos de productos, logotipo de un equipo, etc.
- **Buscador, filtrado o categorización:** La aplicación web deberá ofrecer uno o varias de las siguientes funcionalidades:
  - **Buscador:** Un cuadro de texto que permita introducir un texto y que devuelva sólo aquellos valores de la entidad principal en los que ese texto está incluido en el título o nombre de la entidad principal.
  - **Filtrado:** Un formulario que permita filtrar los elementos en base a uno o varios criterios como por ejemplo precio, marca, año, etc.
  - **Categorización:** Los elementos de la entidad principal se pueden categorizar. Por ejemplo, en un catálogo online se pueden mostrar las categorías de los productos en el menú superior o lateral: Teléfonos, Ordenadores, Televisores, etc.

Documentación

Se creará un fichero README.md usando el formato Markdown en la raíz del repositorio de GitHub. Este fichero se visualizará al entrar en la web de GitHub de ese repositorio y tendrá que estar correctamente formateado usando secciones, subsecciones, tablas, código fuente como texto en formato monoespaciado, imágenes, etc.

La documentación se podrá escribir en castellano, aunque se recomienza que se haga en inglés.

El contenido del fichero README.md deberá contener la siguiente información:

- Nombre de la aplicación web.
- Integrantes del equipo de desarrollo: Nombre, Apellidos, correo oficial de la universidad y cuenta en GitHub.
- Si se utiliza trello o cualquier otra herramienta para la coordinación del equipo, deberá ser pública y se incluirá el link de la misma.
- Una sección llamada “Funcionalidad” cada uno de los aspectos principales de la aplicación web:
  - **Entidades:** Es necesario describir la entidad principal y la entidad secundaria. Se tendrán que definir sus atributos.
  - **Imágenes:** Indicar qué entidades tendrán asociadas uno o varias imágenes por cada objeto/registro.
  - **Buscador, filtrado o categorización:** Describir qué tipo de consultas se realizarán.

Práctica 1: Maquetación de páginas web con HTML y CSS

En la práctica 1 se debe implementar la maquetación de la página web. En esta fase no se implementará un servidor y por tanto se considera que la web no es plenamente funcional.

Características técnicas de la implementación

La aplicación web deberá implementarse cumpliendo con los siguientes requisitos técnicos:

- La página web se implementará utilizando HTML y CSS.
- Los estilos CSS deberán definirse en uno o varios archivos “.css” (no deben incluirse directamente en el HTML).
- Se utilizará Bootstrap. Se permite el uso de una plantilla de Bootstrap, pero será necesario modificarla para cumplir con los objetivos de la práctica.
- Se utilizará el Grid System de Bootstrap, en lugar del Grid Layout nativo de CSS.
- Todas las páginas de la web deben ser responsive (cambio de posición y tamaño de elementos según el tamaño de la pantalla).
- Todas las páginas de la página web deben mantener el mismo diseño gráfico y compartir la cabecera y el pie de página.

- Los elementos del mismo tipo deben tener el mismo estilo gráfico. Por ejemplo, todos los botones deben tener el mismo tipo de letra, el mismo estilo de redondeo de esquinas, etc. Los colores pueden cambiar en función de la función del botón. Los títulos deben ser similares en todas las páginas. Los formularios deben utilizar los mismos estilos gráficos (colocación de etiqueta con el nombre del campo, posición de los botones, etc.)
- Los campos de los formularios serán los más específicos en función del valor que tenga que introducir el usuario: número, fecha, selección, texto largo, etc.
- Se deberá redefinir el estilo gráfico de bootstrap adaptado a la página web. No se puede utilizar el estilo gráfico por defecto.

Funcionalidades de la aplicación web en cliente (frontend)

La página web deberá estar compuesta, al menos, por las siguientes páginas:

- **Página principal:** mostrará todos los elementos de la entidad principal en un grid.
  - Deberá haber al menos 9 elementos de ejemplo para que se pueda visualizar correctamente su disposición. Por cada elemento se mostrará, al menos, su imagen y nombre. Se pueden añadir otros atributos si se considera relevante (precio, parte de la descripción, ...)
  - Al hacer clic en un elemento, se redirigirá a una página de detalle que mostrará la información detallada de ese elemento con sus entidades secundarias asociadas. Por ejemplo, en la web de la liga, en la página de detalle se mostrarán los jugadores del equipo.
  - El grid tendrá 1 columna en móviles, 2 en monitores pequeños y 3 en monitores grandes.
  - Encima del grid, alineado a la derecha, deberá haber un botón llamado "Crear nuevo elemento", que llevará a la página para crear un nuevo elemento.
- **Página de detalle:** mostrará la información detallada del elemento seleccionado en la página principal y sus entidades secundarias asociadas.
  - Incluirá todos los atributos del elemento (nombre, descripción, fecha, etc), además de mostrar la imagen asociada.
  - Deberá contar con los siguientes botones de acción para el elemento:
    - **Borrar:** No tendrá funcionalidad por el momento.
    - **Editar:** No tendrá funcionalidad por el momento.
    - **Volver a página principal:** Redirigirá a la página principal.
  - Se mostrarán también todos los elementos de la entidad secundaria relacionados con el elemento de la entidad principal seleccionado.
  - Además, se incluirá un formulario para crear un nuevo elemento de la entidad secundaria, con todos los campos necesarios y un botón de acción “Crear”.
  - Cada elemento de la entidad secundaria tendrá asociado un botón de ”Borrar” y uno de “Editar” (de momento sin funcionalidad asociada).

  - Solo será necesario crear una única página de detalle (no 10). Todos los elementos de la página principal redirigirán a la misma página de detalle.
- **Página de nuevo elemento:** mostrará un formulario para crear un nuevo elemento de la entidad principal.
  - Deberá contener todos los campos necesarios para recoger los atributos del elemento y su imagen.
  - El formulario tendrá dos botones de acción:
    - **Guardar:** No tendrá funcionalidad por el momento.
    - **Cancelar:** Redirigirá a la página principal.

Documentación

En el README se deberá añadir una sección “Práctica 1” con la siguiente información:

- **Caputas de pantalla de las páginas:** Página principal, página de detalle y página de nuevo elemento.
- **Participación de miembros:** Cada miembro del equipo indicará su participación en esta práctica con:
  - Descripción textual de las tareas realizadas en la práctica.
  - Listado de los 5 commits más significativos durante la práctica (enlazados a dichos commits en GitHub).
  - Listado de los 5 ficheros en los que más haya participado (enlazados a dichos ficheros en GitHub).

Entrega

La entrega de la práctica se compone de los siguientes pasos:

- Crear una “release” en GitHub con el nombre “practica1”
- Subir el .zip del código del repositorio GitHub al aula virtual por uno de los miembros del equipo.

Defensa

Los miembros del equipo mostrarán las páginas diseñadas.

El profesor podrá preguntar en general a todos los miembros del grupo o en particular a uno de ellos. Todos los alumnos deben conocer cómo funciona cada fichero de código aunque no haya sido desarrollado por él.

Rúbrica de autoevaluación

Está disponible una rúbrica de autoevaluación de la práctica 1 para que los alumnos puedan autoevaluar su propia práctica antes del día de la defensa.

Práctica 2: Aplicación web del lado del servidor con Node.js

En la práctica 2, la aplicación web se implementará en el lado del servidor usando **Node.js**.

Se implementará la capacidad de listar, crear, borrar y modificar los elementos. Por tanto, se considera que en la práctica 2 se tendrá una web plenamente funcional.

Características técnicas de la implementación

A continuación se detallan sus características técnicas:

- Los datos de los elementos deben almacenarse en la base de datos MongoDB.
- Las imágenes de los elementos deben almacenarse en la carpeta “uploads”.
- El código y los comentarios deberán estar escrito en inglés (salvo los mensajes que aparezcan mostrados al usuario, que pueden estar en castellano).
- No deberá haber código duplicado (o muy similar) en diferentes partes del servidor.
- Se deben seguir las buenas prácticas en la estructura del código vistas en los ejemplos de clase.
- No se deben subir al repositorio de código las carpetas “node_modules” y “uploads” (porque no se considera parte del código de la aplicación)
- Para el título y el footer de la aplicación será necesario utilizar Mustache Partials para evitar la repetición de código.
- Las páginas de confirmación y error deberán mantener el mismo estilo que el resto del sitio.
- Las páginas de confirmación y error se implementarán sin repetir código (usando Mustache partials)
- Al iniciar la aplicación web deben existir elementos de ejemplo tanto de las entidades principales como de las secundarias. Esta información se insertará en la base de datos al iniciar la aplicación si se detecta que está vacía.
- El formulario de creación de un elemento y el código que procesa el envío de ese formulario debe reutilizarse para la edición del mismo.

Funcionalidades de la aplicación web en el servidor (backend)

A continuación se detallan las funcionalidades que se deben implementar:

- **Página principal:**
  - Se mostrarán los datos de la base de datos paginados (6 elementos en cada página).
  - Debajo de los elementos aparecerán un enlace por cada una de las páginas y resaltada la página actual (por ejemplo 1, 2, **[3]**, 4…)
  - Aparecerá un enlace para navegar a la página anterior y a la página siguiente.
  - Existirá un buscador basado en el Nombre o Título de la entidad principal.
  - Existirán botones para acceder a los elementos principales de una determinada categoría o

sección o género….

- La implementación de un filtro de búsqueda con diferentes campos no es obligatoria.

- **Página de nuevo elemento:** Se debe incluir funcionalidad del botón de **Guardar**.
  - Al hacer clic en el botón, los datos del formulario se enviarán al servidor para crear un nuevo elemento de la entidad principal, el cual se almacenará en la base de datos.
  - Al pulsar el botón de pulsar el elemento, la aplicación deberá navegar a una página intermedia que muestre un mensaje de confirmación indicando que el elemento ha sido creado con éxito y permitirá al usuario ir a la página de detalle de ese elemento. En caso de que se produzca algún error de validación, la página deberá mostrar esos errores y reenviar al usuario de nuevo a la página de nuevo elemento.
  - La validación de los campos deberá ser la siguiente
    - El nombre o título de la entidad principal no puede ser igual al título de otra entidad previa y debe comenzar con una letra mayúscula
    - Los campos obligatorios no pueden estar vacíos
    - Los datos de cada campo tienen un formato adecuado (fechas válidas, valores numéricos adecuados adecuados y dentro de un rango).
    - La descripción de la entidad principal deberá estar entre un mínimo y un máximo de caracteres.
  - El servidor deberá realizar todas las validaciones, pero el navegador también deberá realizar todas aquellas validaciones que sean posibles con HTML.
  - Para comprobar que las validaciones del servidor se realizan de forma correcta, se pueden desactivar temporalmente las validaciones de los campos del formulario cambiando el tipo de campos en el navegador web (en la vista de inspeccionar el árbol DOM).
- **Página de detalle:**
  - Se debe incluir funcionalidad de los siguientes botones de la página:
    - **Borrar:** Al hacer clic en el botón de **Borrar**, se eliminará el elemento de la entidad principal de la base de datos y mostrará una página intermedia con un mensaje de confirmación indicando que el elemento ha sido eliminado y permitirá al usuario ir a la **Página principal**.
    - **Editar:** Al hacer clic en el botón de **Editar**, se redirigirá a una nueva página que mostrará un formulario con los valores actuales del elemento. Al pulsar “Guardar”, los datos del formulario se enviarán al servidor para actualizar el elemento de la entidad principal, donde serán correctamente validados. Finalmente, la aplicación web mostrará una página intermedia con un mensaje de confirmación indicando que el elemento ha sido actualizado con éxito antes de volver a la **Página de detalle**. Si no se selecciona una imagen en el formulario, el elemento mantendrá la imagen previa (si tuviera).
    - **Crear elementos de la entidad secundaria:** El botón de **Guardar** en el formulario de los elementos de la entidad secundaria deberá enviar los datos del formulario al servidor para que se cree el elemento de la entidad secundaria y se vincule a la entidad

principal. Los datos de la entidad sencundaria también deberán validarse correctamente tanto en el navegador como en el servidor (título no repetido, campos obligatorios, etc.). Al crear el elemento se enviará al usuario a una página con un mensaje de confirmación antes de volver a la **Página de detalle**.

- **Borrar elementos de la entidad secundaria:** Al hacer clic en el botón de **Borrar**, se eliminará el elemento de la entidad secundaria de la base de datos y se enviará al usuario a una página con un mensaje de confirmación antes de volver a la **Página de detalle**.
- **Editar elementos de la entidad secundaria:** Al hacer clic en el botón de **Editar**, se redirigirá a una nueva página que mostrará un formulario con los valores actuales del elemento de la entidad secundaria. Al pulsar "Guardar", los datos del formulario se enviarán al servidor para actualizar el elemento, donde será validados. Finalmente, la aplicación mostrará un mensaje de confirmación indicando que el elemento ha sido actualizado con éxito y le permitirá ir a la **Página de detalle**. Si la entidad secundaria tiene imagen y no se selecciona una imagen en el formulario, el elemento mantendrá la imagen previa (si tuviera).

- **Página de error:** Se debe incluir una nueva página de error a la cual se redirigirá cuando haya algún error al procesar un formulario (por ejemplo: nombre o título duplicado, campos vacíos o datos incorrectos).
  - Se mostrará un mensaje de error que explique el error que ha ocurrido.
  - Debe contener un botón que redireccione a la página adecuada en función del error.

Guía de inicio de la implementación

Se recomienda seguir estos pasos para comenzar con la implementación de la práctica 2:

- **Paso 1)** Ejecutar el ejemplo más sencillo que implemente una aplicación web con express en Node.js y con base de datos MongoDB
  - Instalar Node.js y MongoDB
  - Descargar el código de los ejemplos
  - Ejecutar en la consola los comandos:
    - Para descargar las librerías: `npm install`
    - Para ejecutar la aplicación: `npm start`
  - Abrir en el navegador la URL http://localhost:3000 y verificar que funciona correctamente.
- **Paso 2)** Borrar el contenido de la carpeta "public" del ejemplo y copiar los ficheros .css, .js y las imágenes de la Práctica 1.
- **Paso 3)** Borrar el contenido de la carpeta "views" del ejemplo y copiar los ficheros .html de la Práctica 1.
- **Paso 4)** Cargar la URL http://localhost:3000 y verificar que la página se ve como en la

Práctica 1.

- **Paso 5)** Actualizar los datos de ejemplo para que sean los de la Práctica 1 (incluyendo las imágenes).
- **Paso 6)** Modificar la vista index.html y el router.js para que la página principal muestre los datos de la base de datos en vez de los datos estáticos del HTML.
- **Paso 8)** Repartir el resto del trabajo para la implementación de la práctica entre los miembros del equipo.

Documentación

En el README se deberá añadir una sección “Práctica 2” con la siguiente información:

- **Instrucciones de ejecución:** Se indicará qué pasos deben seguirse para poder descargar el código del repositorio, construir y ejecutar la aplicación. También se debe especificar cuales son los requisitos (versión de Node.js, versión de MongoDB, etc.). Las instrucciones se especificarán preferentemente como comandos a ejecutar por la línea de comandos. En caso de que no sea posible, se indicará cómo instalar/configurar aplicaciones de forma interactiva.
- **Descripción de ficheros:** Se indicará la responsabilidad de cada uno de los ficheros creados por el equipo para implementar la página web.
- **Vídeo demostrativo:** Se creará un vídeo en YouTube de al menos 1 minuto mostrando la funcionalidad de la web.
- **Participación de miembros:** Similar a la de la práctica 1.

Entrega

La entrega de la práctica se compone de los siguientes pasos:

- Crear una “release” en GitHub con el nombre “practica2”
- Subir el .zip del código del repositorio GitHub al aula virtual por uno de los miembros del equipo.

Defensa

Los miembros del equipo mostrarán la aplicación web desarrollada mediante una demostración de las funcionalidades de la aplicación similar a la que han grabado en el vídeo.

- Para facilitar la demostración, el alumno que la realice debe tener claros qué pasos seguir, tener preparados datos de ejemplo, imágenes de ejemplo, etc.
- El profesor podrá preguntar en general a todos los miembros del grupo o en particular a uno de ellos. Todos los alumnos deben conocer cómo funciona cada fichero de código aunque no haya sido desarrollado por él.

Rúbrica de autoevaluación

Está disponible una rúbrica de autoevaluación de la práctica 2 para que los alumnos puedan autoevaluar su propia práctica antes del día de la defensa.

Práctica 3: Mejora de la experiencia del usuario con JavaScript

En la práctica 3 se incluirá funcionalidad para mejorar la experiencia del usuario. La parte de cliente es la que sufrirá más cambios, aunque es posible que también sean necesarios cambios en la parte del servidor.

Características técnicas de la implementación:

- En vez de tener páginas completas para mostrar los errores o mostrar que los formularios se han procesado correctamente (crear o actualizar), se utilizarán un cuadro de diálogo de Bootstrap (que no requiere recargar la página completa).
- La validación de formularios a nivel de cliente y la visualización de errores en los campos del formulario se realizará utilizando las herramientas proporcionadas por Bootstrap. Los errores deberán mostrarse debajo de cada campo del formulario.

Funcionalidades de mejora de experiencia de usuario

A continuación se detallan las funcionalidades que se deben implementar:

- **Página principal:**
  - Sustituir la paginación manual con el scroll infinito. Es decir, las páginas se cargarán a medida que el usuario baje la barra de scroll. Para que se pueda comprobar que funciona correctamente, se deberán tener datos de demo para al menos completar 3 páginas.

- **Página de nuevo elemento:**
  - **Mejoras en la validación del formulario en el cliente:**
    - Se deberá usar bootstrap para visualizar los errores detectados por el formulario a nivel de cliente en vez de la notificación nativa que aparece actualmente.
    - Se deberá implementar lógica en JavaScript en el cliente para validar todos aquellos campos que no se pueden validar usando únicamente HTML. La visualización del error debe aparecer como los demás errores de validación (debajo del campo).
    - Se deberá implementar AJAX para aquellas validaciones que requieran información del servidor para realizar la validación (por ejemplo los títulos duplicados). La visualización del error debe aparecer como los demás errores de validación (debajo del campo).

  - **Envío del formulario:**
    - El envío del formulario se realizará utilizando AJAX mostrando un indicador de

procesamiento (*spinner*) mientras se espera la respuesta.

- **Si el servidor devuelve un error al procesar los datos del formulario:**
  - Se mostrará en un cuadro de diálogo informando del error.
  - El cuadro de diálogo se podrá cerrar para que el usuario pueda corregir los datos erróneos y volver a enviar el formulario.
- **Si el envío del formulario es exitoso:**
  - Se redirigirá al usuario a la página de detalle de ese elemento.

- **Página de detalle:**
  - **Borrado del elemento de la entidad principal:**
    - Al pulsar el botón de “Borrar” en la página de detalle, se mostrará un cuadro de diálogo preguntando al usuario si está seguro de realizar la operación. Si acepta, el elemento de la entidad principal se eliminará utilizando AJAX (mostrando un indicador de procesamiento mientras se espera la respuesta)
    - **Si el servidor devuelve un error al borrar:**
      - Se mostrará en un cuadro de diálogo informando del error.
      - El cuadro de diálogo se podrá cerrar.
    - **Si el servidor no devuelve ningún error:**
      - Se redirigirá al usuario a la página principal.

  - **Creación de un elemento de la entidad secundaria:**
    - En la página de detalle, la creación del elemento de la entidad secundaria al pulsar el botón de “Añadir” se realizará con AJAX (mostrando un indicador de procesamiento mientras se espera la respuesta).
    - Se mejorará la experiencia de validación del formulario en el cliente (similar a la del elemento de la entidad principal):
      - Errores debajo de los campos con Bootstrap
      - Validación con JavaScript donde sea necesaria
      - Validación con AJAX donde sea necesaria.
    - **Si el servidor devuelve un error al procesar los datos del formulario:**
      - Se mostrará en un cuadro de diálogo informando del error.
      - El cuadro de diálogo se podrá cerrar para que el usuario pueda corregir los datos erróneos y volver a enviar el formulario.
    - **Si el envío del formulario es exitoso:**
      - Los campos del formulario deben limpiarse automáticamente.

- Los datos del nuevo elemento se añadirán dinámicamente a la lista de elementos secundarios que ya está cargada en la página, sin necesidad de recargar la página de detalle.

- **Eliminación de la página de editar elemento secundario:**
  - En lugar de tener una página exclusiva para editar un elemento secundario, se editará el elemento en la página de detalle de la entidad principal.
  - Para ello, al pulsar el botón editar, la sección de la página que contiene el elemento se transformará en un formulario que incluye los datos correspondientes a ese elemento secundario.
  - Se deben aplicar las mismas validaciones de los campos del formulario que cuando se crea un elemento.
  - **Si el servidor devuelve un error al procesar los datos del formulario:**
    - Se mostrará en un cuadro de diálogo informando del error.
    - El cuadro de diálogo se podrá cerrar para que el usuario pueda corregir los datos erróneos y volver a enviar el formulario.
  - **Si el envío del formulario es exitoso:**
    - El formulario se sustituirá por la visualización normal con los datos actualizados.

- **Borrado de un elemento de la entidad secundaria:**
  - Al pulsar el botón de “Borrar” en un elemento de la entidad secundaria, este se eliminará utilizando AJAX (mostrando un indicador de procesamiento mientras se espera la respuesta).
  - **Si el servidor devuelve un error al borrar:**
    - Se mostrará en un cuadro de diálogo informando del error.
    - El cuadro de diálogo se podrá cerrar.
  - **Si el servidor no devuelve ningún error:**
    - Se eliminará el elemento de la lista de elementos secundarios dinámicamente sin necesidad de recargar la página.

- **Página de editar un elemento:**
  - Las validaciones y envío del formulario serán iguales a la página para generar un nuevo elemento.
  - El envío del formulario se realizará utilizando AJAX (mostrando un indicador de procesamiento mientras se espera la respuesta).

- Mientras se espera la respuesta del servidor se mostrará un indicador de procesamiento (*spinner*).
- **Si el servidor devuelve un error al procesar los datos del formulario:**
  - Se mostrará en un cuadro de diálogo informando del error.
  - El cuadro de diálogo se podrá cerrar para que el usuario pueda corregir los datos erróneos y volver a enviar el formulario.
- **Si el envío del formulario es exitoso:**
  - Se redirigirá al usuario a la página de detalle de ese elemento.

- **Formularios con imagen:**
  - Se deberá previsualizar la imagen seleccionada.
  - Se deberá poder eliminar la imagen seleccionada.
  - Se permitirá arrastrar la imagen desde el explorador de ficheros a un área de la página.
  - Si el formulario es de edición y la imagen se quita y se envía el formulario, la imagen deberá borrarse en el servidor.

Documentación

En el README se deberá añadir una sección “Práctica 3” con la siguiente información:

- **Instrucciones de ejecución:** Se copiará la sección de la práctica 2 y se actualizará si fuera necesario.
- **Descripción de ficheros:** Se copiará la sección de la práctica 2 y se actualizará con los nuevos ficheros y los cambios realizados en los existentes.
- **Vídeo demostrativo:** Se creará el vídeo de nuevo para que se muestren las nuevas funcionalidades.
- **Participación de miembros:** Similar a la de la práctica 2.

Entrega

La entrega de la práctica se compone de los siguientes pasos:

- Crear una “release” en GitHub con el nombre “practica3”
- Subir el .zip del código del repositorio GitHub al aula virtual por uno de los miembros del equipo.

Defensa

Los miembros del equipo mostrarán la aplicación web desarrollada mediante una demostración de las funcionalidades de la aplicación similar a la que han grabado en el vídeo.

Para facilitar la demostración, el alumno que la realice debe tener claros qué pasos seguir, tener preparados datos de ejemplo, etc.

El profesor podrá preguntar en general a todos los miembros del grupo o en particular a uno de ellos. Todos los alumnos deben conocer cómo funciona cada fichero de código aunque no haya sido desarrollado por él.

Rúbrica de autoevaluación

Está disponible una rúbrica de autoevaluación de la práctica 3 para que los alumnos puedan autoevaluar su propia práctica antes del día de la defensa.