

Prácticas

Desarrollo Web Entorno

Servidor

Trimestre 2

Índice

1. Aplicación Web de tipo CRUD en Python/Flask	4
2. Aplicación Web de tipo API REST en Python/Flask	5
3. Aplicación Web de tipo CRUD en Node.js/Express	6
4. Aplicación Web de tipo API REST en Node.js/Express	7

Las siguientes prácticas son de carácter **individual**. Se entregará un documento en formato **Word** o, *preferiblemente*, en **Pdf**. En cada una de ellas se indicará el procedimiento empleado para su resolución mediante **texto** y **capturas de pantalla**. Es importante que en las capturas se aprecie perfectamente nombres de usuarios, textos significativos, hosts, urls, dominios, etc. **De esta forma se diferencia claramente el trabajo de un alumno con respecto al de otro, así como de personas ajenas.**

En el caso de las aplicaciones ubicadas en un servidor de hosting, se debe además indicar claramente la **URL** de acceso a las mismas.

Estas prácticas se **entregarán antes** del **Miércoles, 1 de Marzo de 2023 en el Aula Virtual**.

1. Aplicación Web de tipo CRUD en Python/Flask

Esta aplicación se desarrollará en Windows y consistirá en un **CRUD** (Create-Read-Update-Delete). Por tanto, accederá a una base de datos para realizar estas operaciones.

La base de datos tendrá una o dos tablas. En el caso de dos tablas, estarán relacionadas entre sí de la forma 1:N. Como ejemplos, podrían ser marcas/coches, ciclos/estudiantes, categorías/libros, etc.

Además, al menos una tabla tendrá una columna imagen. Esta columna puede ser un String que contendrá el nombre del archivo imagen. Y el archivo imagen se almacenará en la carpeta *static/img* o similar.

El sistema gestor de BD puede ser MySQL o MongoDB.

En la aplicación se mostrarán claramente las vistas necesarias para listar los datos, darlos de alta y modificación mediante formulario, así como la opción de eliminación.

Se subirá a un servidor de hosting (Vercel u otro que se encuentre). Se pondrá en el documento en **formato texto** (no captura de pantalla) la URL completa de acceso a la misma. No será necesario explicar el procedimiento de subida al hosting en este documento, pero sí realizar las capturas de pantalla necesarias para poder utilizarlas en la práctica del módulo de Despliegue de Aplicaciones Web.

2. Aplicación Web de tipo API REST en Python/Flask

Esta aplicación se desarrollará en Windows y consistirá en un **API REST** que implemente los cuatro verbos del protocolo HTTP: GET, POST, PUT, DELETE, es decir los servicios web **RESTFul**.

La tabla sobre la que actuará será la misma que la del ejercicio anterior. Y en el caso de haber utilizado dos tablas, será aquella que contenga la columna imagen.

Como API REST que es esta aplicación tendrá al menos estas rutas y suponiendo que el recurso son *libros*.

/libros para listar el contenido (método GET)

/libros para dar de alta un elemento (método POST. En este caso será necesario incorporar en el cuerpo de solicitud del mensaje HTTP los datos en formato JSON con los valores a añadir)

/libro/{id-libro} para editar el contenido de un libro con id=id-libro (método GET)

/libro/{id-libro} para eliminar el elemento con id=id-libro (método DELETE)

/libro/{id-libro} para modificar el elemento con id=id-libro (método PUT. En este caso será necesario incorporar en el cuerpo de solicitud del mensaje HTTP los datos en formato JSON con los valores a modificar)

Después de realizar la operación, la aplicación devolverá un mensaje en formato JSON. Puede ser de error. Y en el caso de éxito, los datos que correspondan. Si es listar, el contenido en formato JSON de la tabla. Si es dar alta, modificación o eliminación, los datos en formato JSON del elemento implicado en la operación.

Para probar los datos, se utilizará la utilidad **Thunder Client** de *Visual Studio Code* o la herramienta gratuita **Postman**.

También sería posible hacerlo mediante llamadas **fetch** desde *JavaScript*.

Se subirá a un servidor de hosting.

3. Aplicación Web de tipo CRUD en Node.js/Express

Esta aplicación tendrá exactamente la misma funcionalidad que la del ejercicio nº 1, pero implementada en **Node.js** y el framework **Express** y el SGBD **MySQL**.

Se subirá a un servidor de hosting que aún está por determinar.

4. Aplicación Web de tipo API REST en Node.js/Express

Esta aplicación tendrá exactamente la misma funcionalidad que la del ejercicio nº 2, pero implementada en **Node.js** y el framework **Express** y el SGBD **MySQL**.

Se subirá a un servidor de hosting que aún está por determinar.