

# PROYECTO 1 RESTAURANTE (APP WEB)

Proyecto principal donde se realizará una aplicación web de comercio de restauración (A pesar de que las funcionalidades serán similares, cada estudiante deberá tener el suyo propio) . El sitio web de restauración simulará la creación de un aplicativo web que permita a los usuarios poder seleccionar productos, realizar pedidos, etc. Dispondrán de un panel de usuario donde podrán modificar sus datos personales y ver sus pedidos realizados. La página web ofrecerá si se compra una cantidad determinada de productos.

## Tecnologías

	<b>Editor de texto</b>		<b>Entorno de desarrollo</b>
	<b>PHP</b>		<b>HTML</b>
	<b>CSS</b>		<b>FTP</b>
	<b>Alojamiento</b>		<b>git repositorios</b>
	<b>Bases de Dades</b>		

## Temporización:

La fecha límite para subir el proyecto a **GIT** será el **martes 07/01/2026**; a partir de esa fecha, cualquier cambio realizado no será evaluado.

Las presentaciones de los proyectos se realizarán entre el **08/01/2026** y el **15/01/2026**. Dispondréis de una cuadrícula con la hora y la fecha de cada una de las presentaciones.

## Requisitos de cada parte

### WEB

El usuario mediante sesiones se podrá ir desplazando por los diferentes paneles de la web para ir seleccionando sus productos, que se irán acumulando para posteriormente confirmar el pedido final.

Una vez realizado el pedido, cuando vuelva acceder el usuario, le saldrán los datos de su último pedido, La parte del usuario deberá realizarse únicamente mediante PHP, HTML y CSS.

El proyecto deberá estructurarse como se explica en clase, teniendo en cuenta el modelo vista controlador.

El código deberá estar lo máximo optimizado posible y ser legible, eso implica que deberá:

- Estar organizado bajo el patrón Modelo-Vista-Controlador. En la memoria del proyecto deberá contener un diagrama de clases y cómo interactúan entre ellas.
- Correcta integración php-html
- Programado orientado objeto: haciendo uso de clases, polimorfismo y clases estáticas adicionales.
- Comentado y siguiendo una estructura lógica.
- Encriptación de Pwd al tratar la comunicación de la contraseña con la base de datos.
- El usuario admin podrá realizar operaciones CRUD con productos y pedidos.
- La web deberá incluir ofertas. Estas ofertas deberéis pensarlas individualmente.

A su vez, habrá un rol de usuario “Administrador” que dispondrá de funciones adicionales en un panel de administración. Este panel de administración estará hecho exclusivamente con HTML, CSS y Javascript.

- Ver todos los pedidos
- Filtrar y/o ordenar pedidos por usuario, fecha y precio
- Poder crear, modificar y eliminar pedidos y productos.
- Ver un historial de Logs con todas las acciones que se han realizado en la aplicación web.
- Ver los precios de los pedidos en distintas monedas, dependiendo de la selección del administrador. (utilizar <https://freeCurrencyAPI.com/> o similar)

En el panel de administración habrá que utilizar como mínimo:

- DOM (obtener, crear y/o modificar elementos HTML)
- Eventos (todos los clics a botones, envíos de formularios, escrituras en inputs, etc. son gestionados con eventos de JS)
- Funciones de orden superior (map, filter, reduce y/o otras)

- Clases y Objetos. Habrá que programar al menos 2 clases, de modo que la programación sea orientada a objetos.
- Storage (localStorage o sessionStorage). El alumno decidirá y argumentará dónde y cómo aplicarlo. Se valorará la creatividad y la usabilidad.
- Peticiones asíncronas. Tanto los datos que se envían de formularios, como los datos que se obtienen del backend se realizarán con peticiones asíncronas (fetch o axios).
- Habrá que utilizar como mínimo una API externa

De manera adicional, cada estudiante podrá añadir funcionales adicionales que aporten valor al proyecto, algunos ejemplos de ellos podría ser:

- Productos complejos en los que se puedan seleccionar ingredientes.
- Permitir al usuario recuperar directamente el último pedido realizado.
- Hacer uso del autoloader para que el código sea más limpio.

## DISEÑO

Se deberán hacer los siguientes requisitos respecto al diseño de la web:

- Creación de un **Moodboard** que tiene que responder a color, forma, tipografía y composición según la temática base.
- Creación de un **prototipado** de diseño para escritorio, para pantallas de 1080px.
- Creación **página home** con header, 4 secciones de información (o banner y 3 secciones) y footer. Esta ha de ser responsive para móviles (430x932).
- Creación **Página productos o carta** con todos los productos que se pueden añadir.
- Creación **Páginas del proceso de compra** con los productos añadidos al carrito y opción de finalizar compra con precio e impuestos.
- Uso de **imágenes** correcto (prototipado y web final)
- Uso **tipográfico** adecuado (prototipado y web final)
- Cumplimiento con las siguientes **pautas de diseño**:
  - Principio de Semejanza (prototipado y web final)
  - Principio de Continuidad (prototipado y web final)
  - Principio de Simetría (prototipado y web final)
  - Principio de Proximidad (prototipado y web final)
  - Principio de Contraste (prototipado y web final)
- Uso de **HTML, CSS y BOOTSTRAP** para la creación final del diseño web.
- La página home de la web deberá ser responsive, debe adaptarse tanto para escritorio como para móvil.

## BBDD

Siguiendo los requisitos de la sección de web, se espera que se cree una base de datos para almacenar la información a mostrar en la web, la base de datos deberá estar en un contenedor Docker. Cada base de datos de cada proyecto puede variar, de forma que no necesariamente el diseño de todas será el mismo. En la documentación se pide el diagrama entidad-relación y relacional de la base de datos, primer paso necesario para cualquier despliegue y desarrollo de base de datos.

## SERVICIOS WEB

Los estudiantes deberán complementar su IDE con los plugins que consideren de utilidad y oportunos, instalar y configurar los diferentes servicios o software necesarios para poder ejecutar la web a nivel local (incluido depurador PHP). Posteriormente cada alumno deberá elegir un servidor web para configurarlo y administrarlo. Se deberá poder subir los sitios locales vía FTP y disponer de un dominio público. Se usará el repositorio GIT para poder ir teniendo un control de los avances del proyecto.

## USO DE GITHUB

Se deberá utilizar GitHub. Todo lo desarrollado en el proyecto debe estar subido al repositorio (memoria, código, base de datos, etc...). Todos los profesores deberán tener acceso a los repositorios de Github ([añadir el link del proyecto aquí](#)). Se evaluará que:

- Se hagan correcciones completas abordando todos los problemas
- No se trabaje directamente con la rama principal (main), y se creen ramas de trabajo por funcionalidades, tareas o correcciones
- Que se haya fusionado, exitosamente con la rama principal (main) todos los push

## DOCUMENTACIÓN

Se deberá entregar la memoria final del proyecto en la fecha de entrega del mismo. La documentación debe completarse durante el transcurso del proyecto. Se espera que la memoria final contenga explicaciones detalladas, con un vocabulario adecuado, sin faltas de ortografía y bien estructurada. En la entrega de la memoria, se esperan los apartados que se muestran en [este documento](#). Además, deben incluirse: *título, índice, bibliografía o webgrafía y anexos*.