

04/11/2025

Actividades 4 Aplicaciones Web

Ciclo Formativo de Grado Medio en SMR

Método de entrega

Todas las entregas deberán además de incluir la carpeta con la página web comprimida (ZIP o RAR), también un documento WORD que explique y responda a cada una de las partes descritas en este documento.

Tenéis además tanto en el aula virtual como en el canal de youtube Eurekamps los videos tutoriales de las clases para que podáis usar en el proceso de desarrollo.

Introducción

Esta cuarta y última actividad del bloque de HTML, CSS y JavaScript consiste en crear una **aplicación web completa e interactiva**, integrando todo lo aprendido en las actividades anteriores.

Tu web deberá utilizar **estructuras de datos (arrays y objetos)**, **funciones reutilizables**, **almacenamiento local (localStorage)**, y **peticiones asincrónicas (fetch)**, además de incorporar un **diseño responsive y accesible**.

El proyecto final evaluará tu dominio de HTML, CSS y JavaScript de forma conjunta.

Objetivos de aprendizaje

- Aplicar técnicas avanzadas de programación con JavaScript.

- Manipular el DOM de forma dinámica utilizando estructuras de datos.
- Almacenar y recuperar información con localStorage.
- Realizar peticiones asincrónicas (fetch) para cargar datos externos o JSON.
- Desarrollar una interfaz web interactiva, responsive y accesible.
- Documentar el proyecto de forma técnica y visual en un documento Word.

Requisitos técnicos generales

- Estructura clara con index.html, styles.css, script.js, y carpeta /assets.
- Código JavaScript separado en funciones y comentado.
- Diseño responsive (media queries o flex/grid).
- Navegación funcional y accesible.
- Al menos tres secciones interactivas (DOM dinámico o datos).

Planificación y estructura base

1. Define el **tema del proyecto** (puede ser una web de recetas, tareas, productos, películas, agenda, etc.).
2. Diseña la **estructura HTML y el esquema de navegación** (mínimo 3 páginas o secciones).
3. Aplica una maquetación moderna con **CSS Grid o Flexbox**.

4. Crea los contenedores donde se mostrará la información dinámica generada por JavaScript.
5. Documenta con un esquema (puede ser un boceto o wireframe).

JavaScript avanzado: arrays, objetos y DOM dinámico

1. Crea un **array de objetos** con datos simulados (ejemplo: lista de productos, tareas, usuarios, recetas...).
2. Usa JavaScript para **mostrar dinámicamente los datos** en la web mediante el DOM.
3. Permite **añadir, eliminar o filtrar elementos** del array desde el navegador (inputs o botones).
4. Guarda los datos en localStorage y permite recuperarlos al recargar la página.
5. Añade **efectos visuales o animaciones** al actualizar la información (añadir o eliminar).

Fetch y datos externos

1. Crea o descarga un archivo data.json con información de ejemplo.
2. Utiliza fetch() para **cargar los datos del JSON** y mostrarlos en la web.
3. Muestra mensajes o indicadores de carga (“Cargando datos...”, “Error al conectar”).
4. Crea una función que permita **buscar o filtrar** entre los datos cargados.

5. (Opcional) Conecta tu proyecto con una **API pública sencilla** (ejemplo: frases, clima, imágenes, Pokémon...).

Integración, validación y usabilidad

1. Añade un **formulario de interacción** (registro, búsqueda, comentarios, etc.) validado con JavaScript.
2. Implementa **mensajes dinámicos** (alertas personalizadas, notificaciones o modales).
3. Mejora la usabilidad y accesibilidad:
 - Navegación por teclado.
 - Colores y contrastes adecuados.
 - Feedback visual en botones y formularios.
4. Añade animaciones o transiciones con CSS y JS (por ejemplo, un mensaje flotante al guardar o eliminar).

Cierre y documentación

1. Revisa la funcionalidad general:
 - No deben aparecer errores en la consola.
 - Todo el contenido debe ser accesible y legible.
2. Crea un **documento Word** con:
 - Capturas antes y después de aplicar JavaScript.
 - Breve descripción de cada función y bloque de código.

- Reflexión final sobre el proceso de aprendizaje.
3. Comprime el proyecto final (.zip) con todos los archivos y sube ambos (ZIP + Word).

Rúbrica de evaluación (100 puntos)

| Criterio | Puntos |
|--|---------------|
| Estructura y diseño (HTML + CSS responsive) | 15 |
| Interactividad y manipulación del DOM | 25 |
| Uso de arrays, objetos y localStorage | 20 |
| Carga de datos con Fetch o JSON | 15 |
| Validación y usabilidad (accesibilidad, formularios) | 15 |
| Documentación y reflexión final | 10 |

Bonus: +5 pt por integración de una API externa o diseño avanzado.

Entregables

1. Carpeta comprimida (.zip) con todos los archivos del proyecto.
2. Documento Word con:
 - Capturas de cada fase.
 - Explicaciones de las funciones clave.

- Reflexión final.
3. (Opcional) Publicación en GitHub Pages o Netlify.

Evaluación

Con esta práctica se desarrollan y evidencian parcialmente los criterios a), b), c) d), e), f), g), h), i), j) y k) del RA2.

Instrumentos

- Código funcional y estructurado.
- Documento Word explicativo.
- Observación del proceso en clase.

NOTA: Este proceso es el mismo que se está realizando en los en clase y publicado en el aula virtual en videos, por lo que se recomienda ver los videos paso a paso.