

# PROGRAMACIÓN WEB I



## MANUAL TÉCNICO

ESTUDIANTES:

- Cussi Suxo Hugo Camilo

CARRERA:

- Ing. de sistemas informáticos

GRUPO: “A”

SEMESTRE: 2do

DOCENTE:

- Ing. Miranda Ordonez Henry

02/2025

# Manual Técnico

**Portal Web “Noticiando la Noticia”**

**Versión:** 1.0

**Autor:** Hugo Camilo Cusisuxo

**Fecha:** 7 de diciembre de 2025

**Tecnologías utilizadas:** HTML5, CSS3, JavaScript (ES6), RSS, DOM, Grid y Flexbox

## 1. Descripción General

“Noticiando la Noticia” es un portal web de noticias que presenta información actualizada de fuentes RSS externas (en este caso BBC Mundo). El sitio tiene un diseño responsive, visualmente atractivo, y permite:

- Visualizar noticias destacadas, secundarias y una lista de noticias principales.
- Filtrar noticias por categorías: Internacional, Economía, Tecnología, Deportes, Cultura.
- Suscribirse a un boletín de noticias (simulado con alerta).
- Abrir noticias en un **modal** interno sin salir del portal.
- Navegación en múltiples páginas con **paginación dinámica**.
- Actualizaciones automáticas cada 30 minutos.

El proyecto está pensado como una **portada tipo periódico**, con énfasis en UX moderno, diseño responsive y facilidad de lectura.

## 2. Arquitectura del Sistema

El portal se organiza en tres capas principales:

1. **HTML:** La estructura semántica y contenedores de noticias, aside y footer.
2. **CSS:** El diseño visual, layout con Grid y Flexbox, estilos de tarjetas, menú, responsive.
3. **JavaScript:** La lógica dinámica, manejo de RSS, paginación, modal, menú lateral y actualización en vivo.

## Estructura HTML Principal

- **Header (<header>)**
  - Logo (logoTexto) y menú de navegación (listaMenu).
  - Botón de menú para dispositivos móviles (btnMenu).
- **Main (<main>)**
  - Contenedor principal de noticias (contenedor-noticias).
  - Paginador (paginador) con botones “Anterior” y “Siguiente”.
  - Aside derecho (columnaDerecha) con:
    - Actualizaciones en vivo (tarjetaVivo).
    - Noticias más leídas (tarjetaMasLeidas).
    - Temas populares (tarjetaTemas).
    - Newsletter (tarjetaNewsletter).
- **Footer (<footer>)**

Cuatro columnas: información del portal, secciones, legal y contacto.

## 3. Diseño y Estilos (CSS)

### Variables y Tipografía

```
:root {  
    --color-fondo: #f0f0f0;  
    --color-texto: #333333;  
    --color-header: #E64A19;  
    --color-destacado: #FFB703;  
    --fs-base: 16px;  
    --fs-sm: 0.95rem;  
    --fs-xs: 0.9rem;  
    --fs-h3: 1.25rem;  
    --fs-h2: 2rem;  
}
```

Variables globales para colores y tamaños de fuente, permitiendo consistencia y fácil mantenimiento.

## Layout

- **Header:** sticky, siempre visible, con menú flex para alinear logo y enlaces.
- **Grid principal (zonaContenido):**

```
display: grid;
grid-template-columns: 1fr 340px;
grid-template-areas:
  "dest dest"
  "sec sec"
  "list aside"
  "paginador paginador";
gap: 24px;
```
- **Aside:** Ocupara la columna derecha en desktop, desaparece en páginas >1.
- **Responsivo:** media queries adaptan el grid a una columna en dispositivos <700px.

## Tarjetas de Noticias

- **Destacada (destacadaCard):** Tiene una imagen grande, sombra y animación al hover.
- **Secundarias (cardMini):** Contiene tres columnas, tamaño reducido.
- **Lista principal (cardNoticia):** flex horizontal, imagen a la izquierda, contenido a la derecha.
- Animaciones suaves para hover: transform: translateY(-x) y box-shadow.

## Menú y Botón Móvil

- **btnMenu** se muestra solo en pantallas <900px, alternando la clase .show en listaMenu.

## Paginador

- **Flexbox** centrado, botones estilizados y deshabilitados según contexto ( :disabled ).

## 4. Funcionalidades JavaScript

### Menú Lateral

```
btnMenu.addEventListener("click", () => listaMenu.classList.toggle("show"));
```

- Alternara la visibilidad del menú en dispositivos móviles.

### Suscripción Newsletter

Validara el email, muestra mensaje de agradecimiento con `alert()`, y limpiara el input.

### Manejo de RSS

Se utiliza **proxy** (`api.allorigins.win`) para evitar CORS (para evitar el bloqueo).

- `fetch` obtiene el feed, luego `DOMParser` convierte el XML en objetos noticias.
- Normaliza datos: título, descripción, enlace, imagen y categoría.

### Clasificación de Noticias

La función `clasificarTexto()` detecta la categoría según palabras clave en el título o descripción.

### Paginación

- **Página 1:** Tiene destacada, 3 secundarias y 4 lista.
- **Página 2+:** Tiene una lista que muestra 5 noticias por página (`ITEMS_PER_PAGE`).
- **Botones** `btnPrev` y `btnNext` cambiarian “`páginaActual`” y vuelven a renderizar.

### Modal de Noticias

- Modal creado dinámicamente (`ensureModal()`) al primer click.
- Contiene título, imagen, descripción y botón para abrir la fuente original.
- Se puede cerrar con el botón “X” y con el botón “Cerrar” o clic fuera del modal.

### Actualización en Vivo

- `actualizarTarjetaVivo()` muestra la hora de la última actualización, ya que la pagina se actualiza cada cierto tiempo.
- Feed se refresca automáticamente cada 30 minutos (`setInterval(fetchAndRender, UPDATE_INTERVAL)`).

### Filtrado por Categoría

- Click en menú o chips de temas (`.tema`) cambia `categoriaSeleccionada` y reinicia `páginaActual`.
- Renderiza solo noticias de esa categoría.

## 5. Datos y Eventos

1. Página carga: fetchAndRender() obtiene RSS y construye un array noticias.
2. Página 1: muestra destacada, secundarias y lista principal.
3. Usuario hace click en noticia: abre el modal (abrirModalNoticiaByIndex).
4. Usuario cambia página: El paginador actualiza la lista y aside.
5. Usuario filtra por categoría: renderPagina() filtra y actualizara el contenido.
6. Cada 30 minutos: feed se refrescara automáticamente.

## 6. Buenas Prácticas

- Uso de variables CSS (:root) para colores y tipografías mucho mas rápidas y eficientes.
- Layout con Grid y Flexbox para adaptabilidad, claridad visual y estética.
- Separación de responsabilidades: HTML (estructura), CSS (presentación), JS (lógica).
- Evitar duplicados en feed RSS (Set de IDs).
- Normalización de texto para evitar errores con acentos o mayúsculas.
- Eventos delegados y control de UI responsive.

## 7. Posibles Mejoras Futuras

- **Integrar** un backend para guardar suscripciones reales.
- **Manejar** errores de carga RSS con mensajes más amigables.
- **Añadir soporte** para múltiples feeds y combinación de categorías.
- **Animaciones** más suaves con transition en CSS para modal y cards.
- **Optimización** de imágenes y lazy loading para rendimiento.
- **Filtrar** mejor el contenido, debido a que no es muy preciso.
- **Noticias** en la misma pagina, aun no se logra, debido a eso se creo el modal.

## **8. Conclusión**

El proyecto “Noticiando la Noticia” representa un ejemplo integral de cómo construir un portal web de noticias moderno, interactivo y responsivo, empleando únicamente tecnologías del lado del cliente como HTML, CSS y JavaScript. A través de la integración de feeds RSS externos, el sitio logra ofrecer contenido actualizado de manera dinámica, permitiendo al usuario acceder a noticias relevantes.

El diseño visual combina tarjetas destacadas, secundarias y listas, optimizadas para una lectura cómoda y jerarquizada, mientras que la implementación de un modal interno mejora la experiencia del usuario al ofrecer la posibilidad de leer noticias completas sin perder el contexto de la portada.

El sistema también demuestra buenas prácticas de desarrollo web moderno, tales como el uso de variables CSS para consistencia de diseño, Grid y Flexbox para un layout flexible y responsivo, y métodos de normalización y filtrado de datos para manejar contenido heterogéneo proveniente de múltiples fuentes.

En términos de escalabilidad y mantenimiento, la separación clara entre estructura (HTML), presentación (CSS) y lógica (JS) permite futuras extensiones, como la integración de múltiples fuentes, almacenamiento de suscripciones reales, o mejoras de rendimiento mediante lazy loading y optimización de imágenes.