

Documento técnico del proyecto **RETROTERM.AI HUB**, una plataforma web de estilo retro futurista que integra diseño, desarrollo y comunicación entre usuarios. Se presentan su estructura, fundamentos conceptuales y lineamientos de implementación desde la versión inicial (v0.1).

INFORME WEB

CDM Fp ASIR_2025/26

Iván Brihuega Crespo

DOCUMENTACIÓN TÉCNICA DEL PROYECTO RETROTERM.AI HUB

1. Introducción

RETROTERM.AI HUB es un entorno web en desarrollo concebido como un **sistema retrofuturista de administración, desarrollo y comunicación** entre usuarios. Su concepto combina la estética de las interfaces de terminal clásicas de los años 80–90 con la funcionalidad moderna del desarrollo web actual.

El presente documento describe **el diseño técnico, la estructura de implementación y la justificación conceptual** del proyecto en su estado actual (versión 0.1). Su finalidad es servir como **registro técnico y guía de mantenimiento**, facilitando futuras ampliaciones y documentando las decisiones de diseño que sustentan la arquitectura actual.

2. Objetivos del proyecto

2.1 Objetivo general

Desarrollar una plataforma web modular, accesible y visualmente coherente con el estilo retrofuturista, que funcione como punto de acceso (“HUB”) para distintas funcionalidades:

- Modo desarrollo (editor de código).
- Asistente de inteligencia artificial.
- Red de proyectos compartidos.
- Sistema de publicación de artículos informativos.

2.2 Objetivos específicos

1. Crear una **estructura de navegación fluida** sin necesidad de recarga de página, utilizando únicamente HTML y CSS (sin dependencias JavaScript iniciales).
2. Implementar una **identidad visual unificada** inspirada en consolas CRT: fondo negro, tipografía monoespaciada, paleta verde neón, efectos de brillo y gradientes radiales.
3. Asegurar la **legibilidad, accesibilidad básica y semántica HTML5** para permitir futura indexación, lectura por asistentes y escalabilidad del proyecto.
4. Mantener una **separación estricta entre contenido (HTML) y estilo (CSS)** para simplificar mantenimiento y versiones futuras.
5. Garantizar que el código sea **modular, extensible y documentado**, sirviendo de base para un ecosistema de submódulos integrados (p. ej. red de proyectos, artículos, IA).

3. Fundamentación conceptual y técnica

3.1 Enfoque retrofuturista

El diseño visual y conceptual parte del principio del **retrofuturismo digital**, una corriente estética que combina referencias visuales de la informática temprana (monitores de fósforo verde, interfaces ASCII) con paradigmas tecnológicos contemporáneos.

La elección de:

- **Colores hexadecimales verdes (#00ff66 y #66ffb3),**
- **Tipografía monoespaciada ('Courier New'),**
- **Efectos de neón mediante `text-shadow` y `box-shadow`,**
responde a un objetivo de identidad: recrear la sensación de “terminal inteligente” pero con la ergonomía de un entorno moderno y responsivo.

3.2 Fundamento técnico

El proyecto sigue los principios del **diseño web progresivo** y **mínimo viable**, iniciando con:

- HTML semántico estructurado,
- una hoja de estilo CSS centralizada,
- navegación mediante anclas internas (`#id`) y pseudo-clases (`:target`),
- sin dependencia inicial de JavaScript.

Este enfoque prioriza la **legibilidad, mantenimiento y accesibilidad**, permitiendo incorporar progresivamente interactividad avanzada sin comprometer la base estática.

4. Estructura general del sistema

4.1 Organización de archivos

```
/Proyecto RETROTERM.AI/  
├── index.html           -> Página principal (HUB)  
├── articulos.html      -> Página secundaria de artículos publicados  
├── estilos.css         -> Hoja de estilos unificada  
└── /Imagenes/         -> Recursos visuales
```

4.2 Jerarquía de secciones en `index.html`

Cada bloque principal se define mediante el elemento `<section>` con identificador único y clases compartidas:

Sección	ID	Descripción
Inicio	<code>#inicio</code>	Pantalla principal del HUB
Desarrollo	<code>#desarrollo</code>	Modo de edición de código
Asistente	<code>#asistente</code>	Interacción con IA
Red	<code>#red</code>	Espacio de proyectos compartidos
Artículos	<code>#articulos</code>	Redactor interno de artículos

Clases estructurales:

- `.pantalla` -> unidad modular de contenido (oculta por defecto).
- `.activa` y `:target` -> controlan la visibilidad dinámica mediante CSS.
- `.menu`, `.opcion`, `.volver`, `.pie` -> elementos funcionales de navegación y contexto.

5. Funcionamiento de la navegación

La navegación interna se implementa **sin JavaScript**, utilizando el selector `:target` de CSS.

Cada enlace (``) actualiza el *fragment identifier* de la URL, lo que activa el pseudo-selector `:target` en la sección correspondiente:

```
.pantalla { display: none; }  
:target { display: block; }
```

Esto produce un efecto de **pestañas dinámicas o pantallas intercambiables**, que simulan una aplicación de escritorio retro, sin recargar la página.

6. Diseño visual y principios de estilo

6.1 Paleta cromática

Elemento	Color	Función visual
Fondo base	#000000	Oscuridad absoluta, referencia CRT
Texto principal	#00ff66	Verde neón, legible sobre negro
Subtítulos / enlaces	#66ff66	Variante más clara para jerarquía visual
Fondos secundarios	rgba(0,20,0,0.8)	Profundidad translúcida
Sombras / brillo	box-shadow: 0 0 15px #00ff66	Simulación de resplandor fosfórico

6.2 Tipografía

Se emplea la familia 'Courier New', monospace, elegida por su:

- uniformidad entre caracteres (alineación tipo terminal),
- alto contraste visual,
- evocación estética del código fuente clásico.

6.3 Disposición y estructura visual

- **Flexbox** en el menú principal (.menu) garantiza centrado, adaptación y distribución equitativa de opciones.
- **Gradiente radial** en .pantalla (radial-gradient(circle at center, #001800 0%, #000 80%)) introduce profundidad y textura.
- **Sombreado y bordes verdes** consolidan el lenguaje visual “hardware digital”.

7. Segunda fase: sección de artículos (articulos.html)

Esta página constituye el **módulo informativo** del sistema.

Está diseñada para alojar artículos en formato semántico HTML5 mediante el uso de <article>, <h3>, y .

Cada artículo contiene:

- título principal,
- párrafos descriptivos,
- listas con características u objetivos,
- imágenes centralizadas con borde verde y sombra.

Se conserva la coherencia con el HUB principal mediante las clases `.pantalla`, `.cabecera`, `.volver`, y se integran nuevas (`.lista-articulos`, `.articulo`, `.imagen-articulo`).

El resultado es una interfaz uniforme que facilita tanto la lectura como la navegación retroactiva hacia el HUB (``).

8. Hoja de estilos unificada (`estilos.css`)

El CSS cumple los principios de **modularidad visual** y **consistencia temática**. Los bloques principales son:

1. **Configuración base (`body`)**
-> define tipografía, colores, márgenes, comportamiento de scroll.
2. **Pantallas (`.pantalla`, `.activa`, `:target`)**
-> estructura modular para secciones visibles según la navegación.
3. **Cabecera (`.cabecera`, `h1`, `.subtitulo`)**
-> jerarquía visual, centrado y efectos de texto.
4. **Menú (`.menu`, `.opcion`)**
-> uso de Flexbox, hover animado, equilibrio entre estética y usabilidad.
5. **Enlaces (`.volver`)**
-> consistencia visual con botones del menú.
6. **Pie de página (`.pie`)**
-> información de autoría y separación visual mediante borde superior.
7. **Artículos (`.lista-articulos`, `.articulo`, `.imagen-articulo`)**
-> tarjetas temáticas, uso de sombras, fondos translúcidos y formato legible.

La hoja actual se ha construido sin preprocesadores (como Sass), priorizando la **simplicidad, claridad y bajo coste de mantenimiento**.

9. Consideraciones de accesibilidad y usabilidad

Aunque el diseño prioriza la fidelidad estética, se han tenido en cuenta aspectos mínimos de accesibilidad:

- Contraste texto/fondo suficiente (ratio superior a 5:1 en elementos principales).
- Uso correcto de etiquetas semánticas (`<header>`, `<nav>`, `<section>`, `<article>`).
- Inclusión de atributos `alt` en imágenes.
- Navegación por teclado mediante enlaces ancla (`href="#id"`).

En fases futuras se prevé:

- incorporar `aria-label` en menús y botones,
- mejorar la navegabilidad con `tabindex`,
- permitir cambio de tema para usuarios con baja visión (modo claro alternativo).

10. Perspectiva de desarrollo y futuras ampliaciones

El presente estado (v0.1) constituye la base funcional estática del sistema.

Las etapas siguientes incluirán:

1. **Fase 2 : Interactividad básica:**
Integración de un archivo `app.js` para gestionar visibilidad de secciones, almacenamiento local (`localStorage`) y animaciones.
2. **Fase 3 :Backend y persistencia real:**
Conexión a base de datos o API REST para almacenar artículos, usuarios y proyectos.
3. **Fase 4 : Red de usuarios:**
Implementación de un sistema de perfiles y proyectos compartidos.
4. **Fase 5 : Optimización y accesibilidad total:**
Cumplimiento completo de WCAG 2.1, uso de variables CSS (`--color-primario`), y carga de fuentes personalizadas retro (p. ej. “VT323”).
5. **Fase 6 : Integración IA:**
Conexión del módulo de asistente (`#asistente`) con un modelo de lenguaje o API que permita interacción contextual.

11. Conclusión

El desarrollo de **RETROTERM.AI HUB** responde a una estrategia técnica que prioriza:

- la **claridad semántica**,
- la **identidad visual coherente**,
- la **navegación fluida sin recargas**,
- y la **posibilidad de expansión modular**.

El uso de **HTML5 semántico** y **CSS3 puro** garantiza compatibilidad universal, bajo consumo de recursos y fácil mantenimiento.

La documentación que aquí se presenta constituye la base de referencia para futuras versiones, auditorías de código y procesos de mejora continua.

12. Referencias técnicas y normativas

- W3C. *HTML5 Specification*.
- W3C. *CSS Level 3 (CSS3) Recommendation*.
- WCAG 2.1. *Web Content Accessibility Guidelines*.
- Google Developers. *Responsive Web Design Basics*.
- ISO/IEC 40500:2012. *Information technology — Accessibility guidelines for web content*.