

## HIDDEN SOUND ATLAS – DOCUMENTACIÓN TÉCNICA COMPLETA



Un atlas sonoro oculto dedicado a la música no convencional, donde cada publicación es un viaje a través de sonidos menos mainstream de cualquier género.

### ÍNDICE

1. Descripción del Proyecto
2. Concepto y Filosofía
3. Arquitectura del Sistema
4. Sistema de Roles y Permisos
5. Flujo de Usuario
6. Estructura del Proyecto
7. Tecnologías Utilizadas
8. Características Principales
9. Diagrama UML

### 1. DESCRIPCIÓN DEL PROYECTO

Hidden Sound Atlas es una plataforma web tipo blog especializada en música no convencional y géneros menos mainstream. El proyecto combina una arquitectura técnica basada en el patrón MVC con una experiencia visual inmersiva que evoca sensaciones submarinas y espaciales.

#### Características Clave

- Diseño Visual Inmersivo: Transiciones suaves y efectos lentos que buscan crear una atmósfera particular durante el paso por la web.
- Música Ambiente Integrada: Reproducción de "Sirena" de Robert Rich y Alio Die
- Sistema de Roles Jerárquico: User → Editor → Admin
- Moderación de Contenido: Sistema de aprobación de posts
- Gestión de Usuarios: Perfiles personalizables y solicitudes de permisos

## 2. CONCEPTO Y FILOSOFÍA

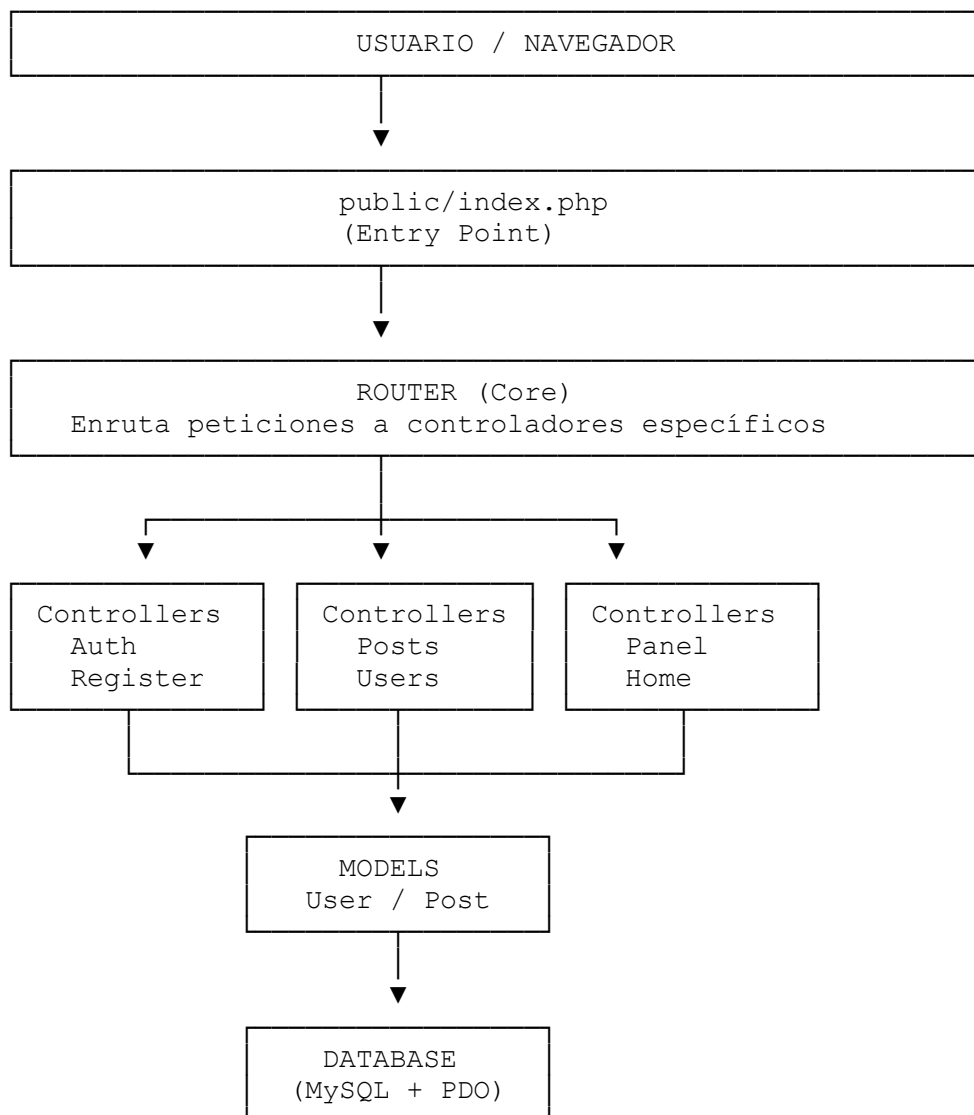
**The Blue Room** es el corazón de Hidden Sound Atlas, la sala principal a la que acceden los usuarios registrados. Este espacio representa un refugio sonoro donde se pueden:

- Descubrir publicaciones sobre música alternativa y experimental
- Sumergirse en una experiencia visual que complementa el contenido auditivo
- Contribuir al atlas con sus propios hallazgos musicales (según permisos)

## 3. ARQUITECTURA DEL SISTEMA (MVC)

El proyecto está organizado mediante el patrón **Modelo–Vista–Controlador**, garantizando separación de responsabilidades, mantenimiento y escalabilidad.

**Flujo General** | Usuario → public/index.php → Router → Controlador → Modelo → Vista



## Componentes

### Core

- Router.php → interpreta URLs y llama al controlador y método correspondiente.
- Database.php → conexión PDO, manejo seguro mediante prepared statements.

### Models

- User.php
  - Registro, login, edición de perfil
  - Solicitudes de editor
  - Cambios de roles, activación/suspensión
- Post.php
  - Creación, edición, borrado de posts
  - Moderación (approved, pending, rejected)
  - Contador de visitas

### Controllers

- AuthController – login/logout
- RegisterController – registro
- HomeController – landing pública y Blue Room
- PostsController – CRUD de posts
- UsersController – perfil personal
- PanelController – administración completa

**PanelController**, gestionando moderación, usuarios, solicitudes, suspensiones y más.

### Views

Divididas en:

- layout\_public.php / layout\_private.php
- Vistas de home, auth, posts, perfiles y panel admin

## 4. SISTEMA DE ROLES Y PERMISOS

### User

- ✓ Accede a The Blue Room
- ✓ Lee posts aprobados
- ✓ Edita su perfil y avatar
- ✓ Puede solicitar ser Editor
- ✗ No crea ni edita posts

### Editor

- ✓ Todo lo de User
- ✓ Puede crear y editar sus posts
- ✓ Sus posts quedan en *pending*
- ✗ No puede moderar

### Admin

- ✓ Aprobación/Rechazo de posts
- ✓ Borrado de contenido
- ✓ Gestión completa de usuarios
- ✓ Modificación de roles
- ✓ Usuarios auto-publican sin moderación

## 5. FLUJO DE USUARIO

### Visitante

→ Página pública → Posts visibles → Login/Registro

### Usuario registrado

→ Blue Room → Lectura ilimitada → Perfil → Solicitar ser editor

### Editor

→ Crear post → Enviar → Estado pendiente → Aprobación por Admin

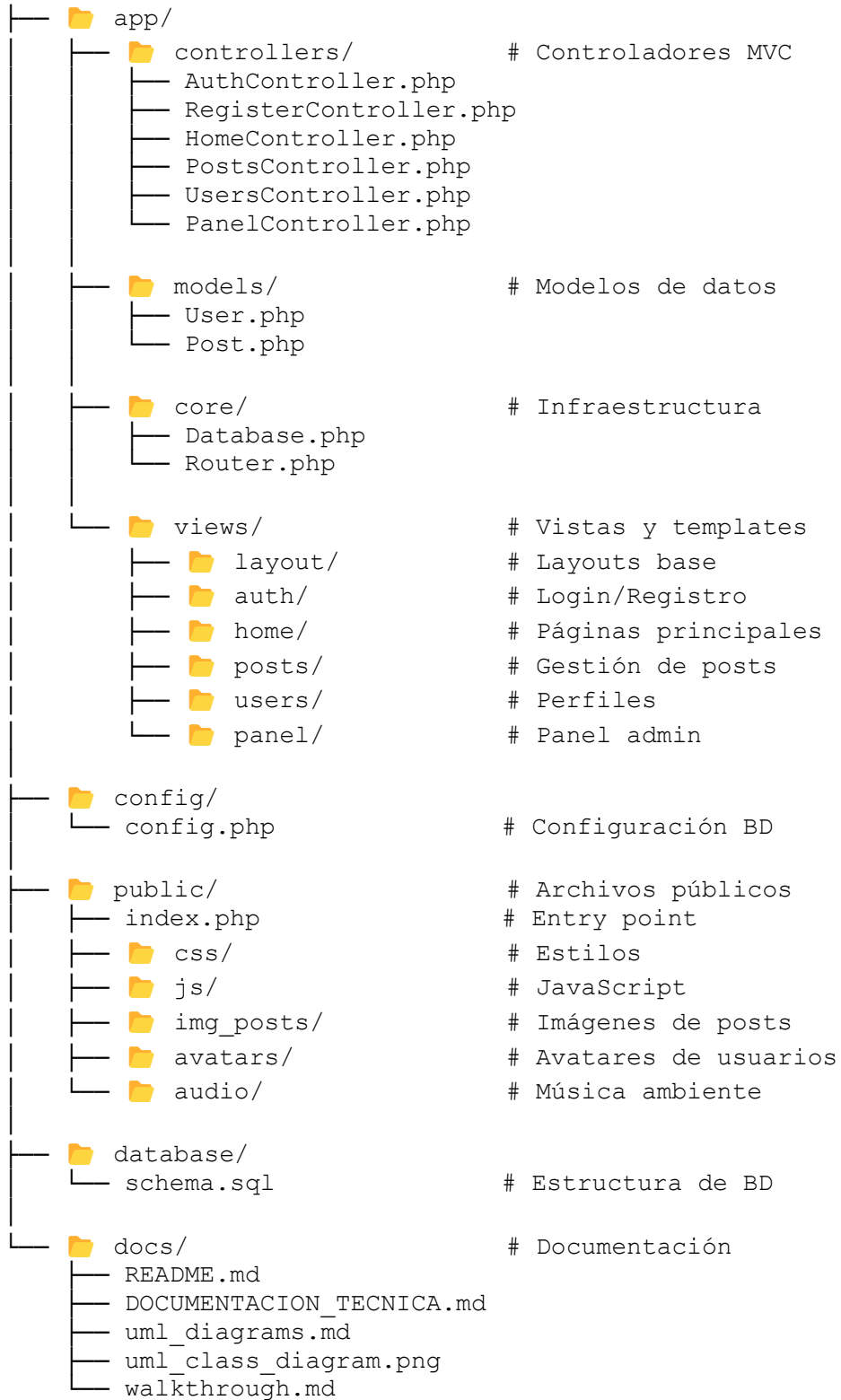
### Admin

→ Panel → Modera posts → Gestiona usuarios → Roles y solicitudes

## 6. ESTRUCTURA DEL PROYECTO

### Estructura del Proyecto

Proyecto\_BlogPHP/



## 7. TECNOLOGÍAS UTILIZADAS

**Backend:** PHP 7.4+, MySQL, PDO

**Frontend:** HTML5, CSS3, JS

**Arquitectura:** MVC

**Seguridad:** password\_hash(), prepared statements, validación MIME, roles

## 8. CARACTERÍSTICAS PRINCIPALES

### Diseño Inmersivo

- Colores oceánicos y cósmicos
- Movimientos lentos, sensación de flotación
- Transiciones suaves
- Integración opcional de música ambient

### Sistema de Posts

- Slugs únicos
- Estados: pending / approved / rejected
- Borrado, edición, moderación
- Contador de visitas

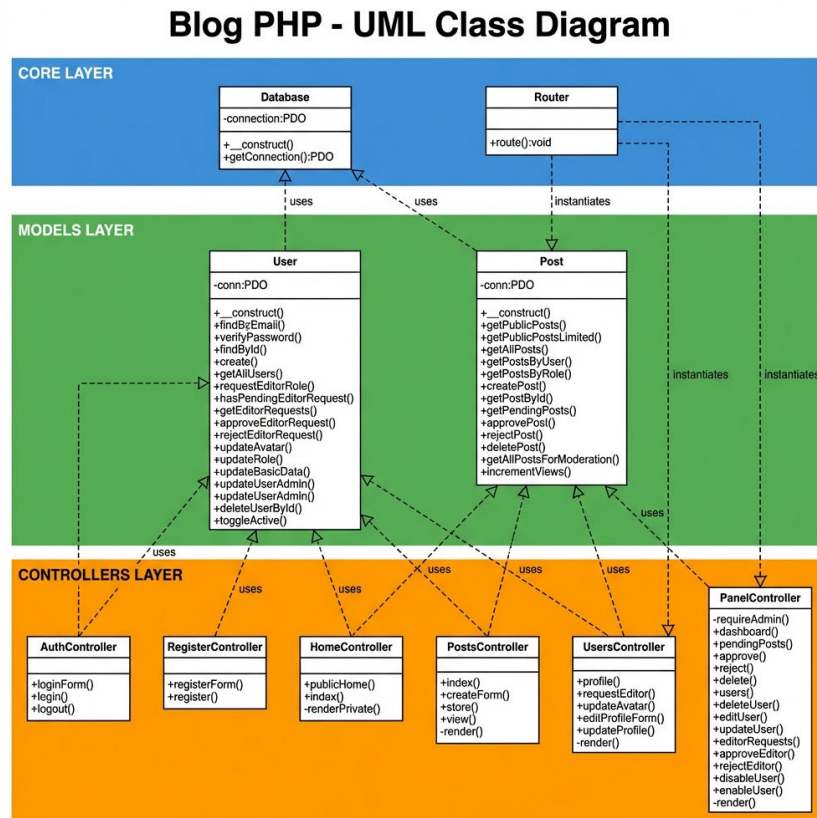
### Gestión de Usuarios

- Cambios de avatar
- Activación/desactivación
- Roles dinámicos
- Solicitud de Editor

## 9. Diagrama UML (Versión Final)

### Diagrama de Clases Completo

El siguiente diagrama muestra la arquitectura completa del sistema, incluyendo todas las clases, sus métodos y las relaciones entre componentes:



### Componentes visualizados:

- ● Core Layer: Database y Router
- ● Models: User y Post
- ● Controllers Layer: 6 controladores especializados

Para ver diagramas adicionales (flujos de autenticación, estados de posts, arquitectura MVC, etc.), es posible consultar el documento completo:

[`uml\_diagrams.md`](./uml\_diagrams.md)

## 10. CREDENCIALES PARA REVISIÓN DEL PROFESOR

Puedes añadir este bloque en el Word:

### Acceso Admin para corrección:

- Usuario: admin
- Password: 1234

### Cuenta de Editor para pruebas:

- Usuario: Martin
- Password: 1234

### Cuenta con solicitud pendiente:

- Usuario: try

## NOTAS DE DESARROLLO

### Convenciones de Código

- PSR-1: Estándar básico de código
- Nombres descriptivos: Métodos y variables auto-explicativos
- Comentarios: Secciones claramente delimitadas
- Prepared Statements: Obligatorio en todas las queries

### Buenas Prácticas Implementadas

- Separación de responsabilidades (MVC)
- Validación de datos en servidor
- Sanitización de inputs
- Manejo de errores
- Código reutilizable (métodos render)
- Seguridad por diseño

### Créditos

#### Música

- "Sirena" - Robert Rich & Alio Die
- Género: Dark Ambient / Drone



## Desarrollo

- Proyecto: Hidden Sound Atlas
- Subtítulo: The Blue Room
- Concepto: Blog de música no convencional
- Arquitectura: MVC con PHP y MySQL

## Licencia

Este proyecto es parte de un trabajo académico para el Grado Superior en Programación.