

Documentación del Proyecto Taller Mecánico

1. Introducción

Este proyecto es una aplicación web para la gestión de un taller mecánico. Permite a los usuarios registrarse, solicitar citas, ver noticias (feed RSS) y consejos de mantenimiento. Incluye un panel de administración para gestionar usuarios, citas, noticias y consejos.

Índice

- [1. Introducción](#)
- [2. Estructura del Proyecto](#)
- [3. Configuración y Base de Datos](#)
- [4. Funcionalidades Principales](#)
- [5. Scripts de Mantenimiento](#)
- [6. Flujo de Usuario](#)
- [7. Despliegue y Credenciales](#)

2. Estructura del Proyecto

Directorios Principales

- `admin/`: Panel de control para administradores. Contiene páginas de gestión (`usuarios.php`, `citas.php`, `noticias.php`, `consejos.php` y `index.php`).
- `api/`: Endpoints para peticiones AJAX, principalmente `citas_api.php`.
- `config/`: Archivos de configuración general y de base de datos (`db.php`, `env.php`).
- `includes/`: Componentes reutilizables (cabecera, pie de página), funciones auxiliares (`functions.php`) y clases de utilidad.
- `scripts/`: Scripts de utilidad para mantenimiento, población de datos y scraping de noticias.
- `database/`: Contiene el esquema de base de datos (`database.sql`).
- `docs/`: Documentación del proyecto.
- `css/`, `js/`, `img/`: Recursos estáticos del frontend.

Archivos Clave en Raíz

- `index.php`: Página de inicio. Muestra servicios destacados, estadísticas y últimos consejos.
- `citaciones.php`: Página unificada para la reserva de citas (calendario interactivo) y visualización del historial de citas del usuario.
- `noticias.php`: Sección de noticias obtenidas de fuentes externas (RSS).
- `consejo.php`: Visualización individual de consejos de mantenimiento.
- `login.php` / `registro.php`: Autenticación y registro de clientes. El registro incluye validación de DNI/NIE, teléfono y dirección completa.
- `perfil.php`: Área personal del usuario para actualizar datos.

3. Configuración y Base de Datos

Conexión (`config/db.php` y `config/env.php`)

El sistema utiliza PDO para la conexión a MySQL. Las credenciales se cargan desde un archivo `config/env.php` (no incluido en repositorios públicos por seguridad).

- Manejo de Errores : Captura excepciones PDO y muestra un mensaje amigable al usuario mientras registra el error real en los logs.

Esquema de Base de Datos

- `users_data`: Almacena información detallada de clientes (nombre, dirección, teléfono, etc.).
- `users_login`: Credenciales de acceso (usuario, hash password, rol).
- `citas`: Registro de reservas (fecha, hora, usuario/invitado, motivo).
- `noticias`: Artículos importados (título, enlace, imagen, fecha, fuente).
- `consejos`: Tips de mantenimiento creados por administradores.

4. Funcionalidades Principales

4.1. Sistema de Reservas (`citaciones.php` + `api/citas_api.php`)

- Interfaz Unificada : Calendario interactivo integrado en `citaciones.php` que permite seleccionar fecha y hora.
- API REST : `api/citas_api.php` maneja la disponibilidad de horarios y la creación de reservas mediante AJAX.
- Validación : Verifica huecos disponibles y evita colisiones en tiempo real.
- Usuarios e Invitados : Permite reservar a usuarios registrados (autocompletando datos) y visitantes (formulario de contacto).

4.2. Noticias y Scraping (`scripts/`)

- Importación Automática : Scripts como `populate_motor_news.php` obtienen noticias de fuentes RSS.
- Almacenamiento : Las noticias se guardan en la base de datos para mostrarlas en `noticias.php`.

4.3. Administración (`admin/`)

Acceso restringido a usuarios con rol `admin`.

- Dashboard (`admin/index.php`): Vista general con estadísticas.
- Gestión : CRUD completo para usuarios, citas, noticias y consejos.

4.4. Seguridad (`includes/functions.php`)

- Autenticación : Verificación de sesión (`isLoggedIn()`, `isAdmin()`).
- CSRF: Generación y validación de tokens para todos los formularios POST.
- Sanitización : Limpieza de entradas de usuario (`sanitize()`) para prevenir XSS.

5. Scripts de Mantenimiento

Ubicados en `scripts/`:

- `setup_check.php`: Verifica requisitos del entorno.
- `run_scraper.bat`: Automatización para actualizar noticias en Windows.

6. Flujo de Usuario

1. Visitante : Puede ver la home, noticias, consejos y reservar cita como invitado.
2. Usuario Registrado : Accede a su perfil, historial de citas y reservas agilizadas.
3. Administrador : Accede al panel de control para gestionar la operación del taller.

7. Despliegue y Credenciales

7.1. Despliegue en Local (XAMPP)

1. Archivos :
 - Copie la carpeta del proyecto dentro de `C:\xampp\htdocs\`.
 - Ejemplo: `C:\xampp\htdocs\trabajo_final_php`.
2. Base de Datos :
 - Inicie Apache y MySQL en XAMPP.
 - Importe `database/database.sql` en phpMyAdmin.
 - Nombre sugerido de la BD: `trabajo_final_php`.
3. Configuración :
 - Copie `config/env.example.php` a `config/env.php` (o créelo) con:

```
return [  
    'DB_HOST' => 'localhost',  
    'DB_NAME' => 'trabajo_final_php',  
    'DB_USER' => 'root',  
    'DB_PASS' => ''  
];
```
4. Acceso:
 - Navegador: `http://localhost/trabajo_final_php`

7.2. Despliegue en la Nube (InfinityFree)

1. Base de Datos :
 - Cree la base de datos en el panel de InfinityFree ("MySQL Databases").
 - Note el `Hostname`, `Database Name`, `Username` y `Password` (del vPanel).
 - Importe el SQL vía phpMyAdmin (disponible en el panel).
2. Archivos :
 - Suba el contenido a la carpeta `htdocs/`.
3. Configuración Final :
 - Cree `config/env.php` con las credenciales de InfinityFree.

7.3. Credenciales de Acceso

Usuario Administrador :

- Usuario : `admin`
- Contraseña : `admin`

Usuario de Prueba :

- Usuario : test
- Contraseña : test