



GARUTO

CUADERNO PISTACHO

1. Introducción

Garuto (Cuaderno de Campo Pistacho) es una aplicación avanzada de tipo **PWA (Progressive Web App)** diseñada para la gestión integral de explotaciones agrícolas. Utiliza un backend ligero en PHP con MySQL y un frontend reactivo basado en JavaScript Vanilla, optimizado para el rendimiento y uso en condiciones de baja conectividad (Offline-First).

2. Arquitectura del Sistema

La aplicación sigue un modelo de **Tres Capas**:

1. **Capa de Presentación (Frontend)**: SPA construida con HTML5, CSS3 (Variables y Flexbox) y JS. Gestiona el estado local y la sincronización a través de un Service Worker (`sw.js`).
2. **Capa de Aplicación (Backend)**: API RESTful transparente en `api.php`. Procesa autenticación, CRUD de datos e integración con servicios externos (SIGPAC, Open-Meteo).
3. **Capa de Datos (MySQL)**: Base de datos relacional para persistencia de registros, inventario y configuración.

3. Requisitos del Servidor

- **Servidor Web**: Apache o Nginx con soporte para HTTPS (Obligatorio para Service Workers).
- **PHP**: Versión 7.4 o superior (Soporta PHP 8.2+).
- **Extensiones PHP**:
 - `pdo_mysql` : Interacción con base de datos.
 - `curl` : Consultas externas (SIGPAC).
 - `mime_content_type` : Validación de archivos subidos.
 - `openssl` y `zlib` : Seguridad y compresión.
- **MySQL / MariaDB**: Versión 5.7+ o 10.3+.

4. Estructura de la Base de Datos

El sistema utiliza el esquema definido en `setup.sql`, compuesto por 10 tablas principales:

Tabla	Propósito
<code>parcelas</code>	Almacena datos geométricos (lat/Ing), referencia SIGPAC y mapas de árboles (JSON).
<code>trabajos</code>	Catálogo de labores (Riego, Poda, Fitos...), categorizadas por tipo legal SIEX.
<code>inventario</code>	Control de stock de insumos (abonos, fitosanitarios, herbicidas).
<code>maquinaria</code>	Listado de equipos y su coste operativo por hora.
<code>registros</code>	El núcleo del cuaderno: vincula parcelas, trabajos y maquinaria.

Tabla	Propósito
fotos	Archivo fotográfico vinculado a parcelas o registros específicos.
planing_progreso	Tracking del checklist anual de tareas de secano.
documentacion	Repositorio de escrituras, contratos y documentos legales en PDF/Imagen.
maquinaria_reparaciones	Registro detallado de costes de mantenimiento y averías.

5. Instalación y Configuración

Paso 1: Base de Datos

1. Crea una base de datos en tu servidor.
2. Importa el archivo `setup.sql`.

Paso 2: Configuración de Conexión

Edita `api.php` para configurar las credenciales o defínelas como variables de entorno (Recomendado):

```
define('DB_HOST', getenv('DB_HOST') ?: 'localhost');
define('DB_NAME', getenv('DB_NAME') ?: 'garuto_db');
define('DB_USER', getenv('DB_USER') ?: 'root');
define('DB_PASS', getenv('DB_PASS') ?: '');
```

Paso 3: Permisos de Archivos

La API necesita crear y escribir en:

- `/uploads/` : Para fotos de la galería.
- `/uploads/docs/` : Para documentación PDF/Word.

Asegúrate de que el usuario del servidor (ej: `www-data`) tenga permisos de escritura (`755` o `775`).

6. Documentación de la API (Endpoints)

La API se invoca mediante `api.php?action=[ACCIÓN]` . Requiere autenticación de sesión para todas las acciones excepto `login` y `checkSession` .

- `login` : Recibe JSON `{username, password}` .
 - `getAll` : Devuelve todos los registros de una tabla (parámetro `collection`).
 - `add` : Inserta nuevos datos (detecta automáticamente cálculos de stock e integración de costes).
 - `uploadPhoto` / `uploadDoc` : Gestiona la subida de archivos binarios al sistema de archivos.
 - `getSigpacInfo` : Proxy que consulta las APIs oficiales del SIGPAC para obtener referencias catastrales y superficies a partir de coordenadas.
 - `export` : Genera un volcado JSON completo para backups.
-

7. Modo Offline y Sincronización

La aplicación implementa una clase `DataStore` en `app.js` que gestiona las operaciones de red:

- Si el navegador está **Offline**, las operaciones `add` , `update` o `borrar` se guardan en una cola persistente (`localStorage`).
 - Al recuperar la conexión (`online` event), el sistema procesa automáticamente la cola subiendo los cambios pendientes al servidor.
 - El `sw.js` asegura que el HTML/CSS/JS se cargue instantáneamente incluso sin internet.
-

Garuto Dashboard

Elegant Roboto, green/white

