

# Supuesto Práctico

Esta práctica cubre los 9 Resultados de Aprendizaje y 42 Criterios de Evaluación del módulo 0613, mediante un proyecto realista, escalable y motivador. El alumnado desarrollará una aplicación web completa para la gestión de una huerta ecológica, evolucionando desde código PHP embebido → arquitectura MVC en Laravel, integrando:

- Generación dinámica de páginas (RA2–3)
- Sesiones, cookies y autenticación avanzada (RA4)
- Separación lógica/presentación con POO y MVC (RA5)
- Acceso seguro a bases de datos (RA6)
- Servicios Web REST/SOAP (RA7)
- Aplicaciones híbridas con repositorios externos (RA9)
- Despliegue profesional con Docker (RA1)

## Descripción narrativa

La Cooperativa Agroecológica “Raíces Vivas”, con sede en Benicarló, gestiona 12 huertos ecológicos distribuidos por la comarca. Actualmente usan hojas de cálculo para registrar cultivos, siembras y cosechas, lo que genera errores, duplicados y falta de visibilidad en tiempo real.

Necesitan una aplicación web interna que permita a sus 25 agricultores:

- Registrar cultivos, siembras y cosechas
- Gestionar usuarios con roles: admin, agricultor, técnico
- Consultar estadísticas de producción (kg/m<sup>2</sup>, rentabilidad por cultivo)
- Integrarse con sensores IoT simulados (humedad, temperatura)
- Exponer APIs para que otras cooperativas consuman sus datos (REST/SOAP)

El alumnado asumirá el rol de equipo de desarrollo interno, entregando iteraciones funcionales por fases, en los plazos indicados más adelante.

# Fase 1: Mi Primera Huerta

Partimos de lo conocido: código embebido en PHP. Es el punto de partida obligatorio según CE RA2.a ("mecanismos de generación de páginas con código embebido"). Permite diagnosticar el nivel inicial y construir una base sólida.

## Objetivos específicos

- Escribir sentencias simples, usar variables/operadores y reconocer ámbitos.
- Usar estructuras de control, arrays, funciones, formularios y recuperar datos del cliente.
- Establecer conexiones, recuperar/editar información y ejecutar sentencias SQL.

## Contenidos básicos:

- Inserción de código en páginas web (PHP, etiquetas, variables)
- Programación con código embebido (if, while, arrays, funciones)
- Acceso a datos (conexión, SELECT/INSERT, sentencias SQL)

## Competencias clave:

- Desarrollar aplicaciones con lógica en servidor (C1)
- Aplicar buenas prácticas de validación (C3)

## Actividades Fase 1:

☐ 1.1. Crear BD `huerta_db`.

☐ 1.2. Crear tabla `cultivos`:

	Nom	Tipus	Atributs	Nul	Comentaris
1	id	INT	PRIMARY KEY, UNSIGNED	No	Identificador únic del cultiu. UTO_INCREMENT
2	nombre	VARCHAR(60)		No	Nom comú o varietat del cultiu.

3	tipo	ENUM('Hortaliza', 'Fruto', 'Aromática', 'Legumbre', 'Cereal')		No	Classificació del tipus de planta. Assegura que el camp només contingui un d'aquests valors predefinits.
4	dias_cosecha	SMALLINT	UNSIGNED	No	Dies (enters) des de la sembra fins a la collita. El tipus <b>SMALLINT UNSIGNED</b> optimitza l'espai i evita valors negatius.

- ☐ 1.2. `index.php`: Listar cultivos con `mysqli_query()` y bucle `while`

**Objetivo:** Mostrar los registros de la tabla `cultivos` utilizando PHP mezclado con HTML para generar una tabla dinámica (código embebido).

- ☐ 1.3. `nuevo.php`: Formulario y Validación con `filter_input()`

**Objetivo:** Crear el formulario HTML y, al procesar la entrada, asegurar que los datos se recuperan y se saneen correctamente.

- ☐ 1.4. `procesar.php`: Insertar con `mysqli_query()` (sin prepared aún)

**Objetivo:** Recibir los datos saneados (del paso 1.3) y construir y ejecutar una sentencia SQL `INSERT`.

- ☐ 1.5. Definir la función `cicloCultivo($dias)`, que devuelve el ciclo del cultivo con los valores "Corto" / "Medio" / "Tardío". La función toma el valor del campo `dias_cosecha`.

## Entregables Fase 1:

- ☐ Carpeta fase1/ con: `index.php`, `nuevo.php`, `procesar.php`, `conexion.php`
- ☐ `huerta_db.sql` con 5 cultivos de ejemplo
- ☐ `Captura`: formulario con mensaje de error al enviar nombre vacío.

## \* Ampliación:

- ☐ Paginación manual (3 cultivos por página)
- ☐ Función `ordenarPorDias($cultivos)` con `usort()`