



# PLAN DE PROYECTO

## Raíces

“Conectando tierra y vida”

**Desarrollo de Aplicaciones Web**

**Proyecto de Desarrollo de Aplicaciones Web**

**Javier Hipólito Rodríguez**

**Versión 1.0**

<b>Nombre del fichero:</b>	Daw_PRW_[]_UT01_PlanDeProyecto.pdf
<b>Fecha de esta versión:</b>	1-12-25

## Historial de revisiones

Fecha	Descripción	Autor
1-12-25	Creación del documento	Javier Hipólito Rodríguez

# Índice

1. Resumen ejecutivo .....	3
2. Alcance del proyecto .....	3
2.1. Objetivo general y objetivos específicos. ....	3
2.2. Límites del proyecto (qué se incluye y qué no). ....	4
3. Requisitos del proyecto .....	4
3.1. Requisitos funcionales. ....	4
3.2. Requisitos no funcionales. ....	4
3.3. Requisitos técnicos. ....	5
4. Estructura de trabajo y cronograma .....	5
4.1. División en fases o tareas principales. ....	5
4.2. Fechas estimadas y responsables. ....	6
4.3. Herramienta o formato de planificación (Gantt, Trello, etc.). ....	6
5. Recursos y roles del equipo .....	6
5.1. Personal asignado y sus responsabilidades. ....	6
5.2. Recursos técnicos y materiales necesarios. ....	6
6. Gestión de riesgos .....	7
6.1. Posibles problemas o retrasos. ....	7
6.2. Plan de contingencia o prevención. ....	7
7. Conclusión .....	7

# 1. Resumen ejecutivo

**“Raíces”** es un proyecto que nace en Fuerteventura con el objetivo de conectar directamente a campesinos locales con consumidores a través de una aplicación web. Su propósito es automatizar la gestión de pedidos semanales de frutas y verduras de temporada, fomentar el consumo responsable de productos de kilómetro 0, preservar las costumbres agrícolas tradicionales y reducir el desperdicio alimentario.

Esta plataforma facilitará:

- La organización de pedidos según la temporada y disponibilidad real.
- La automatización del proceso de compra, preparación y entrega.
- El control del stock y de los beneficios generados.
- La gestión de cuentas, clientes y facturación de forma centralizada.
- La reducción del desperdicio alimentario mediante previsión de demanda.
- Una comunicación ágil y moderna entre productores y consumidores.

# 2. Alcance del proyecto

## 2.1. Objetivo general y objetivos específicos.

El objetivo general que se quiere lograr, es desarrollar una aplicación web funcional que permita a una cooperativa de agricultores locales gestionar de forma eficiente los pedidos semanales de productos de temporada.

Los objetivos específicos para el desarrollo a tener en cuenta serán:

- proporcionar un panel de administración (dashboard) donde los socios puedan gestionar productos, stock, pedidos, clientes y beneficios.
- facilitar al personal de la cooperativa la organización de las entregas y la preparación semanal.
- permitir a los clientes visualizar los productos disponibles y realizar pedidos personalizados.

- automatizar el proceso de generación de listados, resúmenes de pedidos y registros necesarios para facturación.

## 2.2. Límites del proyecto (qué se incluye y qué no).

Incluye:

- Aplicación web responsive accesible desde ordenador y móvil.
- Listado actualizado de productos disponibles semanalmente.
- Registro e inicio de sesión (cliente / administrador).
- Realización, edición y confirmación de pedidos.
- Panel interno para admins con gestión de productos, pedidos, stock y clientes.
- Control de beneficios individuales y globales.
- API REST que comunica frontend y backend.

No incluye:

- Aplicación móvil nativa (Android/iOS).
- Pasarela de pago integrada.
- Servicio logístico de reparto.
- Integración con sistemas externos contables o fiscales.

## 3. Requisitos del proyecto

### 3.1. Requisitos funcionales.

- Registro e inicio de sesión de clientes y administradores.
- Gestión de roles y control de acceso según perfil.
- Visualización del catálogo semanal de productos.
- Añadir productos al carrito y realizar pedidos.
- Modificación y confirmación del pedido.
- Gestión interna de pedidos: estados, preparación, entregas.
- Gestión de productos: CRUD completo (crear, editar, eliminar).
- Panel administrativo con estadísticas: ventas, stock, pedidos, clientes.
- Generación de resúmenes para facturación.

### 3.2. Requisitos no funcionales.

- Interfaz limpia, intuitiva y apta para usuarios de diferentes niveles.
- Diseño responsive adaptado a móviles y ordenadores.
- Seguridad básica en la API (validación de roles, control de acceso).
- Estructura clara: frontend separado del backend.
- Documentación, uso de GitHub y control de versiones.

### 3.3. Requisitos técnicos.

- **Frontend:** HTML, CSS, JavaScript, TailwindCSS, Flowbite.
- **Administración:** SPA ligera con JavaScript.
- **Backend:** PHP con arquitectura API REST.
- **Base de datos:** MySQL/MariaDB para almacenamiento estructurado.
- **Control de sesiones:** localStorage + validación en backend.
- **Herramientas:** XAMPP, Git, GitHub.
- **Entorno:** Servidor local con Apache y MySQL.

## 4. Estructura de trabajo y cronograma

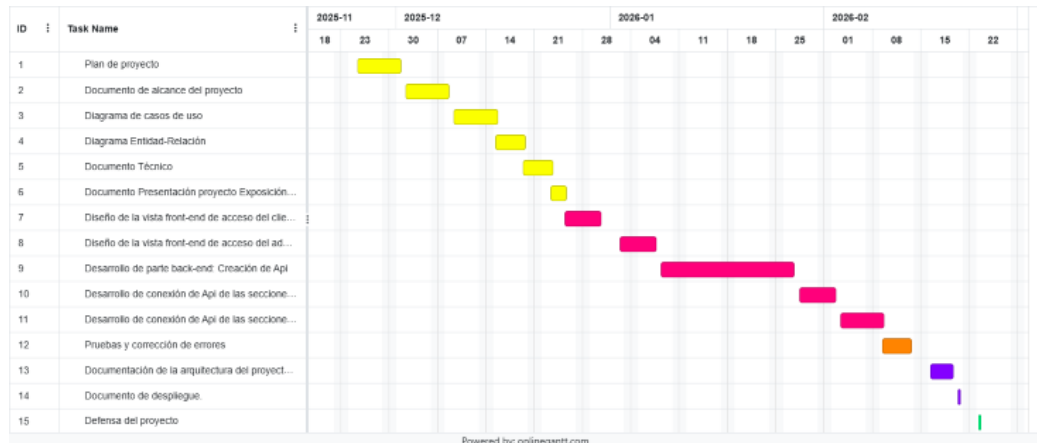
### 4.1. División en fases o tareas principales.

<b>Análisis y diseño</b>	Diseño de requisitos y estructura de datos. Bocetos de interfaz cliente + dashboard.
<b>Configuración inicial del proyecto</b>	Creación de repositorio Git. Estructura de carpetas frontend + backend.
<b>Base de datos</b>	Diseño ER. Implementación en MySQL.
<b>Backend API REST</b>	Endpoints: autenticación, productos, pedidos, clientes. Conexión a BD.
<b>Frontend cliente</b>	Tienda multipágina. Carrito y sistema de pedidos. Login y control de sesión.
<b>Frontend administrador (SPA ligera)</b>	Dashboard. CRUD de productos. Gestión de pedidos. Estadísticas.
<b>Pruebas y validación</b>	Test funcional. Ajustes finales.
<b>Documentación y entrega</b>	Memoria del proyecto.

## 4.2. Fechas estimadas y responsables.

( Esta sección esta desarrollada en el plan de alcance)

## 4.3. Herramienta o formato de planificación (Gantt, Trello, etc.).



## 5. Recursos y roles del equipo

### 5.1. Personal asignado y sus responsabilidades.

Al ser desarrollador único, se abordarán estos puntos:

- Diseño de interfaz
- Desarrollo frontend
- Desarrollo backend
- Modelado de base de datos
- Pruebas y documentación

### 5.2. Recursos técnicos y materiales necesarios.

- Ordenador personal
- XAMPP (Apache, PHP, MySQL)
- Editor de código (VSCode)
- Git + GitHub
- Navegadores web para pruebas
- Frameworks: Tailwind + Flowbite

## 6.Gestión de riesgos

### 6.1. Posibles problemas o retrasos.

- Errores en la base de datos que afecten a la API.
- Complejidad en la SPA del panel admin.
- Problemas de compatibilidad con hosting futuro.
- Sobrecarga de funcionalidades respecto al tiempo disponible.

### 6.2. Plan de contingencia o prevención.

- Priorizar las funcionalidades esenciales.
- Mantener backups de la base de datos.
- Realizar commits frecuentes en GitHub.
- Desarrollar primero la tienda, luego el dashboard.
- Dejar funcionalidades opcionales para una fase final

## 7.Conclusión

*El proyecto “Raíces” aportará una solución real y útil para una cooperativa agrícola local, modernizando la gestión interna y facilitando la conexión con consumidores.*

*El uso de tecnologías modernas y su impacto social, hacen que sea un proyecto sólido y escalable en un futuro.*