

# Propuesta de Proyecto: FoodConnect

## Título del Proyecto

**FoodConnect:** Conectando a la comunidad para reducir el desperdicio de alimentos.



## Datos del Equipo

- **Nombre del Proyecto:** FoodConnect
- **Equipo:** BalmisTech
- **Miembros del Equipo:**
  - María García López (Hacedora)
  - Juan Pérez Martínez (Analista)
  - Laura Sánchez Torres (Divergente)
  - Carlos Ruiz Navarro (Armonizador)
- **Curso:** 2º DAM - IES Doctor Balmis
- **Fecha de inicio:** Octubre 2025

## Alineación con los Objetivos de Desarrollo Sostenible (ODS)

Este proyecto se alinea directa y explícitamente con los siguientes Objetivos de Desarrollo Sostenible de las Naciones Unidas, tal como se requiere en la programación didáctica:

- **ODS 2: Hambre Cero:** Facilita la redistribución de excedentes de alimentos de comercios y particulares a personas que los necesitan, combatiendo el hambre a nivel local.
- **ODS 12: Producción y Consumo Responsables:** Ataca directamente la meta 12.3, que busca reducir a la mitad el desperdicio de alimentos per cápita. La aplicación proporciona una plataforma para dar una segunda vida a los alimentos que de otro modo serían desechados.
- **ODS 11: Ciudades y Comunidades Sostenibles:** Fomenta la creación de redes de apoyo locales y fortalece los lazos comunitarios al conectar a vecinos, ONGs y pequeños comercios con un objetivo común.

# Descripción del Problema

En nuestra comunidad, una gran cantidad de alimentos en perfecto estado es desechada diariamente por pequeños comercios (fruterías, panaderías, restaurantes) al final de su jornada, así como por particulares. Al mismo tiempo, existen familias y personas con dificultades para acceder a alimentos frescos y nutritivos. La desconexión entre quienes tienen un excedente y quienes lo necesitan es el principal obstáculo.

## Descripción de la Solución Propuesta

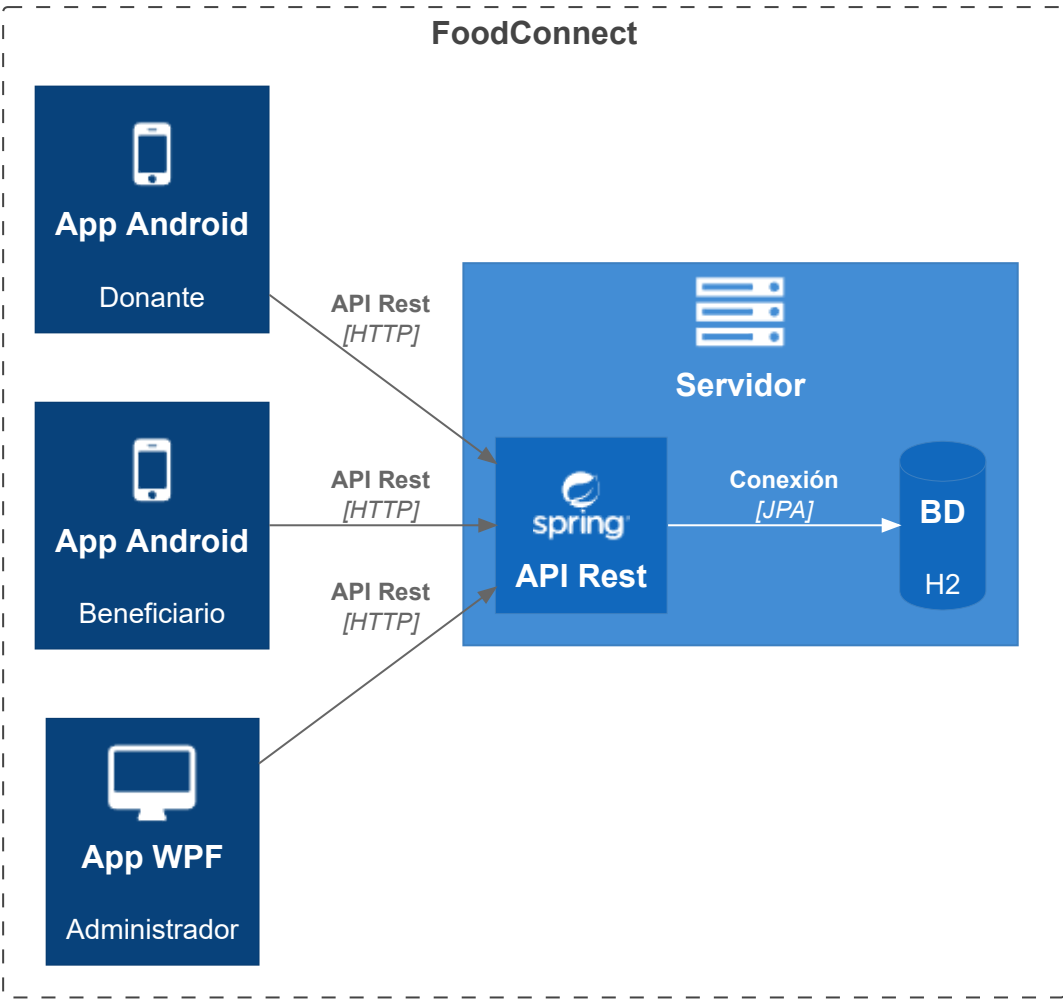
**FoodConnect** es una solución software multiplataforma diseñada para servir de puente entre donantes de alimentos y beneficiarios. El sistema constará de tres componentes principales, siguiendo la arquitectura definida en la programación:

1. **Aplicación Móvil para Clientes (Android con Kotlin y Jetpack Compose):** Será la herramienta principal para los usuarios (donantes y beneficiarios). Permitirá a los donantes publicar "cestas" de alimentos excedentes y a los beneficiarios buscarlas, reservarlas y coordinar su recogida.
2. **Aplicación de Escritorio para Administradores (WPF con C# y MVVM):** Una herramienta de back-office para que los administradores del sistema puedan gestionar usuarios (validar ONGs, bloquear usuarios con mal comportamiento), supervisar las publicaciones y obtener estadísticas básicas sobre el impacto de la plataforma.
3. **Servicio Backend (API REST con Java, Spring Boot y Spring Security):** El cerebro del sistema. Gestionará toda la lógica de negocio, la persistencia de datos en una base de datos H2, la autenticación de usuarios mediante JWT y la comunicación segura entre las aplicaciones cliente y de escritorio.

## Actores y Roles

- **Donante:** Cualquier particular, comercio u organización que desea ofrecer un excedente de alimentos.
- **Beneficiario:** Cualquier persona u organización (ONG, comedor social) que necesita alimentos.
- **Administrador:** Un superusuario que gestiona y modera la plataforma a través de la aplicación de escritorio.

# Arquitectura general



## Casos de Uso Principales

### Para Donantes (App Móvil)

Caso de Uso	Descripción	Prioridad
Registro de Usuario	Permitir a donantes y beneficiarios registrarse en la aplicación móvil.	Alta
Publicar Cesta	Publicar una nueva "cesta de alimentos", especificando contenido, cantidad aproximada, ubicación y horario de recogida.	Alta
Subir Foto	Subir una foto de la cesta para mayor transparencia.	Media

Caso de Uso	Descripción	Prioridad
<b>Ver Estado de Cestas</b>	Ver el estado de sus cestas publicadas (disponible, reservada, recogida).	Alta
<b>Notificaciones</b>	Recibir una notificación cuando su cesta sea reservada.	Baja
<b>Confirmar Recogida</b>	Confirmar que la cesta ha sido recogida por el beneficiario.	Alta
<b>Historial de Donaciones</b>	Consultar su historial de donaciones.	Media

## Para Beneficiarios (App Móvil)

Caso de Uso	Descripción	Prioridad
<b>Registro de Usuario</b>	Permitir a donantes y beneficiarios registrarse en la aplicación móvil.	Alta
<b>Buscar Cestas</b>	Buscar cestas de alimentos disponibles cerca de su ubicación usando un mapa o una lista.	Alta
<b>Filtrar Búsquedas</b>	Filtrar búsquedas por tipo de alimento o distancia.	Media
<b>Ver Detalles de Cesta</b>	Ver los detalles de una cesta y la información del donante.	Alta
<b>Reservar Cesta</b>	Reservar una cesta disponible.	Alta
<b>Chat con Donante</b>	Comunicarse con el donante a través de un chat simple para coordinar la recogida.	Baja
<b>Historial de Recogidas</b>	Consultar su historial de recogidas.	Media

## Para Administradores (App Escritorio)

Caso de Uso	Descripción	Prioridad
<b>Inicio de Sesión Seguro</b>	Permitir a los administradores iniciar sesión de forma segura.	Alta

Caso de Uso	Descripción	Prioridad
Gestión de Usuarios	Ver un listado de todos los usuarios (donantes y beneficiarios).	Alta
Validar ONGs	Validar el registro de nuevas ONGs o comedores sociales.	Media
Bloquear/Eliminar Usuarios	Bloquear o eliminar usuarios que incumplan las normas de la comunidad.	Alta
Gestión de Publicaciones	Visualizar y eliminar publicaciones que sean inapropiadas.	Media
Ver Estadísticas	Ver estadísticas básicas (ej: nº de cestas donadas en la última semana, nº de usuarios activos).	Baja

## Riesgos y mitigación

Riesgo	Mitigación
Chat en tiempo real	Requiere del uso de FLoWs y corrutina en Kotlin + Firebase
Mapas y geolocalización	Uso de Google Maps SDK (API Key) y componentes específico de Maps en compose
Notificaciones push	Uso de Firebase Cloud Messaging (FCM)
Complejidad JWT	Taller extra + plantilla base o usar sesiones
Retraso en integración	Feature-flags para desactivar parte no crítica
Scope creep	Definición clara del <b>Producto Mínimo Viable</b> (MVP) priorizando el backlog
Falta de experiencia	Uso de IA para generación de código y resolución de dudas
Falta de compromiso	Reuniones semanales de seguimiento y revisión del progreso
Conflictos en equipo	Roles claros y comunicación abierta

# Planificación aproximada Sprints por quincena

Sprint	Semanas	Fecha fin	Entregable clave
1	7-8	31 oct	Modelo de dominio Java + diagramas
2	9-10	14 nov	Clases transpiladas a C#/Kotlin + mocks con IA
3	11-12	28 nov	Prototipos UI/UX Stitch/Figma (Material 3 & XAML)
4	13-14	12 dic	App WPF navegable con datos mock (1.ª evaluación)
5	15-16	16 ene	App Android navegable con estados mock
6	17-18	30 ene	Lógica de negocio en WPF (MVVM)
7	19-20	13 feb	Lógica de negocio en Android (MVI)
8	21-22	27 feb	API REST CRUD + BBDD H2 (2.ª evaluación)
9	23-24	13 mar	Servicios de negocio + DTOs + excepciones
10	25-26	27 mar	Seguridad: registro/login JWT + roles
11	27-28	08 may	Autorización por roles + endpoints protegidos
12	29-30	22 may	Integración full-stack (sustitución mocks ↔ API)
13	31-32	05 jun	Memoria final, vídeo 2 min, presentación oral

## Organización repositorio

```
https://github.com/BalmisTech/FoodConnect
├─ docs/
│   ├── PROYECTO.md    ← visión, ODS, casos de uso
│   ├── DISEÑO.md      ← modelo, decisiones arquitectónicas
│   └─ DIARIO.md       ← seguimiento semanal individual
├─ backend/
├─ frontend-wpf/
└─ frontend-android/
```