

# Propuesta tecnológica y diseño técnico

☰ Etiquetas	
📅 Fecha	@February 25, 2026

## Diseño Técnico y Propuesta Tecnológica

### Sistema de Gestión de Actividades

### Biblioteca Municipal de San Gregorio

---

## 1. Contexto del proyecto

La Biblioteca Municipal de San Gregorio presta servicio a aproximadamente 35.000 habitantes y organiza talleres culturales, clubes de lectura, actividades infantiles y eventos formativos durante el año.

Actualmente, la inscripción y seguimiento de actividades se realiza manualmente mediante formularios en papel y hojas de cálculo independientes, generando errores de transcripción, duplicidades, sobreasignación de plazas y dificultades para disponer de información en tiempo real.

El objetivo del proyecto es implantar una solución digital sencilla que permita:

- Centralizar la gestión de actividades e inscripciones
  - Reducir errores administrativos
  - Disponer de información actualizada en tiempo real
  - Mejorar la experiencia de usuarios y personal
  - Cumplir la normativa de protección de datos
  - Ajustarse a recursos limitados y conocimientos tecnológicos básicos
- 

## 2. Alcance funcional

El sistema incluirá:

- Gestión de actividades
- Gestión de participantes
- Inscripciones con control automático de plazas
- Confirmaciones automáticas básicas
- Control de asistencia
- Informes básicos y exportación de datos
- Gestión de usuarios internos y roles

Quedan excluidos:

- Integraciones con otros sistemas municipales
  - Gestión de pagos
  - Aplicación móvil nativa
  - Analítica avanzada
  - Automatizaciones complejas
- 

### **3. Actores del sistema**

Los actores identificados son:

- Personal administrativo
  - Dirección de la biblioteca
  - Usuario/Participante
  - Administrador del sistema
- 

## **4. Diseño técnico**

### **4.1 Principios de diseño**

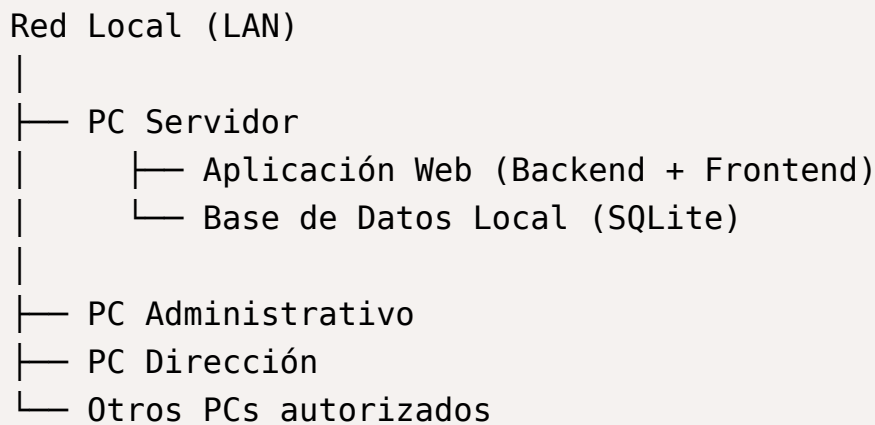
La solución debe:

- Ser simple y fácilmente mantenible
- Funcionar íntegramente en red local

- No depender de infraestructura cloud
  - Ser utilizable por personal sin formación técnica avanzada
  - Adaptarse a recursos limitados
- 

## 4.2 Arquitectura general

Se propone una arquitectura monolítica web en red local (LAN):



### Características

- El servidor ejecuta la aplicación.
- Escucha en IP local (ej. 192.168.1.10).
- Acceso desde cualquier equipo de la red mediante navegador:

`http://192.168.1.10:5000`

- El propio servidor accede vía:

`http://localhost:5000`

No existe exposición a Internet.

---

## 4.3 Componentes del sistema

### 4.3.1 Backend

Responsable de:

- Lógica de negocio
- Validación de reglas

- Control automático de plazas
- Prevención de duplicidades
- Gestión de roles
- Generación de informes

Debe garantizar que:

- No se superen las plazas disponibles
  - No existan inscripciones duplicadas
- 

### **4.3.2 Base de datos**

Base de datos local basada en archivo único (SQLite).

Motivos:

- No requiere servidor independiente
  - Bajo mantenimiento
  - Suficiente para bajo volumen
  - Compatible con entorno de recursos limitados
- 

### **4.3.3 Frontend**

Aplicación web accesible mediante navegador.

Debe permitir:

- Crear y gestionar actividades desde una interfaz única
  - Registrar participantes evitando duplicidades
  - Inscribir usuarios con control automático de plazas
  - Consultar inscritos y plazas disponibles en tiempo real
  - Registrar asistencia
  - Generar y exportar informes
- 

## **4.4 Modelo de datos**

Según el modelo conceptual definido:

- Un usuario puede tener múltiples inscripciones

- Una actividad puede tener múltiples inscritos
- La inscripción vincula usuario y actividad

## Entidades principales

### Participante

- id
- nombre
- email
- teléfono

### Actividad

- id
- título
- descripción
- fecha
- plazas\_maximas

### Inscripción

- id
- participante\_id
- actividad\_id
- fecha
- estado

### Restricción:

```
UNIQUE(participante_id, actividad_id)
```

Evita duplicidades.

### Asistencia

- inscripcion\_id
- asistio

UsuarioInterno

- username
  - password\_hash
  - rol
- 

## 4.5 Lógica crítica del sistema

### Control automático de plazas

Antes de confirmar una inscripción:

```
COUNT(inscripciones confirmadas) < plazas_maximas
```

Si no se cumple → inscripción rechazada.

Este control es requisito obligatorio.

---

### Confirmación automática

Tras una inscripción válida:

- Confirmación inmediata en pantalla
- Opcional: notificación por email

La generación de confirmaciones es requisito del sistema.

---

### Consulta en tiempo real

Las consultas se ejecutan directamente contra la base de datos, permitiendo información actualizada de inscritos y asistencia.

---

## 5. Propuesta tecnológica

### 5.1 Tecnologías seleccionadas

Componente	Tecnología
Backend	Python 3 + Flask
Base de datos	SQLite

Componente	Tecnología
Frontend	HTML + Bootstrap
Autenticación	Flask-Login
Exportación	CSV
Despliegue	PC local en red LAN

---

## 5.2 Justificación técnica

- Software libre → sin costes de licencia
  - Instalación sencilla
  - Bajo mantenimiento
  - Adecuado para equipo reducido
  - Alineado con recursos limitados
  - Usabilidad para personal no técnico
- 

## 6. Seguridad y protección de datos

El sistema debe cumplir normativa de protección de datos.

Medidas técnicas:

- Autenticación obligatoria
  - Hash de contraseñas
  - Control de acceso por rol
  - Copias de seguridad automáticas del archivo SQLite
  - Acceso restringido a red interna
- 

## 7. Beneficios esperados

La solución permitirá:

- Eliminar el sistema manual actual
- Reducir errores y duplicidades
- Disponer de información consolidada para planificación

- Mejorar trazabilidad y organización de actividades
  - Incrementar satisfacción de usuarios
- 

## 8. Conclusión

Se propone una solución web local, simple y económicamente sostenible que cumple con el alcance definido, los requisitos funcionales y no funcionales y las restricciones de recursos del proyecto.

La arquitectura propuesta garantiza simplicidad, control, cumplimiento normativo y acceso desde todos los equipos de la red local, asegurando la viabilidad técnica y operativa del sistema.