

# Manual Técnico: Sistema "Aulas Resilientes"

Versión 1.0 - Plataforma de Comunicación y Gestión Educativa Offline

## 1. Introducción

### 1.1 Propósito

El sistema "Aulas Resilientes" es una solución tecnológica diseñada para garantizar la continuidad pedagógica y administrativa en entornos con conectividad limitada. Está dirigido exclusivamente a docentes y personal administrativo de instituciones educativas.

### 1.2 Alcance

- Comunicación colaborativa entre docentes (chat, pizarra, reuniones).
- Gestión administrativa offline (asistencia, reportes).
- Operación en red local sin dependencia de internet externo.

### 1.3 Tecnologías Clave

Componente	Tecnología
Backend	Python + Flask
Base de datos	SQLite
Frontend	PWA (JavaScript)
Comunicación en tiempo real	WebSocket + CRDT
Videollamadas	Jitsi Meet autoalojado

## 2. Arquitectura del Sistema

## 2.1 Diagrama de Componentes



Syntax error in text  
mermaid version 11.7.0

## 2.2 Flujo de Datos

1. **Datos generados offline** se almacenan en SQLite.
2. **Syncting** sincroniza con el servidor central cuando hay conexión.
3. **Jitsi** opera en la red local (192.168.1.0/24).

## 3. Módulos Detallados

---

### 3.1 Comunicación Docente

#### 3.1.1 Chat por Grado/Materia

- **Protocolo:** WebSocket con persistencia en SQLite.
- **Formato de mensajes:**

```
{  
  "timestamp": "YYYY-MM-DD HH:MM:SS",  
  "sender": "docente_id",  
  "group": "5to-Ciencias",  
  "content": "texto/markdown"  
}
```

#### 3.1.2 Pizarra Colaborativa

- **Algoritmo:** CRDT para resolución de conflictos.
- **Límites:**
  - Máx. 50 ediciones offline.
  - Sincronización cada 5 min en red.

#### 3.1.3 Reuniones Virtuales

- **URL de acceso:** `http://jitsi.aula.local` (DNS local).

- **Requisitos de ancho de banda:** 512 kbps por participante.

## 3.2 Gestión Administrativa

### 3.2.1 Registro de Asistencia

```
# Esquema de la tabla 'attendance'
CREATE TABLE attendance (
    teacher_id TEXT NOT NULL,
    class_id TEXT NOT NULL,
    timestamp DATETIME PRIMARY KEY,
    students JSON # Ej: {"presentes": ["alumno1", "alumno2"]}
);
```

### 3.2.2 Generación de Reportes

- **Dependencias:**

```
pip install weasyprint pandas
```

- **Comando para generar PDF:**

```
python generate_report.py --output informe_mensual.pdf
```

## 4. Instalación y Configuración

---

### 4.1 Requisitos del Servidor

- **Hardware mínimo:**
  - Raspberry Pi 4 (4GB RAM).
  - Almacenamiento: 32GB SD card.

### 4.2 Pasos de Implementación

1. **Preparar el entorno:**

```
sudo apt update && sudo apt install docker.io syncthing
```

2. **Desplegar Jitsi:**

```
docker run -d --restart=always \
  -p 80:80 -p 443:443 \
  -v /etc/jitsi:/config \
  -e PUBLIC_URL=http://jitsi.aula.local \
  jitsi/meet
```

### 3. Iniciar servicios Flask:

```
gunicorn -w 4 -b 0.0.0.0:5000 app:app
```

## 5. Mantenimiento

---

### 5.1 Copias de Seguridad

```
# Backup diario (ejecutar vía cron)
sqlite3 school.db ".backup /backups/${date +%F}.db"
```

### 5.2 Monitoreo

- **Métricas clave:**
  - Uso de almacenamiento SQLite ( `du -sh school.db` ).
  - Estado de WebSocket ( `netstat -tulnp | grep 5000` ).

## 6. Seguridad

---

### 6.1 Controles de Acceso

- Autenticación mediante tokens JWT.
- Cifrado AES-256 para datos sensibles.

### 6.2 Red Local

- **Configuración recomendada:**

```
[network]
subnet = 192.168.1.0/24
firewall = drop external_udp
```

# 7. Soporte Técnico

- **Registro de errores:** logs/app.log (rotación diaria).
- **Contacto:** soporte@aulasresilientes.edu.ar.

## Anexos

### A.1 Códigos de Estado HTTP

Código	Descripción
201	Datos guardados offline
503	Servidor Jitsi no disponible

### A.2 Ejemplo de CRDT en Pizarra

```
// Operación de merge
function mergeUpdates(local, remote) {
  return [...new Set([...local, ...remote])]
    .sort((a, b) => a.timestamp - b.timestamp);
}
```

Documentación actualizada al 10/06/2024

Este manual cumple con los estándares IEEE 830-1998 para especificación de requisitos de software.