Sistema de Votaciones Seguro

Manual de Instalación y Uso

16 de noviembre de 2025

Sistema web de votaciones con autenticación de dos factores (contraseña + clave privada RSA), cifrado de votos y anonimato garantizado.

Índice

1. Características

Sistema completo de votaciones seguras que implementa:

- **Autenticación 2FA**: Contraseña + Clave privada RSA
- **Cifrado de votos**: Cada voto se cifra con la clave pública de la encuesta
- **Anonimato**: Los votos no están vinculados a identidades
- **Roles**: Administrador (crea encuestas y ve resultados) y Votante
- **Tokens de un solo uso**: Cada usuario solo puede votar una vez por encuesta
- **Visualización de gráficas**: Resultados en barras, pastel, dona y línea
- **Base de datos PostgreSQL**: Almacenamiento persistente y seguro

2. Requisitos Previos

2.1. Software Necesario

- **Python 3.8+**
- **PostgreSQL 12+**
- **pip** (gestor de paquetes de Python)

2.2. Verificar Instalaciones

Para verificar que tienes todo instalado correctamente, ejecuta:

```
python3 --version
psql --version
pip3 --version
```

3. Instalación

3.1. Paso 1: Obtener el Proyecto

```
# Si usas git
git clone <url-del-repositorio>
cd sistema-votaciones

# O simplemente descargar y extraer
# los archivos en una carpeta
```

3.2. Paso 2: Instalar Dependencias de Python

```
pip3 install flask flask-cors psycopg2-binary cryptography werkzeug
```

Dependencias Instaladas

- `flask` – Framework web
- `flask-cors` – Manejo de CORS
- `psycopg2-binary` – Conector PostgreSQL
- `cryptography` – Criptografía RSA y cifrado
- `werkzeug` – Hashing de contraseñas

3.3. Paso 3: Configurar PostgreSQL

3.3.1. En Linux/Mac

```
# Iniciar PostgreSQL
sudo service postgresql start

# Acceder a PostgreSQL
sudo -u postgres psql
```

3.3.2. En Windows

```
# Abrir psql desde el menu inicio o:
psql -U postgres
```

3.3.3. Crear Base de Datos y Usuario

```
-- Crear usuario
CREATE USER votaciones_user WITH PASSWORD 'password';

-- Crear base de datos
CREATE DATABASE votaciones_db OWNER votaciones_user;

-- Dar permisos
GRANT ALL PRIVILEGES ON DATABASE votaciones_db
    TO votaciones_user;

-- Salir
\q
```

3.4. Paso 4: Configurar Credenciales

⚠ IMPORTANTE

Edita el archivo `backend.py` y modifica las credenciales de la base de datos en la sección `DB_CONFIG`.

```
DB_CONFIG = {
    'dbname': 'votaciones_db',
    'user': 'votaciones_user',
    'password': 'password',  # CAMBIAR en produccion
    'host': 'localhost',
    'port': '5432'
}
```

4. Ejecución

4.1. Iniciar el Servidor

```
python3 backend.py
```

Deberías ver en la consola:

```
============================================================
   SISTEMA DE VOTACION SEGURO CON AUTENTICACION 2FA
============================================================
Servidor ejecutandose en: http://127.0.0.1:5000
   Autenticacion: Contrasena + Clave Privada RSA
============================================================

Base de datos inicializada correctamente
 * Running on http://127.0.0.1:5000
```

4.2. Abrir en el Navegador

Abre tu navegador web y navega a:

http://localhost:5000

5. Uso del Sistema

5.1. Primer Usuario (Administrador)

El primer usuario registrado en el sistema será automáticamente asignado como **Administrador**.

1. Haz clic en **"Regístrate"**
2. Completa el formulario:
   - Nombre completo
   - Correo electrónico
   - Contraseña
3. Se descargará automáticamente tu clave privada (`.pem`)
4. **¡GUARDA ESTE ARCHIVO EN UN LUGAR SEGURO!**

Advertencia

La clave privada es esencial para iniciar sesión. Si la pierdes, no hay forma de recuperarla y deberás crear una nueva cuenta.

5.2. Usuarios Siguientes (Votantes)

Los usuarios registrados después del primero serán automáticamente **Votantes**, siguiendo el mismo proceso de registro.

5.3. Iniciar Sesión

Para iniciar sesión se requieren **tres elementos**:

1. Correo electrónico
2. Contraseña
3. Archivo de clave privada (.pem)

5.4. Funcionalidades por Rol

5.4.1. Como Administrador

- **Crear encuestas**: Botón "Nueva Votación"
- **Ver resultados**: Botón "Resultados" en cada encuesta
- **Gráficas interactivas**: Barras, pastel, dona, línea
- **Borrar encuestas**: Botón "Borrar"

5.4.2. Como Votante

- **Votar**: Hacer clic en la opción deseada
- **Ver resultados parciales**: Después de votar
- **Una votación por encuesta**: No se puede votar dos veces

6. Estructura del Proyecto

```
sistema-votaciones/
|-- backend.py          # Servidor Flask + logica de negocio
|-- index.html          # Interfaz de usuario
|-- app.js              # Logica del frontend + graficas
|-- styles.css          # Estilos CSS
`-- README.tex          # Este documento
```

7. Seguridad

7.1. Tecnologías Implementadas

1. **RSA-2048**: Generación de pares de claves público/privada
2. **RSA-OAEP**: Cifrado de votos
3. **SHA-256**: Hashing y firmas digitales
4. **Werkzeug**: Hashing seguro de contraseñas (PBKDF2)
5. **Tokens únicos**: Prevención de doble votación
6. **Autenticación 2FA**: Contraseña + clave privada

7.2. Flujo de Seguridad

Registro

1. Genera par RSA (pública/privada)
2. Guarda clave pública en BD
3. Descarga clave privada al usuario
4. Hashea contraseña con PBKDF2

Login

1. Valida contraseña (verifica hash)
2. Valida que clave privada corresponda con pública
3. Ambas verificaciones deben ser correctas

Votación

1. Usuario solicita token (firmado con su clave privada)
2. Cifra voto con clave pública de la encuesta
3. Envía voto cifrado con token de un solo uso
4. Voto es anónimo (no vinculado a identidad)

Conteo (Solo Admin)

1. Descifra votos con clave privada de la encuesta
2. Cuenta resultados
3. Muestra gráficas y estadísticas

8. Solución de Problemas

8.1. Error: "No se pudo conectar a la base de datos"

```
# Verificar que PostgreSQL esta corriendo
sudo service postgresql status

# Iniciar PostgreSQL si esta detenido
sudo service postgresql start
```

8.2. Error: "ModuleNotFoundError"

```
# Reinstalar dependencias
pip3 install flask flask-cors psycopg2-binary \
             cryptography werkzeug
```

8.3. Error: "Permission denied for database"

```
-- Reconectar a PostgreSQL y ejecutar:
GRANT ALL PRIVILEGES ON DATABASE votaciones_db
    TO votaciones_user;
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public
    TO votaciones_user;
GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA public
    TO votaciones_user;
```

8.4. Error: "Port 5000 already in use"

Cambia el puerto en `backend.py` (última línea):

```
app.run(port=5001, debug=True)  # Usar puerto 5001
```

8.5. Perdí mi Clave Privada

Sin Recuperación

No hay forma de recuperar una clave privada perdida. Deberás:

1. Registrar una nueva cuenta
2. Descargar y guardar la nueva clave privada de forma segura

9. Base de Datos

9.1. Tablas Creadas Automáticamente

- **usuarios**: Datos de usuarios y claves públicas
- **encuestas**: Encuestas con claves de cifrado

- **votos**: Votos cifrados
- **tokens_votacion**: Tokens de un solo uso

9.2. Resetear Base de Datos

```
-- Conectar a PostgreSQL
sudo -u postgres psql

-- Eliminar base de datos
DROP DATABASE votaciones_db;

-- Recrear
CREATE DATABASE votaciones_db OWNER votaciones_user;

-- Salir
\q
```

El servidor recreará automáticamente las tablas al reiniciarse.

10. Gráficas (Chart.js)

Las gráficas están disponibles **solo para administradores**:

- **Barras**: Vista vertical clásica
- **Pastel**: Distribución porcentual
- **Dona**: Similar a pastel con centro vacío
- **Línea**: Tendencia visual

Las librerías se cargan automáticamente desde CDN (no requiere instalación adicional).

11. Actualizar el Sistema

Si realizas cambios en el código:

1. Detener servidor: `Ctrl + C`
2. Guardar cambios en los archivos
3. Reiniciar servidor: `python3 backend.py`

Nota

Los cambios en archivos estáticos (HTML, CSS, JS) requieren **recargar el navegador** con `Ctrl+F5` o `Cmd+Shift+R`.

12. Notas Importantes

⚠ Este sistema es para **uso educativo/demostrativo**

⚠ Para **producción**, implementar:

- HTTPS/SSL
- Cambiar contraseña de base de datos
- Usar KMS para claves privadas de encuestas
- Rate limiting
- Logs de auditoría
- Backups automáticos

⚠ **Nunca compartir claves privadas**

⚠ **Hacer backups periódicos de la base de datos**

13. Soporte

Si encuentras problemas:

1. Verifica que PostgreSQL esté corriendo
2. Verifica las credenciales en `DB_CONFIG`
3. Revisa la consola del servidor para errores
4. Revisa la consola del navegador (F12) para errores del frontend
5. Consulta la sección de Solución de Problemas

14. Licencia

Este proyecto es de código abierto para fines educativos y de demostración.

¡Sistema listo para usar!

Para cualquier duda, revisa los logs del servidor
o la consola del navegador.