PROYECTO

Sistema de Votación Digital con Firma Ciega

Grupo 02 Semestre 2026-1

Integrantes:

- Carandia Lorenzo Brenda Fernanda
- Cuadriello Valdés Cynthia Citlalli
- Cuadriello Valdés Diana Sinsuni
- Jose Laguna Daniel
- López Sugahara Ernesto Danjiro
- Rodríguez Kobeh Santiago

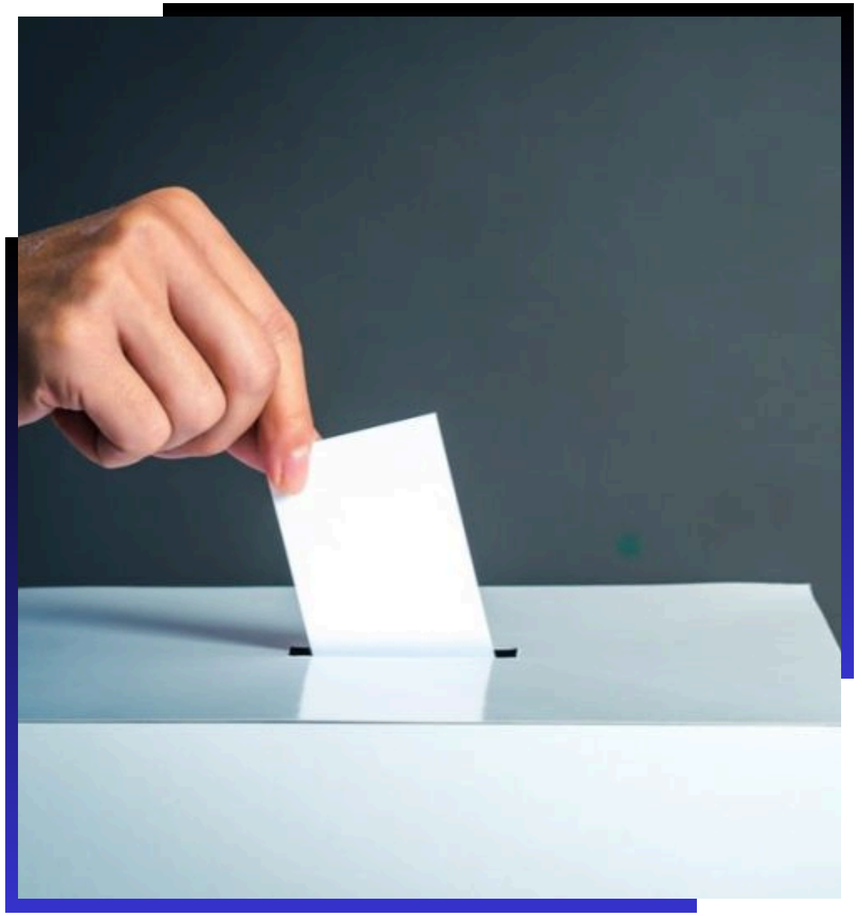
Índice de

CONTENIDOS

INTRODUCCIÓN

El proyecto consiste en el desarrollo de un Sistema de Votación Digital con Firma Ciega, cuyo propósito es demostrar la aplicación práctica de técnicas criptográficas avanzadas dentro de un entorno web.

El sistema busca garantizar la autenticación segura, la privacidad y la verificabilidad del voto, elementos fundamentales en cualquier proceso de votación digital.



OBJETIVOS

El principal objetivo fue construir una aplicación funcional que sirviera como prueba de concepto de un sistema de votación seguro. Entre los objetivos específicos destacan:

01 **Diseñar un sistema de autenticación con gestión de usuarios y roles.**

02 **Implementar la generación y manejo de llaves RSA para cada usuario.**

03 **Utilizar SHAke128 con salt para almacenar contraseñas de forma segura, e integrar la firma ciega RSA para permitir votos anónimos pero verificables.**

El alcance del proyecto se mantuvo dentro de la complejidad esperada para un trabajo de licenciatura

STACK TECNOLÓGICO

Backend

El backend fue desarrollado en Python, utilizando el microframework Flask, que permite un control granular del flujo de la aplicación sin sobrecargar el proyecto con dependencias innecesarias.

Base de Datos

La base de datos empleada fue SQLite, elegida por su simplicidad y por no requerir un servidor externo, ideal para entornos de desarrollo local.

Criptografía

En la parte criptográfica se usaron las librerías hashlib (para SHAke128) y cryptography (para RSA y firma ciega).

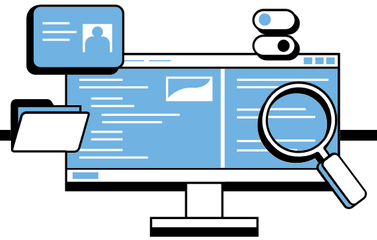
Formularios

El manejo de formularios y autenticación se implementó con Flask-WTF y Flask-Login, mientras que el frontend se desarrolló con HTML, CSS y JavaScript puro, evitando frameworks pesados para mantener la claridad del código.

ESTRUCTURA DEL PROYECTO

01

La estructura del sistema se organizó de forma modular, buscando separar claramente la lógica, la interfaz y las utilidades criptográficas.



MÓDULOS

01 ARCHIVOS PYTHON

El archivo principal app.py inicializa la aplicación Flask, mientras que models.py define las entidades de la base de datos y routes.py contiene las rutas y vistas.

En crypto_utils.py se ubican todas las funciones criptográficas, desde el manejo de llaves RSA hasta la firma y verificación de mensajes.

02 PLANTILLAS HTML

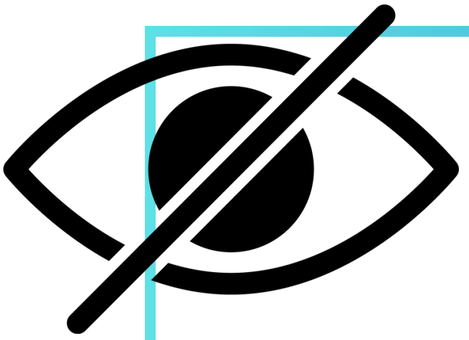
Incluye plantillas HTML dentro del directorio templates/ y archivos estáticos (CSS y JS) en static/. Finalmente, los archivos de claves pública y privada de la autoridad se generan automáticamente para el manejo de la firma ciega.

HASHING Y CRIPTOGRAFÍA RSA

Para proteger las contraseñas de los usuarios se utilizó la función SHAke128, perteneciente a la familia de algoritmos SHA-3. Esta función transforma las contraseñas en valores de longitud fija imposibles de revertir. A cada contraseña se le añade una sal (salt) aleatoria y única antes de ser hasheada, lo que evita que contraseñas idénticas generen el mismo resultado.

El sistema utiliza el algoritmo RSA para implementar criptografía asimétrica basada en un par de llaves: una pública y una privada.

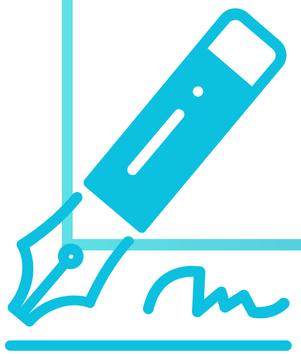
Cada usuario genera sus propias llaves al registrarse. La clave pública se almacena en la base de datos y puede compartirse libremente, mientras que la clave privada se cifra antes de guardarse, utilizando una clave derivada de la contraseña del usuario.



FIRMA CIEGA RSA

La firma ciega RSA es el corazón criptográfico del proyecto. Su función es permitir que la autoridad de votación firme un mensaje, en este caso, el voto, sin conocer su contenido.

El proceso consiste en que el votante “ciega” su voto con un factor matemático, lo envía a la autoridad para su firma, y luego lo “des-ciega” para obtener una firma válida sobre el mensaje original.
Gracias a este protocolo, la autoridad no puede asociar el voto final con el votante que lo solicitó, garantizando el anonimato.



C O N C L U S I O N E S

El proyecto demuestra la viabilidad de un sistema de votación digital seguro y verificable, implementado con herramientas accesibles y tecnologías modernas.

A través del uso de RSA, SHAke128 y firma ciega, se lograron los objetivos de autenticidad, integridad y privacidad del voto.
La arquitectura basada en Flask y SQLite permitió un desarrollo ágil y transparente, adecuado para un nivel académico, pero con fundamentos aplicables a proyectos reales.

GRACIAS POR SU ATENCIÓN