

# Propuesta D como Proyecto Final

por Joan Pastor Vicéns

## Gestor de Contraseñas

19 de octubre del 2024

### Escenario

El mundo actual ha avanzado lo suficientemente rápido como para que más del 90% de las acciones se realicen a través de la computadora siendo la creación de credenciales lo más importante para ello. Aunque el aumento de la ciberseguridad ha proporcionado maneras sencillas de almacenar dichas credenciales, siguen siendo multinacionales privadas las que manejan dichos datos. Las políticas de privacidad implementadas suelen ser unos documentos largos y poco intuitivos para el usuario medio, haciendo que éste las acepte sin tan siquiera terminar de leerlas y obviando, por tanto, qué ocurre con sus datos.

Es por ello que mi propuesta es la realización de una aplicación de escritorio que sea capaz de almacenar contraseñas o información importante de forma cifrada, ya sea el acceso a cualquier tipo de cuenta web, tarjetas de crédito o notas seguras.

### Requisitos mínimos

1. **Base de datos:** Dicho proyecto deberá tener una base de datos, en este caso utilizaremos **SQLite** y se empleará **SQLAlchemy** como *ORM* para la creación de las tablas de datos.
2. **Programación Orientada a Objetos:** Esta aplicación contará con la *PОО* para la creación de usuarios, tarjetas de crédito, notas seguras y direcciones *web*.

3. **El framework elegido será Flet:** Flet\* es un *framework* actual, basado en *flutter* que permite la creación de interfaces multiplataforma. Se puede utilizar para aplicaciones de escritorio y aplicaciones *web* o móviles.
4. **Autenticación:** Para utilizar la aplicación, el usuario deberá crear un usuario y contraseña al cual se asociará cualquier tipo de dato almacenado en la base de datos.
5. **Documentación del proyecto:** Se dispondrá de un proyecto escrito con todas las fases explicadas del mismo donde se tendrá en cuenta la investigación, los ejemplos utilizados y sus elecciones finales.

## Especificaciones

Esta aplicación se encargará de crear un usuario para poder autenticarse. El usuario creará un nombre de usuario y una contraseña maestra para poder realizar cualquier acción. Una vez en su página principal, podrá añadir, ver, modificar o eliminar los datos almacenados (ya sean nombres de usuario, tarjetas de crédito o notas seguras).

Se tendrá en cuenta la encriptación y la desencriptación de los datos entre la petición del usuario y el resultado de la consulta a la base de datos. Es decir: cuando el usuario añada algún tipo de información, esta se encriptará previamente antes de ser almacenada en la base de datos. Si el usuario quiere ver cualquier dato, dicha información se desencripta una vez se solicite y [se mostrará durante un breve periodo de tiempo](#).

Cualquier acción solicitada por el usuario pasará previamente por un filtro que solicitará su contraseña maestra para dar otra capa de seguridad.

## Hitos

### 1. Aprender cómo funciona una API

Al ser una aplicación de uso interno, utilizaremos una [API rest](#) ya que consume menos recursos. Sus peticiones suelen ser más livianas y necesitan menos protocolos de seguridad, haciéndolas más sencillas de configurar.

### 2. Encriptación y Desencriptación

Utilizar el módulo [cryptography](#) nos ayudará a entender los pasos específicos a realizar antes de la encriptación de cualquier tipo de dato o de cómo deben tratarse.

### 3. Tablas Relacionales

Al tener diferentes modelos en este proyecto, aprenderemos a [relacionar cada tabla](#) (usuario, tarjetas, notas y direcciones web) con el usuario creado.

### 4. Nuevo framework

Se ha elegido el framework Flet por la característica principal de ser un framework [multiplataforma](#). Esta opción nos permite una escalabilidad en el proyecto, pudiendo unificar su uso ya sea en web o aplicación móvil.

### 5. Organización de un proyecto

Este ejercicio es una buena oportunidad para aprender cómo [organizar cada elemento](#) dentro de una aplicación python.