

## Proyecto - Modulo 2

### Presentado por:

Andrés Felipe Alarcón Pulido - [analarconp@unal.edu.co](mailto:analarconp@unal.edu.co)

Juan Daniel Jossa Soliz - [jjossa@unal.edu.co](mailto:jjossa@unal.edu.co)

Michel Mauricio Castaneda Braga - [micastanedab@unal.edu.co](mailto:micastanedab@unal.edu.co)

Jaime Darley Angulo Tenorio - [jangulot@unal.edu.co](mailto:jangulot@unal.edu.co)

### Profesor:

Oscar Eduardo Alvarez Rodriguez

[oalvarezr@unal.edu.co](mailto:oalvarezr@unal.edu.co)

11 de Julio



**Universidad Nacional de Colombia**  
**Facultad de Ingeniería**  
**Departamento de Ingeniería de Sistemas e Industrial**  
**2025**

## 1. Lenguaje de Programación y Framework Seleccionado

- **Lenguaje Principal:** Python
- **Framework para la Interfaz Gráfica:** PyQt5

### Comparativo de Tecnologías

Tecnología	Ventajas	Desventajas
<b>Python (PyQt5)</b>	<ul style="list-style-type: none"><li>- Potente y moderno para interfaces complejas</li><li>- Multiplataforma</li><li>- Soporte para diseños avanzados y estilos CSS</li><li>- Integración sencilla con librerías de cifrado y SQLite</li></ul>	<ul style="list-style-type: none"><li>- Mayor complejidad frente a Tkinter</li><li>- Licencia GPL (o comercial para aplicaciones propietarias)</li></ul>
<b>Java (Swing/JavaFX)</b>	<ul style="list-style-type: none"><li>- Mejor rendimiento en GUI más pesadas</li><li>- Mayor control visual</li></ul>	<ul style="list-style-type: none"><li>- Curva de aprendizaje mayor</li><li>- Verbosidad en el desarrollo</li></ul>
<b>C# (.NET WinForms/WPF)</b>	<ul style="list-style-type: none"><li>- Alta integración con Windows</li><li>- Herramientas avanzadas para GUI</li></ul>	<ul style="list-style-type: none"><li>- Dependencia fuerte de entorno Windows</li><li>- Menor portabilidad</li></ul>
<b>Electron (JavaScript + HTML/CSS)</b>	<ul style="list-style-type: none"><li>- Interfaz moderna y atractiva</li><li>- Amplia comunidad</li></ul>	<ul style="list-style-type: none"><li>- Alto consumo de recursos</li><li>- Tamaño considerable del ejecutable final</li></ul>

### Justificación:

Python fue seleccionado por su equilibrio entre simplicidad y poder. La prioridad del proyecto FortiFile es la seguridad local, no la complejidad visual. PyQt5 proporciona una interfaz más moderna y flexible que Tkinter, permitiendo escalar visualmente el producto si es necesario. Además, el equipo tiene experiencia previa con Python, lo que facilita la implementación eficiente y el cumplimiento de estándares como PEP8 y Clean Code.

## 2. Base de Datos Relacional Seleccionada

- SQLite

Comparativo con otras opciones:

Base de Datos	Ventajas	Desventajas
SQLite	<ul style="list-style-type: none"><li>- Ligera y embebida</li><li>- Sin configuración externa</li><li>- Ideal para uso local y monousuario</li></ul>	<ul style="list-style-type: none"><li>- No apta para aplicaciones multiusuario</li><li>- Sin capacidades de red</li></ul>
PostgreSQL	<ul style="list-style-type: none"><li>- Potente y escalable</li><li>- Soporte completo a SQL estándar</li></ul>	<ul style="list-style-type: none"><li>- Necesita servidor y configuración externa</li></ul>
MySQL	<ul style="list-style-type: none"><li>- Amplio soporte y comunidad</li></ul>	<ul style="list-style-type: none"><li>- Requiere instalación de servidor externo</li></ul>

### Justificación:

Dado que FortiFile es una aplicación monousuario y completamente local, SQLite es ideal por su simplicidad y bajo consumo de recursos. Su integración con Python es directa y no requiere servicios en segundo plano.

## 3. Bibliotecas y Herramientas Complementarias Planeadas

Biblioteca / Herramienta	Propósito
cryptography (Python)	Cifrado de archivos mediante AES o RSA
bcrypt + hashlib	Hash y almacenamiento seguro de contraseñas
unittest o pytest	Pruebas automatizadas (unitarias e integración)

logging	Registro de eventos críticos en bitácora local
PyQt5	Creación de la interfaz gráfica
PEP8, flake8	Verificación de estilo de código
Patrones de diseño (Singleton, Factory)	Modularidad y mantenibilidad del código

#### Justificación:

Las bibliotecas fueron seleccionadas por ser estándares de la industria y altamente compatibles con Python. Permiten implementar funcionalidades críticas del proyecto como el cifrado, la autenticación y la seguridad general, sin complejidad adicional para el equipo.

#### 4. Justificación General de Elección Tecnológica

Las elecciones tecnológicas del proyecto FortiFile se fundamentan en los siguientes pilares:

- **Orientación al objetivo del sistema:** Se requiere un sistema seguro, local, simple y multiplataforma. Las tecnologías seleccionadas (Python + PyQt5+ SQLite) permiten cumplir estos requerimientos con bajo costo de implementación y alta portabilidad.
- **Capacidades del equipo:** Todos los integrantes del equipo cuentan con experiencia previa en Python y están familiarizados con sus bibliotecas de seguridad, pruebas y base de datos. Esto garantiza una curva de aprendizaje mínima y enfoque efectivo en la lógica del negocio.
- **Relación con los objetivos del curso:** Se busca aplicar principios de desarrollo seguro, buenas prácticas de codificación, patrones de diseño, y pruebas automatizadas. Python y su ecosistema permiten cubrir ampliamente estos aspectos, fomentando la calidad del software desde el inicio del desarrollo.