

# Estructura de un Programa de Capacitación en Ciberseguridad para PYMEs

Este documento describe la estructura y el plan de desarrollo de una aplicación educativa para capacitar al personal de pequeñas y medianas empresas (PYMEs) en la detección de anomalías de red y respuesta a incidentes cibernéticos. El enfoque está en brindar un entorno interactivo similar a Wireshark para monitoreo, con simulaciones de ataques y material instructivo.

## 1. Estructura de Carpetas

La app estará organizada así:

```
/app/
|
|-- /core/                # Lógica de negocio (detección de anomalías, simulaciones,
etc.)
|   |-- monitor.py        # Captura y análisis de paquetes (parecido a Wireshark)
|   |-- simulator.py      # Simulaciones de ataques
|   |-- utils.py          # Funciones auxiliares
|
|-- /gui/                 # Interfaz gráfica (Tkinter)
|   |-- main.py           # Ventana principal y navegación entre pestañas
|   |-- dashboard.py      # Módulo de dashboard
|   |-- docs_viewer.py    # Visor de documentación
|   |-- monitor_view.py   # Interfaz del monitor de red
|   |-- simulator_view.py # Simulador de ataques
|   |-- info.py           # Información adicional
|
|-- /docs/                # Documentación y recursos
|   |-- *.pdf, *.md
|
|-- /data/                # Logs simulados, configuraciones de red, capturas pcap
|
|-- /assets/              # Imágenes, íconos, etc.
|
-- app.py                 # Punto de entrada principal
```

## 2. Paso a Paso para Empezar a Programar

1. Crea la estructura de carpetas como se mostró arriba.
2. Comienza por el núcleo de la aplicación (/core/):
  - Implementa `monitor.py` para capturar paquetes usando `scapy` o `pyshark`.
  - Luego implementa `simulator.py` para simular escenarios como DDoS, ARP Spoofing, etc.
3. Cuando tengas la lógica lista y probada por consola, comienza la GUI (/gui/):
  - Crea una ventana principal en `main.py` con navegación entre pestañas (usando Frames de Tkinter).
  - Integra las vistas de cada módulo (`dashboard.py`, `monitor_view.py`, etc.).
4. Integra los datos y documentación en `/docs/` y `/data/`.
5. Usa `app.py` para iniciar y coordinar todo.

Consejo: usa clases para que cada parte de la GUI sea modular y fácil de modificar.

### 3. Notas Finales

No empieces con la GUI. Desarrollar primero la lógica te va a ahorrar mucho tiempo y dolores de cabeza.

Una vez que la lógica esté funcional desde consola, lo demás es solo vestirla.

Además, mantener separadas las carpetas ayuda a que el proyecto sea más mantenible y profesional.

Tip extra: guarda las configuraciones en un archivo `.env` o `config.json` si planeas escalarlo.