

Taller Scripts

¿Qué es un *script*?

Un **script** es un conjunto de instrucciones escritas en un lenguaje interpretado que le dicen a una computadora qué hacer paso a paso.

- No necesita ser *compilado*, sino que se **interpreta directamente** por un programa (como Python, Bash, PowerShell, etc.).
- Su objetivo es **automatizar tareas repetitivas**, procesos de mantenimiento o flujos simples.
- Se usa en sistemas operativos, servidores, entornos web, ciencia de datos, y hasta videojuegos.

Lenguajes de *scripting*

No es un tipo de lenguaje, sino un **modo de uso**.

Los más comunes son:

- **Python** → scripts de automatización, análisis, IA, backend, etc.
- **Bash / Shell Script** → administración de sistemas Linux.
- **PowerShell** → automatización en Windows.
- **JavaScript** → scripting web (del lado del navegador o servidor con Node.js).

Un lenguaje se vuelve *de scripting* cuando se usa para **automatizar** o **controlar otros programas**.

¿Qué es un *intérprete*?

El **intérprete** es el programa que lee y ejecuta las instrucciones del script **en tiempo real**.

Ejemplo:

- En Python: `python archivo.py`
- En Bash: `bash script.sh`
- En JavaScript: el navegador o `node archivo.js`

El intérprete traduce las instrucciones humanas a operaciones que entiende la CPU o el sistema operativo.

¿Qué es la automatización?

La **automatización** es el objetivo central de los scripts: dejar que la máquina haga el trabajo repetitivo.

Ejemplo:

- Ejecutar una tarea cada día.
- Limpiar logs antiguos.
- Generar reportes automáticos.
- Monitorear precios, temperatura, o servidores.

Cuanto más predecible sea una tarea, más fácil es automatizarla con un script.

Ejemplos de scripts

- 📁 Limpieza de archivos antiguos.
- 📁 Generar copias de seguridad automáticas.
- 🌐 Descargar datos de una API o web.
- 📊 Procesar hojas de cálculo o logs.
- 📦 Monitorear uso de CPU o red.
- 💬 Enviar alertas o correos automáticos.
- 🎮 Automatizar tareas dentro de juegos o servidores.

Ideas de proyectos

🏁 NIVEL 1: Automatización cotidiana (individual / práctica)

Objetivo: mostrar que un script puede *ahorrar tiempo* o *hacer cosas útiles* sin ser "gigante".

◆ Ejemplo 1: Limpieza automática de archivos

Proyecto: Script que organiza tu carpeta de Descargas.

Qué hace: Clasifica archivos por tipo (PDF, imágenes, ZIPs, ejecutables) y los mueve a carpetas específicas.

Valor competitivo: Muestra *impacto personal inmediato* y noción de automatización local.

Extras para subir nivel: interfaz simple con Tkinter o CLI con logs.

◆ Ejemplo 2: Generador de reportes automáticos

Proyecto: Script que analiza un CSV (ventas, gastos, horas de trabajo) y genera un PDF con gráficos.

Tecnologías: `pandas` , `matplotlib` , `reportlab` .

Valor: demuestra dominio de datos y presentación automatizada.

Idea pro: incluir envío automático del PDF por correo.

◆ Ejemplo 3: Script de respaldo inteligente

Proyecto: "SmartBackup.py" que detecta archivos nuevos y crea un ZIP con fecha.

Valor: útil para oficinas, fotógrafos, o desarrolladores.

Competencia: se puede presentar como *automatización de procesos administrativos*.

NIVEL 2: Automatización conectada (proyectos con APIs y datos)

Aquí se pasa de scripts locales a scripts **que hablan con internet** o con otros servicios.

◆ Ejemplo 4: Monitoreo de precios o clima

Proyecto: Script que consulta una API (por ejemplo, precios de criptos o clima) y envía alertas.

Valor: combina requests, lógica y alertas → se siente como un mini-sistema real.

Competencia: "herramienta de toma de decisiones", "dashboard de información".

◆ Ejemplo 5: Bot de control remoto

Proyecto: Script que escucha un canal de Discord o Telegram y responde a comandos (por ejemplo, encender luces IoT, ejecutar tareas, o dar info del sistema).

Valor: mezcla software y hardware.

Impacto: automatización del hogar o administración remota → excelente para ferias o hackathons.

◆ Ejemplo 6: Scraper de información útil

Proyecto: Script que recolecta datos de sitios web (libros, precios, noticias, etc.) y los guarda en una base de datos local.

Valor: enseña cómo generar datasets propios → ideal para análisis posterior o dashboards.

Extra: visualización web simple con Flask.

NIVEL 3: Mini-proyectos de innovación (competencia seria)

Acá ya entramos a nivel *producto mínimo viable (MVP)*, pero con base en scripts.

◆ Ejemplo 7: "AutoReport"

Idea: un sistema que todos los días lee un conjunto de archivos (ventas, clima, stock, etc.), genera un PDF y lo envía por mail.

Conceptos: tareas programadas, SQLite, formateo de reportes, automatización.

Valor en competencia: combina análisis, persistencia y presentación.

Impacto real: gestión de negocios, automatización contable, monitoreo industrial.

◆ Ejemplo 8: "AlertBot"

Idea: script que monitorea algo (por ejemplo, temperatura del CPU, o disponibilidad de una web) y envía alerta por Telegram o Discord.

Demuestra: integración API, eventos automáticos, respuesta inmediata.

Competencia: proyectos de ciberseguridad o DevOps.

◆ Ejemplo 9: "EcoTrack"

Idea: script que usa datos de una API pública (por ejemplo, contaminación del aire o clima) y genera alertas visuales o gráficas.

Valor: vincula datos públicos, análisis y conciencia ambiental.

Bonus: si lo corrés en Raspberry Pi, tenés *hardware + software + impacto social*.