

# MODULO 2 – Capítulo 4

## Software Libre

El **software libre** es un tipo de software que garantiza a los usuarios ciertas libertades fundamentales. Estas libertades incluyen:

1. **Libertad de uso:** Puedes usar el software para cualquier propósito.
2. **Libertad de estudio:** Tienes acceso al código fuente para analizar cómo funciona y modificarlo si lo deseas.
3. **Libertad de distribución:** Puedes compartir copias del software con otras personas.
4. **Libertad de mejora:** Puedes mejorar el software y distribuir tus mejoras para que otros las utilicen.

Estas libertades promueven el control del software por parte de los usuarios y fomentan la colaboración y la innovación.

### Diferencias entre Software Libre y Software Privativo

- **Acceso al código fuente:** El software libre siempre proporciona acceso al código fuente, mientras que el software privativo no.
- **Control:** El software libre permite a los usuarios modificar el software según sus necesidades, mientras que el privativo está controlado exclusivamente por el desarrollador o la empresa propietaria.
- **Distribución:** Puedes compartir software libre libremente, pero el software privativo suele tener restricciones legales o técnicas para evitar su distribución no autorizada.
- **Costo:** Aunque no siempre es gratis, el software libre suele estar disponible sin costo, mientras que el software privativo generalmente requiere el pago de licencias.

### Diferencias entre Software Libre y Open Source

Aunque ambos conceptos están relacionados, no son idénticos:

#### 1. Enfoque filosófico:

- El software libre tiene un enfoque ético y político, centrado en las libertades y derechos de los usuarios.
- El término **open source** (código abierto) es más práctico y se centra en los beneficios técnicos de tener un código abierto, como una mayor seguridad, calidad y colaboración.

#### 2. Definición:

- El software libre sigue las directrices de la **Free Software Foundation** (FSF), que pone énfasis en las libertades mencionadas.
- El open source se rige por la definición de la **Open Source Initiative** (OSI), que se enfoca en criterios técnicos y prácticos para que el software sea considerado "de código abierto".

### 3. Percepción empresarial:

- "Open Source" es un término más aceptado en el ámbito empresarial porque evita connotaciones políticas, mientras que "software libre" puede percibirse como más radical.

### 4. Compatibilidad: Todo software libre es también open source, pero no todo software open source cumple estrictamente con las libertades que definen al software libre.

En resumen, la diferencia principal radica en el enfoque: el software libre pone las libertades de los usuarios como prioridad, mientras que el open source enfatiza los aspectos colaborativos y técnicos del desarrollo de software abierto.

## ¿Qué es GNU?

GNU es un proyecto de software libre iniciado en 1983 por **Richard Stallman** con el objetivo de desarrollar un sistema operativo completamente libre, similar a Unix, pero compuesto exclusivamente de software libre. El acrónimo GNU significa "**GNU's Not Unix**", un juego de palabras que refleja su conexión conceptual con Unix, pero sin incluir código propietario.

El sistema operativo GNU está compuesto por:

- **Núcleo del sistema:** originalmente, GNU intentaba desarrollar su propio núcleo llamado **Hurd**, aunque no se completó en el tiempo esperado.
- **Herramientas de software:** GNU incluye un conjunto de herramientas esenciales como el compilador **GCC**, el editor **Emacs**, y utilidades básicas (como bash, grep, awk, etc.), necesarias para un sistema operativo funcional.

## ¿Qué es Linux y cómo se relaciona con GNU?

**Linux** es un núcleo (kernel) desarrollado por **Linus Torvalds** en 1991. Aunque Linux es solo el núcleo, frecuentemente se usa en combinación con el sistema GNU, dando lugar a lo que conocemos como **GNU/Linux**, un sistema operativo completo que combina el núcleo Linux con las herramientas y programas del proyecto GNU.

### Diferencias entre GNU y Linux

Aspecto	GNU	Linux
<b>Naturaleza</b>	Sistema operativo completo (sin núcleo)	Núcleo del sistema operativo
<b>Enfoque</b>	Libertad del software	Eficiencia y funcionalidad
<b>Inicio del proyecto</b>	1983	1991
<b>Finalidad</b>	Proveer un sistema libre completo	Proveer un núcleo eficiente y versátil

Aunque ambos son fundamentales para los sistemas operativos libres, el término **GNU/Linux** es más preciso cuando nos referimos al sistema operativo completo que incluye el núcleo Linux y las herramientas GNU.

---

## ¿Quién es Richard Stallman?

**Richard Stallman (RMS)** es un programador y activista del software libre. Nació el 16 de marzo de 1953 en Nueva York, Estados Unidos. Es conocido por ser el fundador del **Movimiento del Software Libre** y de la **Free Software Foundation (FSF)**, una organización dedicada a promover la libertad en el uso y desarrollo de software.

### Aportaciones destacadas

1. **Proyecto GNU (1983):** Stallman lanzó el proyecto GNU para crear un sistema operativo completamente libre.
2. **Creación de licencias libres:** Stallman diseñó la **Licencia Pública General de GNU (GPL)**, que garantiza las libertades de uso, estudio, modificación y distribución del software.
3. **Desarrollo de herramientas:** Escribió varias herramientas clave del software libre, como el editor de texto **GNU Emacs**, el compilador **GCC** y el depurador **GDB**.
4. **Filosofía del software libre:** Stallman estableció las bases éticas del movimiento, argumentando que el software debe respetar las libertades de los usuarios.

### Papel histórico

- Stallman es una figura central en la historia del software libre. En los años 80, cuando la industria del software comenzó a cerrar el acceso al código fuente, él propuso un modelo radicalmente diferente, basado en la cooperación y la libertad.
  - Su activismo inspiró a comunidades y proyectos globales, como el movimiento **open source** (aunque Stallman lo critica por no centrarse en la ética).
- 

### Resumen

- **GNU** es un sistema operativo libre sin núcleo, mientras que **Linux** es un núcleo usado frecuentemente junto con GNU para formar GNU/Linux.
- **Richard Stallman** es el fundador del software libre, conocido por sus aportes técnicos (como GNU y GPL) y por ser el defensor más prominente de la libertad en el software. Su trabajo ha moldeado tanto la ética como la práctica del desarrollo de software en las últimas décadas.

## ¿Qué son las distribuciones de Linux?

Una **distribución de Linux** (o distro) es un sistema operativo basado en el núcleo Linux, combinado con software adicional (como herramientas GNU, gestores de paquetes, entornos gráficos, etc.) para ofrecer una solución completa y lista para ser usada en servidores, escritorios, dispositivos embebidos y más.

Las distribuciones suelen personalizarse según sus objetivos: facilidad de uso, estabilidad, seguridad, soporte para hardware específico o aplicaciones particulares (como desarrollo, servidores, gaming, etc.).

---

## Distribuciones más habituales

### Distribuciones populares para usuarios generales:

1. **Ubuntu:** Fácil de usar, orientada a principiantes y basada en Debian.
2. **Debian:** Estable y altamente configurable, preferida en entornos de servidores.
3. **Fedora:** Innovadora, con tecnologías de vanguardia.
4. **Arch Linux:** Minimalista y personalizable, ideal para usuarios avanzados.
5. **Linux Mint:** Amigable para usuarios de Windows, basada en Ubuntu/Debian.
6. **openSUSE:** Enfocada en estabilidad y herramientas administrativas (YaST).
7. **CentOS Stream (antes CentOS):** Usada en servidores, derivada de Red Hat.

### Distribuciones especializadas:

- **Kali Linux:** Orientada a seguridad informática y pruebas de penetración.
  - **Raspberry Pi OS:** Diseñada para Raspberry Pi, basada en Debian.
  - **Pop!\_OS:** Optimizada para gaming y trabajo, basada en Ubuntu.
  - **Alpine Linux:** Ligera, usada en contenedores y entornos minimalistas.
- 

## Tabla de ramas y derivados principales

Rama Principal	Distribuciones Derivadas	Características
Debian	Ubuntu, Linux Mint, MX Linux, Raspberry Pi OS, Zorin OS	Estabilidad y comunidad activa; base para muchas distros populares.
Ubuntu	Kubuntu, Xubuntu, Pop!_OS, Elementary OS, Linux Lite, Zorin OS	Orientada a principiantes, amplia compatibilidad con software y hardware.
Red Hat	Fedora, CentOS Stream, Rocky Linux, AlmaLinux	Enfocada en servidores y empresas; Red Hat es comercial, los derivados son comunitarios.
Arch Linux	Manjaro, EndeavourOS, Garuda Linux	Filosofía KISS (Keep It Simple, Stupid), rolling release, ideal para usuarios avanzados.
Slackware	Salix, Slackel	Una de las más antiguas; minimalista y para usuarios que prefieren configuraciones manuales.
openSUSE	GeckoLinux, openSUSE Leap	Equilibrio entre facilidad de uso y herramientas avanzadas como YaST.
Gentoo	Calculate Linux, Sabayon	Compilación personalizada para rendimiento y optimización.
Independent	Solus, Void Linux, Clear Linux	No basadas en otras distribuciones; suelen centrarse en un enfoque único o innovador.

---

## ¿Cómo elegir una distribución?

Dependerá del propósito y la experiencia del usuario:

- **Principiantes:** Ubuntu, Linux Mint, Zorin OS.
- **Entornos empresariales:** Red Hat, CentOS Stream, Debian.
- **Usuarios avanzados:** Arch Linux, Gentoo, Slackware.
- **Gaming:** Pop!\_OS, Fedora.
- **Seguridad y hacking ético:** Kali Linux, Parrot OS.

Cada distro tiene su propio enfoque y comunidad, lo que permite una amplia personalización para satisfacer diferentes necesidades.

## ¿Qué es el WSL (Windows Subsystem for Linux)?

El **Windows Subsystem for Linux (WSL)** es una característica de Windows que permite ejecutar un entorno GNU/Linux, incluyendo herramientas y utilidades de línea de comandos, directamente en Windows sin la necesidad de una máquina virtual o configuración de arranque dual.

Con WSL puedes usar distribuciones de Linux (como Ubuntu, Debian o Fedora) para tareas de desarrollo, como ejecutar scripts, administrar servidores o trabajar con herramientas de código abierto, todo integrado en el entorno de Windows.

---

## Versiones de WSL

- **WSL 1:** Más simple, traduce las llamadas al sistema Linux para que las entienda el kernel de Windows. Ofrece menos compatibilidad con características avanzadas de Linux.
  - **WSL 2:** Utiliza un núcleo Linux real mediante una máquina virtual ligera. Es más rápido y compatible con tecnologías como Docker.
- 

## ¿Cómo instalar WSL en Windows 10 y 11?

### Requisitos previos:

- **Windows 10:** Versión 1903 o superior (compilación 18362 o más reciente).
  - **Windows 11:** Viene con WSL preconfigurado para ser más fácil de instalar.
  - Activar la virtualización desde la BIOS (si está deshabilitada).
- 

## Instrucciones para instalar WSL en Windows 10 y Windows 11

### 1. Abrir PowerShell como administrador:

- Busca "PowerShell" en el menú de inicio, haz clic derecho y selecciona "Ejecutar como administrador".

### 2. Ejecutar el comando de instalación rápida de WSL:

- En Windows 11 o en versiones actualizadas de Windows 10:

```
wsl --install
```

Este comando:

- Habilita WSL.
- Instala el núcleo de Linux.
- Descarga e instala la distribución **Ubuntu** por defecto.

### 3. (Opcional) Seleccionar una distribución específica:

- Para ver las distribuciones disponibles:

```
wsl --list --online
```

- Instalar una distribución específica (por ejemplo, Debian):

```
wsl --install -d Debian
```

### 4. (Opcional) Actualizar a WSL 2: Si estás usando una versión más antigua de Windows 10, puede que necesites actualizar manualmente a WSL 2:

- Verifica la versión actual de WSL:

```
wsl --list --verbose
```

- Cambia la versión por defecto a WSL 2:

```
wsl --set-default-version 2
```

### 5. Configurar tu distribución:

- Una vez instalada la distribución, ábrela desde el menú de inicio.
- Se te pedirá que configures un usuario y contraseña para el entorno Linux.

---

## Instalación manual en versiones antiguas de Windows 10 (opcional):

Si no puedes usar el comando `wsl --install`, sigue estos pasos:

### 1. Habilitar características opcionales:

- Ve a "Panel de Control > Programas > Activar o desactivar características de Windows".
- Activa:
  - "Subsistema de Windows para Linux".
  - "Plataforma de máquina virtual" (necesaria para WSL 2).

### 2. Descargar el núcleo de Linux:

- Ve al sitio oficial de Microsoft y descarga el paquete del núcleo de Linux.

### 3. Instalar la distribución:

- Descarga una distribución desde la [Microsoft Store](#).
-

## ¿Por qué usar WSL?

- **Desarrollo cruzado:** Permite trabajar en un entorno Linux desde Windows.
- **Ligero:** No requiere configurar una máquina virtual completa.
- **Compatibilidad:** WSL 2 usa un núcleo Linux real, ofreciendo mayor compatibilidad con herramientas modernas.
- **Integración:** Puedes combinar comandos de Windows y Linux en un mismo entorno, usando herramientas como PowerShell y Bash.

## ¿Qué versión es mejor?

- **WSL 1:** Ideal para sistemas más antiguos o cuando no necesitas un rendimiento avanzado.
- **WSL 2:** Recomendado para la mayoría de los usuarios, especialmente si trabajas con Docker o herramientas avanzadas.