Institut Puig Castellar
Santa Coloma de Gramenet

Decuria: Una distribución de Linux

Proyecto de desarrollo

CFGS Administració de Sistemes Informàtics i Xarxes

Syam Al Rashid Swity | Sergi Leon Cornejo

GRUP: 2ASIX

16/05/2026

GNU Free Documentation License (GNU FDL)

Resumen del proyecto (máximo 250 palabras):

Decuria, una distribución de Linux basada en Ubuntu Server con un entorno gráfico creado desde 0 (en realidad es un gestor de ventanas en este caso pero profundizaremos más adelante sobre este aspecto) por ende con iconos, temas, barra de tareas, gestor de archivos, pantalla de inicio de sesión e incluso animaciones creadas por nosotros mismos, por lo tanto original. Buscamos distribuir esta distribución en formato ISO a través de una página web también creada por nosotros a través de XML y la ISO con Penguins egg's.

La parte más interesante del proyecto se basa en **optimizar** lo máximo posible esta distribución a través de navegar y gestionar el sistema operativo, así como eliminar el bloatware, además de crear herramientas que vendrán incluidas en el propio sistema como un programa/panel más, herramientas encargadas principalmente en que el usuario pueda gestionar el rendimiento según sus necesidades.

La razón de **Decuria** es proporcionar una experiencia ligera, óptima y moderna sin sacrificar de manera notoria el apartado gráfico, pensado para ordenadores con especificaciones más limitadas, así dándoles una nueva vida.

Paraules clau (entre 4 i 8):

Decuria, Linux, Distribución, Rendimiento, Sistema operativo

Abstract (in English, 250 words or less):

Decuria is a Linux distribution based on Ubuntu Server with a graphical environment created entirely from scratch. Although technically based on a window manager, the system includes its own icons, themes, taskbar, file manager, login screen and even animations developed specifically for the project, providing a unique and original visual identity.

The distribution is intended to be distributed in ISO format through a website also developed as part of the project. The ISO image was generated using Penguins Eggs, allowing the complete system to be exported and installed easily on other devices.

One of the most important aspects of the project is the optimization of the operating system. Decuria focuses on reducing unnecessary resource consumption through system management, removal of bloatware and optimization of background services. In addition, several integrated tools were developed so users can manage and adapt system performance according to their own needs.

The main objective of Decuria is to provide a lightweight, modern and optimized experience without significantly sacrificing the graphical aspect of the system. The distribution is especially oriented toward computers with limited hardware specifications, giving older devices a second life through an efficient and functional Linux-based environment.

Keywords (entre 4 i 8):

Decuria, Linux, Distribution, Performance, Operating system

Índice

Lista de figuras

1 Introducción

Este proyecto consiste en la creación de una nueva distribución de Linux basada en Ubuntu Server como ya se ha mencionado antes, la gracia de este sería una distribución con un entorno gráfico pero que consuma recursos tan menores como las de un Ubuntu Server, añadiendo que este contará con un diseño base moderno (barra de tareas, fondo de pantalla, icono de arranque, animaciones, menu, programas) sin que se note demasiado las limitaciones del ordenador, ya que esta distribución va dirigida a aquellos ordenadores con unas especificaciones limitadas en cuanto RAM y procesador, entre otros.

Resumidamente, para lograr crear la distribución a nuestro gusto, tendremos que partir de un Ubuntu Server en donde le instalaremos todas las herramientas necesarias para la creación. Unos ejemplos serían Openbox como gestor de ventanas, Tint2 como barra de tareas, Lightdm como UI de inicio de sesión y muchos más que serán detallados a lo largo de esta documentación. Realmente este será el apartado en donde más deberemos trabajar ya que aparte de hacerlo visual tendremos que hacerlo funcional mediante archivos de configuración, en su mayoría porque hay que recordar que aunque el apartado visual sea importante, en realidad se busca que sea una distribución de lo más optimizado posible, ya que queremos incluso que con poca RAM puedan utilizarse la mayor totalidad de programas básicos y unos cuantos más algo más complejos.

Para crear el tema y los iconos personalizados se hará uso de programas/páginas de edición como GIMP o Canva para luego poder aplicarlos como tema por defecto de la distribución. Para la creación de la ISO simplemente utilizaremos dos máquinas virtuales: uno en donde trabajaremos y el otro donde acabará el producto final para acabar convirtiéndose la ISO.

Para optimizar el sistema, aparte de utilizar herramientas ligeras para la creación del entorno visual, se hará uso de los scripts para crear un menú en donde el usuario pueda gestionar el rendimiento, cara a los aspectos

que el usuario no puede gestionar respecto al rendimiento haremos modificaciones al Kernel.

1.1 Contexto

Actualmente, el ecosistema GNU/Linux ofrece una gran variedad de distribuciones, cada una enfocada a perfiles de usuario y necesidades muy distintas. Existen opciones muy completas y visualmente atractivas, pero que suelen requerir una cantidad considerable de recursos, y otras muy ligeras que sacrifican casi por completo el apartado gráfico y la experiencia del usuario. [1]

En este contexto, consideramos que existe un espacio claro para una distribución que combine eficiencia, ligereza y diseño moderno, especialmente pensada para ordenadores antiguos o con especificaciones modestas. Muchos de estos equipos siguen siendo funcionales, pero quedan relegados porque no pueden ejecutar sistemas actuales de forma fluida.

DECURIA nace como respuesta a esta situación, aprovechando la estabilidad y compatibilidad de Ubuntu Server como base, pero construyendo un entorno gráfico completamente personalizado y optimizado.

1.2 Justificación

Son varios los motivos para la ejecución de este proyecto, al existir tantas distribuciones desde los populares, como Ubuntu, hasta de los más olvidados como Sabily (Ubuntu Islam Edition) [2] normalmente creados por aficionados. Por absurdo que suene esto, da a entender lo accesible que es crear una distribución propia que no quiere decir fácil si se quiere crear algo decente para el uso público. Por otro lado la mayoría de distribuciones populares suelen requerir recursos significativos por el apartado gráfico de

estos y es por ello que hemos decidido crear una versión propia, ligera pero tratando de mantener un apartado gráfico que se vea actual/moderno para revivir ordenadores a los que ya no se les daba uso. Esto viene de tener en nuestros propios hogares ordenadores los cuales se les podría considerar prácticamente obsoletos y no mucho más importante queríamos diferenciarnos de la mayoría de los proyectos los cuales suelen tratar sobre la creación de algún tipo de servidor.

1.3 Objetivos

El objetivo principal del proyecto no se limita únicamente a generar una imagen ISO funcional, sino también a **crear un producto coherente**, documentado y preparado para ser distribuido.

1.3.1 Objetivo general

El objetivo general del proyecto es desarrollar una distribución GNU/Linux completamente funcional capaz de generar una imagen ISO estable, optimizada y preparada para su utilización en equipos con recursos limitados. Esta distribución debe permitir realizar una instalación sencilla y lograr un funcionamiento fluido desde el primer inicio, garantizando que cualquier dispositivo antiguo pueda desplegar el sistema sin penalizaciones de rendimiento.

1.3.2 Objetivos específicos

El proyecto no solo pretende crear un sistema operativo ligero, sino construir una plataforma tecnológica coherente, esmerada y capaz de competir en calidad de experiencia con distribuciones modernas, pero adaptada específicamente a hardware de baja potencia. Esto implica asegurar que:

- El sistema arranque rápidamente y sin procesos innecesarios.
- La interfaz sea intuitiva y agradable visualmente.
- Todas las herramientas esenciales funcionen de forma estable.

- La carga del sistema se mantenga siempre baja, y que la gestión del hardware antiguo sea correcta y eficiente.
- Herramientas propias para la gestión de rendimiento.

1.4 Estrategia y planificación del proyecto

La estrategia elegida ha sido adaptar un producto existente, utilizando Ubuntu Server como base sólida, en vez de desarrollar un sistema desde cero. Consideramos que esta decisión es la más adecuada por motivos de viabilidad, estabilidad y compatibilidad.

La planificación del proyecto se ha dividido en cinco fases claras:

1. Definición de la identidad visual del sistema (colores, tipografía, iconos, estilo general).
2. Selección de herramientas ligeras para el entorno gráfico.
3. Pruebas de rendimiento y estabilidad en máquinas virtuales.
4. Personalización del sistema mediante archivos de configuración y scripts.
5. Generación de la ISO final y validación del resultado.

1.5 Metodología del trabajo

Especificar qué tipo de metodología de trabajo se seguirá a lo largo del desarrollo del proyecto (metodología tradicional estilo “waterfall”, como por ejemplo Métrica 3.0; o metodología ágil, como por ejemplo Scrum). Argumentar por qué se considera la metodología más apropiada para el proyecto e indicar qué herramientas se utilizarán para su seguimiento (diagramas de Gantt, aplicaciones como Trello o Taiga.io, etc.).

Para el desarrollo de DECURIA se ha seguido una metodología de tipo ágil, el trabajo se ha organizado en iteraciones cortas, donde cada cambio se probaba y validaba antes de integrarse definitivamente.

Este enfoque es especialmente adecuado en un proyecto de sistemas operativos, en el que es habitual encontrar incompatibilidades, errores inesperados o limitaciones técnicas que obligan a replantear decisiones.

Como herramientas de soporte se han utilizado:

- Máquinas virtuales para pruebas y validación.
- Documentación técnica oficial.
- Scripts para automatizar tareas.
- Listados de tareas y control de progreso manual.

1.6 Estudio económico y presupuestario

El proyecto DECURIA se ha desarrollado principalmente utilizando software libre y herramientas gratuitas, reduciendo significativamente el coste económico directo. Sin embargo, es necesario realizar una estimación orientativa de los recursos implicados.

Recursos utilizados:

-Hardware: ordenador personal ya disponible (coste 0€).

-Software base: Ubuntu Server, herramientas GNU/Linux (coste 0€).

-Herramientas de personalización: GIMP, Canva (versiones gratuitas).

-Virtualización: VirtualBox.

-Tiempo de desarrollo: principal coste del proyecto.

Coste estimado:

Si se valorara el proyecto en un entorno profesional, el mayor coste sería el **tiempo de desarrollo**, estimado en aproximadamente en 120–150 horas de trabajo técnico.

2 Descripción del proyecto

2.1 Análisis de requisitos [proyecto de desarrollo]

Para que el desarrollo de **DECURIA** pueda considerarse finalizado, se han definido una serie de requisitos funcionales, técnicos y de diseño que garantizan el cumplimiento de los objetivos del proyecto.

Requisitos funcionales

1. **Instalación sencilla y ligera**: El sistema debe poder instalarse fácilmente en equipos con recursos limitados, mediante un instalador gráfico o de texto simple.

2. **Arranque rápido**: El tiempo de inicio del sistema operativo debe ser reducido, incluso en dispositivos con hardware antiguo.

3. **Entorno de escritorio moderno**: DECURIA debe ofrecer una interfaz visual atractiva, intuitiva y actual, adaptada al rendimiento de equipos modestos.

4. **Gestión de software**: Debe incluir un gestor de paquetes accesible y compatible con repositorios Linux estándar.

5. **Compatibilidad básica de hardware**: El sistema debe reconocer de forma automática los componentes más comunes (tarjeta de red, sonido, vídeo, USB, etc.).

6. **Accesibilidad**: La distribución debe ofrecer soporte para varios idiomas, incluyendo opciones de accesibilidad para usuarios con necesidades especiales.

Requisitos técnicos

Requisitos mínimos de hardware:

- CPU: 1 GHz o superior
- RAM: 1GB (recomendado 2 GB)
- Almacenamiento: mínimo 10GB libres
- Base del sistema: Basada en Debian/Ubuntu (por su estabilidad y compatibilidad).
- Entorno gráfico: Utilizar entornos ligeros como XFCE/LXQt
- Gestor de ventanas: Ligero, optimizado para reducir el uso de CPU y memoria.
- Consumo de recursos: El sistema en reposo no debe superar los 500 MB de RAM ni el 10% de CPU en equipos de prueba estándar.

2.1.1 Requisitos funcionales

Listado de requisitos funcionales (funcionalidades o casos de uso descritos de forma sencilla y unitaria).

Los requisitos que consideraremos funcionales serán estos:

- Que se pueda lograr apagar y encender el sistema.
- Lograr gestionar procesos con fluidez y comodidad.
- Gestionar archivos y carpetas, mover cantidades de archivos etc…
- Instalar y usar aplicaciones regulares.
- Gestionar el hardware (sonido, redes WIFI, etc…).

2.1.2 Requisitos no funcionales

Los requisitos no funcionales que se tendrán en cuenta son los siguientes:

- Buen rendimiento en equipos con pocos recursos:
  El sistema debe funcionar de manera ágil en ordenadores antiguos o con limitaciones de memoria RAM y procesador, permitiendo realizar tareas cotidianas sin ralentizaciones excesivas.
- Consumo reducido de recursos:
  Se busca que el uso de CPU, memoria y almacenamiento sea lo más bajo posible, evitando la ejecución de servicios innecesarios en segundo plano.
- Estabilidad del sistema:
  El sistema debe ser fiable y capaz de mantenerse en funcionamiento durante largos periodos sin fallos graves, bloqueos o comportamientos inesperados.
- Facilidad de uso:
  La interfaz y el manejo general deben resultar comprensibles e intuitivos, incluso para usuarios con poca experiencia en sistemas operativos distintos a los habituales.
- Arranque rápido:
  El tiempo necesario para iniciar el sistema debe ser reducido, especialmente en equipos con discos duros tradicionales o hardware antiguo.
- Compatibilidad con distintos tipos de hardware:
  El sistema debe reconocer y funcionar correctamente con periféricos y componentes comunes, incluyendo dispositivos antiguos que aún se encuentren en uso.
- Posibilidad de personalización:
  El usuario debe poder adaptar la apariencia y ciertos comportamientos del sistema según sus preferencias, sin comprometer su funcionamiento.

- Eficiencia energética:

En equipos portátiles, el sistema debe procurar un uso moderado de la batería, reduciendo el consumo cuando no se esté realizando una actividad exigente.

2.2 Previsión de tareas de investigación

1. Análisis inicial del problema y definición de requerimientos:

Lo que se quiere conseguir con el proyecto es una nueva distribución, la cual sea atractiva, cómoda y sostenible para que antiguos equipos informáticos puedan estar en pleno rendimiento usando aplicaciones/herramientas actuales.

Por lo tanto, el proyecto busca equilibrar tres pilares esenciales:

1. **Atractivo visual con identidad propia**

Crear un entorno gráfico coherente visualmente distintivo de otras distribuciones en la que se pueda sentir un producto enteramente moderno sin sobrecargarlo de apartados innecesarios.

2. **Comodidad y facilidad de uso**

Queremos poder brindar a todos los usuarios una experiencia sin complicaciones llevadera y básica de forma en la que todas las funciones sean muy intuitivas.

3. **Sostenibilidad y beneficios de maquinaria existente**

El objetivo primordial es conseguir en equipos antiguos, que muchos se consideran tirar a la basura, poder darles una segunda vida, alargar la vida útil de los equipos debido a la optimización del sistema operativo y el bajo consumo de recursos es algo que tenemos contemplado y queremos que sea un aspecto importante.

2. Componentes tecnológicos involucrados:

Estudio del funcionamiento interno del entorno base

NOMBRE	FUNCIÓN
Openbox	Gestor de ventanas
Tint2	Barra de tareas
Rofi	Lanzador de programas
Thunar	Gestor de archivos
Htop	Panel para inspeccionar memoria
Plank	Dock de apps
Xfce Desktop	Gestor de escritorio
Penguins egg’s	Ensamblador de ISO
Picom	Gestor de transparències
Xfce-settings-manager	Gestor de ajustes del sistema
Viewnior	Lector de pdf
Celluloid	Reproductor de video
PulseAudio System	Gestor de sonido
Mousepad	Bloc de notas

2.3 Tecnologías

2.3.1 Comparativa de las tecnologías valoradas

Comparativa de las diferentes tecnologías que se han tenido en cuenta durante el desarrollo o investigación, especificando las peculiaridades o pros y contras de cada una de ellas.

Las tecnologías que no aparecen en la tabla comparativa y sí en tecnologías escogidas son aquellas con las que no hemos tenido dudas de utilizarlas para el desarrollo de nuestro proyecto.

	Características	Características
Cubic vs Penguins egg's	Programa que sirve para generar una ISO desde un entorno de terminal chroot, se debe crear todo paso a paso.	Programa que sirve para generar una ISO desde el propio sistema, básicamente genera una copia del esqueleto del sistema.
Java vs Python	Java es un lenguaje de programación robusto, orientado a objetos y diseñado para ser multiplataforma	Python se caracteriza por ser un lenguaje interpretado, de alto nivel y con un enfoque total en la legibilidad del código.
IsardVDI vs VirtualBox	Es una plataforma que proporciona máquinas virtuales pre creadas con la posibilidad de trabajar vía internet a través de una página web.	Es un programa potente de virtualización de máquinas de manera local.

2.3.2 Tecnologías escogidas

Descripción de las tecnologías que finalmente se han escogido para estudiarlas en profundidad y/o utilizarlas durante la investigación, argumentando los motivos por los que se han escogido éstas y no otras.

- **Penguins egg's:** Es una herramienta que permite crear una versión personalizada de tu sistema operativo Linux en formato ISO normalmente, aunque posee otros métodos como la clonación del sistema. Este trae Calamares, una herramienta que consiste en ser un instalador gráfico de la ISO creada. También es compatible con las distribuciones más populares y sus derivadas como **Debian** (Ubuntu, Mint, Kali), **Arch** (Manjaro, EndeavourOS), **RPM** (Fedora, openSUSE, Rocky) y **Alpine Linux** además de que soporta las arquitecturas x86_64, i386 y arm64. [3]

El motivo para escoger esta herramienta es debido a su sencillez de uso, tan simple como decirle con un comando que escanee todo el sistema y la convierta en ISO, en cambio su competencia, Cubic ofrece un entorno chroot (una terminal básica) en donde se debe modificar el sistema sin poder ver a tiempo real los cambios realizados, cosa que con el programa escogido podemos controlar siempre lo que se copiara así pudiendo evitar bugs…

- **Kernel Linux:** Esta tecnología es un núcleo de sistema operativo de código abierto y esta es vital para tener control sobre los componentes del hardware y software que componen el sistema. Los núcleos ayudan a prevenir conflictos entre procesos importantes que son esenciales para el correcto funcionamiento del sistema. El código del Kernel se guarda en la memoria del ordenador y permite todas las interacciones entre el software y el hardware. [4]

El principal motivo para escoger esta tecnología es debido a que es de código abierto y a la vez la que permite mayores modificaciones que se adaptan perfectamente con nuestro objetivo y método de trabajo. Además

de la gran popularidad con la que cuenta Linux respecto a la modificación de sistema habiendo así muchos ejemplos de trabajos similares al nuestro, motivo que incita a querer desarrollar un sistema a través de este Kernel contando con un gran soporte de la comunidad.

- **VirtualBox:** Es una herramienta que permite la creación de máquinas virtuales, es decir, simular un ordenador ficticio en nuestro ordenador principal de manera aislada sin comprometer nada importante y dependiente de la potencia de nuestra máquina física. Este permite el uso de cualquier tipo de sistema operativo, siempre y cuando se disponga de su disco de instalación (ISO). También trae muchas otras características, pero la más destacable es que estas máquinas son portables en formato OVA.

El motivo para escoger esta herramienta por delante de Isard VDI es debido a todas las ventajas que ofrece respecto a este. Es más potente ya que utiliza la propia potencia de la máquina física como hemos mencionado con anterioridad, no como Isard, en el que dependes tanto de la potencia de otros servidores y la propia conexión de la red, además de que ambos permiten llevarse el trabajo a casa. Aunque en Isard sea más cómodo, de vez en cuando es posible perder el trabajo debido a mantenimientos de la misma.

- **Python:** Esta tecnología es un lenguaje de programación de alto nivel interpretado y multiparadigma, conocido por su sintaxis limpia y legibilidad, lo que lo hace ideal para principiantes. [5]

El motivo para escoger este lenguaje y no Java es debido a que aún siendo Java nuestra primera opción, debido a la experiencia básica que tenemos del curso anterior, en realidad para los pequeños programas que hemos creado no es nada adecuado. Esto se debe a que unas pocas líneas de Python pueden ser una gran cantidad de líneas en Java, lo que no favorece conociendo el tiempo limitado que tenemos y a la facilidad de aprender este lenguaje respecto a otros como Java que es de un nivel más bajo.

- **Css:** Esta tecnología es un lenguaje de hojas de estilo que por lo general vienen acompañadas de lenguaje de marcas para definir el aspecto y el formato visual de una página web. [6]

Más que escogerlo viene incluido con el programa Rofi, el cual es explicado en el apartado de componentes, pero básicamente Rofi para dar estilo a sus lanzadores usa el estilo css.

- **Bash:** Bash es un lenguaje de scripting y una interfaz de líneas de comandos (shell), se utiliza para interactuar directamente con el sistema operativo (por lo general con Linux o MacOS). Aparte de permitir crear scripts, Bash es la manera de administrar el propio sistema, como ejecutar programas o gestionar permisos y procesos.

Como indica la definición, más que una elección es una obligación al utilizar un sistema operativo derivado del Kernel de Linux.

2.4 Estructura del proyecto

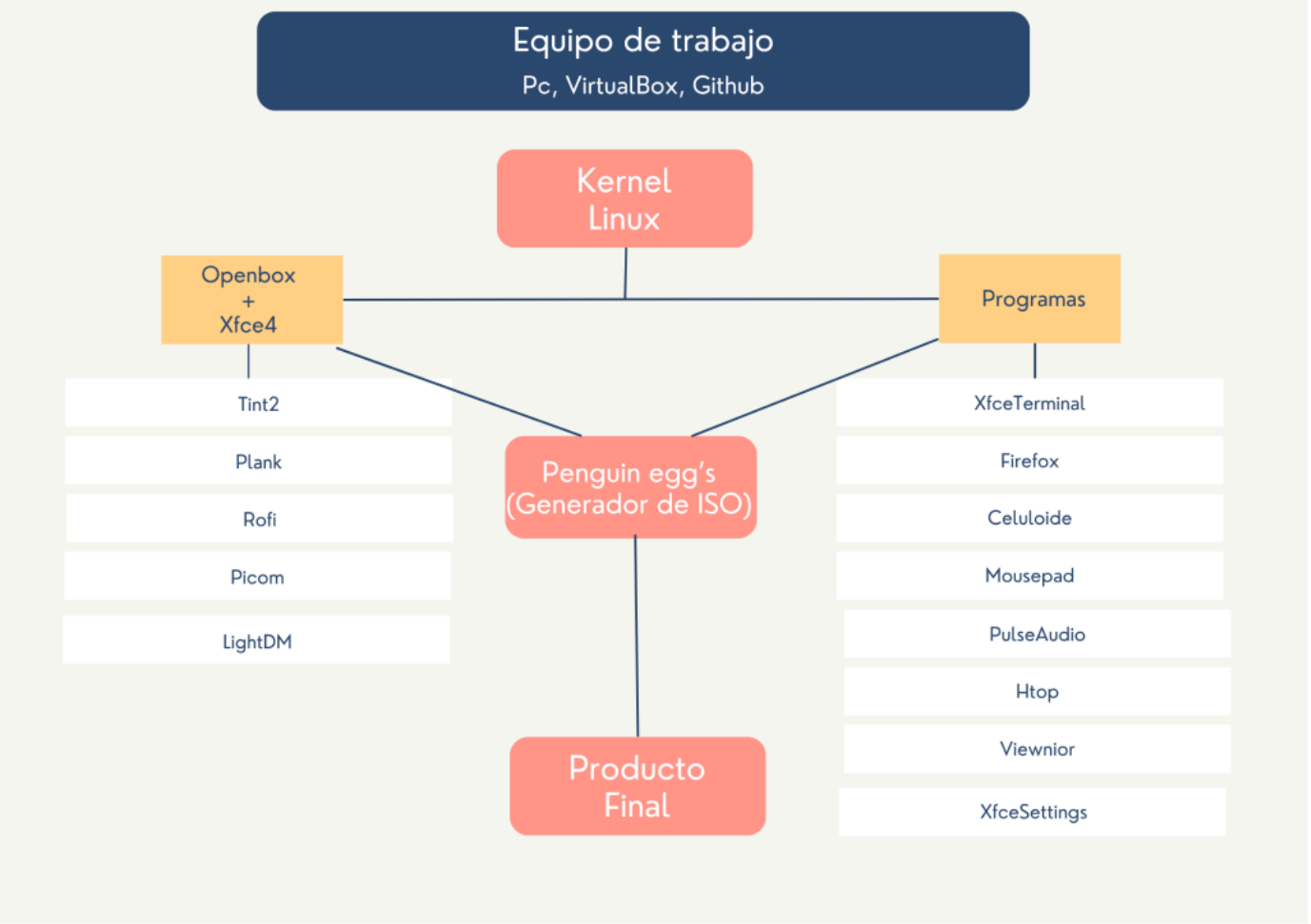


Figura 1. Arquitectura del sistema. Fuente: Propia

2.5 Descripción de los componentes

2.5.1 Componente 1

- **Thunar:** Es el administrador de archivos oficial y predeterminado del entorno de escritorio XFCE. Es conocido por ser extremadamente ligero, rápido y fácil de usar, diseñado específicamente para ofrecer una experiencia de usuario limpia sin opciones innecesarias.

El motivo para escoger esta herramienta es, a pesar de que es igual a cualquier otro gestor de archivos, este no trae ninguna función extra, por lo tanto podemos añadir nosotros mismos las extensiones adecuadas como la de comprimir/descomprimir. Otro punto a favor es la gran compatibilidad que tiene con xfce4 que nos beneficia al ser este nuestro gestor de escritorio.

2.5.2 Componente 2

- **Openbox:** Es un gestor de ventanas altamente minimalista pero personalizable, a diferencia de un entorno de escritorio completo como GNOME o KDE, Openbox unicamente se encarga de manejar las ventanas de manera que tendrias un entorno grafico negro sin nada pero con un menú accesible a través del cursor en donde dispondremos de todos los programas que se encuentren en el sistema como la terminal o un navegador. [7]

El motivo para escoger esta herramienta es debido a que ofrece un entorno gráfico para poder gestionar ventanas pero es exageradamente ligero al no disponer de ninguna función más. Por lo tanto, es perfecto para poder añadir nosotros mismos las herramientas necesarias y también para poder utilizarlo de molde para luego añadirle un gestor de escritorio. En cambio, otras herramientas que ofrecen un entorno de escritorio traen consigo todas sus dependencias de manera que el sistema se haría pesado, cosa que va en contra de nuestra filosofía.

2.5.3 Componente 3

- **Rofi:** Es un lanzador de aplicaciones y cambiador de ventanas extremadamente versátil para Linux que empezó como un clon de run-or-raise (muy similar a d-menu). Ha evolucionado hasta convertirse en una herramienta capaz de lanzar cualquier tipo de lista interactiva como programas del sistema, WI-FI, batería y etc… [8]

Además permite estilizar estas ventanas con CSS.

El motivo para escoger esta herramienta es debido a que actualmente es el lanzador de ventanas más moderno, compatible y personalizable que hay dentro de Linux siendo ligero también.

2.5.4 Componente 4

- **Xfce4-Desktop:** Es el gestor de escritorio oficial del entorno de Xfce4 y es el que permite la posibilidad de tener una experiencia más cómoda para el usuario estándar. [9]

El motivo para escoger esta herramienta es debido a su ligereza y compatibilidad con otros programas del sistema como el Thunar mencionado anteriormente. Lo más importante es que es moderno visualmente pero no trae consigo las dependencias de Xfce, se puede instalar el gestor de escritorio por separado.

2.5.5 Componente 5

- **LightDM:** Es un gestor de inicio de sesión (display manager) ligero, rápido y agnóstico al escritorio. Es el encargado de mostrar la pantalla donde introduces tu usuario y contraseña al arrancar Linux. A diferencia de otros display manager, este no arrastra ninguna dependencia de otros entornos como el de GNOME. [10]

El motivo para escoger esta herramienta es debido a su ligereza, compatibilidad y su modernidad pero sobre todo por su alta personalización tanto en el fondo de inicio de sesión como el propio bloque en donde se

introducen los datos aparte de traer con funciones extra como la de apagar y reiniciar el equipo.

2.5.6 Componente 6

- **Picom:** Es un compositor de ventanas ligero evolucionado del compton, diseñado para añadir efectos visuales (transparencias y sombras) a gestores de ventanas que no los tienen por defecto, como **Openbox**, **i3wm** o **bspwm**. [11]

El motivo para escoger esta herramienta es debido a que es la única viable y compatible para ser utilizado junto a Openbox (nuestro gestor de ventanas utilizado).

2.5.7 Componente 7

- **Xfce4-Settings-Manager:** Es el centro de control unificado del entorno de escritorio **XFCE**. Es el equivalente al "Panel de Control" de Windows, diseñado para gestionar de forma centralizada todas las herramientas de configuración de tu sesión. A partir de este centro de control puedes cambiar la apariencia del sistema (temas) o configurar el hardware (perifericos).

El motivo para escoger esta herramienta es debido a su compatibilidad con el gestor de escritorio de xfce que utilizaremos para el sistema. Esto evitará cualquier problema entre programas, además de que es completo y funcional.

2.6 Definición de las tareas

2.6.1 Prueba 1

A partir de una ISO Ubuntu Server (que sería a lo que nos referimos como **Kernel Linux** en el esquema antes visto) se le crea un entorno gráfico con el componente **Openbox**, el cual se ha mencionado reiteradas veces que es un gestor de ventanas, así permitiendo tener ya un entorno vacío pero capaz de ejecutar ventanas. Para crear este entorno se debe instalar el competente y para ingresar dentro de este, es necesario complementarlo con Xinit, que permite ingresar en entornos.

Para que todos estos cambios se guarden al iniciar sesión y el usuario aparezca en el entorno gráfico iniciado, siempre hay que indicarle al sistema que **Openbox** inicie la sesión con Xinit automáticamente al encenderse, y esto se hace mediante el archivo de configuración del autostart.

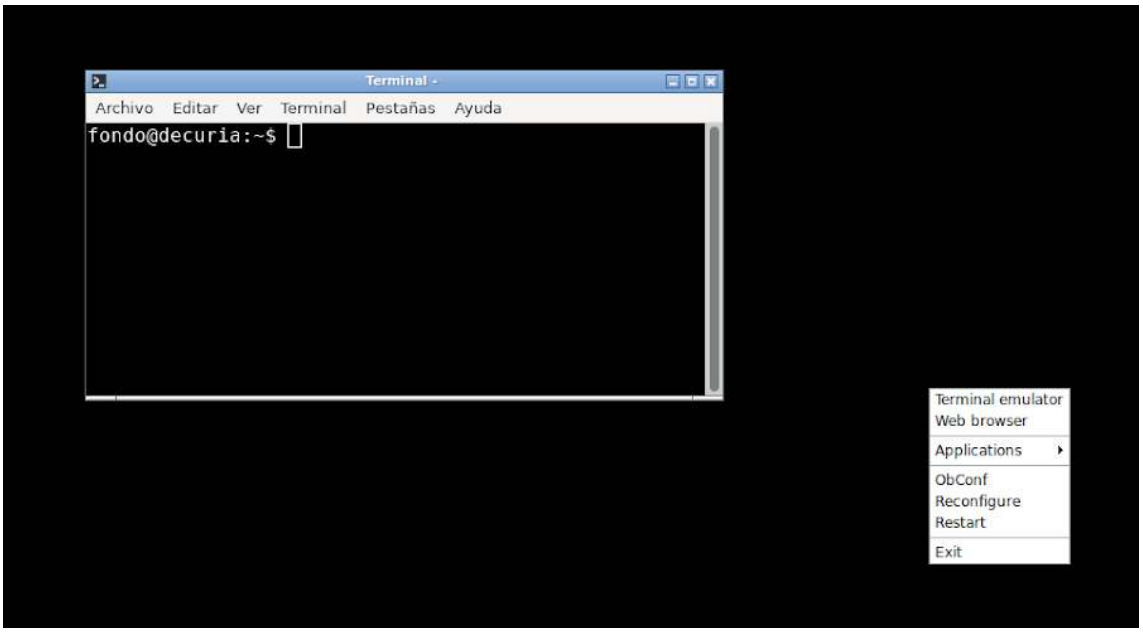


Figura 2.1. Openbox. Fuente: Propia

2.6.2 Prueba 2

Una vez configurado el entorno gráfico de Openbox, ahora debe tener lo esencial como entorno gráfico: una barra de tareas. Para ello se usan dos programas con distintas funcionalidades.

El primero es Plank y también el más sencillo el cual únicamente hay que instalarlo y ejecutarlo, este mostrará una barra de tareas estilo **docker** que muestra las ventanas abiertas en la parte inferior de la pantalla.

El segundo y el más importante es Tint2, una barra de tareas completa con la gran mayoría de funciones básicas (mostrar programas, ventanas abiertas, fecha y hora) y simples pero altamente modificable con su archivo de configuración que se encuentra en la ruta ~/.config/tint2/tint2rc donde se puede modificar absolutamente todo el comportamiento de la barra de tareas. En este caso hemos decidido darle las siguientes propiedades: botón de menú de programas en el lateral izquierdo y seguidamente lo acompaña un botón para abrir el menú de ajustes del sistema. En el lado derecho de la barra se encuentran seguidos los botones del internet, menú de apagado, controlador de volumen/brillo y panel propio de rendimiento del cual se especificará más tarde. Y en medio de la barra se encuentra la fecha junto a la hora.

También se han hecho otras modificaciones en las funciones de esta como impedir que las ventanas se sobrepongan a la barra y que la barra se encuentre de forma horizontal en la parte superior de la pantalla. Aparte de otras configuraciones visuales como la transparencia, iconos y color.

Figura 2.2. Barra de tareas Tint2. Fuente: Propia

Figura 2.3. Barra de tareas Plank. Fuente: Propia

En el apartado de anexos se encuentra el enlace al repositorio de GitHub en donde se puede ver el archivo de configuración.

2.6.3 Prueba 3

Una barra de tareas no es nada sin los programas que esta puede alojar. La idea de la barra de tareas de Tint2 es que no sea modificada, de manera que ofrezca las herramientas ya mencionadas en concreto en el apartado anterior, ya que las herramientas personalizadas al usuario se pueden mantener con el plank. Esto se puede apreciar en la ***figura 2.2*** *(pág. 21).*

La mayoría de herramientas de Tint2 están hechos mediante **rofi** empezando por el menú de aplicaciones el cual como su nombre indica es un panel desplegable con el click o con el botón super/Windows, que muestra todos los programas instalados del sistema que cuenta con un buscador simple. Para la formación del menú se debe crear un archivo de configuración propio con el nombre a nuestra elección pero con la extensión .rasi para indicar que la herramienta rofi debe actuar sobre este, en este caso su archivo de configuración sería ~.config/rofi/real.rasi

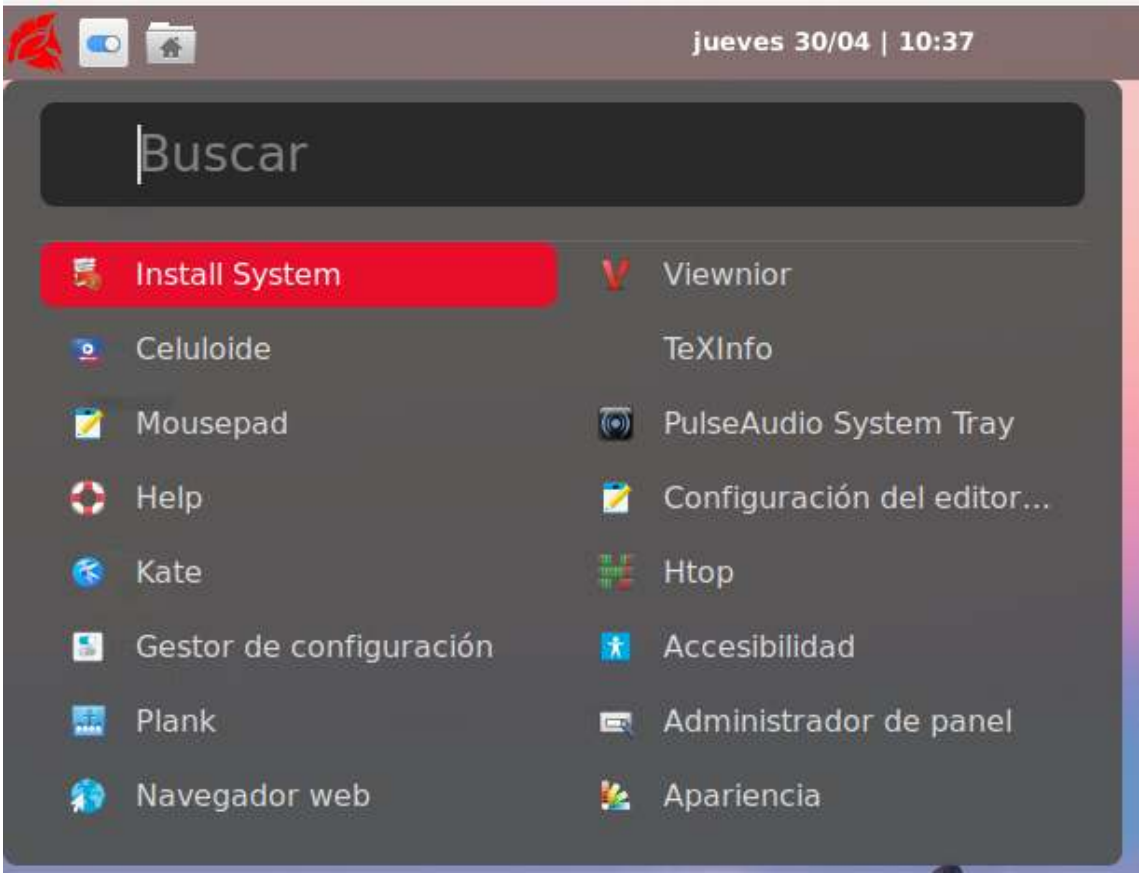


Figura 2.4. Menú de programas Rofi. Fuente Propia

El segundo programa que se encuentra en la barra de tareas es uno de los pocos que no está diseñado con rofi y se trata de xfce4-settings-manager. Cumple con la función de gestionar los parámetros generales del sistema como cambiar el aspecto del sistema, ajustes para los periféricos, etc…

El tercer programa es otro de los que no está diseñado a partir de rofi y sería el thunar, el gestor de archivos de Decuria.

A continuación se presentarán todos los programas diseñados a partir de rofi:

Un panel de rendimiento propio para poder gestionar opciones variadas para que el usuario pueda gestionar como su sistema se comporta según sus necesidades, un panel para gestionar y conectarse a internet/wifi, otro para la batería, otro para gestionar el sonido y el último para mostrar un menú con opciones de apagado. Se darán más detalles de estas funciones más adelante de la memoria.

La manera de incorporar estas herramientas en la barra de tareas con su respectivo icono para que te muestren lo deseado es tan simple como editar el archivo de configuración principal de Tint2 ~.config/tint2/tint2rc y crear esos botones en donde se le indicará la ruta del programa con el que se quiera enlazar.

2.6.4 Prueba 4

Los programas por sí solos son adecuados pero en realidad requieren una mínima configuración para que realmente sean compatibles con el sistema o básicamente cuenten con lo esencial, además de incluir todos los programas vitales que deben venir por defecto.

Para empezar Thunar, el gestor de archivos cuenta con todo lo esencial, pero en este caso no contaba con la opción de comprimir y extraer archivos. Esta función se ha incluido con una extensión llamada File-Roller la cual está pensada para ser un complemento de Thunar, de manera que no dará fallos. La manera de modificar el comportamiento de Thunar es tan simple como navegar dentro de los ajustes del programa para indicar a qué extensiones debe de hacerles caso.

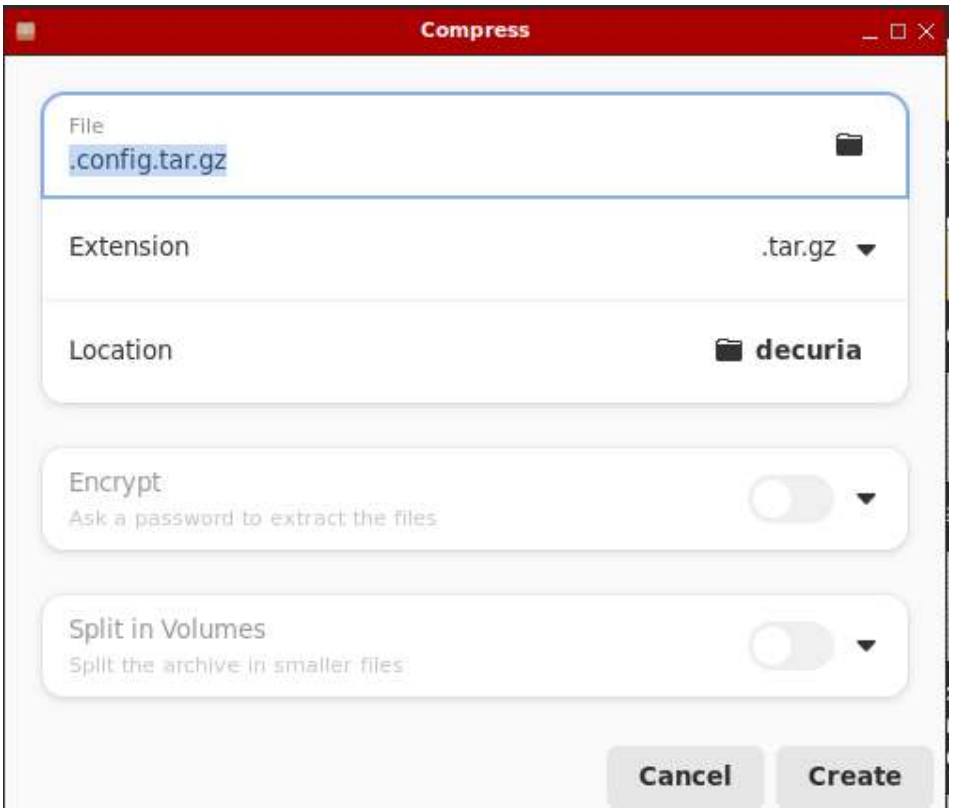


Figura 2.5. Compatibilidad Thunar. Fuente: Propia

Aunque no se haya mencionado antes dentro de este apartado, el instalador de Calamares, que en este caso viene en conjunto con el creador de ISO's Penguin egg's, es altamente sensible y si no se configura correctamente puede generar inconvenientes a más de una persona no permitiendo la instalación del sistema. En este caso se han hecho dos modificaciones principales para evitar la gran mayoría de problemas de compatibilidad con los equipos:

1. La primera modificación consiste en instalar todas las dependencias de calamares que vienen de las librerías de KDE (KDE es un entorno de escritorio) pero no basta con simplemente arrastrar todo KDE, ya que iría en contra de la filosofía del proyecto al aumentar tanto el consumo del sistema, por ende hay que seleccionar milimétricamente la librería encargada de darle soporte a Calamares. Esta librería es importante para que el programa pueda detectar y leer el disco del equipo dentro del instalador.

2. La segunda modificación viene a raíz de pruebas fallidas de instalación debido a un error con el CD/ROM. Para ser breve basta con entender que el programa trataba de instalar el sistema en una ruta no existente /media/CDROM, así que para evitar conflictos la mejor solución ha sido descomentar todas las líneas que referencian a CD/ROM en el archivo de configuración de Calamares:

~/etc/calamares/modules/before_bootloader_context.conf

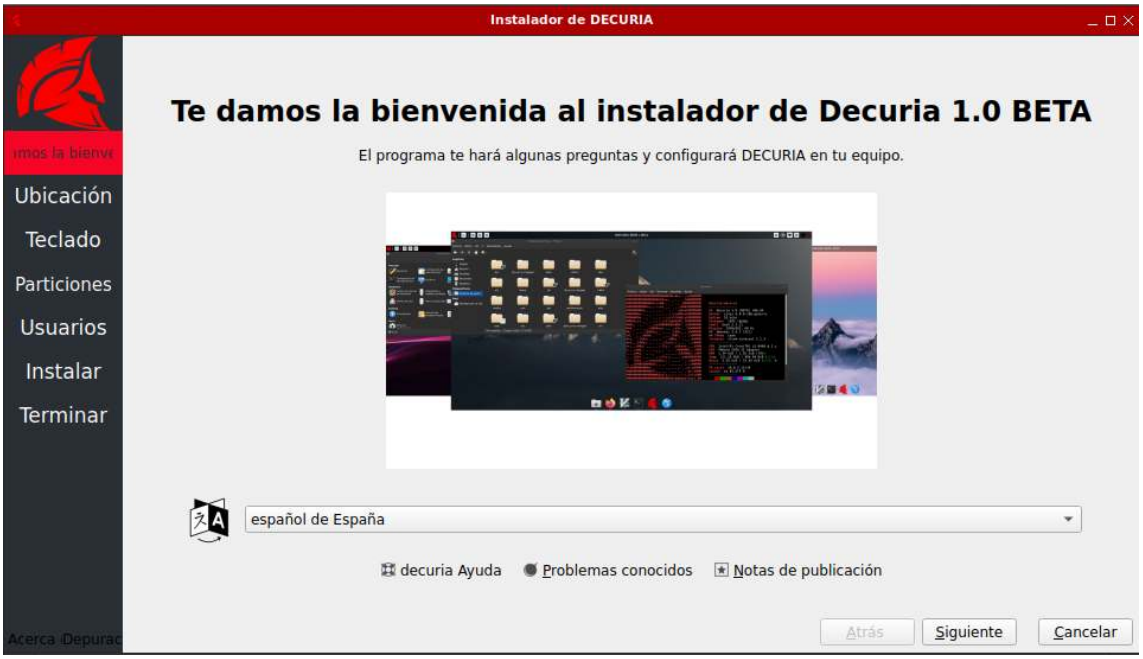


Figura 2.6. Instalador Calamares. Fuente Propia.

Otro programa seria el controlador de volumen nativo del sistema, en este caso más que tocar el propio programa ha sido necesario activar la capacidad de reproducir volumen en el propio sistema, capacidad que no venia por defecto debido a que Decuria parte de Ubuntu Server, el cual no cuenta con esa característica pero es tan simple como activar el volumen mediante este comando: sudo vgchange -ay para luego configurarlo según la lista de volúmenes que aparezcan.

Por último, las configuraciones necesarias para Openbox que en realidad són ajustes más de comodidad para el usuario que de solucionar problemas de compatibilidad, aún así este apartado es el más adecuado para tratar este tema. Los 2 cambios más grandes que han sido necesarios

son sincronizar el teclado con el software creando así ciertos atajos que otros sistemas cuentan con estos como el ya mencionado botón Windows o super para mostrar el menú de programas o el Ctrl + Alt + t para abrir la terminal o que simplemente los botones de los portátiles para controlar el brillo y el volumen funcionen como es debido. Todo esto se puede configurar en el archivo: ~.config/openbox/rc.xml. Y el otro tema es el diseño de las ventanas, ya que por defecto Openbox trae un diseño muy simple e incómodo debido a su tamaño, pero esto se puede solucionar cambiando el tema de la ventana por otro a través del selector de temas de Openbox.

Figura 2.7. Configuración de ventanas Openbox. Fuente: Propia.

2.6.5 Prueba 5

Ya contando con un entorno gráfico y una barra de tareas con sus respectivos programas, es hora de añadir un escritorio y un menú de inicio de sesión para darle forma al sistema.

Como se ha mencionado antes, el entorno de escritorio escogido es el de xfce4 el cual se instala con el comando: apt install xfdesktop4. Para ponerlo en marcha de forma permanente hay que indicarle al sistema que lo ejecute siempre al iniciar sesión; esto se logra añadiendo xfdesktop4 & en la siguiente ruta de configuración: ~.config/openbox/autostart.

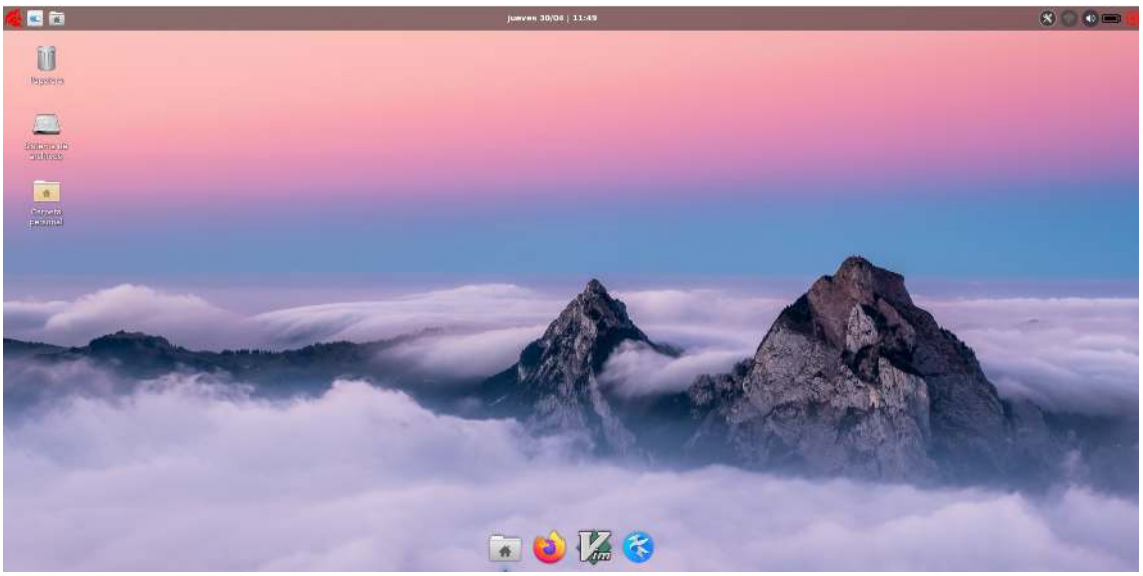


Figura 2.8. Escritorio Decuria. Fuente: Propia.

Para el menú de inicio de sesión en distros de Linux por excelencia se utiliza LightDM, así que en este caso se ha optado por una de sus variantes más ligeras que es lightdm-gtk-greeter. Este programa en realidad funciona como un servicio, por lo tanto con un systemctl status siempre se podrá ver el estado de este. Con su archivo de configuración se le puede dar el aspecto visual deseado: /etc/lightdm/lightdm.conf.

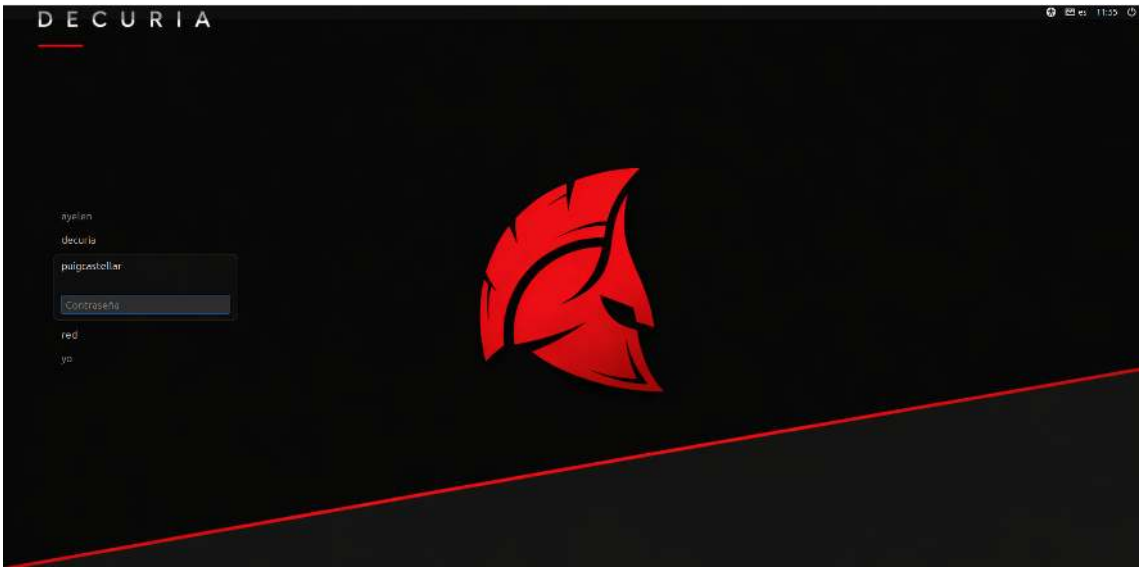


Figura 2.9. Inicio de sesión LightDM. Fuente: Propia.

2.6.6 Prueba 6

Teniendo un sistema parcialmente completo, es hora de modificar el sistema por dentro para incrementar ese rendimiento y reducir el consumo de la memoria RAM.

Lo primero que se ha hecho es inspeccionar el peso del sistema para tener claro todo lo que este lleva. El resultado en este caso fue una gran cantidad de caché, servicios de Ubuntu que Decuria no utiliza, servicios en general innecesarios, gestores de paquetes sobrantes, Kernels expirados y más.

Para hacer una buena optimización lo primero que ha desaparecido del sistema son esos servicios inútiles los cuales solo se utilizarían en un server como todos los complementos de Canonical para reportar mensajes y logs del servidor y junto a estos servicios normales como el de Cups que se encarga de las impresoras o el bluetooth, básicamente servicios que siempre están en segundo plano consumiendo más de la cuenta y son totalmente inútiles si no se utilizan. Esto no es un problema ya que en caso de que el usuario final necesite algún servicio de estos podrá activarlo fácilmente con el controlador de servicios que Decuria trae nativamente.

Tras esos cambios es turno de hacer el cambio real: eliminar el Snap. Es un gestor de paquetes antiguo y muy pesado que únicamente se encuentra vigente a día de hoy debido a que ciertos programas solo son instalables mediante el Snap en principio. Pero, teniendo el gestor de repositorios de APT no vale la pena mantener el Snap de manera que eliminandolo del sistema Decuria se ahorra más de 2Gb de almacenamiento y una considerable cantidad de memoria. El sustituto para los programas de Snap como el propio Firefox sería utilizar Flatpak, otro gestor de paquetes universal que es totalmente más ligero que Snap y lo mejor es que trae consigo una variedad mucho más amplia de programas que Snap como Spotify, Telegram, Chrome, etc.

Por como se ha explicado este asunto parecería que Decuria utiliza el Firefox de Flatpak, pero la realidad es otra, ya que en este caso se ha optado por utilizar la versión original de este navegador instalandolo desde los propios repositorios de Mozilla (creador de Firefox) y incrustando este mismo repositorio al sistema para que se actualice cada vez junto al sistema de manera que siempre el usuario tendría la versión más nueva del navegador y un mejor rendimiento debido a que no depende de ningún gestor como Flatpak o Snap. Este método es más pesado en cuanto a almacenamiento se refiere, pero vale totalmente la pena por la diferencia de rendimiento que ofrece y, siendo realistas, es muy importante ofrecer un navegador veloz y optimizado ya que la mayoría de equipos antiguos notan los problemas de rendimiento normalmente en las webs.

Se ha mencionado que se insertan repositorios como el de Mozilla pero más importante aún: ¿Cuáles son los que se eliminan? Pues todos los repositorios antiguos de Ubuntu para añadir los nuevos pero sin dejar nada relacionado al Server ni a Ubuntu Pro ya que es una versión de pago para recibir actualizaciones más pronto junto a parches de seguridad, aspecto que no interesa nada a Decuria hasta la fecha.

2.6.7 Prueba 7

Un paso antes de culminar con la creación de esta distribución sería ponerle un tema personalizado al arrancar el sistema con el logo de Decuria. Para hacer esto realidad hay que modificar el archivo de configuración de **Plymouth** el cual se encarga de gestionar la imagen de arranque mostrada por el sistema. Para este caso en concreto se ha usado un molde genérico que simula ser la barra de carga de MacOs, pero añadiendo el logo de Decuria por encima.

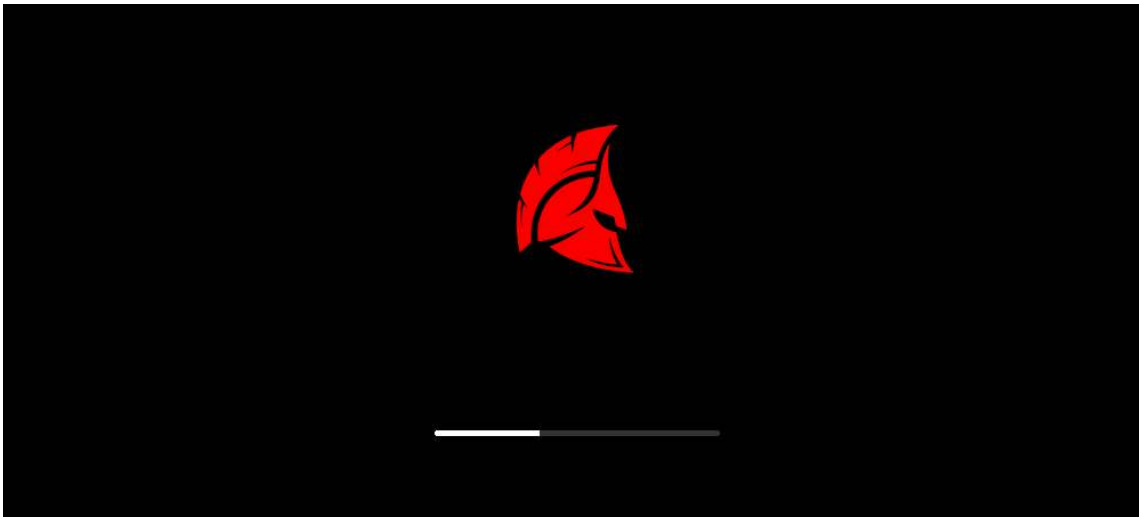


Figura 2.10. Arranque Plymouth Decuria. Fuente: Propia.

2.6.8 Prueba 8

Gracias a la combinación del entorno gráfico ligero y las modificaciones de optimización hechas, Decuria prácticamente tiene solucionado el tema del consumo de memoria excesivo, por lo tanto ya sería hora de implementar todos los programas vitales del sistema mencionados en el apartado de componentes junto a los ya mencionados antes como el Thunar. Hay que tener en cuenta que estos programas deben de ser compatibles con xfce y Openbox para que ofrezcan la experiencia debida; en caso de no ser así tocará modificar el programa o buscar alternativas de la misma.

Otras buenas prácticas són limpiar el historial del sistema, de la terminal y del navegador para ofrecer un entorno limpio. Junto a esto para mejorar la experiencia del usuario Decuria muestra un cartel de bienvenida para explicar brevemente qué es lo que este sistema ofrece. La explicación de la creación del cartel se encuentra más abajo en el apartado de funcionalidades.

Para terminar de crear la distribución asegurando que esta cuenta con todos los pasos mencionados en la sección junto a las funciones propias del apartado siguiente, ya se podría comprimir este sistema en formato ISO con el programa de Penguin egg's que previamente se había configurado para que sea lo más compatible posible. El funcionamiento de este creador de ISO's es muy simple y al ejecutarlo se dispone a crear una copia del sistema pero del /skel de manera que no copia los archivos del usuario actual.

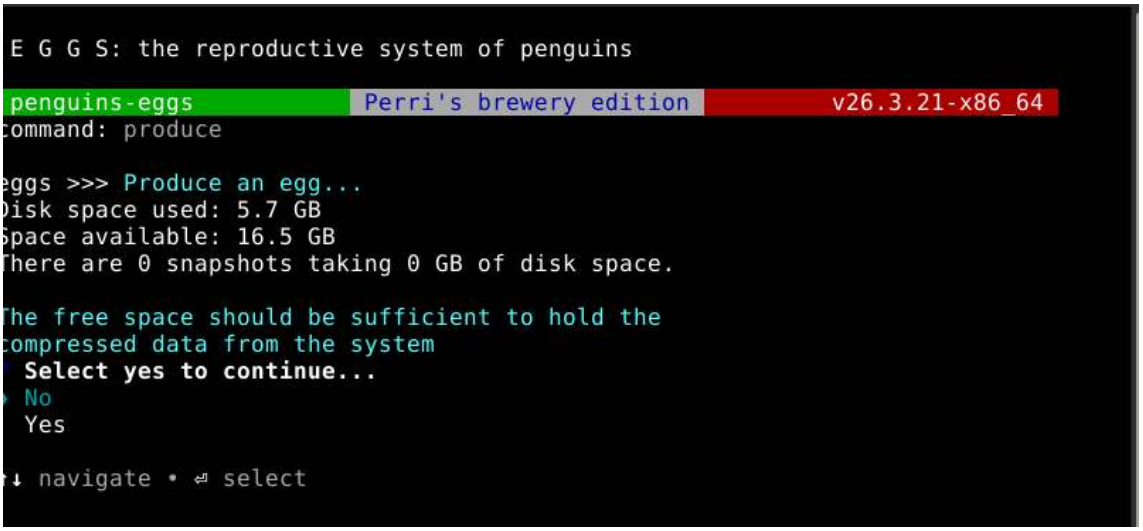


Figura 2.11. Crear ISO Penguin eggs. Fuente: Propia.

2.7 Definición de las funcionalidades

Descripció conceptual i detallada de totes les funcionalitats (casos d'ús) que ofereix la solució que s'està desenvolupant (si els requisits definits són el "què", aquí s'ha de detallar el "com"). Ha de quedar clar què permet fer, quin procés (conceptualment, el detall tècnic de la implementació es pot afegir als annexos com a enllaç al fitxer que conté el codi font) es seguirà per a fer-ho i, finalment, si aquesta funcionalitat ha quedat implementada totalment, parcialment o si es reserva per a una ampliació futura.

2.7.1 Funcionalidad 1

Como se ha mencionado con anterioridad, este panel de rendimiento único está diseñado para poder gestionar opciones variadas para que el usuario pueda gestionar como su sistema se comporta según sus necesidades.

Las opciones que proporciona són: monitorizar el rendimiento con htop y las especificaciones del hardware con Fastfetch, actualizar y limpiar el sistema con un click en lugar de hacer un apt update y upgrade, gestionar la compresión de RAM mediante Zram, controlar el rendimiento de la CPU, ajustar el swappiness, ajustar la latencia de la red y alternar la transparencia con Picom.

Figura 3.1 Panel de rendimiento. Fuente: Propia.

Para la creación de todas estas funciones no solo basta con crear archivos .rasi ya que estos únicamente cuentan con la función de lanzar carteles y mostrar algo pero no de realizar una función en concreto. Para ello se fusionan estos .rasi con scripts en Bash propios de manera que el script hace la función y el .rasi se encarga del aspecto visual y mostrar al usuario la lista de opciones. Todo ello se enlaza señalando la ruta del script en el .rasi de rofi.

Cada uno de estos apartados satisface la principal razón de ser de Decuria, el rendimiento, específicamente:

Rendimiento y Hardware: Esta función muestra dos ventanas: una el rendimiento actual del sistema con el programa htop que muestra parámetros interesantes como el consumo de la memoria RAM y la CPU, y la otra las especificaciones del hardware, sobre todo gracias al programa de Fasttech.

Muy útil debido a que antes de querer optimizar un equipo primero hay que averiguar las condiciones reales en la que este se encuentra.

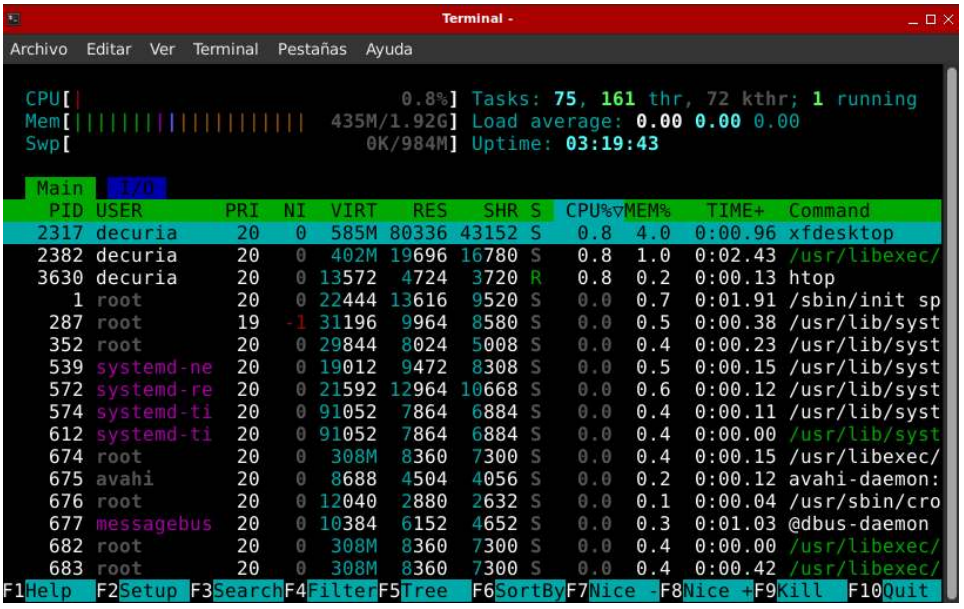


Figura 3.2. Htop. Fuente: Propia.

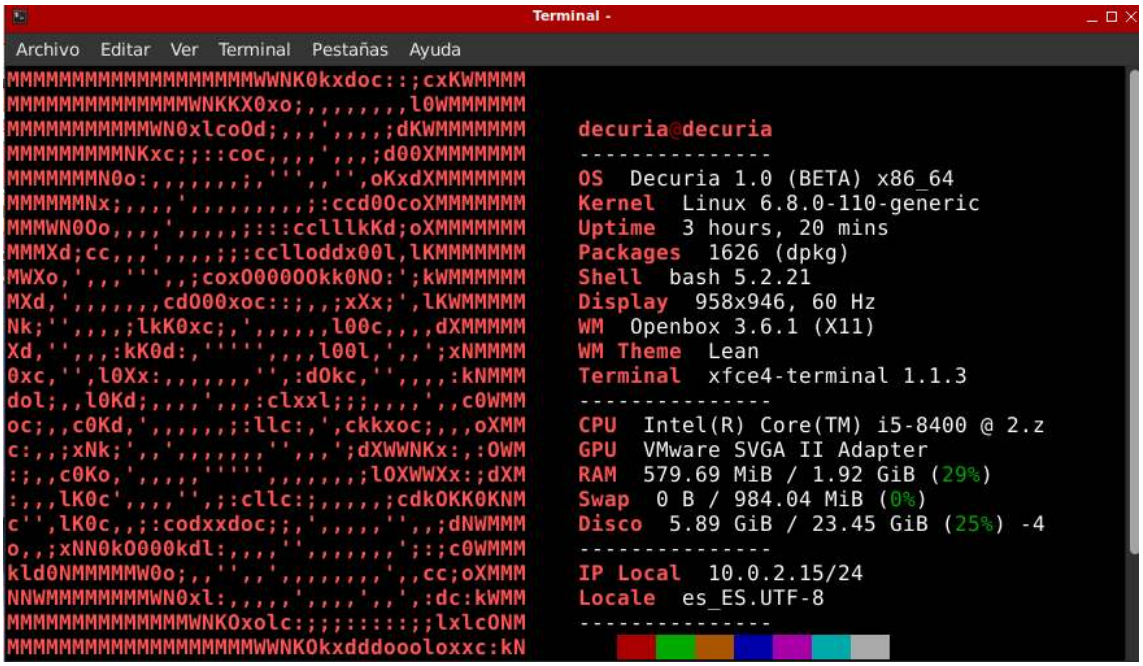


Figura 3.3 Fastfetch. Fuente: Propia.

Gestión del sistema: Una de las maneras más simples de resolver un problema de optimización y rendimiento es actualizar o limpiar el sistema y justo es lo que este apartado permite con un solo paso, poder decidir entre esas dos opciones sin tener que pasar a escribir comandos a través de la terminal.

Muy simple pero vital para comprobar que el problema de rendimiento no se deba a dependencias.

Figura 3.4 Gestión del sistema. Fuente: Propia.

Gestionar Zram: Zram es un programa dedicado a la compresión de RAM el cual te permite ajustar qué tanta memoria se comprimirá de tu memoria total para destinarlo a aumentar su capacidad efectiva. Decuria Zram ya viene con una configuración por defecto la cual sería utilizar el 25% de la capacidad de la memoria total para la compresión.

Esta solución es efectiva debido a que generalmente cuando un ordenador llega a su máxima capacidad de RAM se ralentiza debido a que empieza a guardar esa información sobrante en el disco. En cambio, con Zram se reduce la capacidad de memoria original para que cuando esta llegue a su totalidad de capacidad, en lugar de transportar la información sobrante al disco, la pasa al espacio de la memoria comprimida.

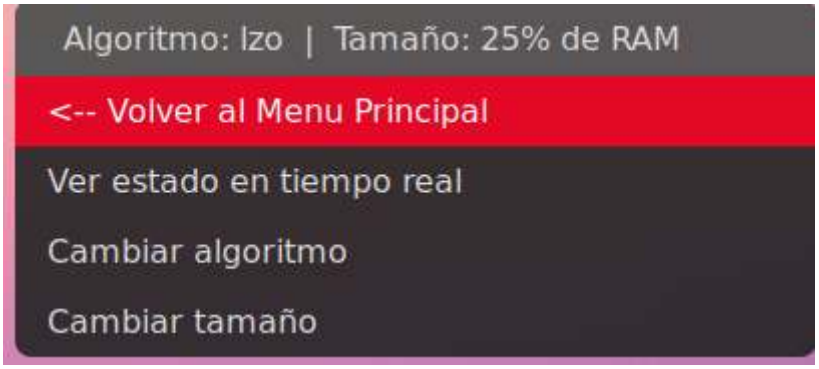


Figura 3.5. Zram. Fuente: Propia.

Cambiar Governor: Modificar el estado del Governor te permite regular la potencia de la CPU para darle el uso más adecuado según el contexto. Las tres opciones ofrecidas són el modo alto rendimiento, el cual utiliza la máxima capacidad de la CPU; un modo equilibrado, el cual sería darle un uso intermedio; y un modo ahorro el cual usaría el mínimo de la potencia del CPU en caso de querer ahorrar batería por ejemplo.

Es sumamente importante poder gestionar el rendimiento de la CPU ya que muchas veces el ordenador suele fallar por problemas de limitación y en otras ocasiones hay que limitar para no sobrecargar a la RAM, todo dependiendo del caso.

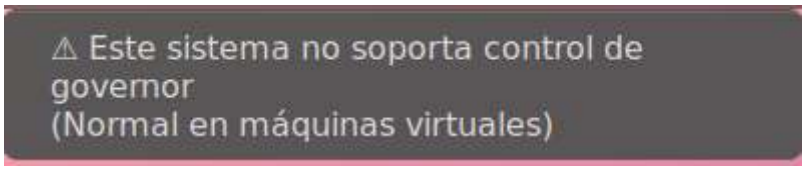


Figura 3.6. Governor. Fuente: Propia.

Ajustar Swappiness: El Swap se encuentra muy asociado con Zram. Esto es debido a que el concepto de Swap se refiere al uso del disco duro como una extensión de la memoria RAM, cosa que hemos mencionado anteriormente que ralentiza mucho todas nuestras tareas. Esta función en Linux ocurre de forma automática para que el sistema no colapse cuando se llega al límite de la capacidad de la memoria RAM a pesar de volver lento la máquina. Para ello, Decuria presenta un panel para ajustar que tanta swap quiere el usuario según sus necesidades.

Poder controlar este aspecto es importante por el mismo motivo que el del Governor, evitar limitaciones e imponer limitaciones a gusto del usuario. Por separado estas herramientas pierden sentido, pero entre Zram y ajustar el Swap pueden generar milagros reales.

Figura 3.7. Swappiness. Fuente: Propia

Controlar servicios: Dentro de Linux los servicios abundan y estos van desde cosas tan importantes como servicios que se encargan de gestionar la conexión a internet como NetworkManager a servicios como Cups que únicamente se encarga de las impresoras, funciones que probablemente el usuario promedio no utilice pero no se puede suponer lo que el usuario puede o no utilizar y eliminar así servicios quizás relevantes para el consumidor. Por ende, Decuria ofrece un panel para poder gestionar todos los servicios del sistema sin tener que pasar por los comandos tediosos de la terminal, así facilitando la gestión de estos para los usuarios menos experimentados. El panel ofrece la opción de ver, iniciar, detener, habilitar y deshabilitar los servicios junto a un buscador simple.

Gestionar los servicios es importante debido a que estos pueden llegar a consumir mucho más de lo esperado, todo dependiendo del servicio, pero siendo realistas no tiene ningún sentido tener servicios activos que no se usan. Gracias a este panel el usuario puede observar simultáneamente todos los servicios instalados del sistema y su estado.

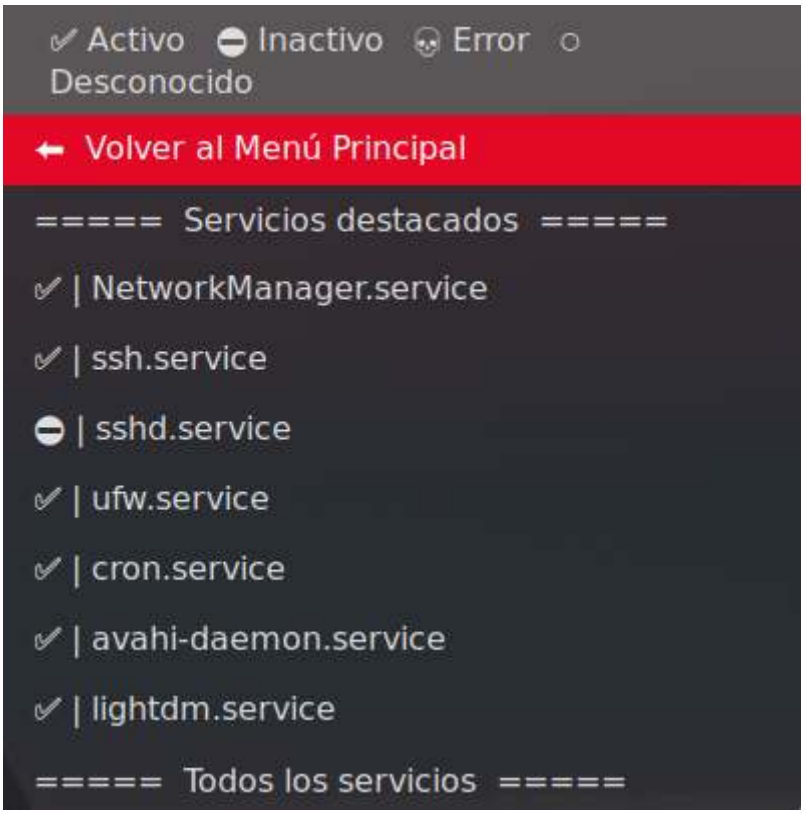


Figura 3.8. Servicios. Fuente: Propia.

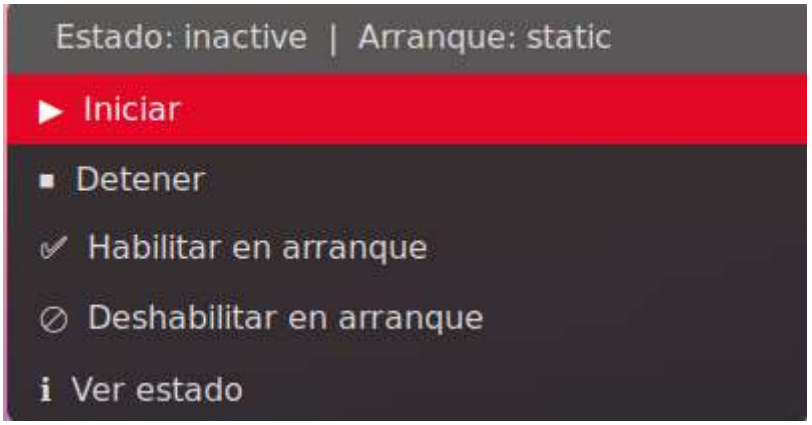


Figura 3.9. Estado de servicios. Fuente

Ajuste de latencia: Muchas veces únicamente se tiene en cuenta al CPU y a la RAM respecto al rendimiento, pero hay que recordar que en ocasiones la lentitud es causada por otros factores como la latencia en la red. Se puede entender como latencia el tiempo de retraso que un paquete de datos puede tardar en entregar al cliente.

Por ello, Decuria proporciona la posibilidad de mejorar la latencia mediante un panel único que permite controlar y monitorizar la latencia junto a los puertos, aunque cabe recalcar que realmente esta opción no mejorará la calidad del internet contratado.

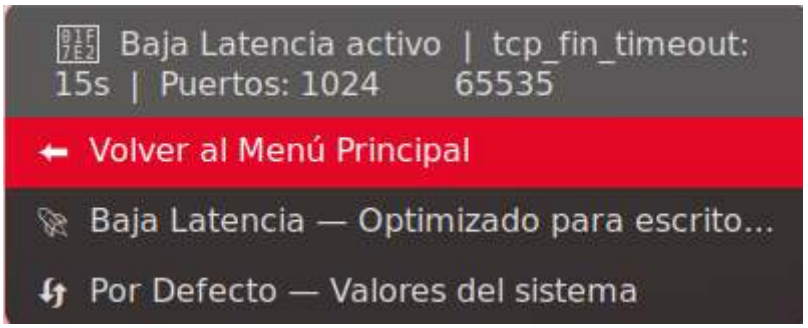


Figura 3.10. Ajustes de latencia. Fuente: Propia.

Alternar compositor: Esta es la última funcionalidad del panel de rendimiento. Trata de poder alternar el compositor que se encarga de la gestión de las transparencias en el sistema, básicamente el usuario puede escoger si usar transparencia o no, lo cual, dependiendo del equipo, puede ser un factor relevante para obtener el rendimiento deseado.

Figura 3.11. Compositor. Fuente: Propia.

Todo el panel de rendimiento se encuentra finalizado en cuanto a funcionalidad se refiere, ya que realmente siempre se pueden mejorar los aspectos técnicos a la hora de crear código y los aspectos visuales.

2.7.2 Funcionalidad 2

La segunda funcionalidad que presenta Decuria son todos los complementos necesarios de la barra de tareas ya mencionadas anteriormente como el panel de internet. Al igual que las funcionalidades del panel de rendimiento, estos apartados están desarrollados en scripts de Bash y fusionados los temas de rofi con los archivos .rasi, excepto el panel de volumen, que es un caso aislado.

Panel de Internet: Como el propio nombre indica, es un panel para poder observar y conectarte a internet vía cable ethernet o vía wifi. Este cuenta con todas las características generales de un gestor de internet como ver una lista de redes. Esto es gracias a que para encontrar las redes disponibles utiliza el programa nmcli, que sería un soporte de texto para el NetworkManager. Básicamente, el script en este caso complementa al programa, que hace la función real.

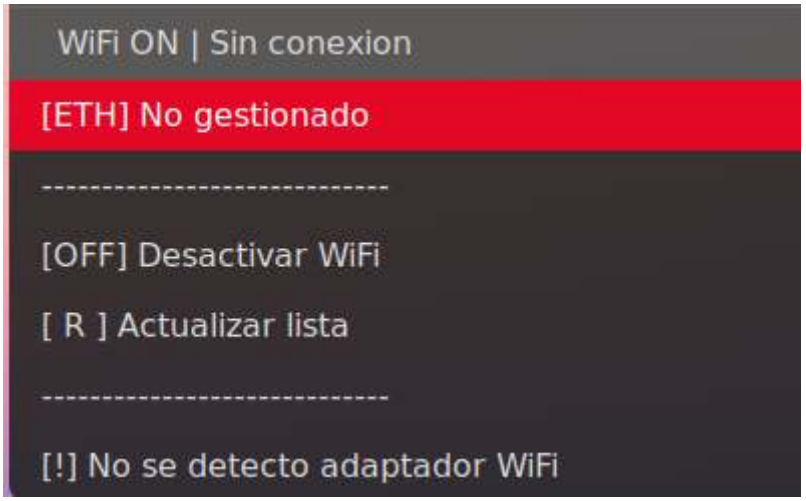


Figura 4.1. Panel de internet. Fuente: Propia.

El panel de internet no está finalizado debido a que la conexión por cable no funciona adecuadamente. Aún así, el sistema es usable al poder conectarse vía wifi.

Panel de sonido: Este panel se encarga de controlar/gestionar el volumen y el micrófono del equipo. En este caso, este es el único panel no hecho por rofi debido a las complicaciones visuales que presenta crear una barra deslizante en Bash, por lo tanto este panel ha sido desarrollado en Python. Para entender de manera breve cómo funciona este código, simplemente hay que saber que hace de complemento para unir el panel visual con el programa PulseAudioSystem, que en segundo plano se encarga de realizar estas funciones.

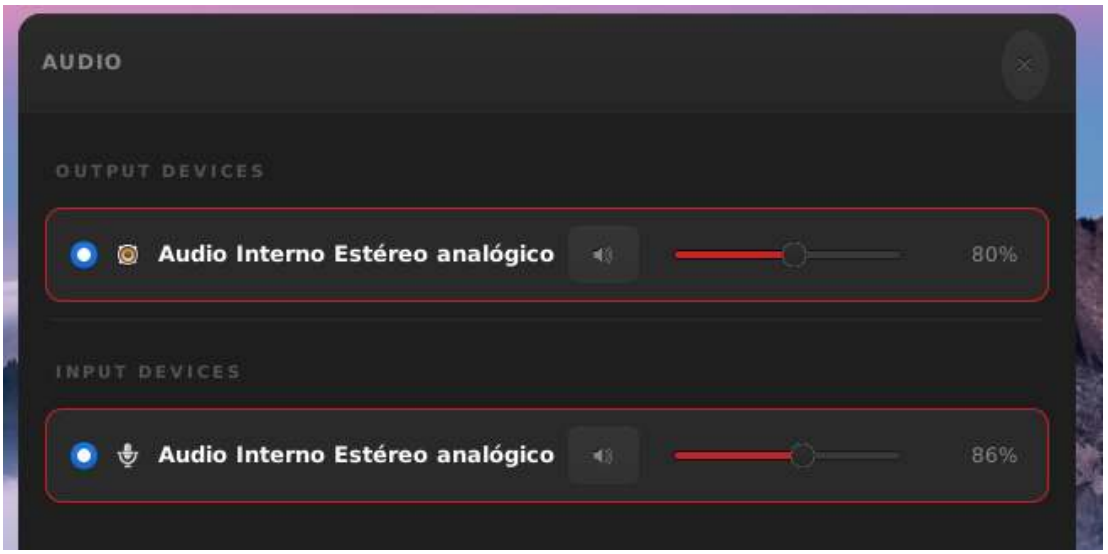


Figura 4.2. Panel de Sonido. Fuente: Propia.

El panel de sonido está totalmente finalizado para cumplir su función.

Panel de batería: Este panel se encarga de monitorizar el estado de la batería o el estado de la corriente. Además, trae la opción de gestionar el brillo de la pantalla en caso de no querer utilizar los botones del teclado, siempre y cuando sea un portátil, en el caso de ser una máquina virtual simplemente no mostrará información. En este contexto, el script se basa en leer el hardware, más específicamente el módulo, encargado de la batería del sistema.

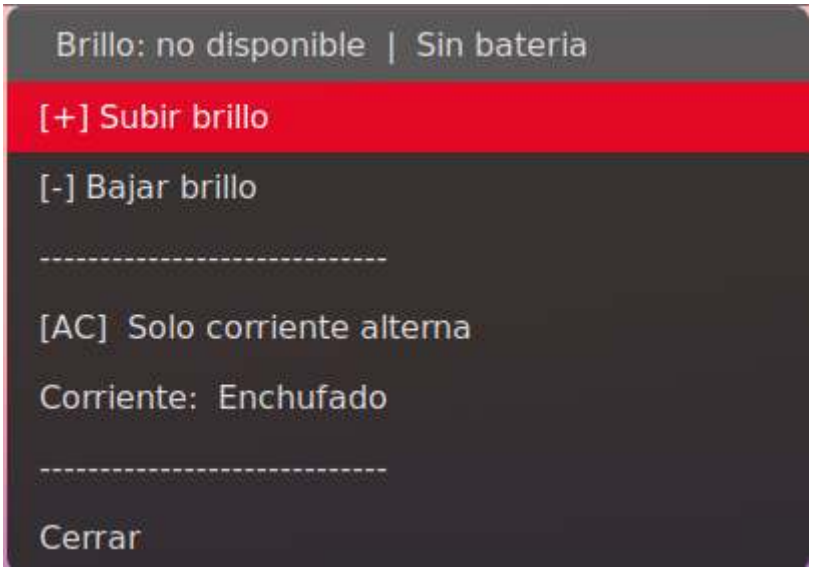


Figura 4.3. Panel de brillo. Fuente: Propia.

Este panel se encuentra finalizado y funciona exactamente como debería.

Menú de apagado: Más que un panel es un menú que muestra diversas opciones de energía, las cuales són: bloquear, suspender, cerrar sesión, reiniciar y apagar. Todas estas opciones no están repartidas en varios scripts si no que se encuentran todas en el mismo script principal con un enlace al tema del rofi porsupuesto.

No hace falta explicar demasiado la importancia de un menú de apagado del sistema. Simplemente es algo vital para poder considerarse una distribución hoy en día y también para poder brindarle la mejor experiencia posible al usuario.

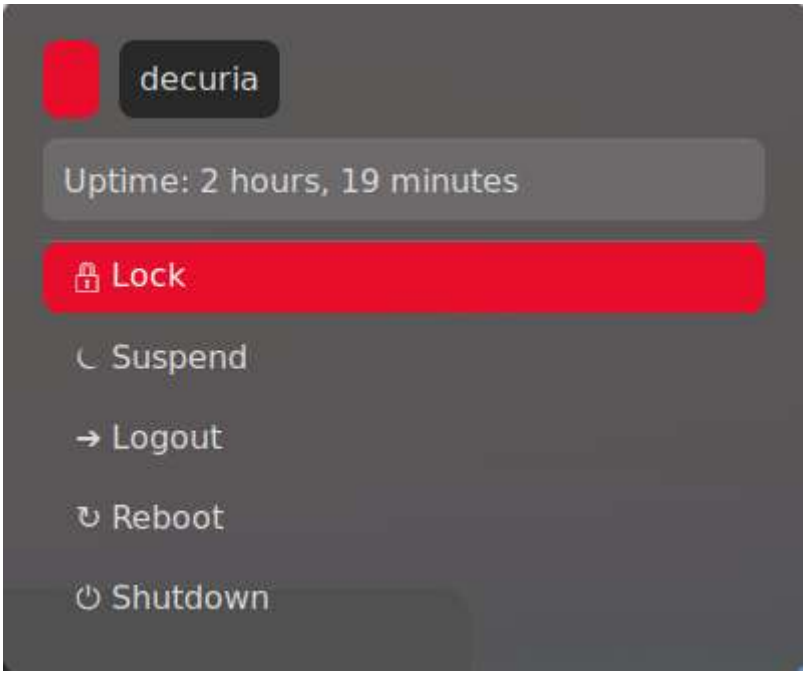


Figura 4.4. Menú de apagado. Fuente: Propia.

Este panel no está finalizado debido a que la opción de suspender la pantalla no hace su cometido dejando la pantalla congelada sin opción de retomar el uso sin tener que recurrir a un reinicio forzoso, pero para el tema principal del proyecto no es algo en lo que se pueda perder tiempo.

2.7.3 Funcionalidad 3

La última funcionalidad desarrollada por nosotros mismos es el cartel de bienvenida, un pequeño programa que muestra una ventana con tres diapositivas en donde se da información al usuario sobre lo que Decuria puede ofrecer. Este es un mensaje que aparece una sola vez por usuario.

La manera de desarrollar este cartel ha sido exactamente igual que con las otras funciones desarrolladas por nosotros, en este caso hemos creado el programa en Python y para que este programa se ejecute una única vez por el usuario ha sido necesario crear un script con esa función.

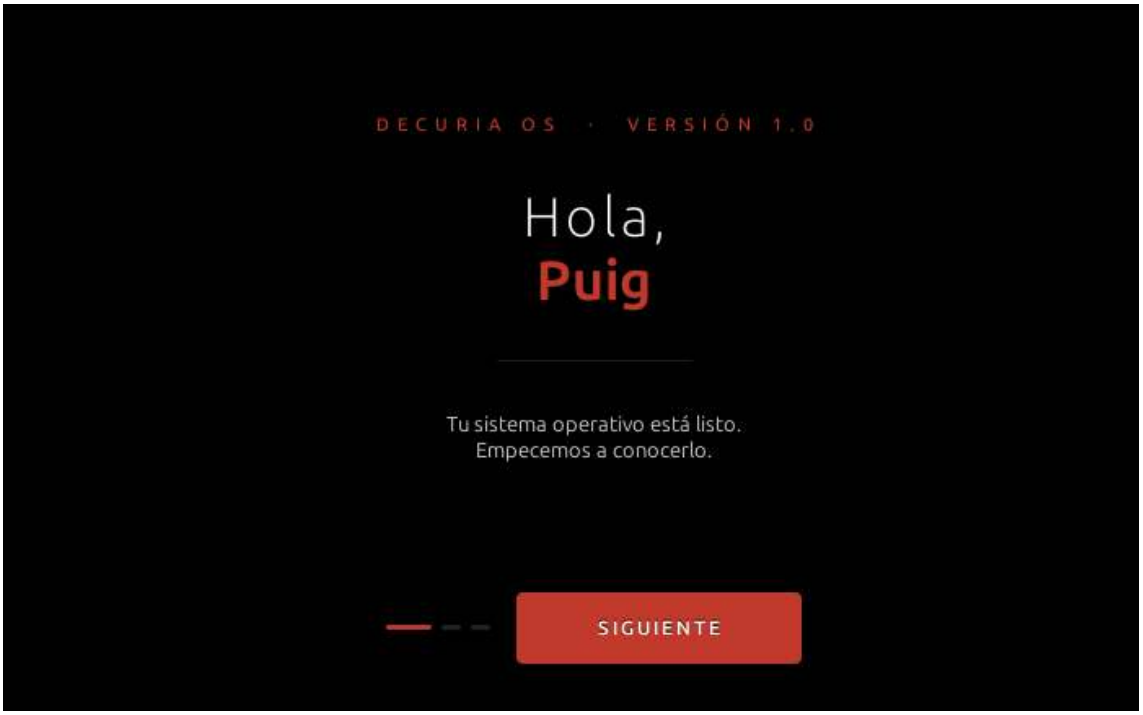


Figura 5.1. Cartel de bienvenida 1. Fuente: Propia.

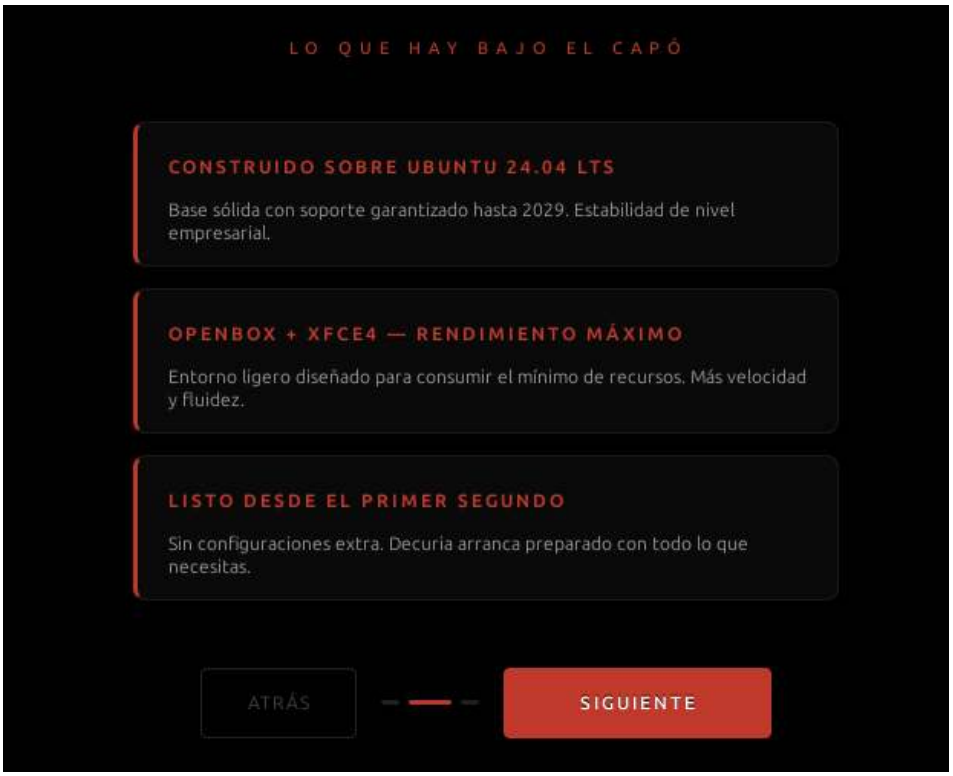


Figura 5.2. Cartel de bienvenida 2. Fuente: Propia.

Ambos códigos se encuentran en el repositorio de github en el apartado de anexos.

3 Otros capítulos

3.1 Comparación

Para realmente demostrar que Decuria cumple con su propósito de ser ligero y veloz, lo ideal es hacer una comparativa entre las distros de Linux más populares, entre ellos Ubuntu Desktop, el sistema preferido por defecto por la mayoría de los usuarios; y Linux mint,un sistema que es la competencia directa de Decuria al ofrecer rendimiento y modernidad justo como nosotros.

Para realizar esta comparativa puntuamos los aspectos más relevantes genéricos que cualquier sistema ofrece y algún detalle en particular que haga destacar a cada distro para así poder comparar justamente a todos. Todos los datos objetivos han sido probados en un mismo equipo para evitar cualquier tipo de trampa y las puntuaciones son en base que tanto puede ofrecer como sistema respecto a todo lo disponible globalmente.

El equipo de prueba es un portátil de la marca Acer con un procesador Intel ® Celeron ® N2940 DE 4 núcleos y 4 Gb de memoria RAM.

	Rendimiento (RAM)	**Modernidad / Intuitivo**	**Compatibilidad**	**Característica única**
Decuria	~400Mb	Openbox+Xfce = 7	7	Panel de control de rendimiento
Ubuntu Desktop	~1Gb	Gnome = 10	8	Enfoque en la facilidad de uso y la accesibilidad
Linux Mint (Xfce)	~700Mb	Xfce (completo) = 9	8.5	Experiencia de escritorio tradicional y familiar

Decuria: Modernidad / Intuitividad un 7 debido a que la combinación de esos 2 entornos visuales por separado presentan una gran calidad para un buen rendimiento pero no tan completos para lo visual.

Ubuntu Desktop: Modernidad / Intuitividad un 10 debido a que Gnome probablemente sea el mejor entorno gráfico de escritorio para Linux no modificable, ya que este está pensado para usuarios novatos ofreciendo un diseño de UX/UI muy intuitivo, pero muy pesado también.

Linux Mint: Modernidad / Intuitividad un 8 debido a que al usar únicamente Xfce, en su totalidad cuenta con todas las dependencias que ligeramente aumentan el peso del sistema, pero ofrece un entorno simple y muy intuitivo debido al gran parecido de escritorio de Windows 10, perfecto para principiantes.

Decuria: Compatibilidad un 7 debido a que al simplemente ser Linux de por sí hay programas a los que ninguna distro puede acceder de manera nativa, pero respecto a los que sí puede es cierto que no cuenta con librerías de 32-bits necesarias normalmente para juegos como los de Steam. Esto se debe a que Decuria no está pensado para ser una distro para jugar, aunque no es imposible, pero depende de la configuración del usuario aunque no cuenta con estas librerías por defecto ya que no se ciñe a nuestro público objetivo.

Ubuntu Desktop: Compatibilidad un 8 debido a que este sistema busca ser más generalista y básicamente ofrece esas librerías.

Linux Mint: Compatibilidad un 8.5 porque cuenta con esas librerías pero es aún más amigable con los usuarios principiantes al proporcionar un instalador de codecs multimedia y drivers de manera más sencilla.

Pruebas visuales:

Decuria: 423M de RAM

Figura 6.1. Test Decuria. Fuente: Propia.

Ubuntu Desktop: 1.47G de RAM

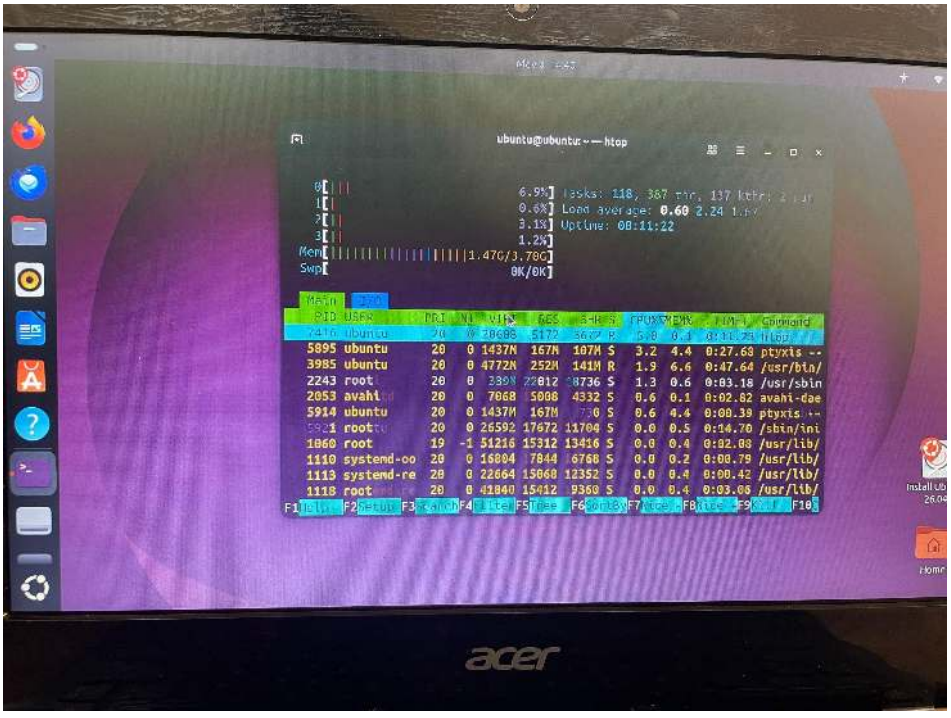


Figura 6.2. Test Ubuntu. Fuente: Propia.

Linux Mint: 914M de Ram

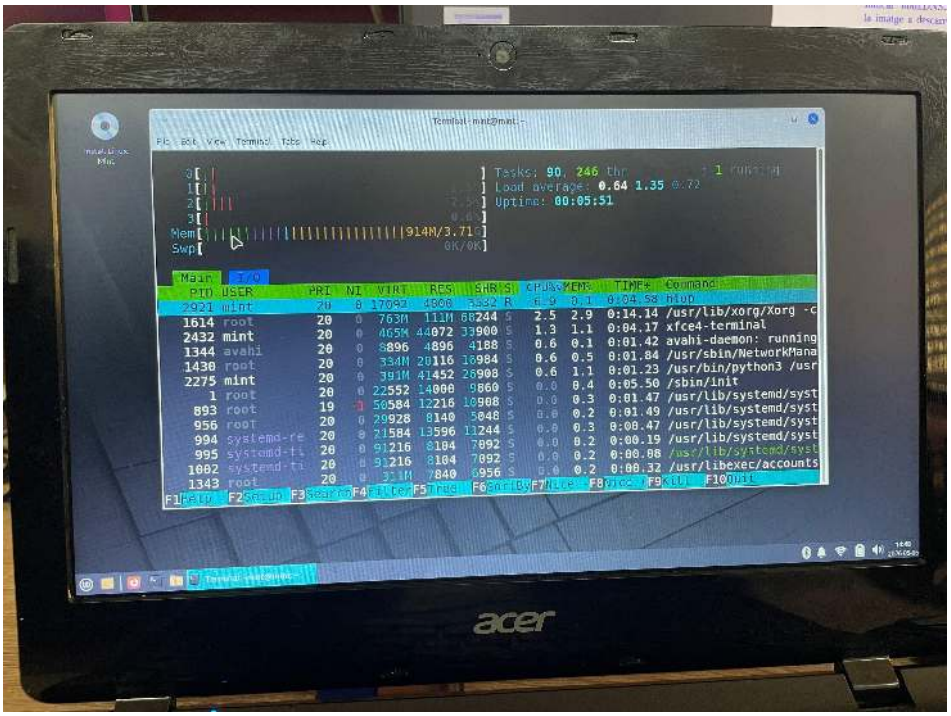


Figura 6.3. Test Linux Mint, Fuente: Propia.

Tras esta breve comparativa, la conclusión de esta es que como una buena distro de Linux cada sistema ofrece diferentes servicios y está pensado para diferentes públicos. Por lo tanto, Decuria no es mejor ni peor distribución, simplemente tiene un público objetivo distinto. Sin embargo, la comparativa ayuda a ver que tanta diferencia hay respecto a los más populares y se puede ver que cumple en ofrecer incluso mayor rendimiento que la mayoría de sistemas populares.

3.2 Web

Una de las ideas principales contempladas durante el desarrollo del proyecto fue la creación de una página web relacionada con Decuria. Esta decisión surgió a partir de una actividad realizada en clase, en la que debíamos desarrollar un sitio web, por lo que consideramos que era una buena oportunidad para integrarlo dentro del proyecto y ampliar su alcance más allá del propio sistema operativo.

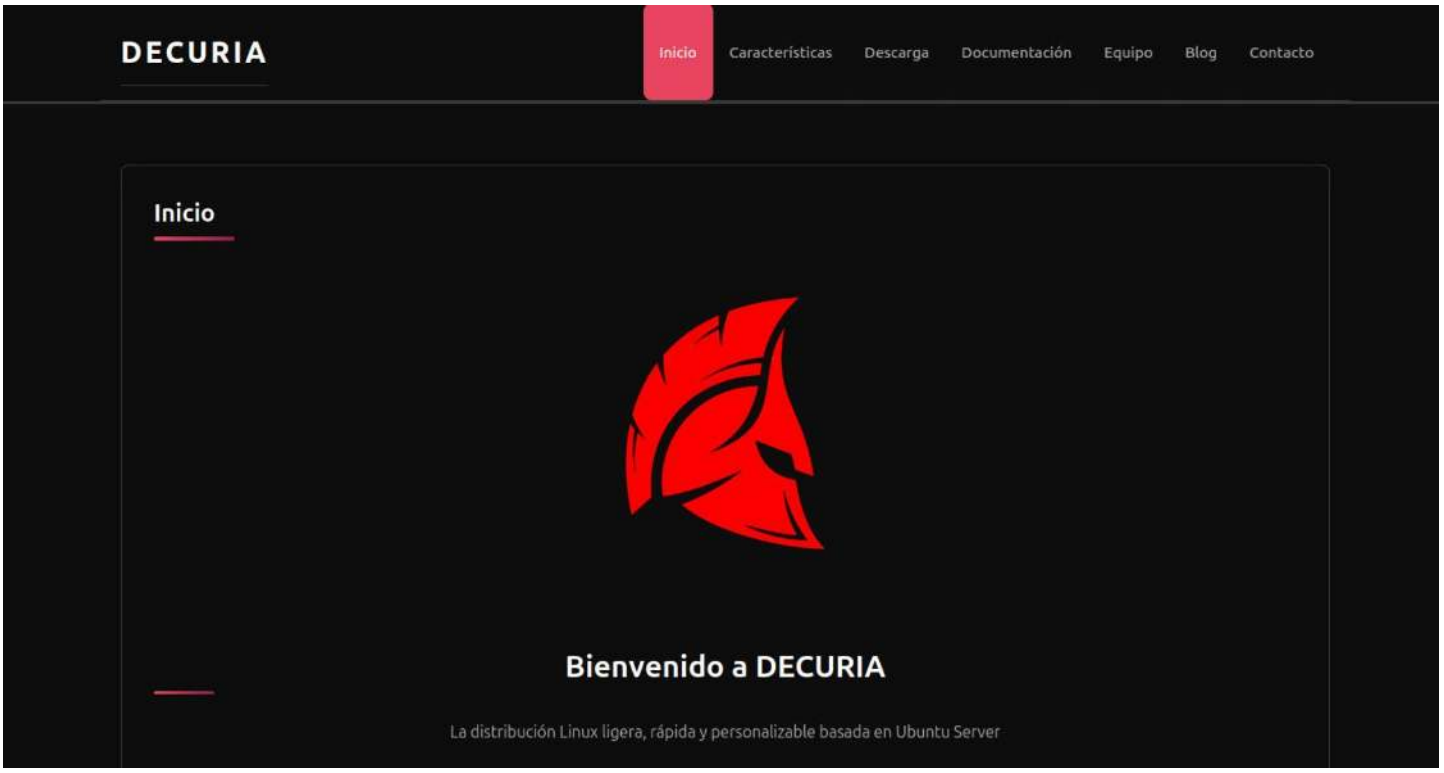
El objetivo principal de esta página web era ofrecer un espacio público donde poder presentar y promocionar Decuria, permitiendo a cualquier usuario conocer más en detalle el proyecto, sus características y su funcionamiento. De esta forma, la web actuaría como una plataforma informativa y de presentación, aportando una imagen más completa y profesional al proyecto.

Para ello, se diseñaron distintos apartados orientados a cubrir diferentes necesidades e intereses de los usuarios. Entre ellos, se desarrolló una sección de Blog destinada a compartir experiencias, novedades y contenido relacionado con el sistema. También se incluyeron apartados centrados en las características principales de Decuria, requisitos mínimos, recomendaciones de uso y diversa información técnica relacionada con el sistema operativo.

El desarrollo del sitio web se realizó progresivamente, ampliando sus funcionalidades y contenido a medida que avanzaba el proyecto. Posteriormente, se contempló la posibilidad de incorporar un apartado de descargas desde el cual los usuarios pudieran obtener directamente la imagen ISO del sistema operativo. Esta funcionalidad pudo implementarse gracias a la flexibilidad y capacidad de adaptación que ofrece WordPress para el desarrollo y gestión de páginas web dinámicas.

En conjunto, la página web permitió complementar el proyecto principal ofreciendo una plataforma accesible, visual e informativa, facilitando tanto la difusión del sistema como la interacción y el acceso al contenido relacionado con Decuria.

A continuación, se mostrarán distintas capturas de la página web desarrollada:



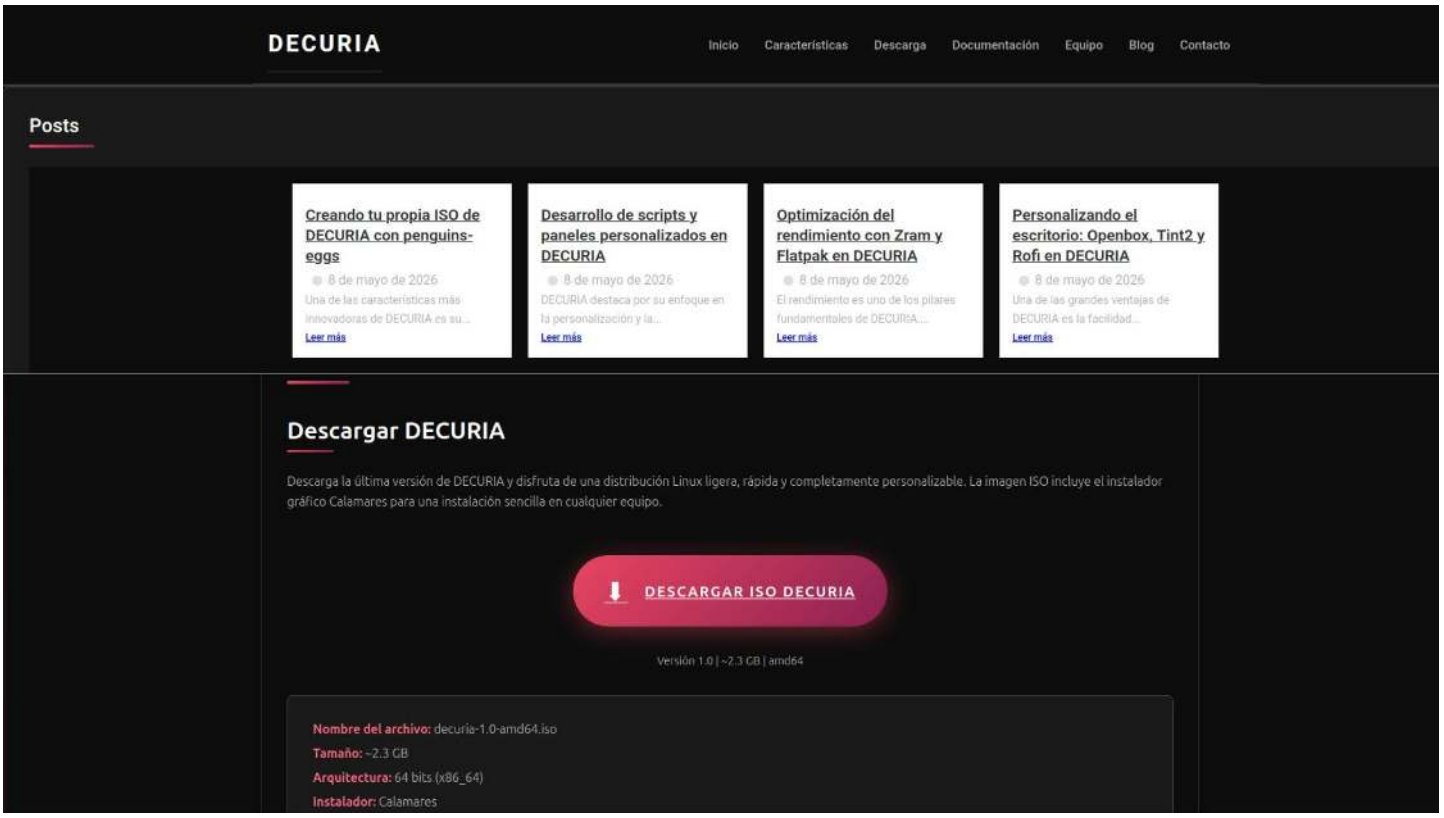


Figura 6.4. Pagina web Wordpress. Fuente: Propia.

3.3 Ideas descartadas

Realizar la ISO final con su compresión máxima

Como el título indica, se refiere a comprimir la ISO lo máximo posible de manera que incluso llegue a ocupar 1Gb, en principio esta idea suena fenomenal y muy complementaria con el propósito de Decuria, pero la realidad es otra. A causa de que la ISO está más comprimida, más tiempo tarda luego en descomprimirse a la hora de arrancar esta ISO en una máquina, por lo tanto lo mejor es un equilibrio entre que el disco ocupe algo más pero así obteniendo una velocidad de arranque mayor.

Añadir librerias de 32-Bits

En el punto de las comparativas ya se mencionó que la razón principal para no incluir estas librerías es debido a que Decuria no está pensado para ser una distro gaming que realmente se puede jugar en él, ya que es algo que debe configurar el usuario. Esta decisión se ha tomado para evitar desviarnos del objetivo principal y simplemente para no hacer más pesado el sistema.

4 Conclusiones

4.1 Conclusiones generales del proyecto

Decuria ha sido un proyecto ambicioso y arriesgado al alejarse tanto de la dinámica común de proyectos del centro, pero es también por lo que destaca y llama la atención. El crear una distribución de Linux es una tarea en donde al principio avanzas rápido, pero una vez llegas a la mitad es cuando te estancas en la misma pantalla configurando por meses un solo apartado, hecho que realmente puede llegar a ser desesperante. Por mal que pueda llegar a sonar, en realidad también es un proyecto divertido al prácticamente solo tocar archivos de configuración que cambian totalmente la apariencia de lo que se observa en el mismo instante en el que se modifica hace que se tenga una idea clara de lo que puede estar fallando y a su vez motiva a seguir al ver progreso en tiempo real. El nivel de desarrollo sobre este proyecto ha sido del nivel esperado y no algo imposible, eso sí, es un trabajo muy laborioso, por eso se debe ser constante para no perder el hilo.

Respecto a cómo nos ha enriquecido el desarrollo de este proyecto académica y profesionalmente, podemos decir que hemos adquirido muchísimos conocimientos nuevos sobre la gestión de sistemas, tener un mayor entendimiento de cada carpeta que viene por defecto en un sistema, tener una noción básica sobre los scripts en Bash, explorar el mundo de la programación al utilizar Python viniendo de saber Java de manera básica, muy por encima el diseñar logotipos o fondos, saber gestionar el trabajo en equipo mediante herramientas que permiten clasificar / ordenar las tareas como el Trello o el Excel, aprender a gestionar el tiempo y las ideas a partir de un diagrama y de un documento funcional, aprender a cómo estructurar una buena memoria para plasmar adecuadamente nuestro trabajo, aprender a preparar una buena presentación y sobretodo, a tener un mayor conocimiento sobre el Kernel de Linux.

4.2 Consecución de los objetivos

Objetivo inicial	Argumento
El sistema arranque rápidamente y sin procesos innecesarios.	Objetivo cumplido debido a que en comparación a otras distribuciones de Linux, Decuria junto a Linux Mint ha sido el primero en arrancar con una duración aproximada de 30s desde que se enciende el equipo.
La interfaz sea intuitiva y agradable visualmente.	Objetivo cumplido parcialmente, ofrece una interfaz visualmente agradable pero es demasiado simple y en realidad es un aspecto que se podría haber pulido más. Por otro lado, sí que es intuitivo pero no tanto como otros sistemas, básicamente se requiere acostumbrarse un poco.
Todas las herramientas esenciales funcionen de forma estable.	La gran mayoría de herramientas / programas esenciales del sistema funcionan correctamente demostrados en el apartado de prueba de compatibilidad. Es posible que alguno muy concreto no sea compatible con algunos equipos o máquinas virtuales, pero estos son herramientas específicas del xfce4-settings-manager.
La carga del sistema se mantenga siempre baja y que la gestión del	Este objetivo está totalmente completado demostrado en la sección de comparación en donde

hardware antiguo sea correcta y eficiente.	muestra que por diferencia es el sistema con mejor rendimiento y también demostrado que arranca en hardware antiguo.
Herramientas propias para la gestión de rendimiento.	Este objetivo también se encuentra totalmente completado y demostrado con todas las herramientas en el apartado de funciones propias en donde se muestra la función de cada una.

4.3 Valoración de la metodología y planificación

Tras un curso entero trabajando en este proyecto, podemos decir que estamos satisfechos con el resultado final y todo esto ha sido gracias a que hemos seguido la metodología de trabajo de tipo ágil que estipulamos al principio. El hecho de implementar y probar en el mismo instante cada nueva función permite tener un control mayor sobre los posibles errores que se puedan presentar. Esta metodología se ha seguido durante todo el curso ya que consideramos que para nuestro proyecto es la más práctica de probar e implementar.

Respecto a la planificación del trabajo hecho en Trello, se ha seguido al pie de la letra ya que hemos ido añadiendo y descartando ideas y tareas, mientras que a la planificación del Excel la realidad es que no hemos cumplido siempre con los tiempos de cada tarea debido a que muchas veces una tarea aparentemente sencilla termina estancando más, así que principalmente ha sido cosa de apartarse.

4.4 Visión de futuro

Para el futuro de Decuria lo principal es hacerlo más profesional antes de seguir trabajando en el rendimiento que este puede ofrecer. La razón de esto es poder realmente lograr que se distribuya globalmente y para ello, para que un usuario le de gusto usar este sistema, lo primero que habría que lograr mejorar es la experiencia del usuario de manera que aspectos dejados de lado en esta ocasión se incluyan en un futuro como los siguientes:

- Poder configurar el idioma a través de un panel gráfico una vez que el sistema haya sido instalado (el instalador si que ofrece esta opción).

- Personalizar el instalador con nuestra marca para dar ese aspecto más profesional.

- Pulir la función y aspecto visual del entorno gráfico, sobretodo la barra de tareas y las herramientas de rendimiento nativas de Decuria.

- Arreglar el bug de la conexión de internet por cable ya que actualmente muchas veces falla, cosa que no ocurre con el wifi. (No arreglado por falta de tiempo).

- Arreglar el bug del brillo de pantalla, consiste en evitar que el brillo al ser bajado al mínimo llegue a apagar la pantalla por completo. (No arreglado por falta de tiempo).

- En un futuro nos gustaría crear una versión de Decuria destinada al uso más general, permitiendo así compatibilidad con juegos que requieran librerías específicas.

5. Glossario

Distribución	En el contexto de Linux, una distribución es aquel sistema que utiliza el Kernel de Linux o deriva de uno que utilice este mismo Kernel.
Kernel	Un Kernel es el puente de comunicación entre el software y el hardware, sin este un sistema no podría ser utilizado al no detectar ningún componente.
Script	Un script es un archivo que contiene un conjunto de instrucciones escritas en lenguaje de programación que se ejecutan de forma secuencial sin necesidad de ser compiladas.
ISO	Una ISO en el contexto informático es un tipo de archivo que contiene una imagen exacta de un sistema. Muy útil para distribuir software.
RAM	La RAM es un tipo de memoria volátil de manera que almacena temporalmente los datos de los programas en uso para que el procesador acceda a ellos rápidamente.

6. Bibliografia

[1]https://www.stackscale.com/es/blog/distribuciones-linux-populares/ - 15/10/25

[2] https://blogubuntu.com/ubuntu-muslim-edition - 16/10/25

[3] https://penguins-eggs.net/ - 20/10/25

[4] https://www.ibm.com/es-es/think/topics/linux-kernel - 20/10/25

[5] https://www.w3schools.com/python/ - 01/11/25

[6] https://www.w3schools.com/css/default.asp - 12/11/25

[7] https://www.youtube.com/watch?v=uLQTvjzd2gI - 01/11/25

[8] https://github.com/davatorium/rofi - 04/11/25

[9] https://www.xfce.org/ - 05/11/25

[10] https://wiki.debian.org/es/LightDM - 20/11/25

[11] https://github.com/yshui/picom - 10/11/25

7 Anexos

Código fuente de las funciones propias del sistema en github:

https://github.com/68sm68/Proyecto-Decuria