

Semana 4

Software Libre

Unidad 4

Estudio de casos

LINUX FREE BSD.

Material compilado con fines académicos, se prohíbe su reproducción total o parcial sin la autorización de cada autor.

4. Estudio de casos LINUX FREE BSD

Linux es sin duda la aplicación que caracteriza al desarrollo de software libre, al grado de confundir el propio software libre con Linux. Linux es parte de un sistema operativo en el cual funcionan diversas aplicaciones que no necesariamente son software libre (Acrobat Reader, Matlab, etc.).

En la actualidad Linux está en su versión 4.17. Sin duda se espera que continúe mejorando en aspectos de rapidez, rendimiento y eficiencia. Este tipo de avances ocurren rápidamente, en aproximadamente 2 o 3 semanas es posible obtener una nueva versión que ofrece las ventajas antes citadas.

La siguiente infografía ilustra información relevante del núcleo de Linux, así como las etapas de desarrollo que ha tenido hasta la fecha. *(Ver imagen en la siguiente página)*

FreeBSD es el más antiguo y conocido de los sistemas BSD, el objetivo de este proyecto, es la creación de un sistema operativo que sea utilizado sin ningún tipo de restricción, pero con la posibilidad de disponer del código y de utilizarlo en función de sus necesidades (Barahona, 2003).

El modelo de desarrollo está basado en el sistema de versiones CVS y el sistema de información de errores GNATS. El proyecto FreeBSD distribuye su software de dos maneras:

- A través de un sistema que descarga las fuentes, las compila e instala la aplicación de forma local.
- Con el uso de paquetes, que consisten en las fuentes de los *ports* precompilados y, por lo tanto, en formato binario. La ventaja de los ports es que, a partir de los paquetes, permiten al usuario configurar y optimizar el software en la computadora. La desventaja se presenta cuando el sistema de paquetes, al estar ya precompilado, permite instalar el software de una forma más rápida.

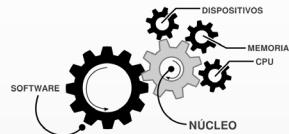
En seguida se presenta a las organizaciones y desarrollos de software más representativos, que trabajan bajo la modalidad de código abierto.

El Núcleo Linux

En 1991, un estudiante finlandés llamado Linus Torvalds concibió lo que hoy se conoce como el núcleo Linux. En poco tiempo, la iniciativa se popularizó y gran cantidad de programadores a nivel mundial se unieron para contribuir con su desarrollo. Es así como el núcleo Linux se convierte en uno de los fenómenos de desarrollo colaborativo más exitosos de la historia del software.

¿QUÉ ES UN NÚCLEO?

El núcleo de un sistema operativo es el componente que interactúa con el hardware y administra la comunicación entre el software y éste.



¿CÓMO NACIÓ EL NÚCLEO LINUX?

1984

Richard Stallman inicia el proyecto GNU para la creación de un sistema operativo libre. Para 1991, ya existían muchos componentes pero hacía falta el núcleo.



1991

Linus Torvalds publica los avances preliminares del desarrollo del núcleo Linux. Cuando anuncia el desarrollo del proyecto dice: "Estoy haciendo un sistema operativo. Es sólo un pasatiempo, no será nada grande ni profesional como GNU".

LICENCIA DEL NÚCLEO LINUX

El núcleo Linux utiliza la Licencia Pública General GNU, la cual permite la distribución y comercialización de versiones modificadas y sin modificar del núcleo Linux, pero requiere que todas esas copias se distribuyan bajo la misma licencia y estén acompañadas por el código fuente correspondiente. Este tipo de licencias es conocida como "copyleft".

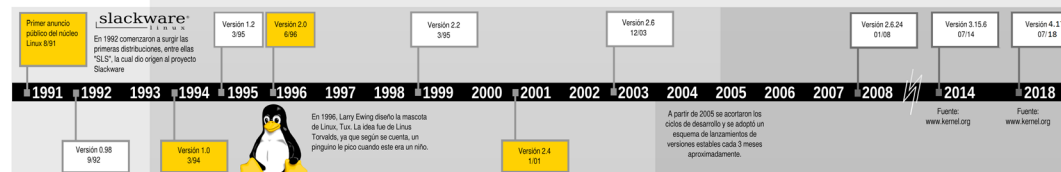


Todos los contribuyentes del núcleo Linux retienen sus derechos de autor y liberan sus contribuciones bajo una licencia tipo copyleft.

¡Esto significa que el núcleo Linux posee miles de dueños!



LÍNEA DE TIEMPO DEL NÚCLEO LINUX



Infografía Núcleo Linux por CIDETYS
www.cidetys.org.pa



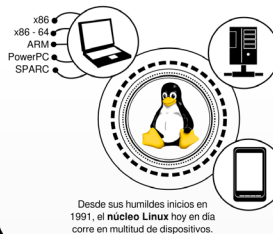
Infografía Núcleo Linux por CIDETYS
está licenciada bajo Creative Commons
Reconocimiento 4.0 Internacional License.

Imágenes de Wikimedia Commons
Logo Linux: Larry Ewing
Foto Linux Torvalds: linuxmag.com
Foto Richard Stallman: autor Gisle Hannemyr

Arte y diseño
Juan Barba: fedoraproject.org/wiki/User:Shaksx
Mónica J. Mora: ayalita.wordpress.com

Textos y diagramas
Mónica J. Mora: ayalita.wordpress.com

Revisión
David Narváez: blog.dmaggot.org



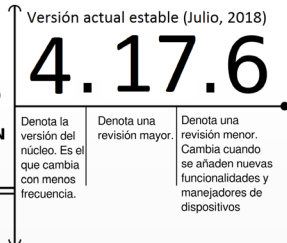
Desde sus humildes inicios en 1991, el núcleo Linux hoy en día corre en multitud de dispositivos.

LENGUAJE DE PROGRAMACIÓN

El núcleo Linux está escrito mayormente en el lenguaje de programación C en conjunto con la colección de compiladores GNU GCC.

C
Language
GNU C Compiler

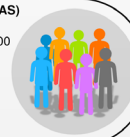
SIGNIFICADO DE LA NUMERACIÓN DEL NÚCLEO LINUX



¿QUIENES DESARROLLAN EL NÚCLEO LINUX?

VOLUNTARIOS(AS)

Se estima que 2400 profesionales y académicos desarrollan el núcleo Linux



THE LINUX FOUNDATION

Promueve y fomenta el ecosistema Linux. Además brinda soporte legal al núcleo Linux y auspicio a la labor de Linus Torvalds.



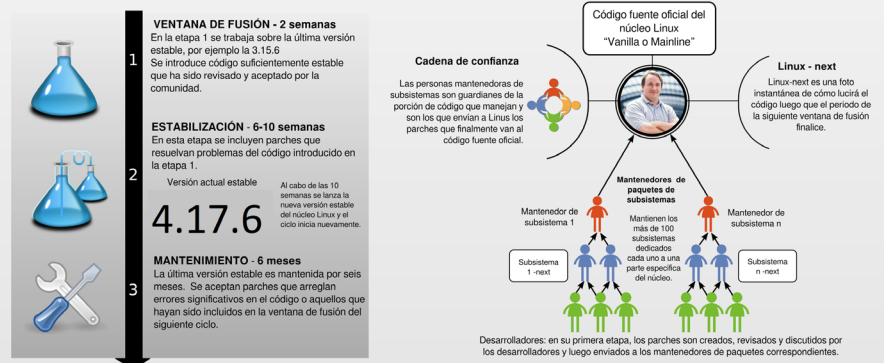
EMPRESAS

Aproximadamente 75% de los cambios en el núcleo Linux provienen de una pequeña cantidad de empresas.

¿CÓMO TRABAJA LA COMUNIDAD LINUX?

El núcleo Linux posee un ciclo de desarrollo en el que se lanza una versión estable aproximadamente cada tres meses. ¿Cómo lo hacen?

Existe una sola persona que controla lo que entra al código fuente oficial del núcleo Linux: Linus Torvalds. He aquí cómo lo hace.



4.1 GCC

Originalmente era conocido como “GNU C Compiler”. Es la colección de compiladores del proyecto GNU. Proporciona analizadores de código fuente para varios lenguajes de programación, incluidos C, C++, Objective-C, Fortran, Java, Ada y Go. También incluye bibliotecas de soporte de tiempo de ejecución para estos lenguajes (Foundation, Free Software Directory, 2016).

GCC proporciona muchos niveles de comprobación de errores de código fuente proporcionados por otras herramientas, produce información de depuración y puede realizar optimizaciones al código objeto resultante.

El compilador GNU para Java está integrado y es compatible: GCJ puede compilar recursos de Java, código nativo o archivos de clase. Los resultados se reflejan al ejecutar más rápido y permitir a los usuarios ejecutar un sistema gratuito sin necesidad de un compilador de Java propio. Otra ventaja de GCC es que admite muchas arquitecturas y sistemas operativos diferentes.

La sintaxis de GCC es:

Para programas en lenguaje C:

```
gcc [ -opción [argumento(s)_opción]] nombre_archivo
```

Para programas en lenguaje C++:

```
g++ [ -opción [argumento(s)_opción]] nombre_archivo
```

La declaración en “opción” debe ir precedida de un guion medio y pueden tener varias letras; se debe considerar que, no pueden agruparse varias opciones en un mismo guion.

Algunas opciones pueden requerir un nombre de archivo o directorio. Enseguida veremos algunos ejemplos de las distintas declaraciones de un archivo, para realizar el proceso de compilación desde GCC:

gcc hola.c	Compila el programa en lenguaje C y devuelve un ejecutable de nombre: a.out
gcc -o hola hola.c	Compila el programa en lenguaje C y devuelve un ejecutable de nombre: hola
g++ -o hola hola.cpp	Compila el programa en lenguaje C++ y devuelve un ejecutable de nombre: hola
gcc -c hola.c	No genera un ejecutable, sino un código de tipo objeto, devuelto con el nombre: hola.o . Si no es indicado un nombre al archivo, usa el nombre del archivo en C, cambiándole la extensión por .o
gcc -c -o objeto.o hola.c	Genera un código objeto, pero este caso si tiene un nombre de archivo asignado: objeto.o
g++ -o ~/bin/hola hola.cpp	Genera un ejecutable con el nombre: hola en el subdirectorío bin del directorío propio del usuario

El proceso de compilación consta de cuatro etapas continuas: *preprocesamiento*, *compilación*, *ensamblado* y *enlazado*. Los comandos gcc y g++ son capaces de realizar todo el proceso de una sola vez.

Hemos incluido el siguiente enlace que te permitirá seguir el proceso de compilación desde GCC.

<https://iie.fing.edu.uy/~vagonbar/gcc-make/gcc.htm#Ejemplos>
Compilación con GCC

También podrás consultar en el siguiente enlace los manuales de GCC:

<https://gcc.gnu.org/onlinedocs/>

GCC mantiene el objetivo de mejorar su compilador utilizado en los sistemas GNU. Usa desarrollo abierto y posee la capacidad de soportar varias plataformas, lo cual fomenta el uso de un compilador-optimizador global. Estas cualidades abren la posibilidad de atraer mayor colaboración para extender y mejorar sus características, lo cual permitiría ampliar su funcionamiento en diferentes arquitecturas y entornos.

4.2 Make

GNU Make es una herramienta que controla la generación de archivos ejecutables y otros archivos que no son archivos fuente de un programa.

Make obtiene su conocimiento de cómo crear su programa de un archivo llamado Makefile, que enumera cada uno de los archivos que no son fuente y cómo ejecutarlos desde otros archivos.

Las capacidades de Make son (System, 2016):

- Permitir al usuario final construir e instar su paquete sin conocer los detalles de cómo se realizan estos procedimientos.
- Establecer de forma automática los archivos que necesitan actualización, según los archivos origen que hayan cambiado. También determina el orden en las actualizaciones.
- Make no está limitado a un lenguaje en particular.
- Además de la creación de paquetes, make es utilizado para controlar la instalación o desinstalación de paquetes.

Vamos a revisar un ejemplo de compilación de un programa escrito en lenguaje C. Este ejercicio se basa en el ejemplo tomado del libro de Kernighan y Ritchie, *El lenguaje de programación C*.

El programa consiste en una notación polaca, la cual es una forma de indicar operandos y operadores que son implementadas a través de una pila o “stack”. En una operación como $4 + 2$, se ingresa inicialmente los operandos 4, 2 y luego el operador.

Ejemplo 1: Para calcular la operación $10 - 8$ se digita,

10 8 -

Ejemplo 2: Para calcular la operación $(10 - 8) * (4 + 5)$ se digita,

10 8 - 4 5 + *

Se puede observar en los ejemplos que, no es necesario un signo de *paréntesis* o de *igual*, lo cual es una ventaja debido a que necesita menos símbolos y permite más facilidad de implementación.

Un programa, con la anterior notación, puede implementarse en un solo archivo. Para este ejemplo distribuiremos el código en diversos archivos con extensión .c y un archivo

de encabezado con extensión .h. Por lo tanto, tendríamos los siguientes archivos:

- calc.h
- getch.c
- getop.c
- main.c
- stack.c

La compilación de este programa utilizando un archivo make básico sería con las siguientes instrucciones:

```
polaca : main.o stack.o getop.o getch.o
        gcc -o polaca main.o stack.o getop.o getch.o

main.o: main.c calc.h
        gcc -c main.c
stack.o: stack.c calc.h
        gcc -c stack.c
getop.o: getop.c calc.h
        gcc -c getop.c
getch.o: getch.c
        gcc -c getch.c

clean:
        rm polaca \
        main.o stack.o getop.o getch.o
```

Cada instrucción comienza con un nombre seguido de “:”. Las líneas iniciadas con tabulador son continuación de la misma regla.

La ejecución desde make se realiza con los siguientes pasos:

1. Compilar el programa con el comando: **\$ make** *observará los mensajes indicativos de las tareas realizadas.*
2. Enseguida se debe teclear el comando: **\$./polaca** con el que podrá verificar su funcionamiento.
3. Enseguida se digita el comando: **\$ make clean** que borrará el ejecutable y los archivos objeto

Para consultar más información de esta herramienta, puedes ingresar al siguiente enlace:

<https://www.gnu.org/software/make/>

Como hemos notado, con el uso de make podemos realizar la compilación completa de diferentes archivos que pertenezcan a un mismo programa. También es posible regenerar, usar y luego eliminar archivos intermedios que no necesitemos guardar.

4.3 Bison

GNU Bison es un programa generador de analizadores sintácticos de propósito general (Foundation, GNU Operating System, 2014). Este programa pertenece al proyecto GNU y está disponible casi para todos los sistemas operativos.

Bison convierte la descripción formal de un lenguaje, escrita como una gramática libre de contexto, en un programa en lenguaje C, C++, Java que realiza análisis sintáctico. El uso de esta herramienta requiere experiencia con la sintaxis usada en la descripción de gramáticas.

La forma general de la gramática de Bison es:

```
%{  
Declaración en lenguaje C  
%}  
Declaraciones de Bison  
%%  
Reglas gramaticales  
%%  
Código adicional en C
```

- Las cadenas: “%{”, “%}”, sirven para delimita el encabezado que pudieran ser bibliotecas.
- Las cadenas: “%%”, sirven para indicar las reglas.
- Al final se incluye la programación en C o llamadas a funciones, en caso de ser necesario.

Ejemplo: El siguiente código debería estar en un archivo “nombre.y”.


```
%{
#include <stdio.h>
}%

/* Declaraciones de BISON */
%token ARTICULO
%token SUJETO
%token VERBO
%%
cadena: ARTICULO SUJETO VERBO;
%%

int main( ) {
    yyparse( );
}
yyerror (char *s)
{
    printf ("%s\n", s);
}
int yywrap()
{
    return 1;
}
```

La compilación del archivo de Bison da como resultado dos archivos, un archivo con extensión **.h** y otro **.c**. El archivo **.h** (biblioteca) debe incluirse en el archivo de Lex. Donde los token ARTICULO SUJETO VERBO están dentro del archivo de Lex que, para este ejemplo, corresponde al archivo con extensión **.l**:

```
/* Archivo en lex */
%{
#include "reglas.tab.h"
}%

%%
El|lo|la|los|las {return ARTICULO;}
perro|perros|carro|carros|nino|ninos {return SUJETO;}
come|comen|corre|corren|juega|juegan {return VERBO;}
%
```

La compilación de ambos códigos con GCC, devolverían un ejecutable que funciona como un analizador sintáctico o parser.

Los analizadores lexicográficos y sintácticos generados por Bison son aprovechados como parte de los compiladores de los lenguajes de programación; estas funciones en

la herramienta no son las únicas, pero son las más usadas.

Para consultar más información de esta herramienta, puedes ingresar al siguiente enlace:

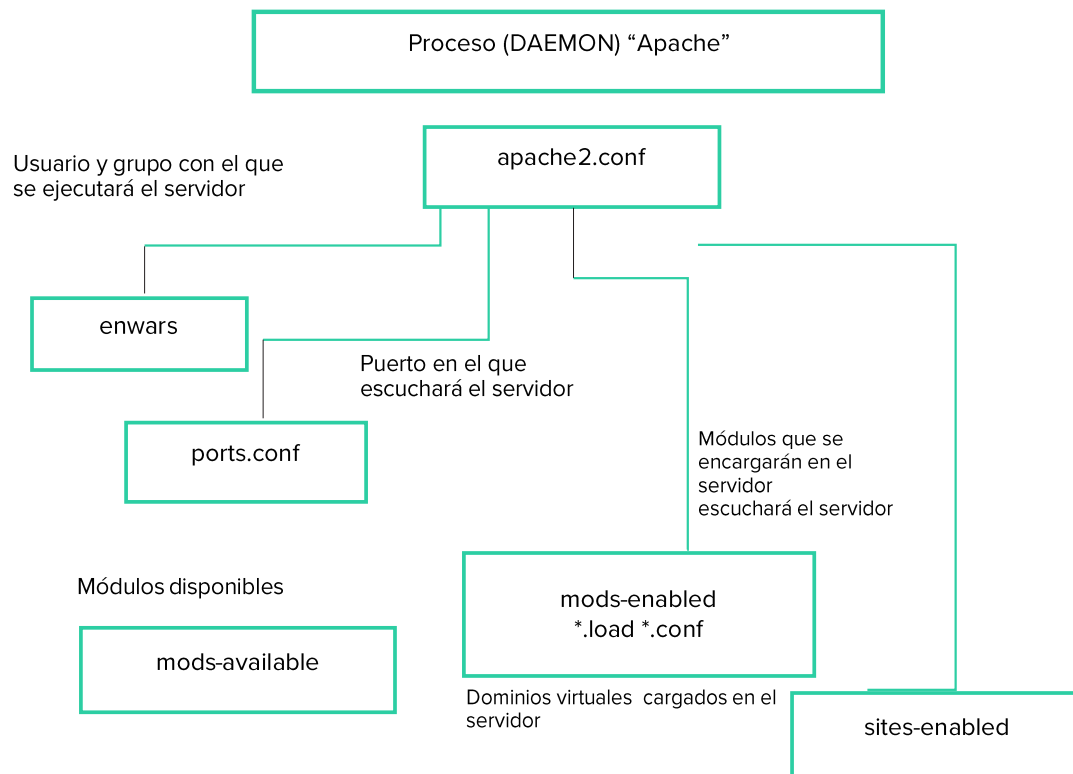
<https://www.gnu.org/software/bison/>

4.4 Apache

Apache Software Foundation (ASF) da soporte a los proyectos de software que están bajo el nombre de Apache, incluyendo al popular servidor HTTP Apache. Esta organización sin ánimo de lucro trabaja bajo el modelo basado en el consenso, la colaboración y con una licencia de software abierta.

Apache es utilizado para enviar páginas web estáticas y dinámicas en la World Wide Web (WWW). Muchas aplicaciones web están diseñadas asumiendo como ambiente de implantación a Apache, o que utilizarán características propias de este servidor web. Apache es el componente de servidor web en la plataforma de aplicaciones LAMP, junto a MySQL y los lenguajes de programación PHP/Perl/Python y Ruby.

El siguiente diagrama te ayudará a entender cómo está conformado un servidor Apache:



El archivo de configuración principal de Apache 2 es “apache2.conf” y se encuentra en el directorio “/etc/apache2”. Además, todos los módulos se encuentran almacenados en los directorios de configuración “mods-enabled” y “sites-enabled”.

Si estás interesado en crear un servidor web utilizando Apache, el siguiente enlace te proporciona los pasos para realizar dicha actividad:

Generación de un servidor en Apache

http://www.juntadeandalucia.es/empleo/recursos/material_didactico/especialidades/materialdidactico_administrador_servidores/Content/3-servicios_web/5-ApacheWebServer.pdf

La ASF es un importante ecosistema de software de código abierto, que proporciona algunas de las soluciones de software más utilizadas y más importantes que impulsan la economía actual.

Para consultar más información de esta fundación y de sus desarrollos, entre ellos Apache, puedes dirigirte al siguiente enlace:

<https://www.apache.org/>

4.5 GNOME

GNOME es desarrollado por The GNOME Project y provee los siguientes productos:

- El escritorio GNOME.
- Una interfaz gráfica de usuario.
- Un conjunto de aplicaciones centrales.
- Plataforma de desarrollo GNOME.
- Un entorno para crear aplicaciones que se integran con el escritorio.

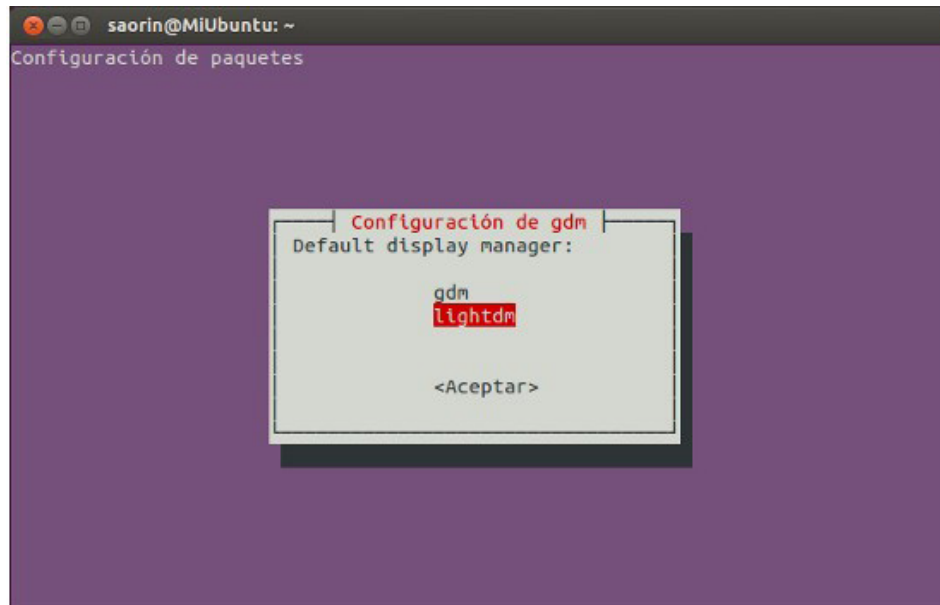
GNOME está disponible para ser instalado en las siguientes distribuciones de GNU/Linux:



- Fedora
- openSUSE
- Ubuntu

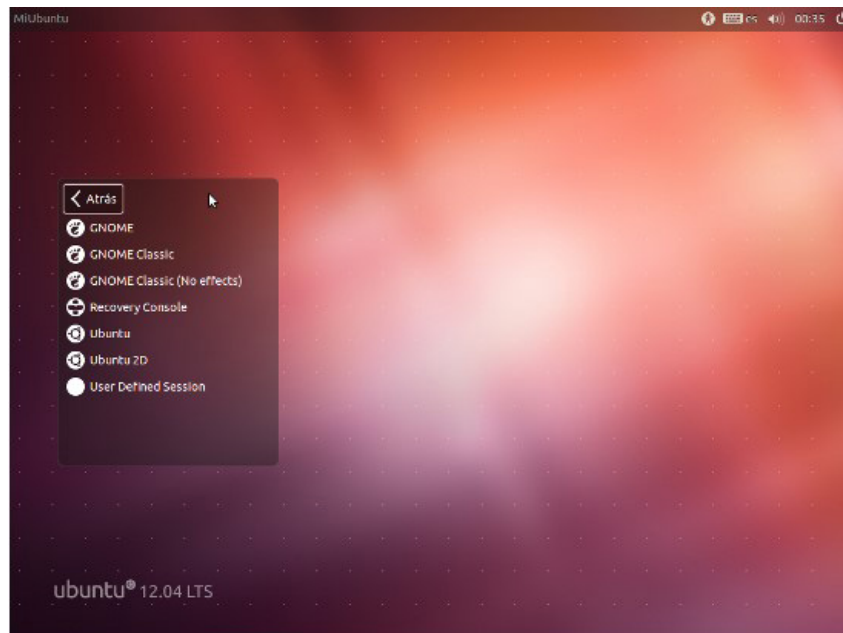
Enseguida revisaremos cómo se realiza la instalación del escritorio GNOME en Ubuntu. En general, la instalación del entorno no tiene mucha variación con respecto a la distribución en la que se haga.

1. Se abre la terminal en Ubuntu y se escribe el siguiente comando:
 - `sudo apt-get install gnome`

La instalación podría llevar tiempo, debido a la cantidad de paquetes que son instalados. Durante la instalación se observará la siguiente ventana.



2. Al finalizar la instalación, se debe dar clic en el ícono  y seleccionar la opción **Cerrar sesión**. Nos devolverá la siguiente pantalla, en la que debemos dar clic en el ícono  y seleccionar el escritorio con el que se quiera trabajar



3. Finalmente, al seleccionar GNOME, la interfaz y sus funciones estarán listas para usarse.



Lo primero que se recomienda hacer, es la actualización de paquetes de Linux y los complementos de GNOME. Esta opción se encuentra en: **Herramientas del sistema -> Administración -> Centro de software.**

Para consultar más información del proyecto GNOME, pueden ingresar al siguiente enlace:

<https://www.gnome.org/>

GNOME proporciona una plataforma de desarrollo para crear aplicaciones que se integran en un entorno de escritorio. El proyecto GNOME enfatiza en la simplicidad, facilidad de uso y eficiencia, creando aplicaciones de escritorio de código abierto para su reutilización bajo la licencia de software libre.

4.6 KDE

KDE Community es una comunidad de software libre dedicada a crear productos informáticos en modalidad abierta y fáciles de usar (Webmasters, 2016), ofreciendo:

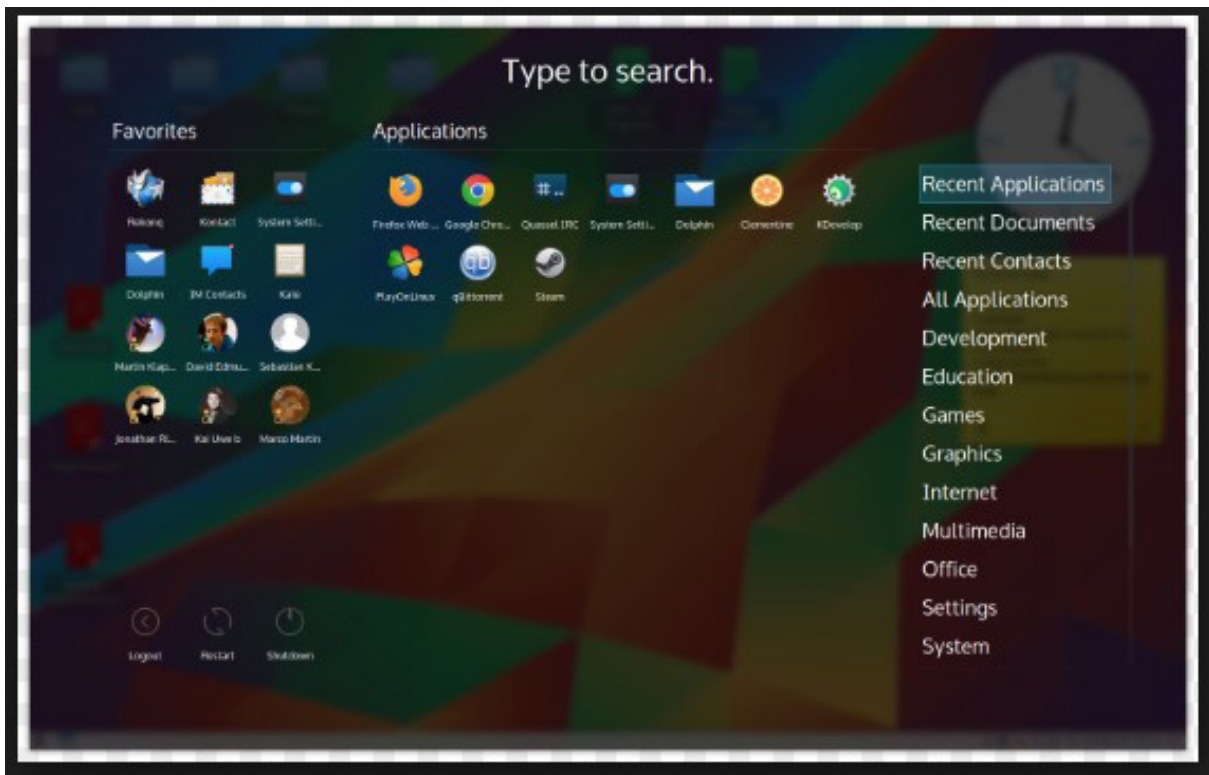
- Un escritorio gráfico avanzado.
- Una amplia variedad de aplicaciones para comunicación, trabajo, educación y entretenimiento.
- Una plataforma para crear nuevas aplicaciones de forma fácil.

Las aplicaciones KDE están traducidas en aproximadamente 75 idiomas y están desarrolladas bajo los principios de facilidad, uso y accesibilidad. Las aplicaciones KDE funcionan en GNU/Linux, BSD, Solaris, Windows y MAC OS X.

Enseguida revisaremos cómo se realiza la instalación del escritorio KDE plasma en Ubuntu.

1. Se debe abrir una terminal desde Ubuntu y ejecutar en ella el siguiente comando:
 - `sudo apt-get install plasma-desktop`

La instalación podría llevar tiempo, debido a la cantidad de paquetes que son instalados. Al finalizar la instalación se observará una interfaz del siguiente tipo:



2. Posteriormente, debe actualizarse los siguientes comandos, uno seguido del otro:
 - `sudo apt-get update`
 - `sudo apt-get update`

Para consultar más información de este proyecto y sus productos, ingresar al siguiente enlace:

<https://www.kde.org/>

El manifiesto KDE indica:

“Somos una comunidad de tecnólogos, diseñadores, escritores y defensores, que trabajamos para garantizar la libertad de todas las personas a través de nuestro software. Por este trabajo hemos llegado a valorar los elementos: gobernanza abierta, el software libre, inclusión, innovación, propiedad común y enfoque del usuario final”.

4.7 Mozilla

Mozilla es una organización sin ánimos de lucro que consiste en una suite de internet de código abierto y multiplataforma que incluye un navegador web, un editor HTML, un cliente de correo electrónico y un cliente de noticias.

Firefox es el navegador web desarrollado por Mozilla, el cual actualmente posee un potente motor creado para un rendimiento veloz al cargar páginas web más rápido y con menos uso de memoria, además, de proporcionar un diseño atractivo y con características “inteligentes” para una mejor navegación.

A través del siguiente enlace podrás encontrar un completo manual de instalación y uso de uno de los productos más importantes de la organización Mozilla, el conocido navegador Firefox:

Manual de Firefox

http://www.aeemt.com/contenidos_socios/Informatica/Manuales/Manual de firefox.pdf

La cantidad de líneas de código fuente que se encuentra en los desarrollos de Mozilla, lo posicionan como una aplicación rival, en cuanto tamaño, con el propio núcleo de Linux y OpenOffice.org. El lema de esta organización es promover la apertura, la innovación y la oportunidad en internet.

Para consultar más información de esta organización y sus productos, puedes ingresar al siguiente enlace:

<http://www.mozilla.org>

4.8 Mono

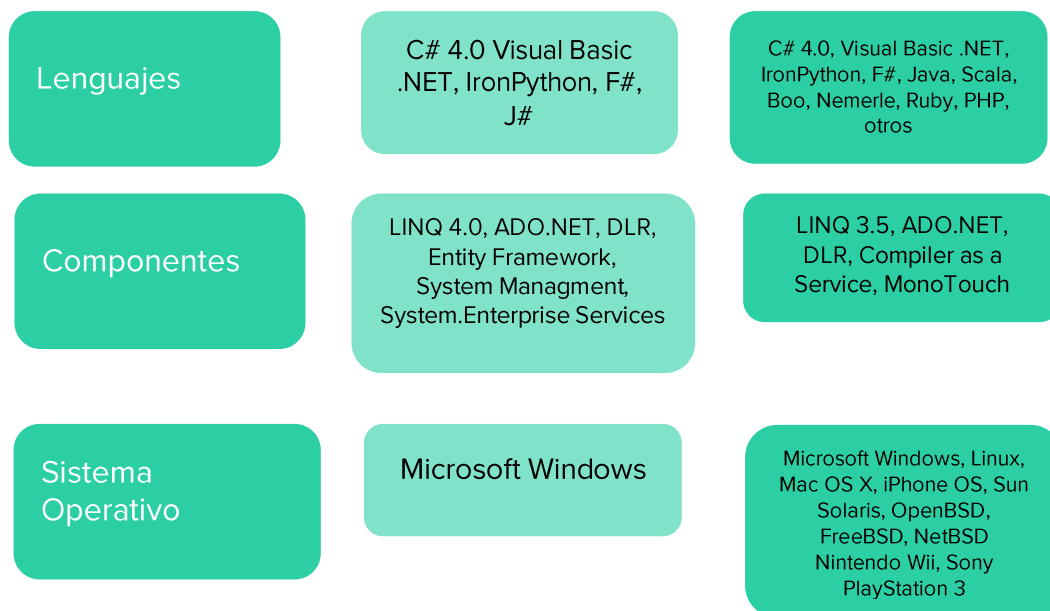
Mono es un proyecto de código abierto iniciado por la Ximian, empresa proveedora de software libre para Linux y Unix. Este proyecto actualmente es impulsado por la

compañía Novell, tras adquirir a Ximian, para crear herramientas libres basadas en GNU/Linux y compatibles con .NET.

Las utilerías de Mono en desarrollo de software son:

- Una máquina virtual con la infraestructura de lenguaje común.
- Una biblioteca de clases que funciona en cualquier lenguaje, siempre que funcione con CLR (Common Language Runtime).
- Un compilador para lenguaje C#, Visual Basic, Java y Python.
- Un sistema de objetos único, sistema de hilos, biblioteca de clases y sistema recolector de memoria.

En el siguiente diagrama se ilustran algunas semejanzas y diferencias entre Microsoft .NET y Mono.



Como se puede observar, Mono funciona en una variedad de dispositivos, incluyendo móviles y sistemas operativos. En el desarrollo de aplicaciones sobre dispositivos móviles, Mono cuenta con un componente no gratuito llamado MonoTouch. Este componente es útil en el manejo de la pantalla táctil en los dispositivos. Mono incorpora lenguajes ya existentes para Microsoft .NET como C#, Visual Basic, C++, IronPython y F#, permitiendo la interoperabilidad y la migración hacia .NET.

El proyecto Mono reconoce las fortalezas con las que cuenta .NET en el desarrollo de aplicaciones. Sin embargo, Mono tiene las propias, como la incorporación de caracterís-

tivas únicas, la ejecución multiplataforma, un entorno para el desarrollo de dispositivos móviles, servicio de compilación para C#, lenguajes como Boo y Java y la capacidad para desarrollar GUI multiplataforma utilizando GTK#. En contraparte, Microsoft .NET brinda herramientas útiles, pero son propias de Windows.

Para consultar más información acerca de Mono puedes ingresar al siguiente enlace:

<http://www.mono-project.com/>

4.9 Sistemas APM (Apache-Php y MySQL)

La combinación del servidor web Apache, con el módulo de PHP y la interacción con el servidor MySQL, proporciona la implementación de sitios web basados en software libre y gratuito.

Apache:

Es un servidor web muy utilizado actualmente en Internet. Una de las características importantes de Apache es que tiene una licencia, que, aunque no es GPL (Licencia Pública General), es posible tener acceso al código fuente. Apache está implementado en más del 50% de los servidores web en el mundo, este servicio es comúnmente encontrado en cualquier hosting.

PHP:

PHP (en español, Preprocesador de Hipertexto), permite la programación del lado del servidor. Funciona de tal forma que cuando un cliente envía una petición GET al servidor de una determinada página en PHP, antes de que el servidor entregue la petición al cliente, interpretará el código PHP y enviará el resultado. Una de las ventajas de usar PHP, es que su potente API (en español, Interfaz de Programación de Aplicaciones), le permite conectar con servidores de bases de datos como PostgreSQL y MySQL.

MySQL:

Es un sistema de gestión de bases de datos (BD) relacional que ha sido desarrollado bajo la licencia GPL/Licencia comercial por Oracle Corporation. Es considerado el gestor de BD código abierto más popular del mundo.

A diferencia de Apache, que es desarrollado por una comunidad pública, MySQL es patrocinado por una empresa privada, la cual posee los derechos de la mayor parte del código. MySQL es utilizado en desarrollos web, entre los que destacan, Joomla, Wordpress, Drupal y también en plataformas **Linux/Windows-Apache-MySQL-PHP/Perl/Python**, por ello es considerado uno de los cuatro componentes del desarrollo web **LAMP** y **WAMP**.

El siguiente enlace proporciona una guía básica del proceso de instalación y configuración de los servicios Apache, PHP, MSQl y phpMyAdmin, sobre el sistema operativo Linux. Todas estas herramientas, en unión y siendo de uso gratuito, permiten el establecimiento de un primer sitio web.

Instalación y configuración de Apache, MySQL, PHP y phpMyAdmin en Ubuntu

https://pensamientosaztlek.files.wordpress.com/2013/07/linux_instalaryconfigurarapachephpmysqlphpmyadminlenubuntu.pdf

Las aplicaciones Apache, PHP y MySQL proporcionan el soporte perfecto en el desarrollo de aplicaciones de software de código libre. Por separado son herramientas muy potentes y en unión, proporcionan soluciones muy eficientes.

Recapitulando en el contenido del último bloque de la materia “Software libre”. Se han revisado diversos proyectos que trabajan con la filosofía del desarrollo libre, los cuales han logrado tener amplia aceptación y éxito. Éstos y muchos más casos, siguen despertado el interés de la comunidad para integrarse a colaborar bajo esta modalidad de desarrollo.