



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

**ACTUALIZACIÓN Y REMODELACIÓN DE INFRAESTRUCTURA PARA EL IT
CENTRE OF EXCELLENCE ITCoE, SAE SAP, FACULTAD DE INGENIERÍA DE LA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

Cristian Alexander Aguilar López

Asesorado por el Ing. Marlon Antonio Pérez Türk

Guatemala, febrero de 2013

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**ACTUALIZACIÓN Y REMODELACIÓN DE INFRAESTRUCTURA PARA EL IT
CENTRE OF EXCELLENCE ITCoE, SAE SAP, FACULTAD DE INGENIERÍA DE LA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

CRISTIAN ALEXANDER AGUILAR LÓPEZ
ASESORADO POR EL ING. MARLON ANTONIO PÉREZ TÜRK

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, FEBRERO DE 2013

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Ing. Alfredo Enrique Beber Aceituno
VOCAL II	Ing. Pedro Antonio Aguilar Polanco
VOCAL III	Inga. Elvia Miriam Ruballos Samayoa
VOCAL IV	Br. Walter Rafael Véliz Muñoz
VOCAL V	Br. Sergio Alejandro Donis Soto
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Ing. Ludwing Federico Altán Sac
EXAMINADOR	Ing. Pedro Pablo Hernández Ramírez
EXAMINADOR	Ing. Juan Álvaro Díaz Ardavin
SECRETARIA	Inga. Marcia Ivónne Véliz Vargas

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

ACTUALIZACIÓN Y REMODELACIÓN DE INFRAESTRUCTURA PARA EL IT CENTRE OF EXCELLENCE ITCoE, SAE SAP, FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha 10 de febrero del 2011.

Cristian Alexander Aguilar López

ACTO QUE DEDICO A:

- Dios** Por darme vida, salud, inteligencia y la oportunidad de crecer como persona.
- Mis padres** Ramón Conrado Aguilar Reyes y Ana Lisseth López, por el apoyo que me brindaron para completar mis estudios.
- Mis hermanos** Kenny Arnold y Allan Wilfredo Aguilar López, por ser parte de mi vida.
- Mi novia** Mónica Marina León Herrera por ayudarme y apoyarme a alcanzar mi meta.
- Mis amigos** Francisco Castro, César Hernández, Joel Bojorquez, Werner Morales, Axel Hernández, Marco López, Fernando Ramírez, por acompañarme en el camino, a Hugo Mena y Gabriela Juárez por ofrecerme sus consejos.
- Mi asesor** Ing. Marlon Pérez Turk, quien con sus consejos acompañó mi trabajo de graduación.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	V
LISTA DE SÍMBOLOS	IX
GLOSARIO	XI
RESUMEN.....	XIII
OBJETIVOS.....	XV
INTRODUCCIÓN	XVII
1. MARCO TEÓRICO.....	1
1.1. Reseña histórica.....	1
1.2. UPS.....	2
1.2.1. Tipologías	3
1.2.1.1. Offline/stand by.....	3
1.2.1.2. Line-Interactive	4
1.2.1.3. Conversión doble/en línea	6
1.2.1.4. Tipología híbrida doble conversión	7
1.2.1.5. Ferro-Resonant.....	8
1.3. PostgreSQL.....	9
1.3.1. Arquitectura de la base de datos	10
1.3.2. Instalación.....	11
1.3.3. Creación de la base de datos	21
1.3.4. Tipos de datos	24
1.3.5. Creación de tablas.....	25
1.3.6. Insertar y seleccionar registros.....	27
1.3.7. Eliminar registros	29
1.3.8. Actualizar registros	29

1.3.9.	Operadores racionales	30
1.3.10.	Operadores Lógicos	31
1.3.11.	Columnas calculadas	31
1.3.12.	Manejo de cadenas	32
1.3.13.	Funciones matemáticas.....	33
1.3.14.	Funciones de fecha y hora	34
1.3.15.	Funciones de agrupamiento	35
1.3.16.	Llave primaria.....	37
1.3.17.	Llave foránea.....	38
1.3.18.	Vistas.....	39
1.3.19.	Gestión de usuarios	40
1.3.20.	Modificación de usuarios	43
1.4.	Procesadores Sparc.....	44
1.4.1.	Historia	45
1.4.2.	Principales características de arquitectura Sparc....	47
1.4.3.	Ventanas de registros	47
1.4.4.	Categoría de instrucciones.....	49
1.4.5.	Direccionamiento en memoria.....	50
2.	CONFIGURACIÓN Y ACTIVACIÓN DEL UPS	53
2.1.	Diagnóstico.....	54
2.1.1.	Procedimiento de instalación y activación	54
2.1.2.	Pruebas	58
3.	HABILITACIÓN DEL SERVIDOR	59
3.1.	Diagnóstico.....	59
3.1.1.	Mantenimiento del hardware	60
3.2.	Selección del sistema operativo.....	61
3.2.1.	Instalación del sistema operativo	65

CONCLUSIONES 75
RECOMENDACIONES 77
BIBLIOGRAFÍA 78

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Funcionamiento Offline/stand by.....	4
2.	Funcionamiento line-interactive.....	6
3.	Arquitectura cliente-servidor.....	10
4.	Pantalla inicial de bienvenida de instalación de PostgreSQL.....	12
5.	Selección del directorio instalación	13
6.	Selección de directorio de datos	14
7.	Selección de contraseña	14
8.	Selección del puerto.....	15
9.	Selección configuración regional.....	16
10.	Iniciar instalación.....	16
11.	Progreso de la instalación	17
12.	Finalización de la instalación.....	17
13.	Menú en Linux.....	18
14.	Menú Windows XP	18
15.	Menú Windows 7	19
16.	Pantalla de psql.....	20
17.	Pantalla PgAdmin.....	20
18.	Consola psql	21
19.	Conectándose por medio de PgAdmin.....	22
20.	Creación de nueva base de datos por medio de PgAdmin	23
21.	Ingresar nombre base datos por medio de PgAdmin	23
22.	Creación de usuarios por PgAdmin.....	43
23.	UPS Powerware 9355.....	53

24.	Retiro de la tapa frontal.....	55
25.	Retiro cubiertas internas	56
26.	Estructura eléctrica del UPS	56
27.	Modulo de los breakers.....	57
28.	Servidor Ultra Enterprise.....	60
29.	Vista lateral del servidor.....	61
30.	Carga de Open Boot	65
31.	Selección del idioma para la instalación	66
32.	Selección de zona horaria	67
33.	Elección de la distribución del teclado.	68
34.	Carga de componentes.....	68
35.	Configuración de red.....	69
36.	Configuración de usuarios y contraseñas.	69
37.	Creación de usuario y contraseña	70
38.	Particionando discos.....	71
39.	Formateo de particiones	71
40.	Instalación de sistema base.....	72
41.	Selección de programas.	72
42.	Instalación del GRUB.....	73

TABLAS

I.	Tipos de datos SQL3 en PostgreSQL	24
II.	Tipos de datos extendidos en PosgreSQL	25
III.	Estructura de registros y campos	25
IV.	Ejemplo de creación de tabla	26
V.	Ejemplo visualizar estructura de la tabla	26
VI.	Ejemplo eliminar tabla	27
VII.	Ejemplo del comando select.....	27

VIII.	Consulta con condición <i>where</i>	28
IX.	Ejemplo insertar datos	28
X.	Ejemplo de delete y truncate	29
XI.	Sintaxis de comando update.....	29
XII.	Operadores racionales.....	30
XIII.	Ejemplo de operadores racionales.....	30
XIV.	Ejemplo operadores lógicos.....	31
XV.	Ejemplo columnas calculadas.....	32
XVI.	Funciones manejo de cadenas	32
XVII.	Ejemplo de concatenación	33
XVIII.	Funciones matemáticas de PostgreSQL.....	34
XIX.	Funciones de fecha y hora.....	35
XX.	Ejemplo función count.....	35
XXI.	Estructura de group by.....	36
XXII.	Ejemplo de función sum.....	36
XXIII.	Ejemplo función avg.....	37
XXIV.	Sintaxis básica de creación de tabla	38
XXV.	Ejemplo de llave foránea	39
XXVI.	Sintaxis básica de creación de vistas	40
XXVII.	Opciones para crear usuarios	41
XXVIII.	Sintaxis modificación de usuarios	44
XXIX.	Versiones de Solares.....	62
XXX.	Distribuciones de Linux.....	64

LISTA DE SÍMBOLOS

Símbolo	Significado
AC	Corriente Alterna
CD	Corriente Directa
kW	Kilovatio
kVa	Kilovoltioamperio
Hz	Herzio
V	Voltaje
W	Vatio

GLOSARIO

<i>Boosting</i>	Aumento continuo de tensión eléctrica.
<i>Boost</i>	Aumento de voltaje.
<i>Buck</i>	Caída de voltaje.
<i>Benchmarks</i>	Proceso de compararse con el líder de la industria.
<i>Offline</i>	Fuera de línea, desconectado.
Linux	Sistema operativo gratuito, modificable.
<i>Proxy</i>	Servidor que provee internet y filtra su contenido.
<i>SQL</i>	Lenguaje estructurado de consulta de base de datos.
<i>Stand by</i>	En espera.
Tics	Tecnologías de la información y comunicación.
<i>UPS</i>	Sistema de alimentación ininterrumpida.
Webex	Software para impartir cursos online.

RESUMEN

El IT Centre of Excellence ITCoE es una institución que tiene como objetivo principal, capacitar a la población en general sobre Tecnologías de la Información y Comunicación (TIC's). Para darle continuidad a dicho objetivo es necesario actualizar y remodelar parte de la infraestructura existente. Por medio de la instalación del sistema de alimentación eléctrica ininterrumpida UPS, la habilitación del servidor SUN Ultra Enterprise 450 y el desarrollo del material para impartir el curso de introducción a PostgreSQL.

La habilitación del UPS tiene como función principal regular el voltaje, para proteger el equipo. Asimismo, brindar las condiciones óptimas para utilizar el equipo de cómputo. Por otro lado, la vida útil del equipo de cómputo se puede reducir significativamente por las sobrecargas y bajas de voltaje.

No obstante, el servidor SUN Ultra Enterprise 450, ya tiene poca vida útil, por lo cual es necesario habilitarlo en la brevedad posible, para aprovechar sus capacidades antes de que lleguen a ser obsoletas.

OBJETIVOS

General

Instalar y configurar el servidor ULTRA Enterprise 450 y activar la unidad UPS Powerware PW9355.

Específicos

1. Gestionar el proceso de configuración y activación del UPS del ITCoE.
2. Adquirir el teclado del servidor SUN modelo Ultra Enterprise 450.
3. Instalar el sistema operativo al servidor SUN de la Escuela de Ciencias y Sistemas que se encuentra en el ITCoE.
4. Apoyar en las tareas administrativas del ITCoE e impartir cursos virtuales por medio de la plataforma Webex.

INTRODUCCIÓN

En consecuencia del convenio suscrito entre la República de la India y la Universidad de San Carlos de Guatemala, en el 2006, se inauguró el IT Centre of Excellence ITCoE, en cual se instalaron dos laboratorios de computo completamente equipados para impartir cursos de capacitación en TIC's. A partir de este acuerdo se donó un UPS para proteger el equipo de cómputo de las variaciones de la corriente eléctrica, por falta de presupuesto no se pudo gestionar el proceso de instalación del UPS y toda la infraestructura quedo desprotegida de las variaciones de la corriente eléctrica, estas variaciones pueden causar daños al equipo de computo.

A la Escuela de Ciencias y Sistemas, se le donó un servidor ULTRA Enterprise 450. El cual no se puede utilizar, debido a que no cuenta con teclado, y no es posible instalarle un sistema operativo sin el teclado, el servidor se destinará para impartir nuevos cursos y ampliar la infraestructura del ITCoE.

Por lo tanto, este proyecto de Ejercicio Profesional Supervisado (EPS) en el ITCoE, tiene como finalidad: instalar el UPS, habilitar el servidor Ultra Enterprise 450 para instalarle el sistema operativo y la base datos PostgreSQL.

1. MARCO TEÓRICO

Básicamente se mencionará el proceso desde sus inicios hasta la actualidad, en el cual se pretende buscar solución para el óptimo manejo de los laboratorios y servidores, promoviendo las mejoras y el buen aprovechamiento de ello.

1.1. Reseña histórica

El Gobierno de la India instaló un Centro de Capacitación en Tecnologías de Información y Comunicación ITCoE el 28 de septiembre del 2006. Dotado por dos laboratorios provistos de 26 computadoras cada uno y una oficina administrativa con 3 computadoras y 2 servidores. Un servidor cuenta con sistema operativo Windows Server 2003, el otro con un sistema operativo Linux y desempeña la función de Proxy. Los laboratorios y oficinas se encuentran ubicados el cuarto nivel del edificio T3 de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala. Actualmente este equipo de cómputo se encuentra desprotegido de sobrecargas eléctricas y fallas de energía, debido a que no cuenta con un sistema eléctrico de respaldo funcional.

El Gobierno de la India también donó un UPS marca Powerware modelo 9355 que proveerá este soporte, pero no se encuentra conectado por los altos costos de asesoría para instalación. Dentro de las donaciones también incluyeron un servidor SUN modelo Ultra Enterprise 450 que pertenece a la Escuela de Ciencias y Sistemas, el cual no es usado por falta de instalación de sistema operativo y periféricos para su utilización.

1.2. UPS

La palabra UPS es acrónimo de Uninterruptible Power Supply. Los UPS son dispositivos electrónicos diseñados para regular y resguardar la corriente eléctrica generada por una descarga de tensión eléctrica, caída de la tensión eléctrica o por la interrupción temporal de la energía eléctrica. Las variaciones de voltaje se clasifican en:

- Alto voltaje: cuando el voltaje en la línea es mayor al que debería ser
- Bajo voltaje: cuando el voltaje de la línea es menor al que debería ser
- Pérdida de energía: cuando se corta la energía o se quema un fusible
- Variaciones en frecuencia: cuando la energía oscila en rangos diferentes a 60 hertz.

Existen diferentes tipos de UPS clasificados por su funcionalidad y se describirán a continuación:

- Un UPS en espera: se mantiene la computadora suministrada de energía de la red eléctrica hasta que detecte un problema, en ese momento en menos de 5 milisegundos enciende un convertidor que alimenta el equipo directamente de una batería. Un convertidor simplemente transforma la corriente directa suministrada por la batería en corriente alterna de 120 volts y 60 hertz.
- En un UPS continuo: la computadora opera directamente de la batería, y esta es recargada continuamente por la red eléctrica. Alguien podría construir un UPS continuo con un cargador de batería, una batería y un convertidor. El cargador de batería produce corriente directa que alimenta a la batería y de esta el convertidor cambia a 120 volts de

corriente alterna. Si la energía falla, la batería la entrega al convertidor, no hay cambio en el suministro de energía, por lo que este tipo de UPS entrega una muy estable alimentación de energía.

1.2.1. Tipologías

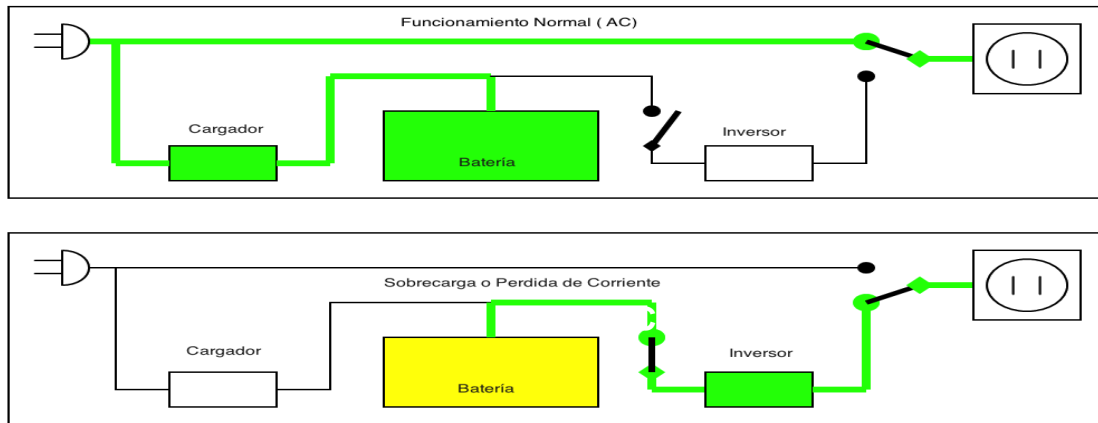
La tipología se refiere a la configuración interna y la forma en la cual se mantienen la regulación y corriente eléctrica.

1.2.1.1. Offline/stand by

Los UPS offline/stand by ofrecen las características más básicas, como regulación de voltaje y batería de corta duración, para proteger al equipo durante una sobrecarga o baja de la corriente eléctrica. El equipo electrónico es conectado directamente al UPS, normalmente cuando el voltaje cae por debajo del nivel requerido el UPS activa su mecanismo interno de inversión de CD-CA. Por consiguiente, suministra energía a partir de la generación, por medio de la bacteria interna de almacenamiento, cuando la corriente eléctrica regresa a la normalidad.

El UPS deja de utilizar la batería automáticamente e inicia el proceso de recarga. Este tipo de UPS fue diseñado para proteger equipos personales y ofrece poca autonomía a la hora de sufrir un corte de energía eléctrica.

Figura 1. **Funcionamiento Offline/stand by**



Fuente: en.wikipedia.org/wiki/Uninterruptible_power_supply. Consulta: 12 de diciembre 2011.

1.2.1.2. **Line-Interactive**

El UPS line-interactive opera de manera similar al UPS stand by. Incluso con una variable y transformador de voltaje. Es un tipo de transformador especial que puede adicionar o sustraer la potencia de la corriente de bobina, para poder así, incrementar o disminuir el campo magnético y la salida de voltaje del transformador.

Este tipo UPS tiene la capacidad de tolerar caídas de baja tensión y sobre tensiones, sin consumir el límite de reserva de la energía de la batería. En caso contrario, lo compensa automáticamente seleccionando el tipo de medición en el auto-transformador. Dependiendo del diseño, cambiando la medida del auto-transformador que puede causar una breve salida y desconectar la energía, éste UPS está equipado con una alarma de pérdida de potencia puede causar chirrido en el momento.

Se ha hecho popular al igual que los UPS más baratos porque toma ventaja de los componentes que ya están incluidos. El transformador principal de 50-60 Hz usado para convertir entre línea de voltaje y voltaje de batería necesita proveer dos tipos, poco más de dos porciones, para convertir el voltaje de salida de la batería (más común con un múltiple de 12 CD) para línea de voltaje, y el segundo para convertir la línea de voltaje en un cargador de voltaje de batería un poco más alto (como el múltiple de 14 V). Igualmente, éste es más fácil para hacer el cambio en la línea de voltaje del lado del transformador, por las más bajas corrientes en ese lado.

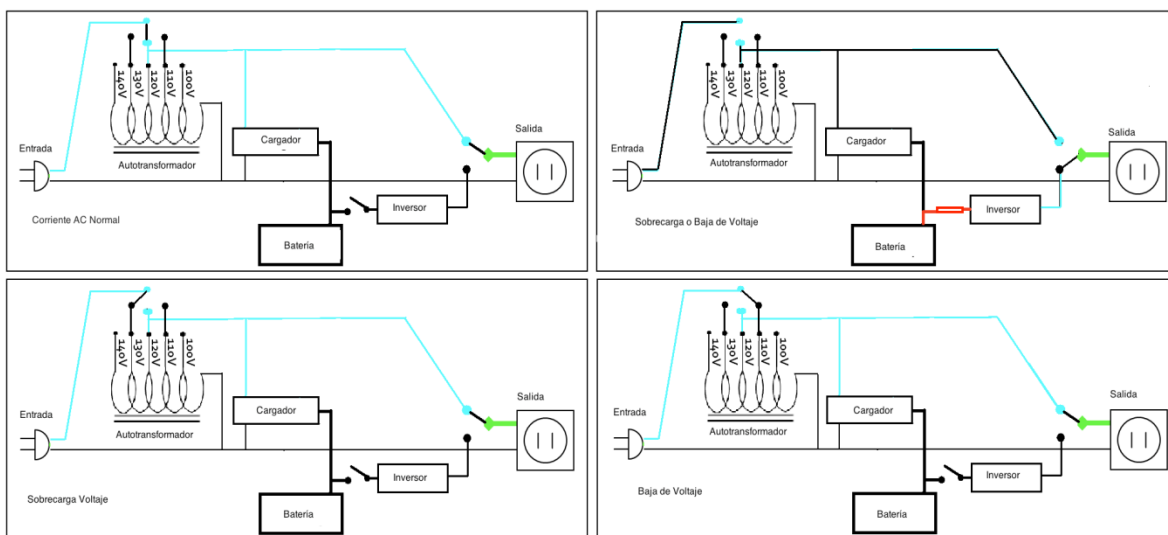
Para incrementar la característica de buck/boost se requiere de todo, son dos cambiadores separados. Al mismo tiempo, la entrada CA puede estar conectada a una de las dos corrientes, mientras puede estar conectada al otro, luego usar el transformador principal de las primeras bobinas como un auto-transformador. La batería puede estar cargada en el sobre voltaje bucking y cuando esta la baja tensión del boosting, la salida del transformador es muy baja para cargar las baterías.

Los auto-transformadores pueden estar diseñados para cubrir el rango de variación de la entrada de los voltajes del cable. Pero, eso requiere más entradas y el incremento de complejidad y costo del UPS. Es común para un auto-transformador que cubra un rango únicamente de 90V a 140V para la energía de 120V, y luego cambiar la batería si el voltaje va mucho más alto o más bajo que el rango.

En el voltaje bajo las condiciones del UPS se usaran más frecuentes que el normal. No obstante, necesitará fluencia más alta de circuito que un dispositivo normal. Por ejemplo, para darle energía a un dispositivo de 1000W en 120V, el UPS marcará 8.32 amperios.

El UPS va brindar 10 amperios para compensar, en caso ocurra una caída de tensión de voltaje de 100 voltios. También, trabaja como una reserva, así en las condiciones de subidas de tensión, el UPS trabajará con menos corriente.

Figura 2. Funcionamiento line-interactive



Fuente: en.wikipedia.org/wiki/Uninterruptible_power_supply. Consulta: 12 de diciembre 2011.

1.2.1.3. Conversión doble/en línea

El UPS en línea es ideal para ambientes en los cuales la insulación eléctrica es necesaria o para equipo que es muy sensible en fluctuaciones de energía.

En cierto modo está previamente reservado para largas instalaciones de 10kW o más. Ahora los avances en la tecnología han permitido que esté funcionando como un dispositivo sea de consumo común, administrando 500W o menos. El costo inicial del UPS puede ser un poco más alto, pero es el costo

total de la obtención, generalmente es más bajo debido a la duración de vida de la batería. UPS online o en línea es necesario cuando la línea de energía trae mucho ruido, especialmente si existen bajas y altas de intensidad, apagones y otras anomalías en el fluido eléctrico.

En el UPS online, las baterías están siempre conectadas al inversor, de esta manera, la transferencia de energía cambia si es necesario. Cuando hay pérdida de energía, el rectificador simplemente, retira del circuito y las baterías se mantienen con la energía estable, y sin cambios. Cuando se restablece la energía, el rectificador reanuda el circuito, llevando la mayor parte de la carga para que las baterías se carguen. Igualmente, la corriente de carga puede estar limitada para prevenir sobrecalentamiento por la alta energía del rectificador de las baterías y la evaporación del electrolito.

La mayor ventaja del UPS online es la habilidad de proveer una pared eléctrica entre la entrada de la energía en uso y la sensibilidad del equipo electrónico. Mientras que el *stand-by* y el *line-interactive* son meramente filtrar la entrada de la utilidad de energía. El *double conversion* UPS provee una capa de insulación de la calidad de problemas de energía. Permite el control de salida de voltaje y la frecuencia sin importar la frecuencia y entrada del voltaje.

1.2.1.4. Tipología híbrida doble conversión

Estos diseños híbridos no tienen una designación oficial, por consiguiente, sus nombres son usados por HP y Eaton que son conversiones de doble demanda. Este estilo de UPS es específico, se dirige hacia la alta eficiencia en aplicaciones mientras que aún mantienen las características y la protección del nivel ofrecido por la conversión doble.

Un híbrido (UPS de doble conversión de demanda) opera como un UPS *off line/stand by*, cuando las condiciones de energía están dentro de ciertas ventanas preestablecidas, le permite al UPS alcanzar una muy alta eficiencia de puntuaciones. Cuando las condiciones de energía fluctúan afuera de las ventanas predefinidas, ocurren cambios de operación y conversión del UPS de online. En modo de doble conversión, el UPS puede ajustarse para las variaciones del voltaje sin tener que usar la energía de la batería, puede filtrar la línea de salida del ruido y controlar la frecuencia.

1.2.1.5. Ferro-Resonant

Unidades que operan en la misma forma que lo hace la unidad de UPS, por consiguiente, están en línea con excepción que el transformador ferro-resonant es usado para filtrar la salida. Este transformador está diseñado para mantener la suficiente energía y cubrir el tiempo entre el cambio de línea de energía y el de energía de batería. Eliminando efectivamente la transferencia del tiempo. Muchos UPS ferro-resonant son del 82 al 88% eficientes (CA / CD-CA) ya que, ofrecen excelente insulación.

El transformador tiene tres bobinas, una para mantener la energía ordinaria principal. Segunda para la energía de la batería. Tercera para cargar la energía AC, una vez el tipo dominante de UPS esté limitada cerca del rango 150 kVA.

Estas unidades aún se mantienen y se utilizan principalmente en cuestiones industriales (aceite, gas, químicos, petroquímicos, servicios públicos e industrias fuertes de mercado). Debido a la naturaleza robusta del UPS. Muchos UPS ferro-resonant utilizan tecnología ferro controlada que no puede interactuar con el factor de corriente y equipo de energía.

1.3. PostgreSQL

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos, publicado bajo la licencia BSD. Comenzó como un proyecto de la Universidad Berkeley en 1982, denominado Ingres. Fue uno de los primeros intentos de implementación de un motor de base de datos relacional. El proyecto fue liderado por Michael Stone Braker, que después de haber trabajado un largo tiempo en Ingres y de incursionar comercialmente con el mismo, decidió volver a la universidad en 1985 para trabajar en un nuevo proyecto, basado sobre la experiencia recopilada en el proyecto Ingres, dicho proyecto fue llamado post-ingres o simplemente POSTGRES.

El proyecto evolucionó continuamente, a mediados de los años noventa le añadieron soporte a SQL. Continúo su evolución hasta que se liberó en el año de 1997 PostgreSQL 6.0. Con el desarrollo del Internet, muchos investigadores se unieron al proyecto, conformando una comunidad que continúa mejorando las características de PostgreSQL, realizando liberaciones y lanzamientos menores de reparación de bugs anualmente.

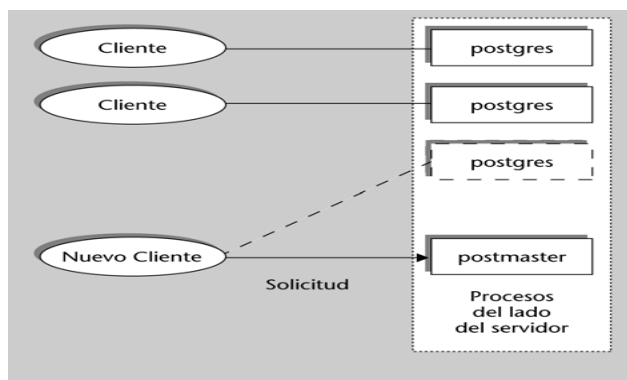
Postgres está considerado como la base de datos de código abierto más avanzada del mundo. Postgres proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales tales como DB2 u Oracle. Ofreciendo control de concurrencia multi-versión y soportando la sintaxis SQL. Además cuenta también con un amplio conjunto de API's para la mayoría de lenguajes de programación (incluyendo los más usados, tales como C, C++, Java, PHP, perl, tcl y python). En este manual se encuentran las funciones básicas de Postgres con las cuales el lector será capaz de utilizar esta potente base de datos de manera básica.

1.3.1. Arquitectura de la base de datos

Postgres está basado en una arquitectura cliente-servidor. Un proceso servidor postgres puede atender exclusivamente a un solo cliente; es decir, hacen falta tantos procesos de servidores postgres como clientes.

El proceso postmaster es el encargado de ejecutar un nuevo servidor para cada cliente que solicite una conexión.

Figura 3. **Arquitectura cliente-servidor**



Fuente: <http://www.dataprix.com/72-arquitectura-postgresql>. Consulta: 03 de diciembre 2011.

Se llama sitio al equipo anfitrión (host) que almacena un conjunto de base de datos Postgres. En un sitio se ejecuta solamente un proceso postmaster y múltiples procesos Postgres.

Los clientes pueden ejecutarse en el mismo sitio o en equipos remotos conectados por TCP/IP.

1.3.2. Instalación

Postgres se encuentra disponible para el sistema operativo Windows, Linux y MacOS, en el sitio oficial www.postgresql.org/download, el instalador es universal.

Para GNU/Linux, se puede obtener de los repositorios de cada distribución si se desea, en el caso de Debian, Ubuntu o cualquier diestro que utilice apt ejecutamos el siguiente comando:

```
“apt-get install postgresql-8.3 postgresql-client-8.3 postgresql-contrib-8.3”
```

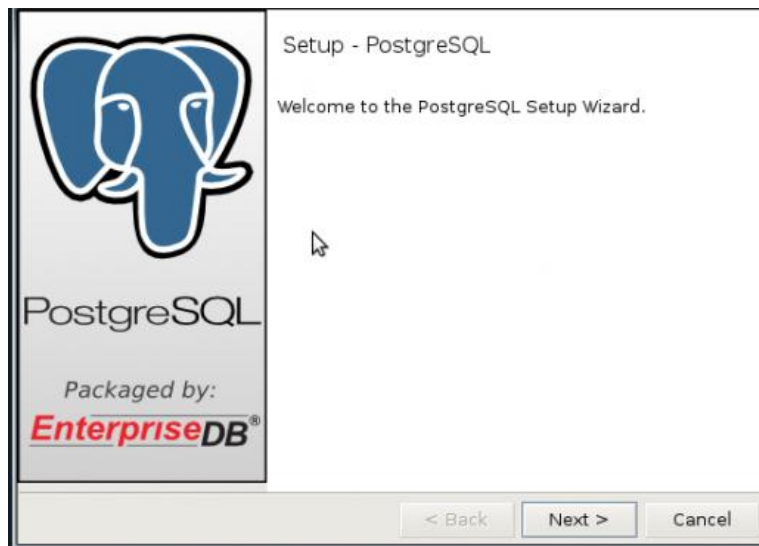
En distribuciones que utilicen los paquetes RPM, se ejecuta el siguiente comando:

```
“yum install postgresql-8.1.9-1.el5.i386.rpm postgresql-contrib-8.1.9-1.el5.i386.rpm postgresql-libs-8.1.9-1.el5.i386.rpm ”
```

Si se baja el instalador de la página oficial antes mencionada, el procedimiento será igual para cualquier sistema operativo, se ejecuta el paquete de descarga y aparece el programa de instalación asistida.

La pantalla inicial muestra la bienvenida al instalador de PostgreSQL. A partir de ahora tendrá que pulsar Next cada vez que se quiera avanzar al siguiente paso.

Figura 4. **Pantalla inicial de bienvenida de instalación de PostgreSQL**



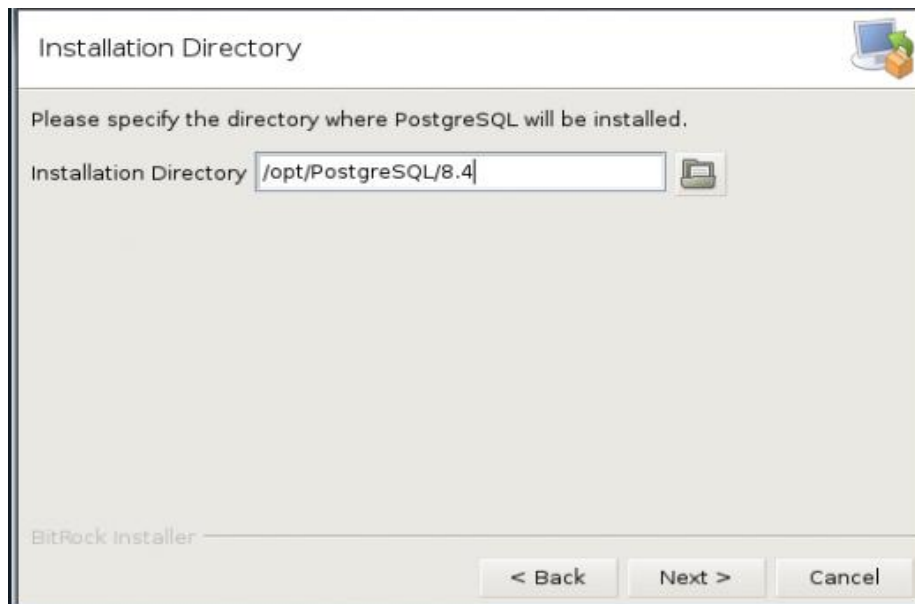
Fuente: elaboración propia.

En el siguiente paso se tendrá que definir el directorio donde va a instalar todos los programas que vienen con esta versión de PostgreSQL, generalmente incluye el manejador de base de datos, el programa para conectarse en consola y un administrador gráfico.

En el presente caso, se utilizará el valor por defecto, dependiendo el sistema operativo se listan las ubicaciones por defecto del programa:

- Windows XP, se ubicará en "C:\Program Files\PostgreSQL\8.4".
- Windows 7, se ubicará en "C:\Program Files (x86)\PostgreSQL\8.4".
- Linux, se ubicará en "/opt/PostgreSQL/8.4".

Figura 5. Selección del directorio instalación



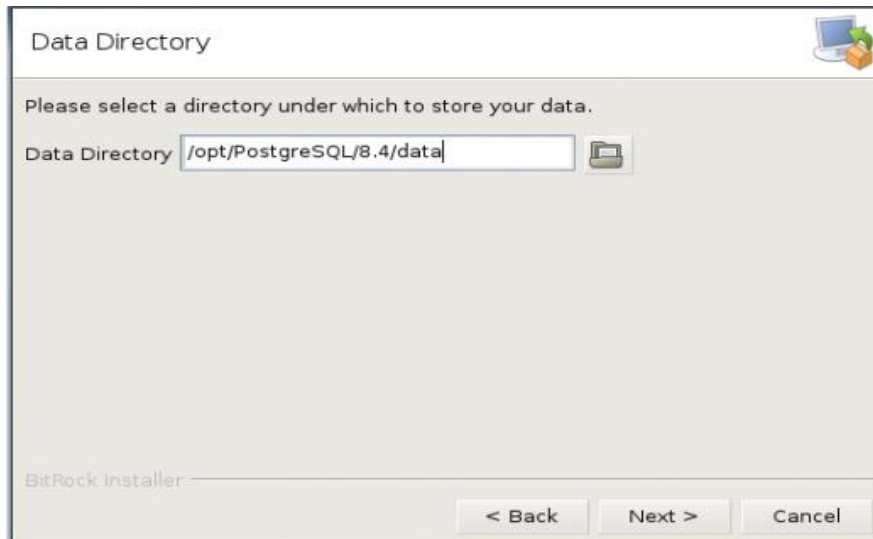
Fuente: elaboración propia.

Una vez se ha escrito el directorio que se quiere utilizar para instalar el programa, se pasa a lo siguiente.

En la pantalla a continuación hay que definir el directorio de datos en donde se crearán las bases de datos. De nuevo, en el siguiente caso se utilizará el valor por defecto, listados a continuación:

- Windows XP, se ubicará en "C:\Program Files\PostgreSQL\8.4\data".
- Windows 7, se ubicará en "C:\Program Files (x86)\PostgreSQL\8.4\data".
- Linux, se ubicará en "/opt/PostgreSQL/8.4/data".

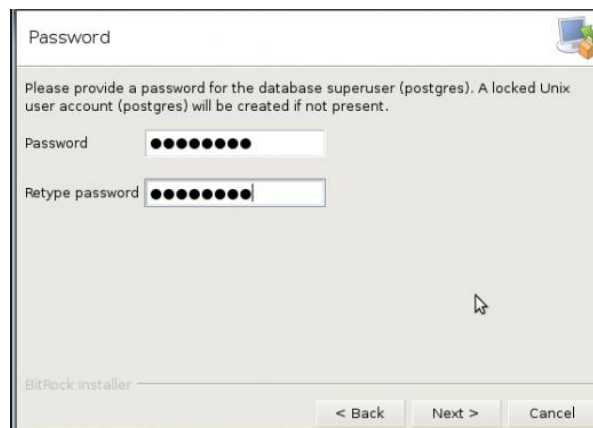
Figura 6. **Selección de directorio de datos**



Fuente: elaboración propia.

En el siguiente paso se tendrá que definir una clave de acceso para el usuario administrador de la base de datos PostgreSQL. Esta clave es totalmente independiente de la clave de la cuenta de servicio postgres en el sistema operativo.

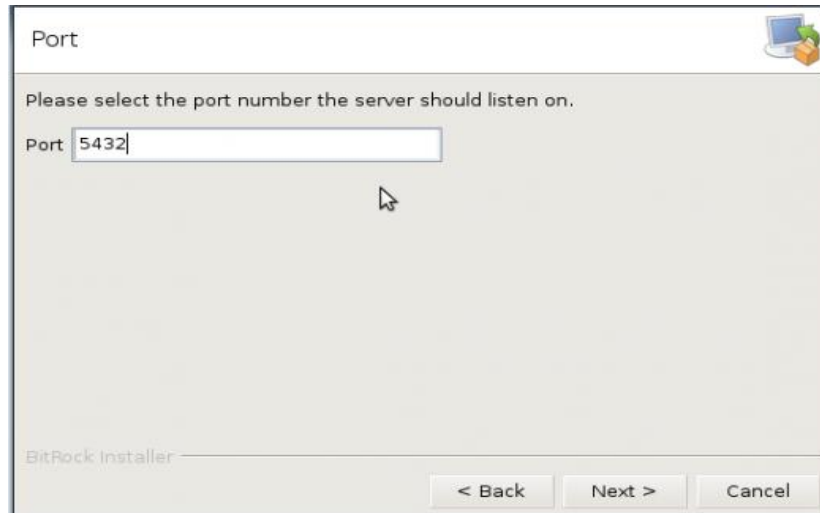
Figura 7. **Selección de contraseña**



Fuente: elaboración propia.

Paso siguiente. Hay que especificar el puerto que PostgreSQL, utilizará para escuchar por conexiones. Se deja el valor por defecto 5432.

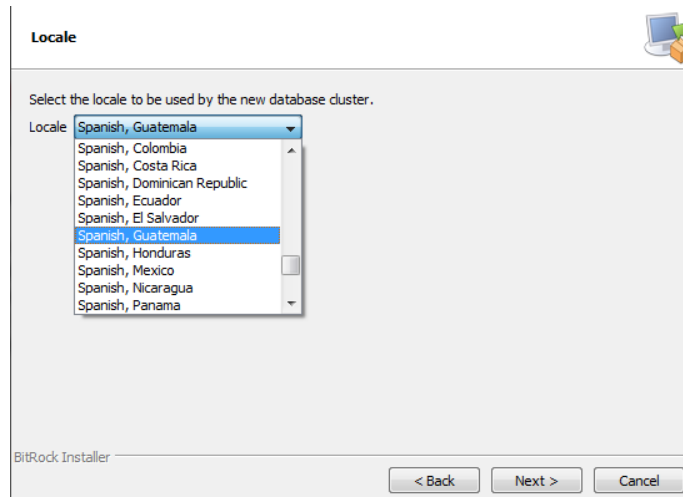
Figura 8. **Selección del puerto**



Fuente: elaboración propia.

Paso siguiente. Se selecciona el conjunto de caracteres alfabéticos y preferencias culturales para la base de datos, esto comprende el alfabeto de la región en donde pertenece el usuario y el formato de los números.

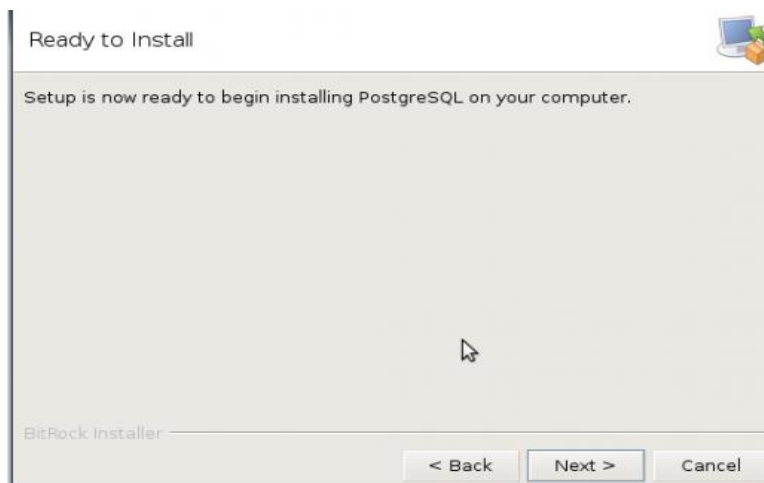
Figura 9. Selección configuración regional



Fuente: elaboración propia.

Paso siguiente. Como ya se terminó la configuración básica, se procede a instalar la aplicación.

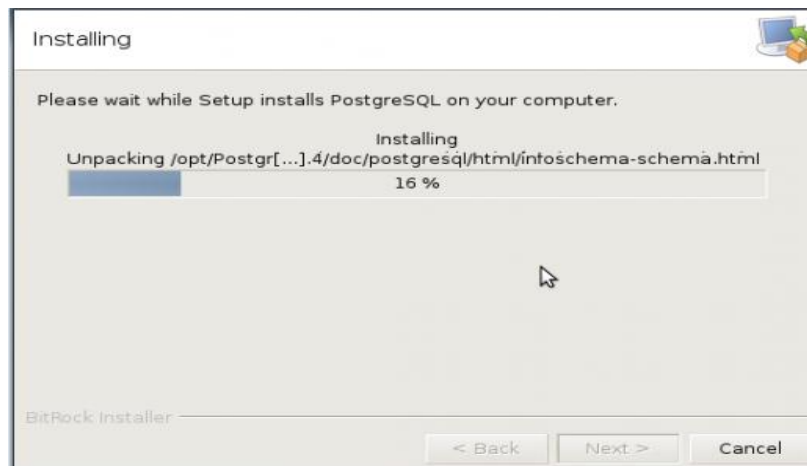
Figura 10. Iniciar instalación



Fuente: elaboración propia.

Pulsar por última vez *Next* y esperar hasta que el programa termine de instalar todo.

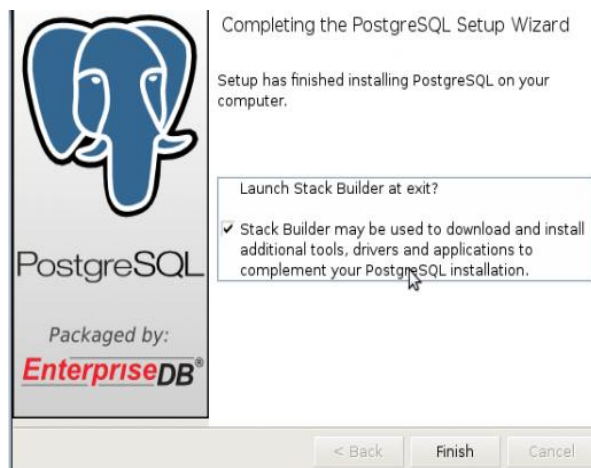
Figura 11. **Progreso de la instalación**



Fuente: elaboración propia.

Una vez terminada la instalación, se puede salir del instalador pulsando *Finish*.

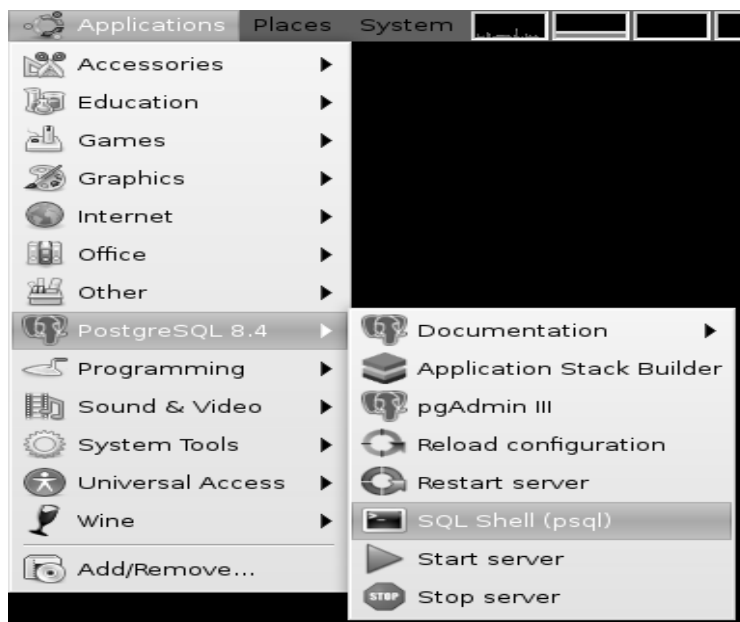
Figura 12. **Finalización de la instalación**



Fuente: elaboración propia.

Al finalizar el procedimiento de instalación, se podrá acceder a la documentación, a diversos programas clientes, a la administración básica. El menú variará dependiendo del sistema operativo que utilice el usuario.

Figura 13. Menú en Linux



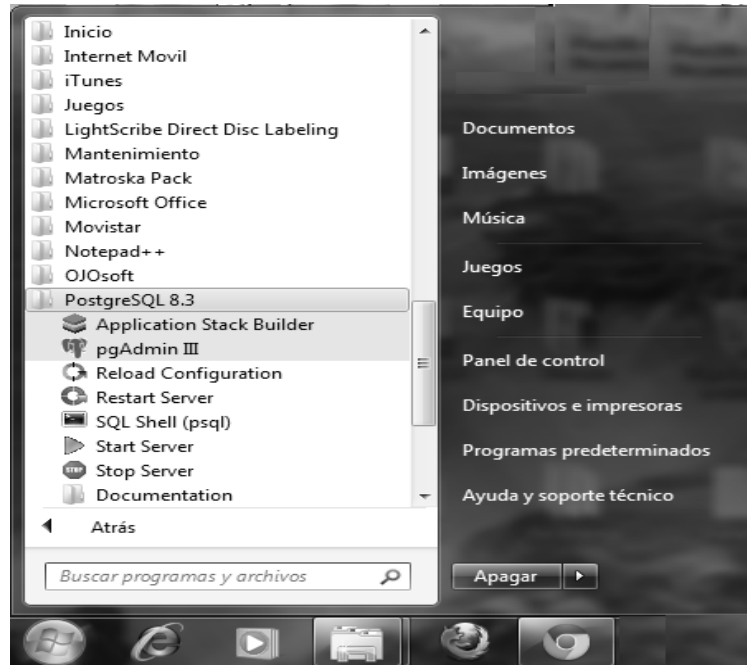
Fuente: elaboración propia.

Figura 14. Menú Windows XP



Fuente: elaboración propia.

Figura 15. Menú Windows 7



Fuente: elaboración propia.

Los clientes (psql y PgAdminIII) en GNU/Linux, en Windows XP y en Windows 7; se acceden de la misma manera y tienen el mismo aspecto, dando click sobre el icono de psql o PgAdmin III.

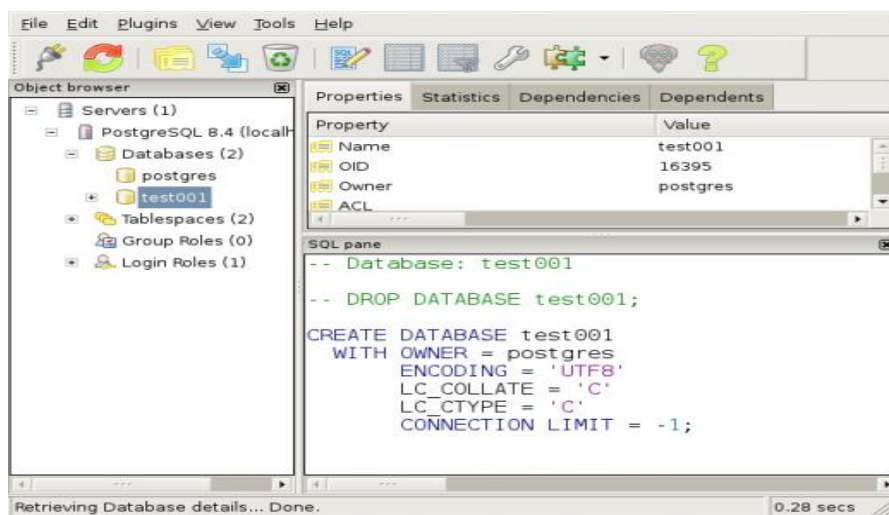
Generalmente solo varia el marco de las pantallas en donde se encuentran los botones de minimizar, maximizar y cerrar, las variaciones se deben a las configuraciones de los temas del sistema operativo.

Figura 16. Pantalla de psql



Fuente: elaboración propia.

Figura 17. Pantalla PgAdmin



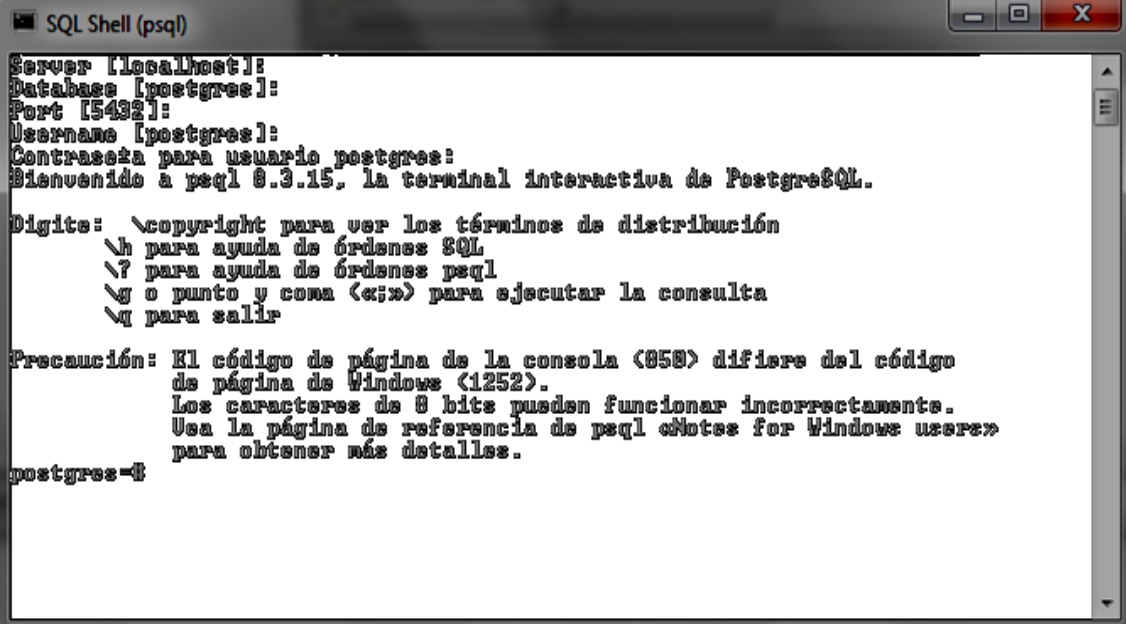
Fuente: elaboración propia.

La instalación creará por defecto el usuario Postgres y la base de datos postgres que es la base de datos del sistema.

1.3.3. Creación de la base de datos

Ingresar a la consola psql que se encuentra en el menú éste mismo preguntará el nombre del servidor, la base de datos, el puerto, el usuario y por último la contraseña, la consola muestra los valores por defecto y se presiona la tecla *enter* para que los introduzca automáticamente.

Figura 18. Consola psql



```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Contraseña para usuario postgres:
Bienvenido a psql 0.3.15, la terminal interactiva de PostgreSQL.

Digite: \copyright para ver los términos de distribución
        \h para ayuda de órdenes SQL
        \? para ayuda de órdenes psql
        \g o punto y coma (<ej>) para ejecutar la consulta
        \q para salir

Precaución: El código de página de la consola (050) difiere del código
de página de Windows (1252).
Los caracteres de 8 bits pueden funcionar incorrectamente.
Use la página de referencia de psql «Notes for Windows users»
para obtener más detalles.

postgres->
```

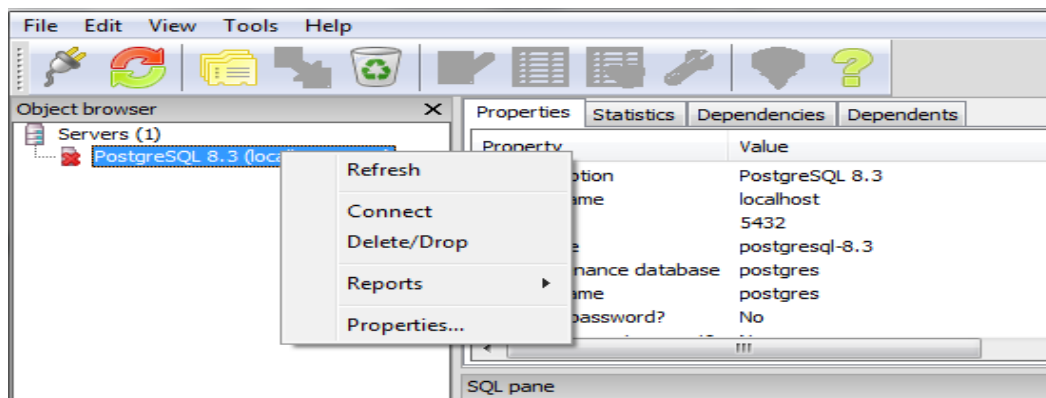
Fuente: elaboración propia.

Ingresar el siguiente comando en la consola psql “Create database foo with owner=postgres;”, el comando creará la base datos, para este ejemplo se

creará la base de datos con nombre foo en la que el usuario postgres será el dueño de la base de datos.

En el caso de que se desee utilizar la herramienta gráfica PgAdmin, ingresará a ella y se conectará a la base de datos, dando *click* derecho sobre la base de datos y seleccionamos *Connect*.

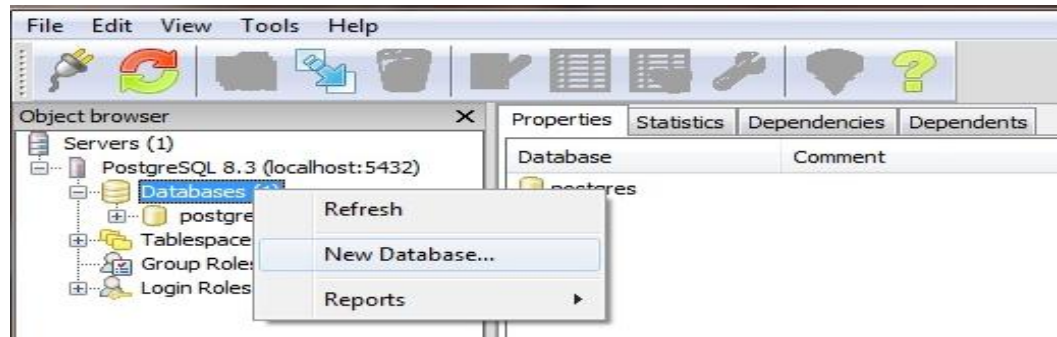
Figura 19. Conectándose por medio de PgAdmin



Fuente: elaboración propia.

Después de conectar, dar *click* derecho sobre *databases* y seleccionar *New Database*.

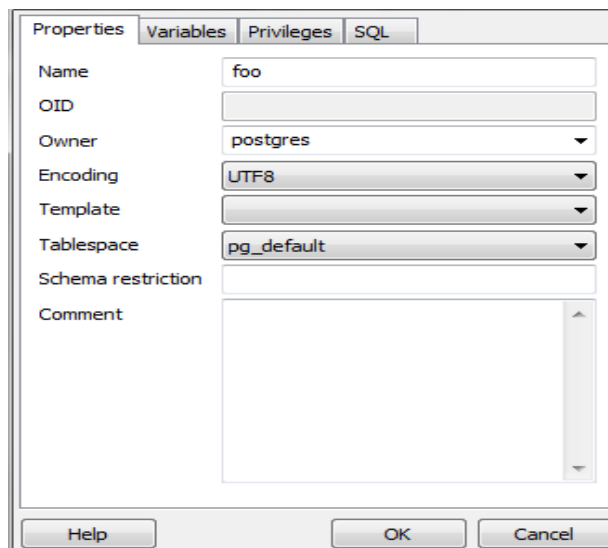
Figura 20. Creación de nueva base de datos por medio de PgAdmin



Fuente: elaboración propia

Luego aparecerá un cuadro de diálogo donde se introduce los datos necesarios para crear dicha base de datos.

Figura 21. Ingresar nombre base datos por medio de PgAdmin



Fuente: elaboración propia.

1.3.4. Tipos de datos

Como todos los manejadores de bases de datos, PostgreSQL implementa los tipos de datos definidos para el estándar SQL3 y aumenta algunos otros. El tipo de valor de un dato define y limita el las operaciones que se pueden hacer sobre él. A continuación se presenta una tabla con los tipos de datos estándar definidos en SQL3 y su equivalente en PostgreSQL:

Tabla I. Tipos de datos SQL3 en PostgreSQL

Postgres	SQL3	Descripción
Bool	Boolean	valor lógico o booleano (true/false)
char(n)	character(n)	cadena de caracteres de tamaño fijo
Date	Date	fecha (sin hora)
float4/8	float(86#86)	número de punto flotante con precisión 86#86
float8	real, doublé precisión	número de punto flotante de doble precisión
int2	Smallint	entero de dos bytes con signo
int4	int, integer	entero de cuatro bytes con signo
int4	decimal(87#87)	número exacto con 88#88
int4	numeric(87#87)	número exacto con 89#89
Money	decimal(9,2)	cantidad monetaria
Time	Time	hora en horas, minutos, segundos y centésimas
Timespan	Interval	intervalo de tiempo
Timestamp	timestampwith time zone	fecha y hora con zonificación
varchar(n)	Carácter varying(n)	cadena de caracteres de tamaño variable

Fuente: elaboración propia

Tabla II. **Tipos de datos extendidos en PostgreSQL**

Tipo	Descripción
Box	caja rectangular en el plano
Cidr	dirección de red o de <i>host</i> en IP versión 4
Circle	círculo en el plano
Inet	dirección de red o de <i>host</i> en IP versión 4
int8	entero de ocho bytes con signo
Line	línea infinita en el plano
Lseg	segmento de línea en el plano
Path	trayectoria geométrica, abierta o cerrada, en el plano
Point	punto geométrico en el plano
Polygon	trayectoria geométrica cerrada en el plano
Serial	identificador numérico único

Fuente: elaboración propia.

1.3.5. Creación de tablas

Las tablas son estructuras de datos que organizan los datos en columnas y filas, cada columna es un campo (o atributo) y cada fila, un registro.

Tabla III. **Estructura de registros y campos**

	Campo 1	Campo 2	Campo N
Registro 1			
Registro N			

Fuente: elaboración propia.

Para crear una tabla se utiliza el comando “create table” acompañado de la lista de los campos y su tipo de dato, en la siguiente tabla se muestra un ejemplo de la creación de una tabla con nombre “usuario”, en la que se almacenara el nombre y la clave de cada usuario.

Tabla IV. **Ejemplo de creación de tabla**

```
create table usuario( nombre varchar(30), clave varchar(10));
```

Fuente: elaboración propia.

Para ver la estructura de una tabla se puede utilizar el comando *select* en la tabla siguiente se coloca el comando y el resultado que devuelve la terminar psql.

Tabla V. **Ejemplo visualizar estructura de la tabla**

Comando			
Select Table_name, column_name, udt_name, character_maximum_length From Information_schema.columns WHERE table_name = 'usuario';			
Resultado			
table_name	column_name	udt_name	character_maximum_length
usuario	clave	varchar	10
usuario	nombre	varchar	30

Fuente: elaboración propia.

Para eliminar una tabla se utiliza el comando “*drop table*” junto al nombre de la tabla que se desea eliminar, en la siguiente tabla se presenta el ejemplo para eliminar la tabla creada anteriormente.

Tabla VI. **Ejemplo eliminar tabla**

```
Drop table Usuario;
```

Fuente: elaboración propia.

Antes de correr el comando *drop*, es necesario asegurar que no existan referencias a la tabla que se desea eliminar.

1.3.6. **Insertar y seleccionar registros**

El comando *select* nos permite consultar la información que se encuentra almacenada dentro de una tabla, el comando nos devuelve un conjunto de registros, teniendo en cuenta que un registro es una fila de la tabla que contiene los datos. Cada registro esta compuesto de campos que almacenan datos, los datos representan las columnas (campo) de la tabla. La tabla menciona usuarios que consta de 2 campos, nombre y clave. En la siguiente tabla se muestra un ejemplo de la sintaxis básica del comando *select*.

Tabla VII. **Ejemplo del comando select**

```
Select nombre, clave from usuario;
```

Fuente: elaboración propia.

El comando *select* puede ser mas especifico al utilizar predicados, los predicados son condiciones o filtros por medio de los cuales se restringe el resultado de la consulta, en la siguiente tabla se muestra un ejemplo de la condición *where*, en el ejemplo se buscaran los usuarios que tengan en un nombre igual a caaguila.

Tabla VIII. **Consulta con condición *where***

```
Select clave from usuario where nombre='caaguila'
```

Fuente: elaboración propia.

El comando *select* puede mostrar toda la información de una tabla al colocar asterisco (*) en lugar de listar los campos.

El comando *insert* se utiliza para almacenar los datos dentro de la tabla, la sintaxis se compone por el comando, el nombre de la tabla, el listado de los campos y los valores que deseamos insertar, en la siguiente tabla se muestra un ejemplo de inserción de datos.

Tabla IX. **Ejemplo insertar datos**

```
Insert into usuario (nombre, clave) values ('caaguila','Xx4ll');
```

Fuente: elaboración propia.

1.3.7. Eliminar registros

Para eliminar registros se utiliza el comando *delete*, seguido de la palabra reservada *from*, el nombre de la tabla y las condiciones en los casos que aplique. Si se desea eliminar todos los registros, podemos utilizar el comando *truncate*, en la siguiente tabla se presenta ejemplos de los comandos mencionados.

Tabla X. **Ejemplo de delete y truncate**

Ejemplo delete:
Delete from usuario where nombre = 'caaguila';
Ejemplo truncate:
Truncate table usuario;

Fuente: elaboración propia.

1.3.8. Actualizar registros

La actualización, es el proceso de modificar el valor de un campo previamente almacenado, se realiza por medio del comando *update*, seguido del nombre de la tabla, la palabra reservada *set*, el listado de campos que deseamos actualizar y en los casos que aplique condiciones. En la siguiente tabla se muestra la sintaxis básica y un ejemplo de actualización de registros.

Tabla XI. **Sintaxis de comando update**

Sintaxis básica:
Update nombre_tabla set Nombre_columna = valor_dato Where predicados;
Ejemplo:
Update usuario set clave='12345' where usuario='caaguila';

Fuente: elaboración propia.

1.3.9. Operadores racionales

Los operadores racionales nos permiten delimitar de mejor manera los datos que deseamos consultar, borrar o actualizar, en la siguiente tabla se presenta el listado de los operadores racionales validos en PosgreSQL.

Tabla XII. **Operadores racionales**

=	Igual
<>	Distinto
>	mayor
<	Menor
>=	mayor o igual
<=	menor o igual

Fuente: elaboración propia.

Se tiene una tabla en la que se almacena el nombre del empleado y su salario y queremos realizar una consulta en la que queremos obtener los datos de los empleados que tengan un salario mayor o igual a mil quinientos, en la tabla siguiente se muestra este ejemplo:

Tabla XIII. **Ejemplo de operadores racionales**

```
Select nombre_empleado, salario From empleado Where salario >= 1500;
```

Fuente: elaboración propia.

1.3.10. Operadores Lógicos

Los operadores lógicos nos permiten evaluar diferentes condiciones basadas en la lógica y no en la aritmética, a continuación se listan los operadores lógicos validos en PostgreSQL.

- *And*, significa "y", ambos elementos de comparación deben coincidir.
- *Or*, significa "y/o", uno de los elementos de comparación debe coincidir
- *Not*, significa "no", invierte el resultado
- *Is null*, significa la ausencia de valor
- *Between*, trabajan con intervalos de valores.
- *In*, compara si el valor de un campo esta incluido en una lista de valores.

A continuación se presenta en la siguiente tabla un ejemplo en el que buscamos los empleados con salario entre Q1,000 y Q2,000.

Tabla XIV. **Ejemplo operadores lógicos**

```
Select empleados where between salario between 1000 and 2000;
```

Fuente: elaboración propia.

1.3.11. Columnas calculadas

Los operadores aritméticos (+, -, *, /, %) permiten realizar cálculos con valores numéricos. Es posible obtener salidas en las cuales una columna sea el resultado de un cálculo y no un campo de una tabla.

En la siguiente tabla se muestra un ejemplo en el que se desea calcular el aumento de un 20% del salario de los empleados.

Tabla XV. **Ejemplo columnas calculadas**

Select Nombre_empleado, salario, (salario*0.20) as aumento From Empleado;
El resultado será: cristian, 1000, 200

Fuente: elaboración propia.

1.3.12. Manejo de cadenas

En muchas ocasiones se almacenan textos en los campos de las tablas, por eso es necesario utilizar algunas funciones que nos permiten manipular los datos.

Tabla XVI. **Funciones manejo de cadenas**

Función	Descripción
Char_length(string)	Retorna la longitud del texto.
Upper(string)	Retorna el texto en mayúsculas.
Lower(string)	Retorna el texto en minúsculas.
Position(string in string)	Retorna la posición de un string dentro de otro.
Substring (string [fromint] [forint])	Retorna un substring, indica la posición inicial y la cantidad de caracteres a extraer desde dicha posición.
Trim (string)	Elimina caracteres en blanco de una cadena.
Ltrim (string,string)	Elimina los caracteres de la izquierda según el dato del segundo parámetro de la función.
Substr (text,int,int)	Retorna una subcadena a partir de la posición que se le indicó en el segundo parámetro hasta la posición indicada en el tercer parámetro.

Continuación de tabla XVI

Lpad (text,int,text)	Rellena de caracteres por la izquierda. El tamaño total de campo es indicado por el segundo parámetro y el texto a insertar se indica en el tercer parámetro.
Rpad (text,int,text)	Rellena de caracteres por la derecha. El tamaño total de campo es indicado por el segundo parámetro y el texto a insertar se indica en el tercero.

Fuente: elaboración propia.

Entre los operadores mas utilizados para el manejo de cadenas tenemos el operador de concatenación “ || ”, en la siguiente tabla se presenta un ejemplo para concatenar el nombre y apellido de una tabla, separado por una coma.

Tabla XVII. **Ejemplo de concatenación**

Select (nombre ', ' apellido) From Empleado; El resultado del comando será: cristian,aguilar

Fuente: elaboración propia.

1.3.13. **Funciones matemáticas**

Las funciones matemáticas realizan operaciones con expresiones numéricas y retornan un resultado, operan con tipos de datos numéricos. En la siguiente tabla se listan las funciones matemáticas de PostgreSQL. En la siguiente tabla se listan, las principales funciones matemáticas.

Tabla XVIII. **Funciones matemáticas de PostgreSQL**

Función	Descripción
abs(x)	Retorna el valor absoluto.
Cbrt(x)	Retorna la raíz cubica.
Sqrt(x)	Retorna la raíz cuadrada.
Ceiling(x)	Retorna el número redondeado hacia arriba.
Floor(x)	Retorna el número redondeado hacia abajo.
Power(x,y)	Retorna el valor de x elevado a la y potencia.
Round(x)	Retorna el número redondeado al valor más próximo.
Sign(x)	Retorna 1 si el valor es positivo, 0 si es cero y -1 si es negativo.
Mod(x,y)	Retorna el resto de dividir x con respecto a y.
Pi()	Retorna el valor de pi.
Random()	Retorna un valor aleatorio
Trunc(x)	Retorna la parte entera del valor de x.
Trunc(x,y)	Retorna la parte entera x junto con su parte decimal del tamaño y
Sin(x)	Retorna el seno de x.
Cos(x)	Retorna el coseno de x.
Tan(x)	Retorna la tangente de x.

Fuente: elaboración propia.

1.3.14. Funciones de fecha y hora

Las funciones de fecha y hora tienen gran importancia, a la hora de realizar consultas por lo general se comparan fechas para realizar calculos.

En la siguiente tabla se listan las principales funciones para el manejo de fecha y hora de PostgreSQL.

Tabla XIX. **Funciones de fecha y hora**

Función	Descripción
Current_date	Retorna la fecha actual.
Current_time	Retorna la hora actual con la zona horaria.
Current_timestamp	Retorna la fecha y la hora actual con la zona horaria.
Extract (val from timestamp)	Retorna el valor del parámetro val (año, mes, día) del valor del timestamp.

Fuente: elaboración propia.

1.3.15. **Funciones de agrupamiento**

En PostgreSQL existen funciones que permiten contar registros, calcular sumas, promedios, obtener valores máximos y mínimos. Estas funciones se denominan funciones de agregado y operan sobre un conjunto de valores (registros), no con datos individuales y devuelven un único valor.

La tabla usuario contiene muchos registros. Para averiguar la cantidad sin necesidad de contarlos manualmente se usa la función *count* en la tabla siguiente se presenta el ejemplo de como utilizar la función mencionada.

Tabla XX. **Ejemplo función count**

```
Select count(*) from usuario;
```

Fuente: elaboración propia.

La función *count* cuenta la cantidad de registros de una tabla, incluyendo los que tienen valor nulo. Se pueden usar en una instrucción *select* y combinarlas con la cláusula *group by* que nos permite juntar registros de un mismo tipo, la sintaxis básica de esta función se describe en la siguiente tabla.

Tabla XXI. **Estructura de group by**

Select	Column1, column2, ... column_n, función_de_agrupamiento FROM tables
Where	Predicado
Group by	Column1, column2, ... column_n;

Fuente: elaboración propia.

Todas estas funciones retornan *null* si ningún registro cumple con la condición del *where*, excepto *count* que en tal caso retorna cero. El tipo de dato del campo determina las funciones que se pueden emplear con ellas. Las relaciones entre las funciones de agrupamiento y los tipos de datos es la siguiente:

La función *sum()* retorna la suma de los valores que contiene el campo especificado. Si se quiere calcular el valor de la nomina de la tabla empleados se sumarán todos los salarios, en la siguiente tabla se presenta el ejemplo .

Tabla XXII. **Ejemplo de función sum**

Select sum(salario) from empleado;

Fuente: elaboración propia.

Para averiguar el valor máximo o mínimo de un campo se usan las funciones max() y min() respectivamente.

La función avg() retorna el valor promedio de los valores del campo especificado. Si se quiere saber el salario promedio de la tabla empleado, aplica la función avg al campo del salario, el ejemplo se presenta en la siguiente tabla.

Tabla XXIII. **Ejemplo función avg**

```
Select avg(salario) from empleado;
```

Fuente: elaboración propia.

1.3.16. Llave primaria

En el diseño de bases de datos relacionales, se llama llave primaria a un campo o a una combinación de campos que identifica de forma única a cada fila de una tabla. Una llave primaria comprende de esta manera una columna o conjunto de columnas. No puede haber dos filas en una tabla que tengan la misma llave primaria.

Una llave primaria debe identificar unívocamente a todas las posibles filas de una tabla y no solo a las filas que se encuentran en un momento determinado.

Si se tiene la tabla usuario, el nombre de cada usuario puede establecerse como llave primaria, es un valor que no se repite; puede haber usuarios con

igual llave, pero su nombre de usuario será siempre diferente. La sintaxis básica se presenta en la siguiente tabla.

Tabla XXIV. **Sintaxis básica de creación de tabla**

Sintaxis básica:
Create table Nombre_tabla (Nombre_campo Tipo_dato, primary key (nombre_campo));
Ejemplo:
Create table Usuario(Nombre varchar(20), Clave varchar(10),Primary key (nombre));

Fuente: elaboración propia.

Una tabla sólo puede tener una llave primaria. Cualquiera puede ser llave primaria, sólo debe cumplir con un requisito, que sus valores no se repitan ni sean nulos. Por ello, al definir un campo como llave primaria, automáticamente PostgreSQL lo convierte a *not null*.

Luego de haber establecido un campo como llave primaria, al ingresar los registros, PostgreSQL controla que los valores para el campo establecido como llave primaria no estén repetidos en la tabla; si estuviesen repetidos, muestra un mensaje de error y la inserción no se realiza.

1.3.17. Llave foránea

Un campo que no es llave primaria en una tabla y sirve para enlazar sus valores con otra tabla en la cual es llave primaria se denomina llave foránea, externa o ajena.

Las llaves foráneas y las llaves primarias deben ser del mismo tipo para poder enlazarse. Si se modifica una, se debe modificar la otra para que los valores se correspondan.

En la siguiente tabla se muestra el ejemplo de llave foranea, en la que almacenamos los correos electrónicos de que pertenecen usuarios, tendrá como llave foránea el nombre del usuario, de la tabla usuario.

Tabla XXV. **Ejemplo de llave foránea**

```
Create table correo ( Usuario Varchar(20),  
correo varchar(20),  
foreign key (usuario)  
References  
Usuario (nombre) );
```

Fuente: elaboración propia.

1.3.18. Vistas

Una vista es una alternativa para mostrar datos de varias tablas. Una vista es como una tabla virtual que almacena una consulta. Los datos accesibles a través de la vista no están almacenados en la base de datos como un objeto. Las vistas permiten:

- Ocultar información: permitiendo el acceso a algunos datos y manteniendo oculto el resto de la información que no se incluye en la vista. El usuario solo puede consultar la vista.
- Simplificar la administración de los permisos de usuario: se pueden dar al usuario permisos para que solamente pueda acceder a los datos a través

de vistas, en lugar de concederle permisos para acceder a ciertos campos, así se protegen las tablas base de cambios en su estructura.

- Mejorar el rendimiento: se puede evitar teclear instrucciones repetidamente almacenando en una vista el resultado de una consulta compleja que incluya información de varias tablas.

Podrá crear vistas con: un subconjunto de registros y campos de una tabla; una unión de varias tablas; una combinación de varias tablas; un resumen estadístico de una tabla; un subconjunto de otra vista, combinación de vistas y tablas. La sintaxis básica para crear una vista se muestra en la siguiente tabla.

Tabla XXVI. **Sintaxis básica de creación de vistas**

Sintaxis básica:
Create view Nombre_vista as Sentencia_select;
Ejemplo:
Create view usuario_correo as Select usuario.nombre,usuario.clave,correo.correo From usuario, correo Where usuario.nombre = correo.usuario

Fuente: elaboración propia.

1.3.19. Gestión de usuarios

En PostgreSQL los usuarios pueden tener capacidades de crear bases de datos y más usuarios, dependiendo de los permisos que se le definan a la hora de su creación. En PostgreSQL se utilizará el comando create rol y create user.

Create user es un alias de create role, la diferencia se da que el comando create user, el login se asume por defecto mientras que el comando create role asume por defecto el no login, el comando para crear usuarios se muestran en la tabla siguiente.

Tabla XXVII. **Opciones para crear usuarios**

Sintaxis básica:
create user name [[with] opciones [...]].
Opciones:
SYSID uid CREATEDB NOCREATEDB CREATEUSER NOCREATEUSER IN GROUP groupname [, ...] [ENCRYPTED UNENCRYPTED] PASSWORD 'password' VALID UNTIL 'abstime'
Ejemplo:
Create role "fooadmin" login Password 'foo' ;

Fuente: elaboración propia

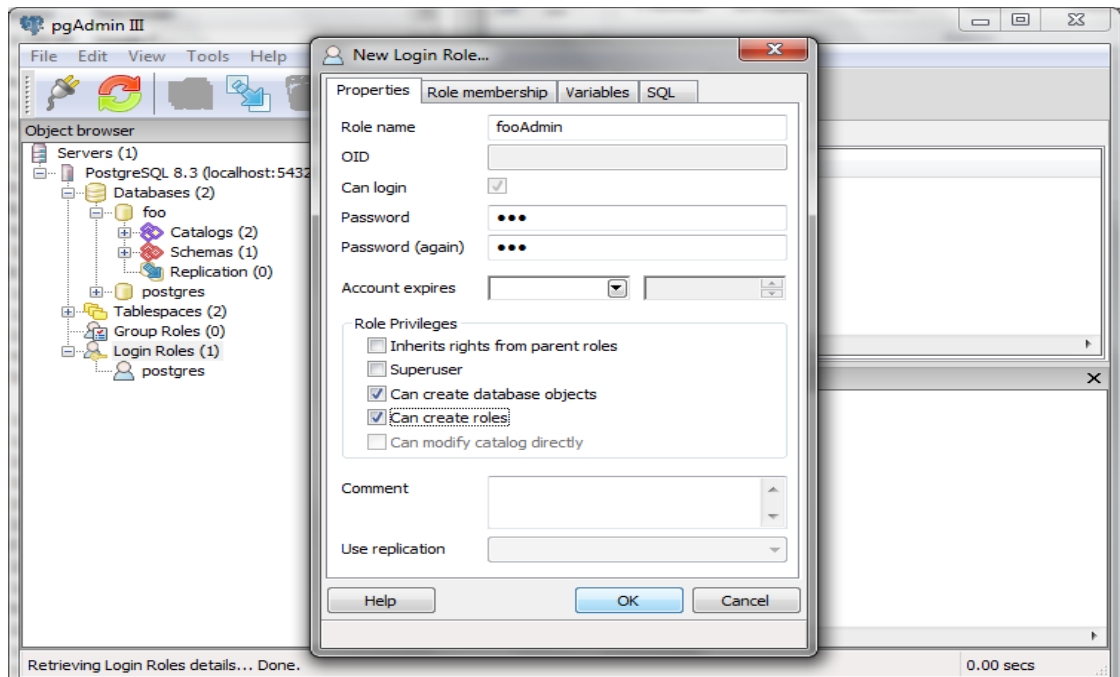
Las opciones se describen a continuación:

- Sysid uid: especifica que el id que va a definirse debe establecerse al valor de uid. Si se omite, un razonable y único valor numérico por defecto es escogido.
- Password- password: establece la nueva contraseña del usuario. Si no se especifica, la contraseña por defecto es *null*.

- `Createdb` | `nocreatedb`: usando la palabra clave `createdb` se le garantiza al nuevo usuario el privilegio de crear nuevas bases de datos, así como el de destruir las de su propiedad. Usando `nocreatedb` se niega este permiso (que es lo que ocurre por defecto).
- `Createuser` | `nocreateuser`: certifica el privilegio de crear nuevos usuarios. Si un usuario tiene los privilegios de crear a otros usuarios tendrá además todos los privilegios, en todas las bases de datos (incluyendo los permisos para crear una base de datos, aunque se haya especificado `nocreatedb`).
- `Nocreateuser`: explícitamente fuerza a la situación por defecto, que deniega el privilegio.
- `In group nombre_grupo [, ...]`: añade al nuevo usuario al grupo llamado `nombre_grupo`. Pueden ser especificados múltiples nombres de grupo, separándolos mediante comas. El o los grupos deben existir para que funcione la condición.
- `Valid until abstime`: establece que la contraseña del usuario expirará el `abstime`, el cual debe ser un formato reconocible de fecha/hora (time stamp). Atrás esa fecha, la contraseña se restablece, y la expiración se hace efectiva.

La creación de usuarios es mas amigable por medio de la interfaz grafica que nos ofrece la herramienta PgAdmin, en la siguiente imagen se muestra la pantalla para crear roles y usuarios.

Figura 22. Creación de usuarios por PgAdmin



Fuente: elaboración propia.

1.3.20. Modificación de usuarios

Los usuarios existentes sólo pueden ser modificados por súper usuarios o usuarios administradores, por esta razón se realizan las modificaciones mediante el comando *Alter User*, y para eliminar el usuario se utiliza el comando *drop user*, la sintaxis se muestra en la siguiente tabla.

Tabla XXVIII. **Sintaxis modificación de usuarios**

Alter user nombre_usuario [with password' password'] [createdb nocreatedb] [createuser nocreateuser] [valid until 'abstime']
Drop user nombre_usuario;

Fuente: elaboración propia.

1.4. Procesadores Sparc

La palabra SPARC viene del inglés *Scalable Processor Architecture*. Un procesador con arquitectura escalable es aquel en el que se pueden definir versiones posteriores de la misma con mayor cantidad de características, siempre salvaguardando la compatibilidad con los programas de versiones anteriores. Tiene una arquitectura basada en un juego de instrucciones RISC (*Reduced Instruction Set Computer*) desarrollado por la empresa Sun Microsystems y lanzado a mediados de 1987.

Fue originalmente diseñada por Sun Microsystems y dirigido por el ingeniero Kaa en 1985, se basa en los diseños RISC I y II de la Universidad de California en Berkeley que fueron definidos entre los años 1980 y 1982. La empresa Sun Microsystems diseñó esta arquitectura y la licenció a otros fabricantes como Texas Instruments, Cypress Semiconductor, Fujitsu, LSI Logic entre otros.

SPARC es la primera arquitectura RISC abierta y como tal, las especificaciones de diseño están publicadas, así otros fabricantes de microprocesadores pueden desarrollar su propio diseño. Una de las ideas innovadoras de esta arquitectura es la ventana de registros que permite hacer fácilmente compiladores de alto rendimiento y una significativa reducción de

memoria en las instrucciones *load/store* en relación con otras arquitecturas RISC. Las ventajas se aprecian sobre todo en programas grandes.

La CPU SPARC está compuesta de una unidad entera, IU (*Integer Unit*) que procesa la ejecución básica y una FPU (*Floating-Point Unit*) que ejecuta las operaciones y cálculos de reales. La IU y la FPU pueden no estar integradas en el mismo chip.

Aunque no es una parte formal de la arquitectura, las computadoras basadas en sistemas SPARC de Sun Microsystems tienen una unidad del manejo de memoria (MMU) y un gran caché de direcciones virtuales (para instrucciones y datos) que están dispuestos periféricamente sobre un bus de datos y direcciones de 32 bits.

1.4.1. Historia

Sparc es una marca registrada de SPARC International Inc, una organización establecida en 1989 para promover la arquitectura Sparc, y proporcionar exámenes de estabilidad. La implementación de la arquitectura original de 32 bits de Sparc, inicialmente fue diseñada en estaciones de trabajo Sun-4 (son una serie de estaciones de trabajo y servidores Unix, lanzados en 1987), reemplazando los sistemas anteriores (Sun-3) basados en la familia de procesadores Motorola 68,000.

Posteriormente, los procesadores Sparc fueron utilizados en servidores SMP (*Symmetric multiprocessing* donde dos procesadores de características idénticas están conectados a una única memoria central compartida) desarrollados entre otros por Sun y Fujitsu, y diseñados para operaciones de 64 bits. Ha habido tres grandes versiones de la arquitectura Sparc.

- La primera de ellas (publicada en 1986) fue la versión V7 (32 bits). Sparc v8 (una versión superior) fue lanzada en 1990. La principal diferencia entre estas dos versiones fue la introducción de operaciones de producto y cociente de números enteros. La versión V8 de SPARC ha servido como base para Estándar IEEE 1754-1994, un estándar IEEE para una arquitectura de microprocesadores de 32 bits. La versión V9 de Sparc fue lanzada en 1993, esta ya poseía una arquitectura de 64 bits.
- En el 2002, Fujitsu y Sun liberaron JpS1 (*Joint Programming Specification*). Describiendo las funciones de procesador que de modo idéntico fueron puestas en práctica en varias CPUs de ambas empresas. Los primeros CPUs en las que se implementa JPS1 fueron el Ultra SPARC III por Sun y el SPARC64 por Fujitsu.
- A principios del 2006, Sun liberó Ultra SPARC (Arquitectura 2005) Esta incluía no sólo las características de SPARC V9, sino también todas las extensiones arquitectónicas (como CMT, *hiper privileged*, VIS 1, y VIS 2) el presente en los procesadores Ultra SPARC de Sun los cuales se ponen en práctica con Ultra SPARC T1. Ultra SPARC (Arquitectura del 2005) incluye las extensiones estándar de Sun.

En el 2007, Sun publica una nueva actualización; la arquitectura Ultra SPARC 2007. Entre tantas implementaciones de Sparc, las mas populares fueron la Súper SPARC y la Ultra SPARC-I, estas fueron usadas como sistemas de referencia para los *benchmarks*. El Ultra SPARC 296 Mhz es el sistema de referencia para el Benchmark de SPEC CPU2006, actualmente han alcanzado gran auge por el desarrollo de los teléfonos inteligentes.

1.4.2. Principales características de arquitectura Sparc

Su característica distintiva es utilizar ventanas de registros. De la cual hablaremos a continuación, consiste en 32 registros de enteros de 32 bits. 16 registros de punto flotante de 64 bits (para el caso de doble precisión) que se pueden utilizar como 32 registros de 32 bits (para precisión simple). A continuación se listan sus principales características:

- Modos de direccionamiento
 - Inmediato, (constantes de 13 bits)
 - Directo, (offset de 13 bits)
 - Indirecto, (registro + offset de 13 bits o registro + registro)
 - Utiliza instrucciones retardadas (saltos, *load* y *store*)

- Manejo de memoria:
 - Espacio virtual de 4 Gigabytes
 - Unidad de manejo de memoria (MMU) que trabaja con páginas de tamaño configurable.

1.4.3. Ventanas de registros

Los procesadores SPARC realizan un sistema de ventanas. Cada ventana son 24 registros (a los cuales se le suman 8 más que son globales, ósea, accesibles por todas las ventanas; la suma de 32 registros es visible en todo momento) que pertenecen a un procedimiento o subprograma. A medida que se van creando nuevas ventanas los registros de salida de la ventana actual se solapan sobre los de entrada de la ventana siguiente. Los CPU registros locales

y de salida de la ventana actual son diferentes a los registros locales y de salida de la ventana siguiente.

Para verificar si un número de ventana es válido se utiliza el registro WIN. (InvalidMask) de 32 bits. Si el bit *i*-ésimo está en 1 indica que esa ventana inválida y si se intenta acceder esta, se genera un error.

El registro G0 está fijo por medio del hardware con el valor 0. En él, los almacenamientos no alteran su valor. Los registros G1 a G7 son globales y pueden contener variables enteras, apuntadores a tablas u otros datos importantes. Los registros de salida son de O0 a O7 y son usados por los procedimientos para pasar parámetros a los procedimientos que llaman.

El primer parámetro va al registro O0, el siguiente al O1 y así sucesivamente. El registro O6 (AE) se utiliza como apuntados a la pila de memoria. La pila se usa para los parámetros que exceden los registros, espacio de asignación dinámica de la pila, ventanas apiladas en memoria, así como para salvar registros de punto flotante, apuntadores a buffers en donde los procedimientos llamados pueden enviar estructuras y arreglos de datos, etc.

La instrucción CALL (llamada) deposita la dirección de retorno en el registro O7. Los 8 registros para variables locales L0 a L7, se pueden usar en la forma que el programador o el compilador consideren más conveniente. Las 8 variables de entrada I0 a I7 son los parámetros que son pasados al procedimiento actual por el que le llama: Los registros no utilizados pueden contener variables locales adicionales. I6 (FP) es el apuntador de marco que se usa para direccionar variables en el marco de pila.

A diferencia de SP que solo puede cambiar mientras el procedimiento se ejecuta, FP apunta la misma palabra en memoria durante toda la ejecución del procedimiento, siendo más apropiado para la indexación que el SP. La dirección de retorno del procedimiento se encuentra contenida en el registro I7. Por lo tanto y en general, existen 4 grupos de registros diferentes:

- Registros Globales: G0-G1
- Registros de Salida: O0-O7
- Registros Locales: L0-L7
- Registros de Entrada: I0-I7

1.4.4. Categoría de instrucciones

La arquitectura SPARC tiene cerca de 50 instrucciones enteras. Estas se pueden clasificar en cinco grandes categorías. *LOAD* y *STORE* (La única manera de acceder a la memoria). Además usan dos registros o un registro y una constante para calcular la dirección de memoria a direccionar. Instrucciones Aritméticas/Lógicas/Shift. Ejecutan operaciones aritméticas, lógicas y de desplazamiento de bits. Estas instrucciones calculan el resultado si es una función de dos operandos y guardan el resultado en un registro.

Operaciones del Coprocesador. La IU (unidad de instrucciones) extrae las operaciones de punto flotante desde las instrucciones del bus de datos y las coloca en la cola para la FPU (Unidad de punto flotante). La FPU ejecuta los cálculos de punto flotante con un número fijo en unidad aritmética de punto flotante, (el número es dependiente de la aplicación).

Las operaciones de punto flotante son ejecutadas concurrentemente con las instrucciones de la IU y con otras operaciones de punto flotante cuando es

necesario. La arquitectura SPARC también especifica una interfaz para la conexión de un coprocesador adicional.

Instrucciones de Control de Transferencia. Estas incluyen saltos, llamadas, etc. El control de transferencia tiene retardo hasta después de la ejecución de la próxima instrucción, así el pipeline no es vaciado porque ocurre un control de tiempo. De este modo, los compiladores pueden ser optimizados por ramas retardadas.

Instrucciones de control de registros Lectura/Escritura. Estas instrucciones se incluyen para leer y grabar el contenido de varios registros de control, generalmente la fuente o destino está implícito en la instrucción.

1.4.5. Direccionamiento en memoria

A diferencia de los procesadores 80x86 el espacio de direcciones de los procesadores SPARC es lineal y no segmentado. Por otra parte el bus de direcciones es de 32 bits lo que permite direccionar hasta 32 bytes = 4 Gbytes de memoria virtual. Es decir que para especificar una dirección de memoria se necesitan 32 bits.

Como ya se mencionó anteriormente otra característica importante de los procesadores SPARC es que son procesadores RISC. Para que un procesador sea considerado RISC debe cumplir cierto conjunto de propiedades entre las cuales hay una que nos interesa resaltar: la codificación de todas las instrucciones ocupan exactamente 32 bits.

De lo anterior queda claro que no es posible, en los procesadores SPARC, codificar una dirección de memoria absoluta de 32 bits junto con la instrucción

que la utilizará ya que no queda espacio para colocar el valor de la dirección mas los bits necesarios para codificar la instrucción.

Para esto el procesador SPARC provee una instrucción especial llamada SETHI que permite colocar un valor de hasta 22 bits en los 22 bits más significativos de algún registro. De esta forma luego utilizando adecuadamente la instrucción OR es posible colocar los 10 bits menos significativos de la constante en los 10 bits menos significativos del registro.

2. CONFIGURACIÓN Y ACTIVACIÓN DEL UPS

El UPS se encuentra en el salón 010 del edificio T3 de la Facultad de Ingeniería de la Universidad de San Carlos, el modelo del UPS pertenece a la serie Powerware 9355 del fabricante Eaton, se basa en una Tipología Híbrida doble conversión.

Figura 23. **UPS Powerware 9355**



Fuente: www.eaton.com. Consulta: 02 de diciembre 2011.

2.1. Diagnóstico

El UPS es nuevo y nunca se ha conectado, cuenta con instalación de aire acondicionado y la infraestructura eléctrica adecuada, tiene más de dos años apagado, lo que afecta la vida útil de las baterías internas, por consiguiente, es necesario encenderlo e iniciar el proceso de carga para determinar si las baterías son funcionales.

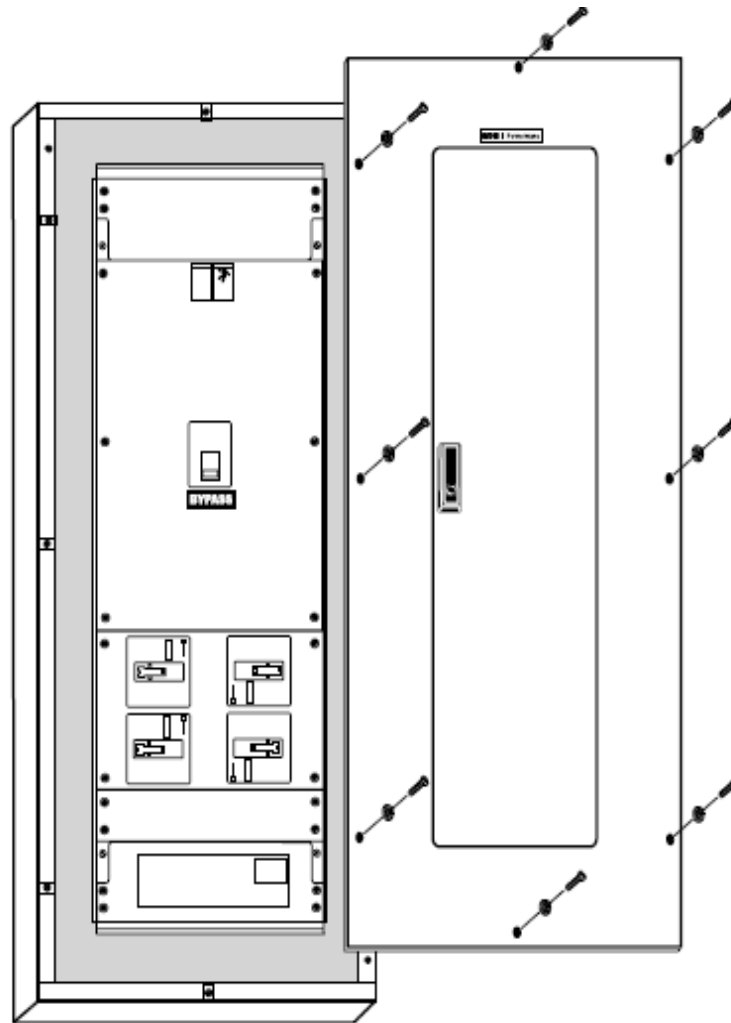
No se tiene ningún esquema de conexión eléctrica y se desconoce que flipón de la caja eléctrica alimenta la infraestructura eléctrica del UPS, dentro de la infraestructura eléctrica se tiene una roseta que alimenta directamente la infraestructura eléctrica del ITCoE, dejando al equipo de computo desprotegido de variaciones de corriente y cortes de energía eléctrica.

2.1.1. Procedimiento de instalación y activación

El procedimiento de conexión y activación se realizó con base en las especificaciones del Manual de instalación del UPS que se encuentra disponible en la página web de la empresa, los pasos realizados fueron los siguientes:

- Realización de pruebas para determinar el tipo de flipón que alimenta la infraestructura eléctrica del UPS.
- Mantenimiento del equipo, abrir, realizar aspirado y limpieza del hardware
- Verificación, que las conexiones eléctricas del lugar de instalación hayan sido debidamente colocadas y tengan las especificaciones de acuerdo al manual de la instalación.
- Suspensión del flujo de electricidad hacia el dispositivo.

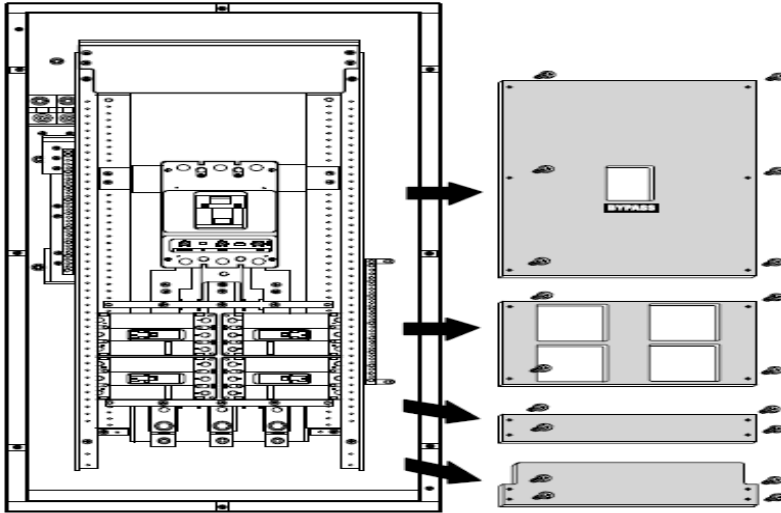
Figura 24. Retiro de la tapa frontal



Fuente: *Eaton 9355 Installation Guide*.

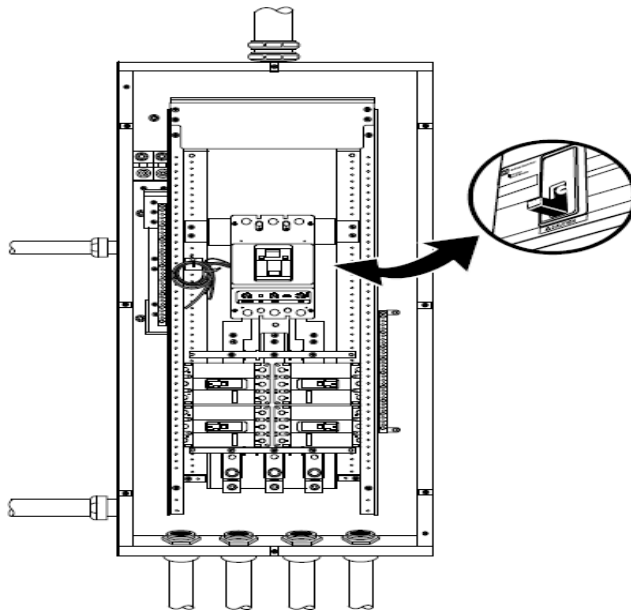
- Retiro de la tapa frontal del UPS.
- Retiro de cubiertas internas para acceder a los interruptores internos.

Figura 25. Retiro cubiertas internas



Fuente: Eaton 9355 Installation Guide. 22 de diciembre 2011.

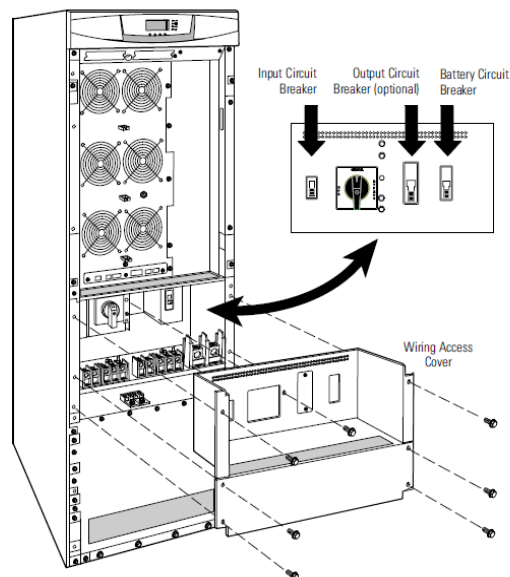
Figura 26. Estructura eléctrica del UPS



Fuente: Eaton 9355 Installation Guide.

- Realización de agujeros para el conducto (entrada de CA, salida de la UPS, conexión de carga, y mantenimiento de los hilos de contacto del bypass).
- Verificación que el interruptor de *bypass* Gabinete está en la posición OFF.
- Revisión de estado “OFF” en *Input circuit Breakery* el *Battery Circuit Breaker*.
- Retiro de cubierta de acceso cableado del UPS.
- Introducción de cables de corriente y se conectaron al UPS
- Colocación de cubierta de acceso cableado del UPS.
- Comprobación de ON en *Input Circuit Breaker* y el *Battery Circuit Breaker*.

Figura 27. **Modulo de los breakers**



Fuente: Eaton 9355 Installation Guide.

- Instalación de cubiertas internas de los interruptores internos
- Liberación de corriente eléctrica que alimenta el UPS
- Procedimiento de encendido de UPS y se dejó cargar por una semana.
- Constatación de funcionamiento
- Cambio de dirección del bypass (roseta) para que las instalaciones eléctricas del ITCoE se alimentaran del UPS

2.1.2. Pruebas

Se procedió a realizar pruebas para determinar si el UPS retenía la carga y si regula la corriente eléctrica de los laboratorios del ITCoE, se determinó que el UPS perdió capacidad de almacenamiento de energía debido a que al estar apagado por más de dos años se atrofiaron parte de las baterías y es necesario reemplazarlas para aprovechar al máximo las capacidades del UPS.

3. HABILITACIÓN DEL SERVIDOR

El servidor SUN pertenece a la serie ULTRA Enterprise 450, cuenta con dos procesadores de 450Mhz de 64 bits, basados en una arquitectura Sparc, es un servidor tipo torre, equipado con fuente eléctrica redundante.

3.1. Diagnóstico

El servidor se encontraba desconectado, no tenía cable de alimentación eléctrica, llave de encendido y periféricos de entrada (teclado).

No contaba con la llave para encender el equipo, está conformado por seis discos duros de 30 Gb, de los cuales únicamente cuatro funcionan correctamente, estaba instalado Solaris 6, pero las claves de administración estaban perdidas, por lo que era imposible tener acceso al servidor.

El servidor cuenta con seis tarjetas de red, las cuales funcionan correctamente. El servidor también cuenta con unidad de disquete y CD Room, ambas funcionan correctamente.

El servidor posee fuentes redundantes de energía, las cuales se encuentran en perfecto estado y no tienen ningún desgaste o símbolo de envejecimiento a simple vista, no se cuenta con el cable de alimentación que debe soportar altas corrientes por lo cual no se puede colocar un cable genérico de computadora ya que no soportaría las exigencias del servidor.

Figura 28. **Servidor Ultra Enterprise**



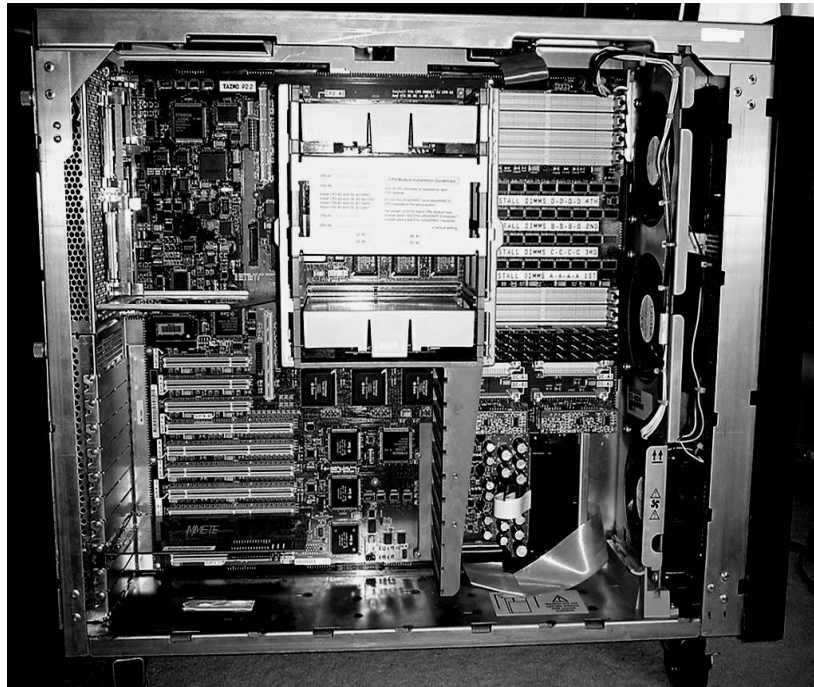
Fuente: www.ITrecycle.com. Consulta: 03 de diciembre de 2011.

3.1.1. Mantenimiento del hardware

Debido a que no se contaba con la llave de encendido se anuló en sistema de encendido por llave y se habilitó únicamente el encendido por interruptor.

Para realizar una limpieza profunda del servidor se procedió a desarmarlo completamente y limpiar cada una de las piezas de manera individual, aplicándoles aire comprimido y limpia contacto.

Figura 29. **Vista lateral del servidor**



Fuente: www.Sunstuff.org. Consulta: 03 de diciembre de 2011.

Se retiraron los discos duros que no funcionaban correctamente y se configuró el arreglo de discos para que funcionara correctamente, se armó el servidor y se reinició Bios.

3.2. Selección del sistema operativo

El servidor fue diseñado para operar sistemas operativo Unix, especialmente de la rama Solaris, se analizaron cada una de las versiones existentes y su viabilidad, el soporte y la licencia.

Tabla XXIX. **Versiones de Solares**

Versión de Solaris, Sun OS y Fecha de Publicación	Descripción
<p>Solaris 10 Sunos 5.10 31 de Enero, 2005</p>	<p>Incluye soporte AMD64/EM64T, Dtrace, Solaris Containers, Service Management Facility (SMF) para reemplazar al sistema init.d, NFSv4.</p> <p>Eliminado soporte para procesadores sun4m y Ultra SPARC I a frecuencia menor que 200 Mhz. Se ha añadido Java Desktop System como escritorio por defecto. Se ha añadido GRUB como cargador de arranque para plataformas x86. Se ha añadido soporte iSCSI. Se ha añadido soporte para el nuevo sistema de archivos, ZFS (versión 1/06).</p>
<p>Solaris 9 SunOS 5.9 28 de mayo, 2002 SPARC 10 de enero, 2003(x86)</p>	<p>iPlanet Directory Server, Resource Manager, Solaris Volume Manager.</p> <p>Añadida compatibilidad con Linux. Eliminado Open Windows, y soporte para sun4d. La actualización más reciente Solaris 9 9/05.</p>
<p>Solaris 8 SunOS 5.8 Febrero de 2000</p>	<p>Incluye Multipath I/O, IPv6 y IPsec. Introduce RBAC (control de acceso basado en roles).</p> <p>Soporte para sun4c eliminado. La actualización más reciente es Solaris 8 2/04.</p>
<p>Solaris 7 SunOS 5.7 Noviembre de 1998</p>	<p>La primera versión de 64 bits para plataforma Ultra SPARC. Añadido soporte nativo para registro de metadatos en el sistema de archivos (UFS logging).</p>
<p>Solaris 2.6 SunOS 5.6 Julio de 1997</p>	<p>Incluye protocolo Kerberos, PAM, TrueType, WebNFS, y soporte de archivos grandes.</p>

Continuación de la tabla XXVIII.

Solaris 2.5.1 SunOS 5.5.1 Mayo de 1996	Ésta fue la primera y única versión que soportó la plataforma PowerPC. También se añadió soporte Ultra Enterprise, y los identificadores de usuario (uid_t) se expandieron a 32 bits.
Solaris 2.5 SunOS 5.5 Noviembre de 1995	Primera versión en soportar Ultra SPARC e incluir CDE, NFSv3 y NFS/TCP.
Solaris 2.4 SunOS 5.4 Noviembre de 1994	Primera versión unificada SPARC/x86. Incluye soporte de ejecución OSF/Motif.
Solaris 2.3 SunOS 5.3	Noviembre de 1993, OpenWindows 3.3 cambia de NeWS a Display PostScript y elimina soporte de SunView. Esta versión fue sólo para SPARC.
Solaris 2.2 SunOS 5.2 Mayo de 1993	Primera en soportar la arquitectura sun4d. Esta versión fue sólo para SPARC.
Solaris 2.1 SunOS 5.1 Diciembre de 1992 (SPARC), Mayo de 1993 (x86)	Soporte para arquitectura Sun-4 y sun4m. Primera versión para Solaris x86.
Solaris 2.0 SunOS 5.0 Junio de 1992	Primera versión preliminar, soporte solamente para la arquitectura sun4c.

Fuente: [es.wikipedia.org/wiki/Solaris_\(sistema_operativo\)](http://es.wikipedia.org/wiki/Solaris_(sistema_operativo)). Consulta: 03 de diciembre 2011.

Se determinó que la última versión de Solaris no es compatible con la arquitectura Sparc y las versiones anteriores tienen un alto costo de licenciamiento, por esta razón se inició a evaluar las versiones Unix de GNU/Linux compatibles con la arquitectura del servidor.

Tabla XXX. **Distribuciones de Linux**

Distribución Linux	Características
Aurora SPARC Linux	Está basada en redhat 7.3, únicamente existen 3 maillist y hay poca documentación.
Debian GNU/Linux.	Tiene una comunidad muy extensa, existe soporte para arquitectura Sparc y tiene repositorios actualizados. Su fortaleza es el sistema de instalación de paquetes APT.
Gentoo	Tiene una comunidad grande, pero la instalación es complicada y no hay variedad de paquetes disponibles.
PLD.	Es una variación de redhat desarrollada en Polonia, no hay documentación.
RedHat Linux	El proyecto para soportar arquitectura Sparc no está activo y únicamente cuenta con versiones antiguas del kernellinux.
Slackware	No hay una comunidad que de soporte a la arquitectura Sparc.
SuSE Linux	Ya no da soporte a la arquitectura Sparc.
Vine Linux	Es una distribución japonesa con soporte a paquetes rpm, existe poca documentación.

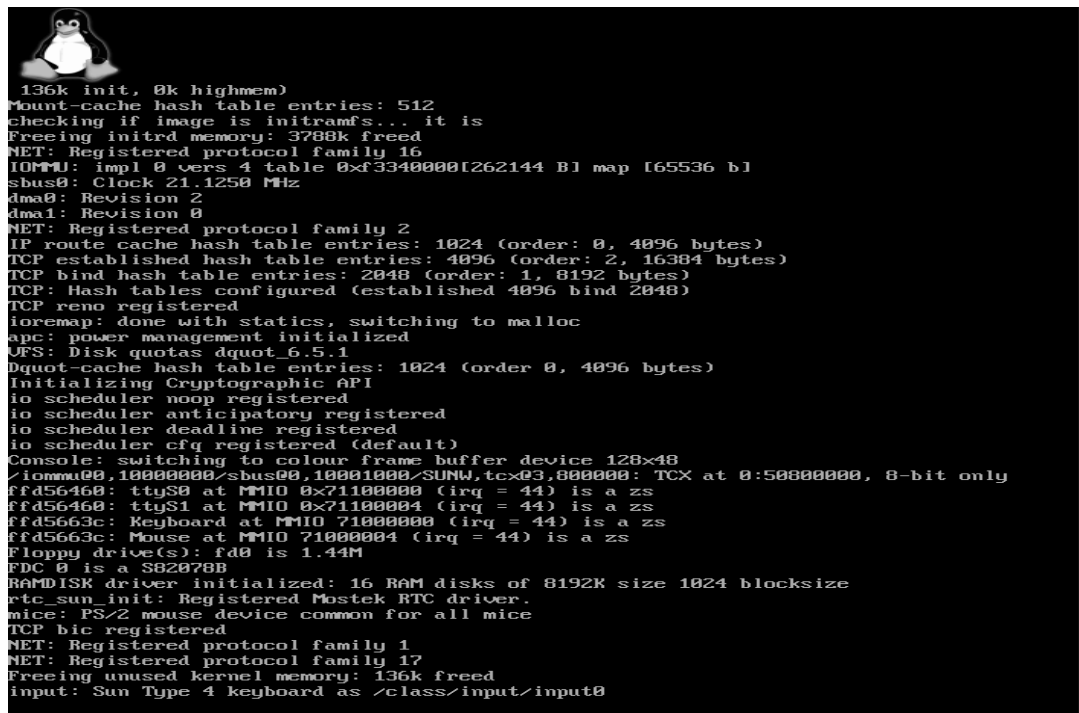
Fuente: elaboración propia.

3.2.1. Instalación del sistema operativo

El servidor soporta Solaris 2.5.1 o alguno superior, como no se tienen las licencias y el sistema operativo que soporta, es obsoleto para las necesidades actuales se consideró GNU/Linux, específicamente Debian 6.

El servidor cuenta con Open Boot una variante de Sparc Improved Boot Loader (SILO), un Bios basado en comandos, leyendo el manual de referencia, se logró acceder al mismo al teclear simultáneamente las teclas Stop y A, el menú el servidor se despliega y se debe teclear la siguiente instrucción: bootcdrom.

Figura 30. Carga de Open Boot



```

 136k init, 0k highmem)
Mount-cache hash table entries: 512
Checking if image is initramfs... it is
Freeing initrd memory: 3788k freed
NET: Registered protocol family 16
IOMMU: impl 0 vers 4 table 0xf3340000[262144 B] map [65536 b]
sbus0: Clock 21.1250 MHz
dma0: Revision 2
dma1: Revision 0
NET: Registered protocol family 2
IP route cache hash table entries: 1024 (order: 0, 4096 bytes)
TCP established hash table entries: 4096 (order: 2, 16384 bytes)
TCP bind hash table entries: 2048 (order: 1, 8192 bytes)
TCP: Hash tables configured (established 4096 bind 2048)
TCP reno registered
ioremap: done with statics, switching to malloc
apc: power management initialized
VFS: Disk quotas dquot_6.5.1
Dquot-cache hash table entries: 1024 (order 0, 4096 bytes)
Initializing Cryptographic API
io scheduler noop registered
io scheduler anticipatory registered
io scheduler deadline registered
io scheduler cfq registered (default)
Console: switching to colour frame buffer device 128x48
/iommu00,10000000/sbus00,10001000/SUNW,tcx03,800000: TCX at 0:50000000, 8-bit only
ffd56460: ttyS0 at MMIO 0x71100000 (irq = 44) is a zs
ffd56460: ttyS1 at MMIO 0x71100004 (irq = 44) is a zs
ffd5663c: Keyboard at MMIO 71000000 (irq = 44) is a zs
ffd5663c: Mouse at MMIO 71000004 (irq = 44) is a zs
Floppy drive(s): fd0 is 1.44M
FDC 0 is a SB2078B
RAMDISK driver initialized: 16 RAM disks of 8192K size 1024 blocksize
rtc_sun_init: Registered Mostek RTC driver.
mice: PS/2 mouse device common for all mice
TCP bic registered
NET: Registered protocol family 1
NET: Registered protocol family 17
Freeing unused kernel memory: 136k freed
input: Sun Type 4 keyboard as /class/input/input0

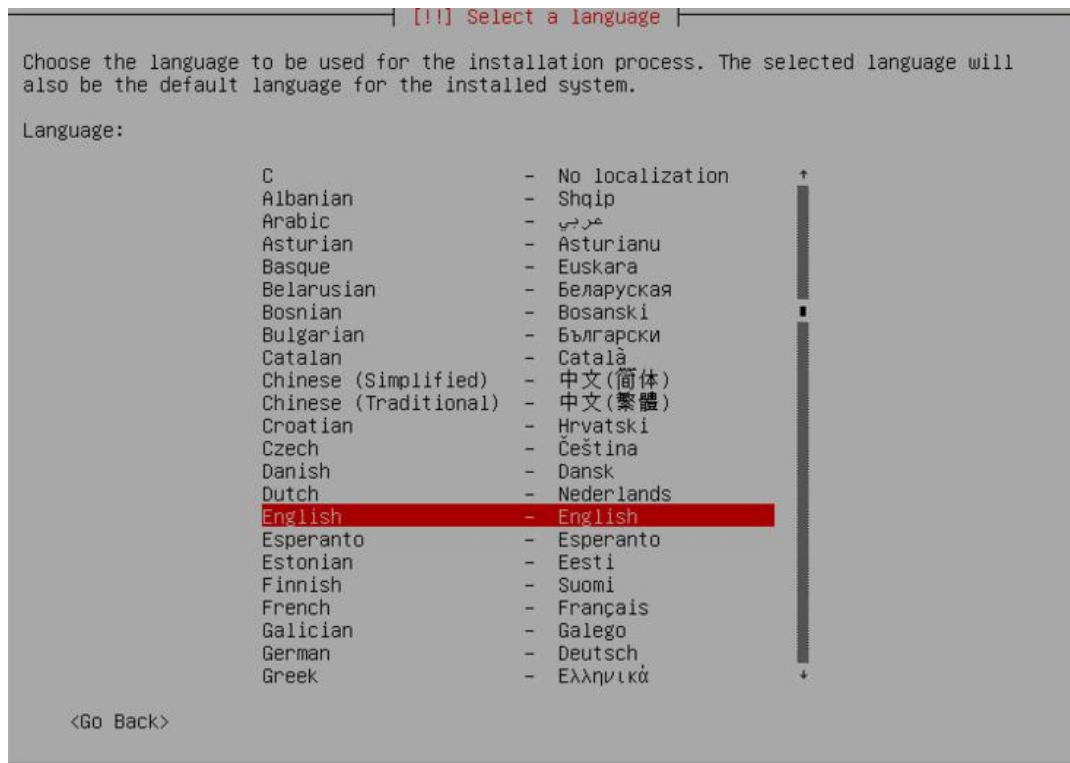
```

Fuente: elaboración propia.

La instrucción openboot permite que el servidor lea la unidad de CD-ROM, cargando el programa de instalación en la memoria del servidor, permitiendo instalar Debian 6 en el servidor.

Al iniciar el programa de instalación, debemos seleccionar el idioma de la instalación que se utilizara durante la instalación.

Figura 31. Selección del idioma para la instalación

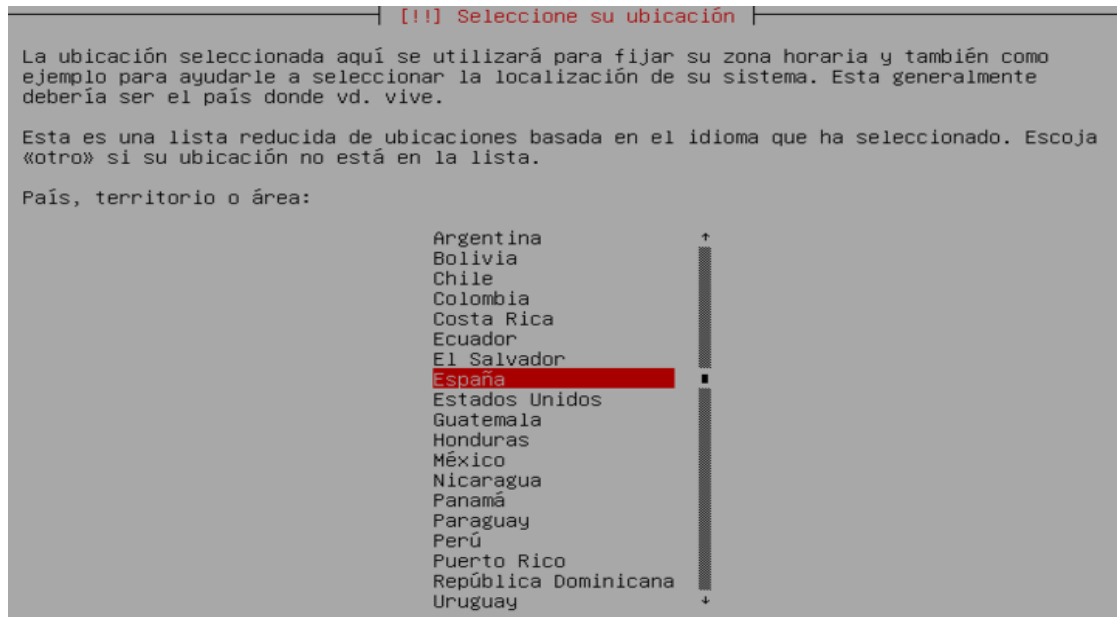


Fuente: elaboración propia.

Después de seleccionar el idioma de la instalación, procedemos a seleccionar la ubicación de nuestro servidor, es importante seleccionar nuestro

país, para que nos precargue las configuraciones de zona horaria, notación decimal y elementos de la regionalización.

Figura 32. Selección de zona horaria

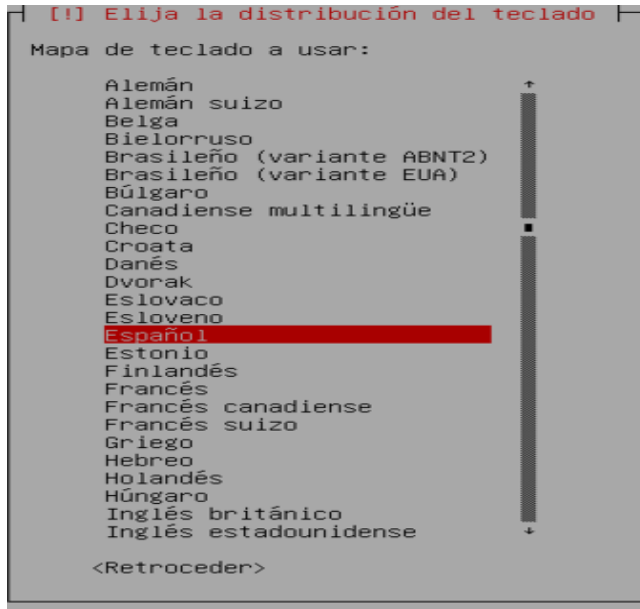


Fuente: elaboración propia.

Al finalizar la selección de la configuración regional, es necesario seleccionar que tipo de teclado estamos utilizando, para poder utilizar de manera correcta la combinación y configuración de teclas que posee nuestro teclado.

En muchas ocasiones esta mala configuración provoca problemas al utilizar la consola, porque se ingresan caracteres distintos a los que se seleccionan en el teclado y que provocan la ejecución de comandos no requeridos.

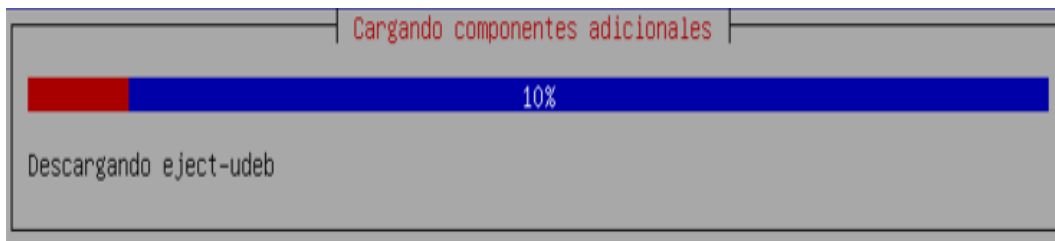
Figura 33. Elección de la distribución del teclado.



Fuente: elaboración propia.

Se espera que finalice el inicio de la carga de componentes y otras herramientas necesarias en la instalación.

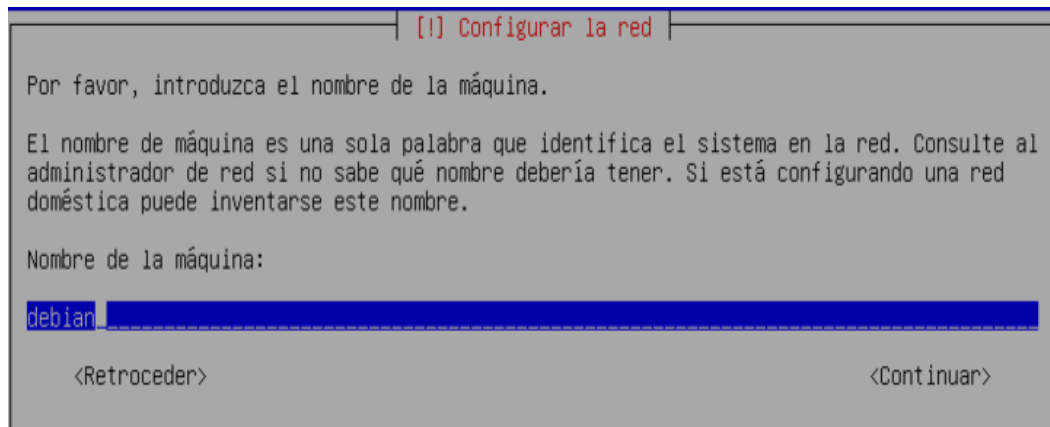
Figura 34. Carga de componentes.



Fuente: elaboración propia.

Se introduce el nombre por medio del cual identifica el servidor, dentro de la red.

Figura 35. **Configuración de red.**



[!] Configurar la red

Por favor, introduzca el nombre de la máquina.

El nombre de máquina es una sola palabra que identifica el sistema en la red. Consulte al administrador de red si no sabe qué nombre debería tener. Si está configurando una red doméstica puede inventarse este nombre.

Nombre de la máquina:

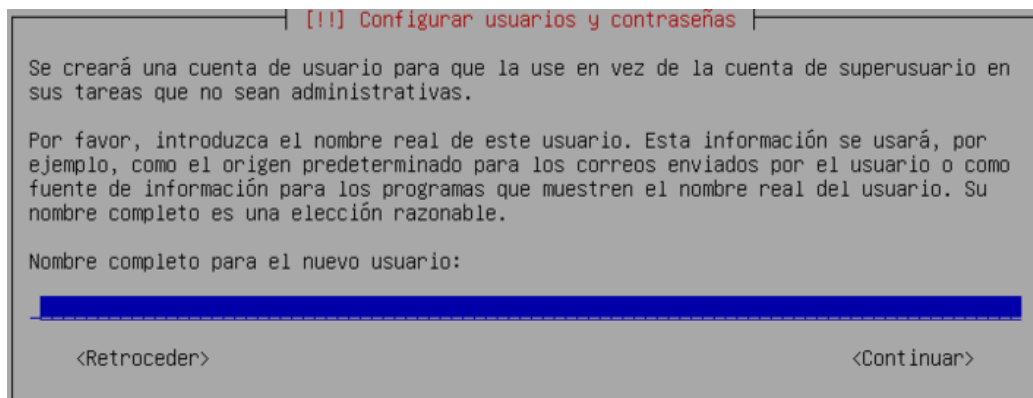
debian

<Retroceder> <Continuar>

Fuente: elaboración propia.

Se introduce la clave de administrador, esta será la clave que utilizará el usuario root.

Figura 36. **Configuración de usuarios y contraseñas.**



[!!] Configurar usuarios y contraseñas

Se creará una cuenta de usuario para que la use en vez de la cuenta de superusuario en sus tareas que no sean administrativas.

Por favor, introduzca el nombre real de este usuario. Esta información se usará, por ejemplo, como el origen predeterminado para los correos enviados por el usuario o como fuente de información para los programas que muestren el nombre real del usuario. Su nombre completo es una elección razonable.

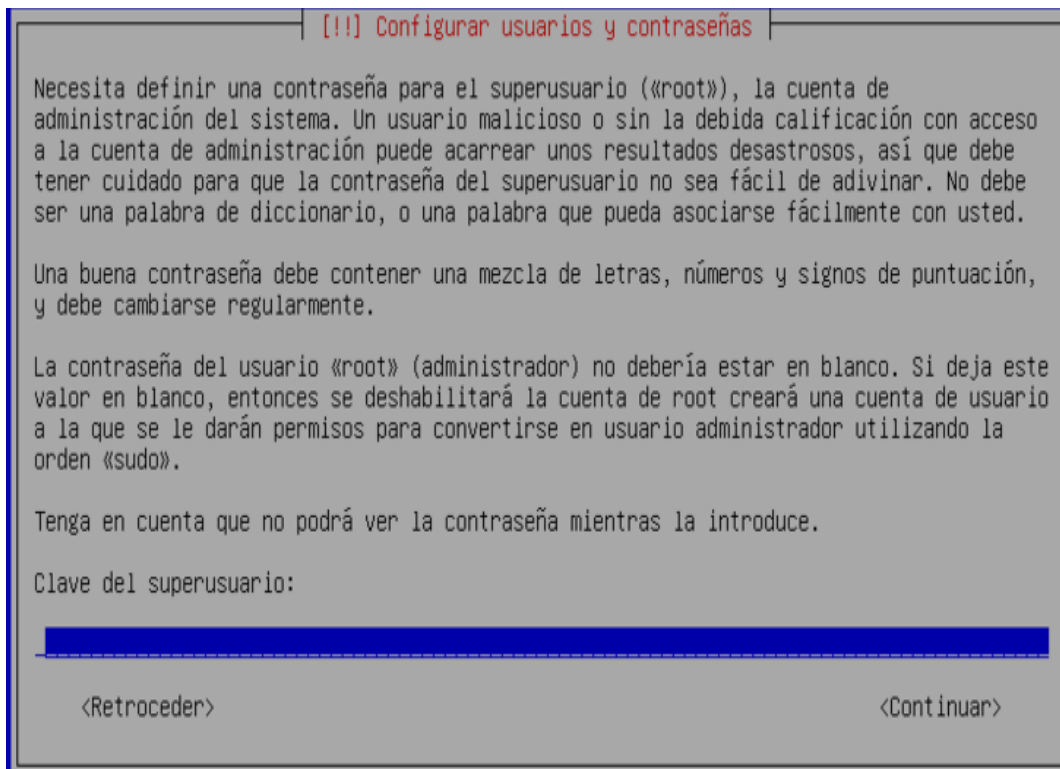
Nombre completo para el nuevo usuario:

<Retroceder> <Continuar>

Fuente: elaboración propia.

Se procede a crear a los usuarios con permisos básicos con su respectiva contraseña.

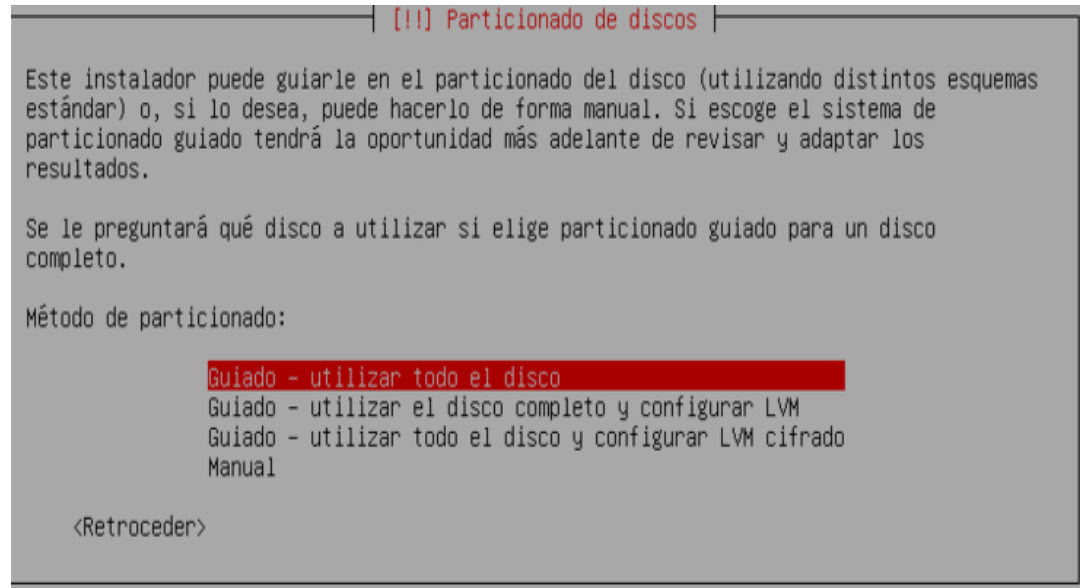
Figura 37. Creación de usuario y contraseña



Fuente: elaboración propia.

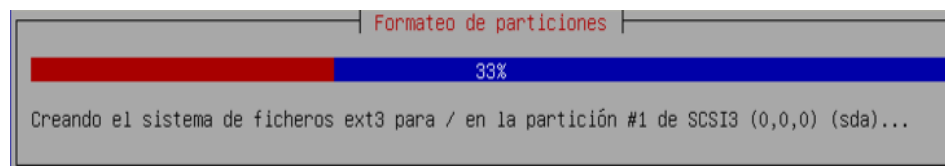
Se procede a particionar el disco y realizar las configuraciones básicas para el sistema de archivos.

Figura 38. **Particionando discos**



Fuente: elaboración propia.

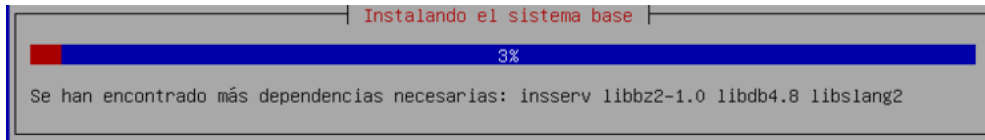
Figura 39. **Formateo de particiones**



Fuente: elaboración propia.

Al finalizar las configuraciones, el programa de instalación inicia la carga de los archivos en el servidor, realiza las configuraciones del sistema operativo y el sistema de archivos.

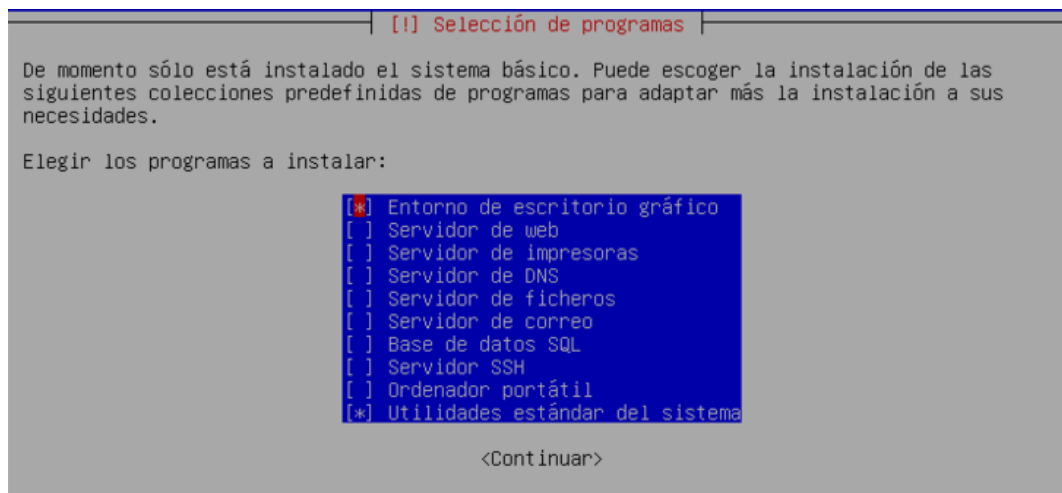
Figura 40. **Instalación de sistema base**



Fuente: elaboración propia.

Se instalan los programas básicos del servidor, como correo electrónico, dns, servidor web y cualquier otro programa que se requiera en el servidor.

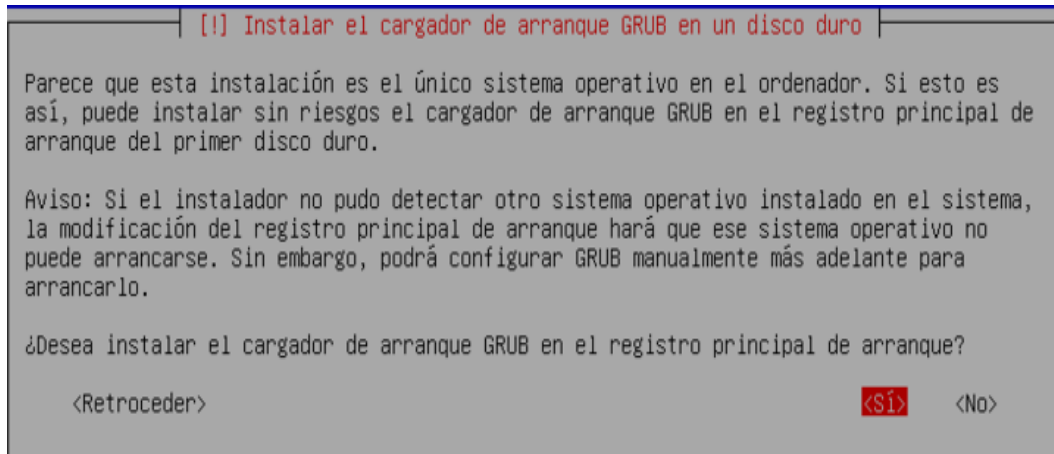
Figura 41. **Selección de programas.**



Fuente: elaboración propia.

La instalación se finaliza al seleccionar la instalación del cargador de arranque.

Figura 42. **Instalación del GRUB**



Fuente: elaboración propia.

Al terminar el asistente de instalación se procede a reiniciar el servidor e ingresar a Debian 6, para configurar los repositorios para actualizar el Kernel y los componentes básicos para administrar el servidor como SSH y la base de datos PostgreSQL.

CONCLUSIONES

1. Se adquirió el teclado del servidor ULTRA Enterprise450, con el teclado se puede acceder al menú de arranque del servidor e instalar el software necesario para su correcto funcionamiento.
2. El UPS conectado regula la corriente eléctrica de los laboratorios y del área administrativa del ITCoE, pero ya no opera a su capacidad completa, debido a que al estar desconectado por más de un año, las baterías perdieron su capacidad de almacenar la carga.
3. Se le realizó mantenimiento al servidor ULTRA Enterprise 450 y se instaló el sistema, GNU/Linux Debian 6, se removió el hardware defectuoso del servidor ULTRA Enterprise que consiste en dos discos duros de 30Gs.
4. Se reubicó el servidor ULTRA Enterprise 450, dentro del área de servidores del ITCoE del cuarto nivel del edificio T3.
5. Se desarrolló el manual para impartir el curso de introducción a PostgreSQL y se entregó al ITCoE.
6. Se impartió el curso de introducción a Java a estudiantes del Cunoc-USAC por medio de la plataforma Webex.

RECOMENDACIONES

1. Reemplazar las baterías internas del UPS para devolverle capacidad operativa total.
2. Cambiar la chapa del salón 010 en donde se encuentra almacenado el UPS por armellas, debido a que, por las bajas temperaturas del salón congela la chapa y es difícil abrir el salón para acceder al UPS.
3. Darle uso al servidor SUN, ya le queda poco tiempo de vida y no existe soporte de hardware y software por parte de SUN Microsystems para este modelo.
4. Utilizar el servidor para aplicaciones no críticas, porque si un componente de hardware falla, no es posible conseguir un reemplazo nuevo, tendría que recurrir a utilizar piezas de segunda mano, ya que se encuentra discontinuado.
5. Grabar los laboratorios impartidos en el ITCoE para que se puedan replicar por medio de Webex para aumentar el alcance de las cátedras de expertos.

BIBLIOGRAFÍA

1. Data Prix. *Arquitectura de Postgre SQL* [en línea]. 2007. <http://www.dataprix.com/72-arquitectura-postgresql>. [Consulta: junio de 2011].
2. DE DIEGO VARONA, Ruben; ROMAY LOPEZ, Oscar Manuel; VEGA MARTINEZ, Jorge. *Arquitectura Sparc: conceptos generales* [en línea]. 2004. www.infor.uva.es/~bastida/OC/GeneralSPARC.pdf [Consulta: junio de 2011].
3. Eaton Corporation, *Powerware 9355 Tie Cabinet Installation Guide*, Eaton Corportation, [ref: 29 de junio de 2010]. Disponible en web: <http://powerquality.eaton.com/Products...services/Backup-Power-UPS/9355.aspx?cx=119>
4. TROLLHUNDER, Lionel. *Sparc-howto* [en línea]. 2004. <http://tldp.org/HOWTO/SPARC-HOWTO.html#toc14> [Consulta: mayo de 2011].
5. Wikipedia Foundation, *Uninterruptible power supply* [en línea]. 2010. wikipedia.org/wiki/Uninterruptible_power_supply [Consulta: junio de 2011].
6. _____. *Postgre SQL* [en línea] 2007. es.wikipedia.org/wiki/PostgreSQL. [Consulta: junio de 2011].

7. _____. *Solaris* (Sistema Operativo [en línea]. 2009.
es.wikipedia.org/wiki/PostgreSQL [Consulta: junio de 2011].