**Unidad2**

**Ejercicio 1**

Se entregará este documento en el aula virtual en la tarea **ENTREGA EJERCICO1 – UNIDAD2** . Para contestar a la mayoría de preguntas se pegarán los pantallazos o recortes que estime oportunos, donde se vea claramente las acciones que se piden en cada pregunta o apartado.

Al final de la actividad tendremos dos elementos:

Un documento en PDF con el resultado de la actividad. Llevará el nombre del ejercicio y vuestro nombre, ejemplo**: EJERCICIO1-UNIDAD2 JUAN.DOCX**

Una máquina virtual con Windows 8, postgreSQL instalado y con la base de datos **EMPRESA1** con su tabla **CLIENTES** y la base de datos **INSTITUTO** y sus tablas correspondientes.

Utilice la máquina virtual creada en Windows 8.1

1.-Conteste a las siguientes preguntas:

1. Diferencia entre partición primaria y extendida

Las particiones primarias son las particiones básicas del disco. Los discos están limitados a 4 particiones primarias, con lo que solo se podrían instalar 4 SO en el disco. Para saltarse esa limitación se crearon las particiones extendidas, en las cuales se pueden instalar hasta 23 particiones lógicas.

1. Cuántas particiones primarias puede haber en NTFS

4

1. Cuántas particiones lógicas pueden crearse en NTFS y donde pueden crearse.

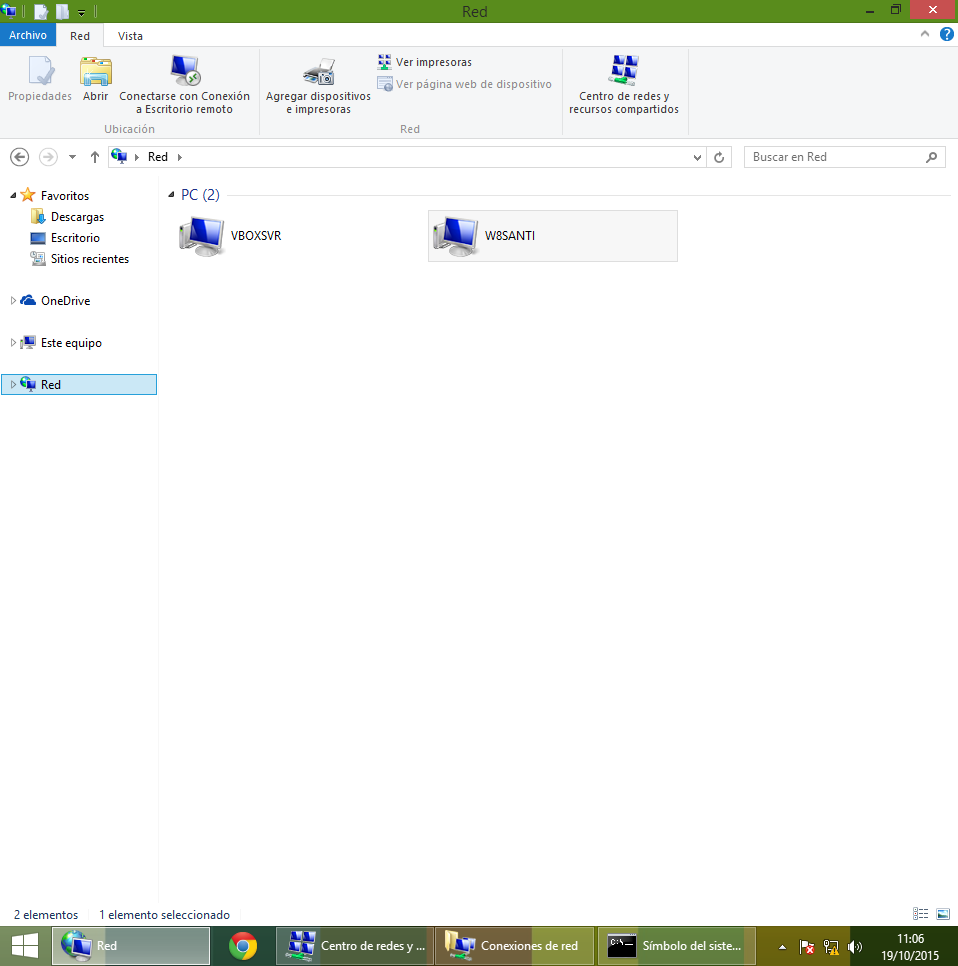
Se pueden crear hasta 23 particiones lógicas

1. Cuál es la diferencia fundamental entre NTFS y FAT32.

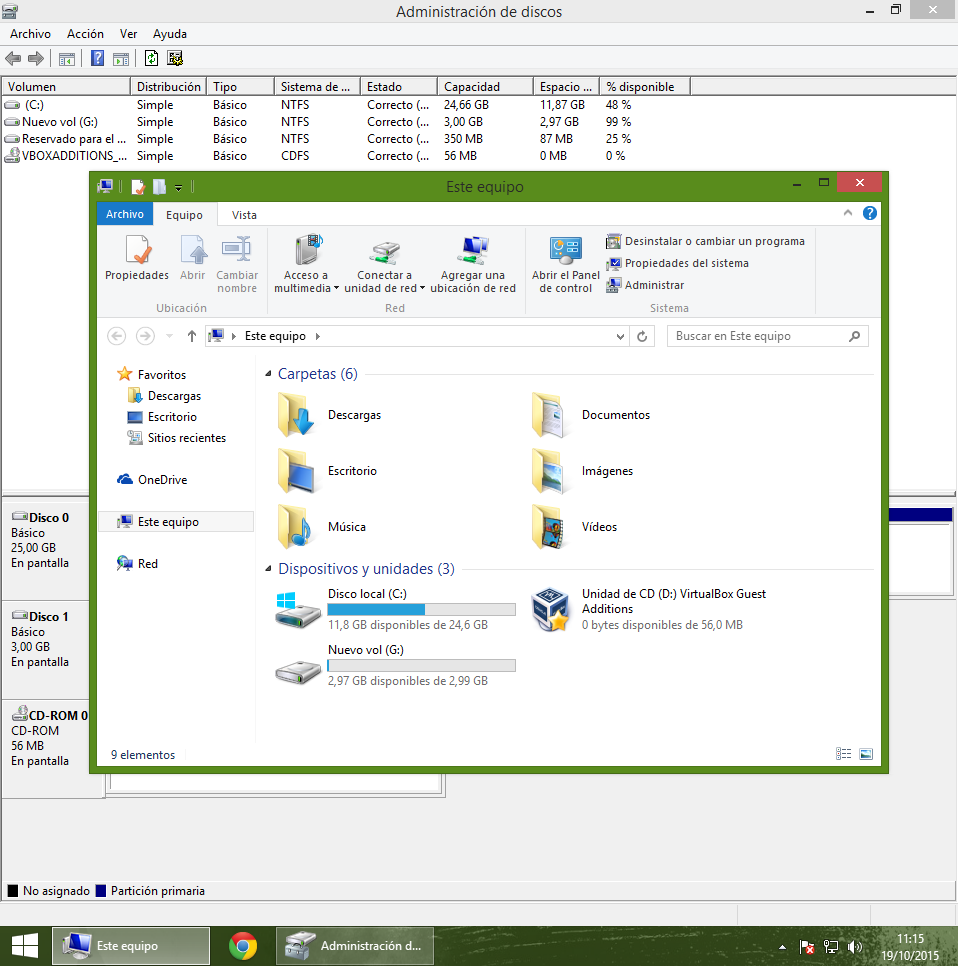
La diferencia fundamental es el tamaño máximo de disco, así como el tamaño máximo de archivos (limitado a 4GB en fat32) aunque existen varias diferencias de seguridad y compatibilidad.

2.- Se hará una copia de la máquina mediante la opción de **CLONACIÓN ENLAZADA**. A la nueva máquina se le pondrá el mismo nombre pero con el número 2 al final y se inicializará su mac. Las máquinas estarán dentro de una red interna, pero la máquina original o primera tendrá salida a internet. Las IP’s de las máquinas a nivel red interna podrán ser las que desee, dentro del rango 192.168.x.x

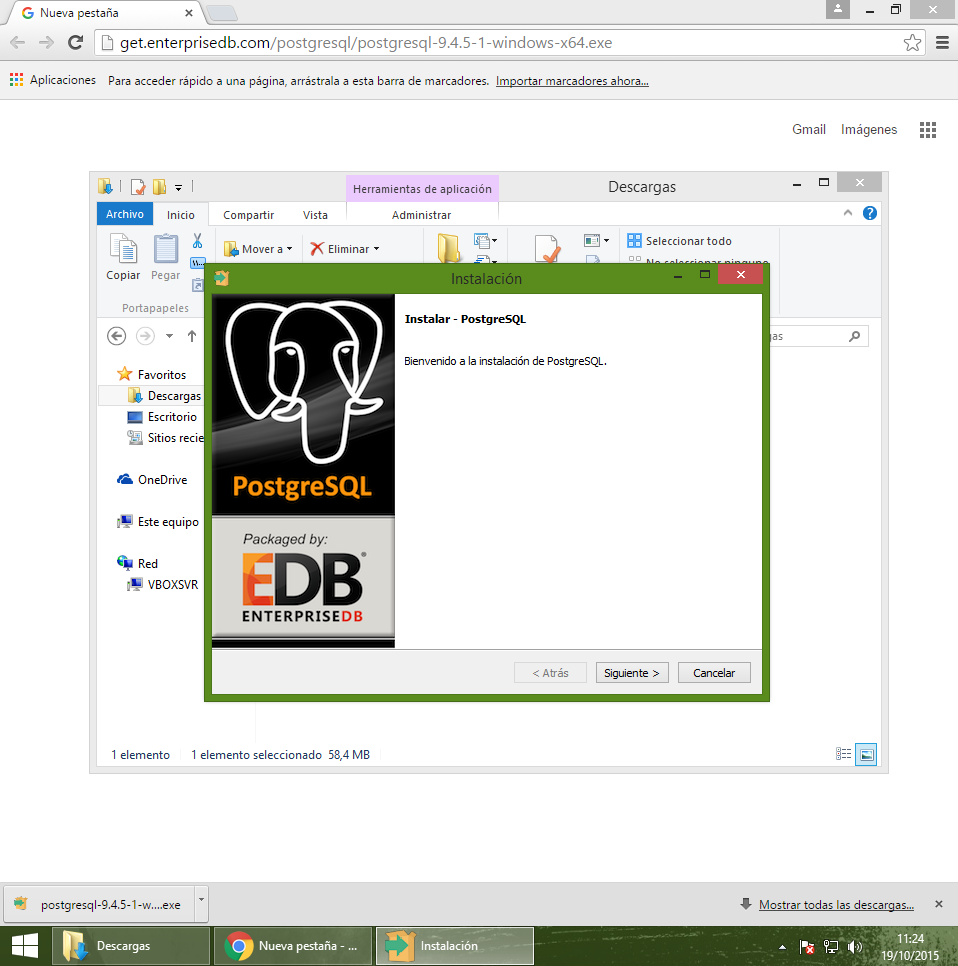
3.- Conseguir que las dos máquinas se vean entre sí. Compartiendo una carpeta por ejemplo.



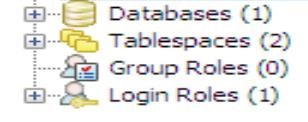
4.- Crea un nuevo disco virtual dentro de la primera máquina del mismo tipo que el anterior, es decir, IDE o SATA con **3 GB** de capacidad. Haz que aparezca en el explorador de Windows como unidad **G**



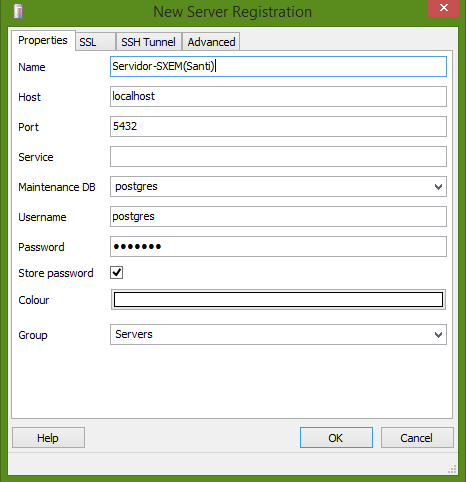
5.- Entrar en la web de [www.postgresql.org.es](http://www.postgresql.org.es/) y baja la versión para windows de 64 bits. Se instalará PostgreSQL sin instalar Stack Builder. La contraseña que le pide la instalación será: abc123. Se pegará el pantallazo inicial del instalador y pgAdmin III ya funcionando. Se instalará en la primera máquina.



6.- En este entorno gráfico de pgAdmin III, conectarse con el servidor local creado utilizando la contraseña indicada en la instalación. Verá dentro del servidor los elementos que se muestran en la imagen.



Ahora vamos a eliminar el servidor que existe y vamos a crear un servidor nuevo con los siguientes datos:

Una vez creado entre en él.

Identifique los componentes o elementos que tiene. Defina **Bases de Datos, Catálogos, Esquemas.**

7.- Intente crear una base de datos llamada **EMPRESA1-vuestronombre**. Se puede crear en sql o a nivel gráfico.

*-- Database: "Empresa1-Santi"*

*-- DROP DATABASE "Empresa1-Santi";*

*CREATE DATABASE "Empresa1-Santi"*

*WITH OWNER = postgres*

*ENCODING = 'UTF8'*

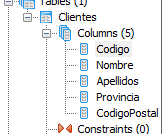
*TABLESPACE = pg\_default*

*LC\_COLLATE = 'Spanish\_Spain.1252'*

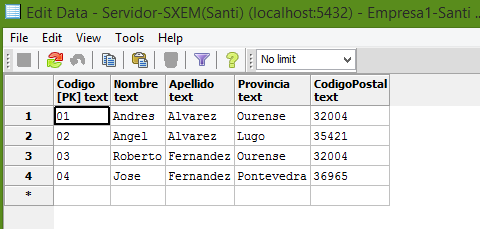
*LC\_CTYPE = 'Spanish\_Spain.1252'*

*CONNECTION LIMIT = -1;*

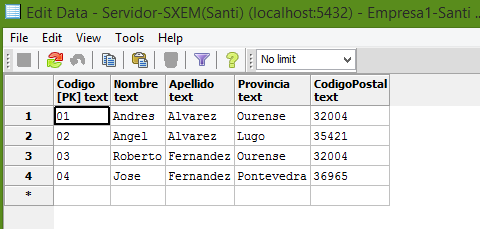
8.- Dentro de esa base de datos cree una tabla de **CLIENTES** con los campos: **CODIGO, NOMBRE, APELLIDOS, PROVINCIA, CODIGO POSTAL**, poniendo claves, tipo de datos y si son nulos o no.



9.-Rellene mediante sql o el entorno gráfico de la tabla creada anteriormente, al menos 4 registros.



10.- Cómo haría una consulta donde nos muestre todos los registros de la tabla. Hágalo de forma gráfica y en sql.

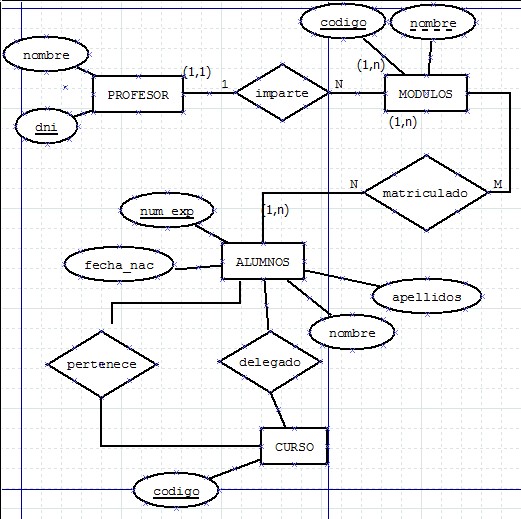


SELECT \* FROM Clientes

11.- Lo mismo que en el caso anterior pero ahora los que son de una determinada provincia.

SELECT \* FROM Clientes WHERE Provincia = “Ourense”

12.- Dado el siguiente Entidad – Relación:



1. Identifica las tablas que salen del mismo y los campos de cada una, incluyendo las claves primarias y foráneas si hubiera.
2. Implementa esas tablas en PostgreSQL en una nueva base de datos llamada **INSTITUTO\_VUESTRONOMBRE**.
3. Rellena con datos coherentes las tablas al menos 5 registros en cada una de ellas.
4. Realiza varias consultas simples donde entren parte de las tablas creadas. Consultas coherentes.

CREATE TABLE Profesor(

DNI text PRIMARY KEY,

Nombre text not null);

CREATE TABLE Modulos(

codigo text PRIMARY KEY,

nombre text not null,

dniProfesor text);

CREATE TABLE Alumnos(

numExp text PRIMARY KEY,

nombre text not null,

apellidos text not null,

fechaNac date,

curso text);

CREATE TABLE Curso(

codigo text PRIMARY KEY,

numExpDelegado text);

CREATE TABLE Matriculado(

codigoModulo text,

numExpAlumno text);

ALTER TABLE Matriculado

ADD CONSTRAINT PK\_Matriculado PRIMARY KEY(codigoModulo, NumExpAlumno);

ALTER TABLE Alumnos add

CONSTRAINT "FK\_Alumnos\_Curso" FOREIGN KEY (Curso)

REFERENCES Curso (codigo) MATCH SIMPLE

ON UPDATE NO ACTION ON DELETE NO ACTION;

ALTER TABLE Modulos add

CONSTRAINT "FK\_Alumnos\_Curso" FOREIGN KEY (DniProfesor)

REFERENCES Profesor (dni) MATCH SIMPLE

ON UPDATE NO ACTION ON DELETE NO ACTION;

ALTER TABLE CURSO

ADD CONSTRAINT "FK\_Cusro\_Alumno" FOREIGN KEY (numExpDelegado)

REFERENCES Alumnos(numExp) MATCH SIMPLE

ON UPDATE NO ACTION ON DELETE NO ACTION;

ALTER TABLE Matriculado

ADD CONSTRAINT "FK\_Matriculado\_Modulo" FOREIGN KEY (codigoModulo)

REFERENCES Modulos(codigo) MATCH SIMPLE

ON UPDATE NO ACTION ON DELETE NO ACTION,

ADD CONSTRAINT "FK\_Matriculado\_Alumno" FOREIGN KEY (numExpAlumno)

REFERENCES Alumnos(numExp) MATCH SIMPLE

ON UPDATE NO ACTION ON DELETE NO ACTION;

INSERT INTO Curso VALUES ('01'),('02'),('03'),('04'),('05');

INSERT INTO Alumnos VALUES

('02','Nombre1','Apellido1','1993-10-08','01'),

('03','Nombre2','Apellido2','1963-08-06','02'),

('04','Nombre3','Apellido2','1973-11-05','03'),

('05','Nombre4','Apellido3','1983-10-02','04'),

('06','Nombre5','Apellido2','1999-12-12','05');

INSERT INTO Profesor VALUES

('12345678a','Nombre1'),

('12345678b','Nombre5'),

('12345678c','Nombre4'),

('12345678d','Nombre3'),

('12345678r','Nombre2');

INSERT INTO Modulos VALUES

('01','Modulo1','12345678a'),

('02','Modulo2','12345678a'),

('03','Modulo3','12345678a'),

('04','Modulo4','12345678a'),

('05','Modulo5','12345678a');

INSERT INTO Matriculado VALUES

('01','01'),

('01','02'),

('01','03'),

('01','04'),

('01','05');

**Resumen de la práctica:**

* **Clonación de máquinas en VirtualBox**

* **Creación de un grupo de trabajo entre dos máquinas.**
* **Instalación de PostreSQL**
* **Primeros pasos dentro de PostreSQL y pgAdmin III**
* **Creación de la base de datos INSTITUTO**