

Rayniel Ramos González | Base de datos II

SBDE (Sistema de Base de Datos de Estudiantes)

# Objetivo del proyecto:

El proyecto tiene por objetivo crear un sistema de base de datos para la Escuela de Cuadros, en este sistema se debe poder guardar la información referente a las tesis realizadas por los estudiantes, así como información referente a los propios estudiantes, entre ésta, los cursos en los que se encuentran matriculados, los organismos a los que pertenecen, etc. Además de poder realizar consultas estáticas acerca de los estudiantes que se encuentran en un tipo de curso determinado, y realizar consultas dinámicas acerca de información referente a las tesis.

La aplicación que se realizó cumple con los requisitos pedidos, provee una interfaz simple e intuitiva, es de fácil instalación y fue realizada completamente con software libre lo cual redunda en un fuerte ahorro presupuestario, en el caso de que se tuviese que pagar licencia, y en el uso de nuevas prácticas y políticas de programación que se encuentran en auge por el cooperativismo que fomentan y la innovación.

# Requerimientos:

* **Requerimientos técnicos:**

No se requiere un gran poder de cómputo. Basta un servidor con 1 GHz de procesamiento y 4 GB de RAM, y una cantidad de disco duro en dependencia del tamaño de la base de datos, o sea, la capacidad actual de la mayoría de las computadoras personales.

* **Requerimientos del software:**

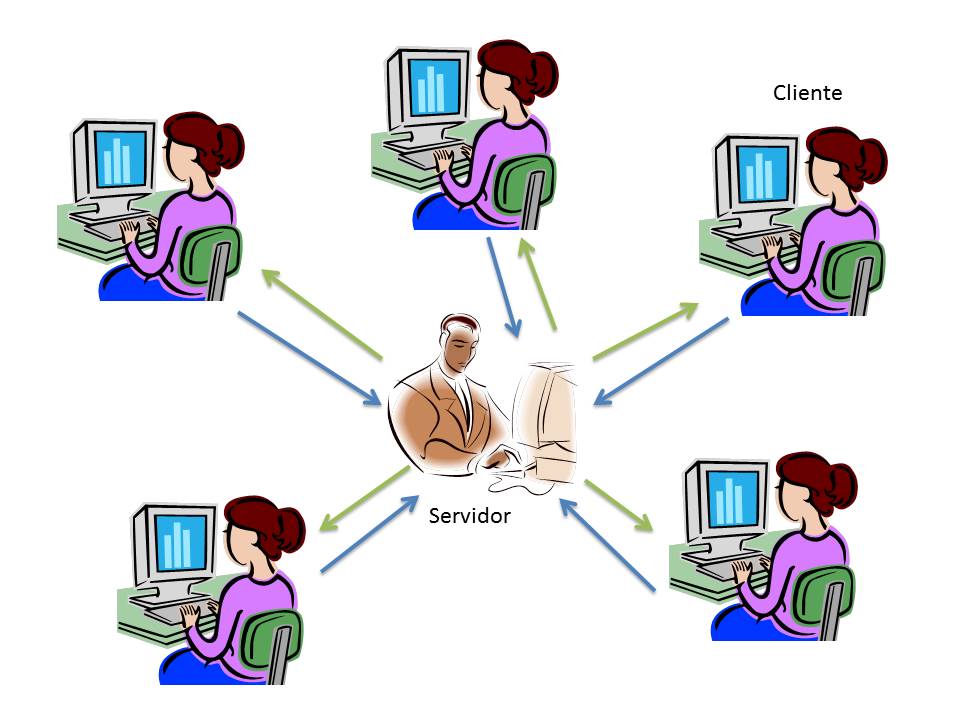
Se requiere de SQLite 3 (incluido en la instalación de Django 1.8)

Se requiere de Django 1.8 (descarga gratuita desde su sitio hospedero o desde Pypi)

Se requiere de Python 3.4 (descarga gratuita desde su sitio hospedero)

# Estructura de la red:

La siguiente imagen muestra el esquema que seguiría una red que contenga el Sistema de base de datos, la forma en que los distintos clientes conectados a la red, podrían hacerle consultas, agregar información, editarla o eliminarla. Como bien se aprecia es un diseño cliente - servidor, bastante simple, puesto que este diseño basta para satisfacer las necesidades de la aplicación, este diseño presenta distintas ventajas, entre ellas, centralización de la información, no es necesario, si hubiera que hacer alguna modificación, ya sea en la base de datos o en la aplicación, conectarse a distintos servidores, ni trabajar con distintos sistemas operativos o protocolos de red, problemas comunes en la descentralización, basta que podamos acceder a una sola máquina donde estará nuestro sistema y modificar todo lo necesario, además del bajo consumo de recursos, puesto que al ser necesario un solo servidor, no hay porque luchar con una gran diversidad de hardware y software.



# Esquema de conexión al sistema de base de datos:

La siguiente imagen muestran distintas capas de la aplicación, y cómo interactúan los módulos más importantes de la aplicación. Se puede apreciar cómo los distintos clientes interactúan con el sistema de base de datos a través de un servidor Apache, el cual ejecutará un intérprete de Python, a través de un plug-in, y como éste ejecutará Django, y éste último trabajará con una base de datos SQLite 3, el cual ya incorpora por defecto. Esta arquitectura provee de una gran modularidad y especialización por parte de sus distintos componentes.



Servidor Apache

Django

Python

SQLite3

# Diseño conceptual:

La imagen a continuación muestra el esquema del diseño conceptual de la base de datos, en muchos casos, no se especifica un atributo como llave primaria, Django en ese caso genera uno automáticamente.

