TP N°2

BASE DE DATOS II

*“Backup continuo y coneccion BD con pagina web”*

Alumnos:

* Vargas, Lucas
* Pesado, Gonzalo

Materia:

* Base de datos II

Profesor:

* Ignacio Bisso

Semestre:

* 1er Semestre 2017

Fecha de entrega:

* 24-06-2017

**Informe**

Para comenzar, hay que aclarar que el objetivo del tp es el de realizar una sencilla página web, la cual se conectará con nuestra base de datos y realiza algunas operaciones básicas sobre ella, además de, implementar un sistema de backup continuo ya que simularemos una falla en la base de datos y tendremos que recuperar la totalidad de la información.

Para realizar esto pasaremos a enumerar las herramientas que utilizamos:

* **POSTGRESQL**: Lo utilizamos para crear y gestionar la base de datos, además de utilizar sus funciones de recuperación de datos para realizar y levantar el backup luego de la falla.
* **HTML/JS/CSS/bootstrap**: Lo utilizaremos para realizar la página web.
* **PHP**: Lo utilizaremos para lograr la coneccion con la base de datos.Utilizamos una **versión 5.6**.
* **Prevención de inyección sql**: Utilizando statement.
* **Encriptación de contraseñas**: Se utilizó el método password\_hash and password\_verify con la encriptación en PASSWORD\_BCRYPT para garantizar esto. Desistimos de utilizar MD5 debido a que el programa se hacía muy lento debido a que no es un tipo de encriptación sino que la técnica es un hash mezclado con salts y caracteres especiales, los cuales sin dudas tardan mucho más en transformar la contraseña.
* **Diseño en clases:** Garantiza concentrar la lógica en pequeñas clases y así poder tener mejor mantenibilidad del código.
* **Sistema instalado completamente desde 0:** En la máquina de presentacion.

Además se pidió que cuando entre al programa muestre las personas que miraron las mismas películas, por lo que la query pensada será la siguiente:

select distinct pqv1.usuario, pqv2.usuario

from pelis\_que\_vio pqv1, pelis\_que\_vio pqv2

where pqv1.usuario < pqv2.usuario

and not exists

( select \* from pelis\_que\_vio pqv3 where pqv1.usuario = pqv3.usuario

and not exists

(select \* from pelis\_que\_vio pqv4 where pqv2.usuario = pqv4.usuario and pqv3.pelicula\_id=pqv4.pelicula\_id))

and not exists

( select \* from pelis\_que\_vio pqv3 where pqv2.usuario = pqv3.usuario

and not exists

(select \* from pelis\_que\_vio pqv4 where pqv1.usuario = pqv4.usuario and pqv3.pelicula\_id = pqv4.pelicula\_id));

Donde la idea general será buscar una tupla de usuarios que cumpla que:

no existe una película que haya visto el usuario1 que no haya visto el usuario 2

además de no existir una película que haya visto el usuario 2 que no haya visto el usuario 1.

Por último se añadió una condición extra donde se resuelve el problema de que no aparezca la tupla inversa, la idea se basa en tratar que las tuplas que aparecen son solamente en las que el usuario1 encabeza la tabla resultante en la query (por ende no aparecería ninguna tupla inversa, la cual estaría encabezada por el usuario 2).