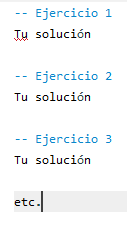
# **Instrucciones**

Importa el script “universidad2.sql” en MySQL Workbench.

Renombra la base de datos con tu nombre de esta manera: “nombre\_universidad”.

La entrega serán las soluciones de cada apartado en un solo script, de este modo:



Al final de la clase debes subir al servidor ese script como lo tengas, aunque falten ejercicios por resolver.

# **Bloque 1**

1. Lista de profesores ordenados alfabéticamente.
2. Lista de alumnos ordenados por ciudad y por apellido, alfabéticamente.
3. Lista de alumnos solo de Barcelona, alfabéticamente en sentido descendente.
4. Lista de alumnos no matriculados en ninguna asignatura:   
   nif, nombre, apellido1
5. Alumno hombre más joven matriculado en 2017:   
   nombre, apellido1 (atención, que hay personas no matriculadas)
6. Profesor de más edad que dio cursos en 2019
7. Asignatura con más alumnos por año
8. Asignatura con más alumnos mujeres y cuantas son
9. Asignatura con menos alumnos hombres en 2018
10. 10 nombres de alumnos más usuales, ordenados de mayor a menor
11. Ciudad con más alumnos hombres en 2016: ciudad, cantidad de alumnos
12. Las tres asignaturas con mayor número de alumnos inscritos en toda la historia de más a menos, indicando la cantidad:   
    Nombre de la asignatura, cantidad de alumnos
13. Se ha decidido conceder una beca a los tres alumnos casados de mayor edad. Identifica quienes son, mostrando su nif, nombre y apellido1.
14. Mostrar por parejas los alumnos que viven en la misma ciudad. Deben aparecer los apellidos de cada uno y la ciudad, sin parejas duplicadas. El orden debe ser por el nombre de la ciudad y el apellido de la primera columna. Ejemplo: 
15. Gestión de usuarios. Has de guardar de código de cada acción:
    1. Muestra los usuarios y sus permisos.
    2. Crea un usuario llamado cliente que solo se pueda conectar por localhost y permisos para hacer select y update. Su contraseña será ‘1234abcd’.
    3. Quítale el permiso de update.
    4. Borra el usuario.

# **Bloque 2**

1. Procedimiento Almacenado para obtener la ciudad con más alumnos por año y sexo (serán los parámetros). Se llamará pa\_ciudad\_sexo\_anyo:   
   ciudad, cantidad de alumnos hombres, cantidad de mujeres, año
2. Procedimiento almacenado para obtener los alumnos matriculados por asignatura, sexo y año de inicio (serán los parámetros).   
   Se llamará pa\_alumnos\_asig\_sexo\_anyo.   
   Por ejemplo, cuando tendrá esta respuesta cuando se ejecute así :   
   call pa\_alumnos\_asig\_sexo\_anyo (‘Estadistica’, ‘M’, 2017):
3. Función para obtener el profesor (nombre, apellidos y asignatura ) con más alumnos por año y sexo (serán los parámetros). Se llamará fu\_profesor:  
   La respuesta será : “El profesor nombre\_profesor apellido\_profesor ha tenido Y alumnas y X alumnos en XXXX”.
4. Crea un procedimiento para inscribir(que no matricular) alumnos, introduciendo todos los datos de la tabla “persona”. Se llamará pa\_inscripcion(todos\_los\_datos):  
   Si se introduce un nif repetido debe de avisar del error.
5. Crea otro procedimiento para matricular alumnos a partir de su nif. Los parámetros serán nif, asignatura, año de inicio.   
   Se llamará pa\_matricula(nif, asignatura, anyo\_inicio)  
   Si ya está matriculado de esa asignatura ese año debe avisar del error.
6. Crea una función que muestre de qué asignaturas y en que año se ha matriculado un alumno. El parámetro de entrada será el nif. La salida debe mostrar nif, nombre, apellido1, asignatura, año de inicio.   
   Se llamará fu\_info\_matriculas\_alumno.