

# Tarea Base de Datos SQL

*Alejandro Sierra Fernández*

En este documento se desarrollarán dos de los tres objetivos del proyecto final “Creación de una base de Datos”: El análisis y el diseño de esta. El tercer objetivo, la implementación, se entregará a través de un script SQL, tal y como se requiere.

Empezamos por mostrar el enunciado, pues a través de éste obtenemos la información necesaria que definirá la estructura y las características de nuestra Base de Datos:

*El móvil se ha convertido en algo no sólo cotidiano, sino necesario para la mayoría de las actividades que realizamos cada día, bien sean sociales o personales. Se ha convertido en el centro de nuestra vida, imprescindible en el ámbito profesional o de ocio. Las empresas de tecnología han descubierto ahí un mercado, por eso han proliferado las empresas de creación de aplicaciones para móviles y las tiendas de aplicaciones (app stores). El modelo de negocio de las tiendas de aplicaciones se basa en una comisión, por lo general del 30%, sobre apps descargadas y las ventas que se realicen desde las mismas. Si tenemos en cuenta que estamos ante un mercado de millones de apps cuya única manera de descarga es a través de las tiendas de aplicaciones, no cabe duda que las tiendas de aplicaciones son una de las mayores y más robustas fuentes de ingresos del ecosistema móvil.*

*Las aplicaciones las realizan empresas de servicios y las suben a las plataformas o tiendas de aplicaciones para ser descargadas por los usuarios de móviles, cada aplicación puede tener o no un coste.*

*Vamos a crear una base de datos donde guardar la información de las tiendas de aplicaciones: Apple (App Store), Google Android (Google Play Store), BlackBerry (App World), WindowsPhone (MarketPlace), Nokia (OVI Tienda), HP (AppCatalog), Amazon (Appstore), ... De la tienda sabemos el nombre, que es distinto para cada tienda, quien la gestiona (Android, Apple, Amazon, ....) y dirección web.*

*De las empresas de servicios que realizan las aplicaciones (apps), conocemos su nombre, país en el que paga sus impuestos, año de creación, correo electrónico y página web, sabemos que estas empresas no tienen ninguna conexión con las tiendas de aplicaciones.*

*En la empresa hay empleados, pero debido al dinamismo en estos tipos de trabajo y, a la oferta y la demanda en el sector, el empleado puede haber trabajado en varias empresas del sector e incluso puede trabajar en la misma empresa en distintos periodos de tiempo, nos interesa conocer la experiencia profesional del empleado. Además, del empleado nos interesa el DNI, dirección (calle, número, código postal), correo*

electrónico y teléfono. En la empresa se realizan aplicaciones, de las que conocemos su nombre que es único, el código de aplicación, la fecha en la que se comenzó a realizar y fecha de terminación, la categoría o categorías en las que se puede incluir (entretenimiento, social, educación, ...), espacio de memoria, precio.

Las aplicaciones son subidas a las tiendas o plataformas. Una misma aplicación puede ser subida a varias tiendas, por supuesto una tienda tiene muchas aplicaciones. Cada aplicación está realizada por un grupo de empleados. La aplicación es dirigida por un empleado y un empleado puede dirigir varias aplicaciones.

Un usuario puede descargar o no aplicaciones, pero no puede descargar dos veces la misma aplicación. El usuario puede puntuar de 0 a 5 cada una de las aplicaciones que se descarga y hacer comentarios referentes a la misma. Del usuario conocemos el número de cuenta que es único, nombre, dirección y si se descarga la aplicación en el teléfono conocemos el número de móvil.

Entre otras informaciones, nos interesa saber la fecha en la que se realizan más descargas, el país de los usuarios que más aplicaciones se han descargado y la puntuación media de cada una de las apps.

Teniendo el enunciado siempre presente pasamos al diseño conceptual de la base de datos, en concreto del modelo entidad-relación. Al ser el modelo de un tamaño considerable, lo vamos describiendo por partes para facilitar su estudio y visualización.

En el texto encontramos que las aplicaciones se suben a las tiendas, y entendemos que una tienda puede tener distintas aplicaciones. Por tanto, nos encontramos con una relación de tipo  $N:N$ . Además, se nos describen los atributos que deben caracterizar a cada una de estas dos entidades. De esta forma la primera relación la describiríamos del siguiente modo:

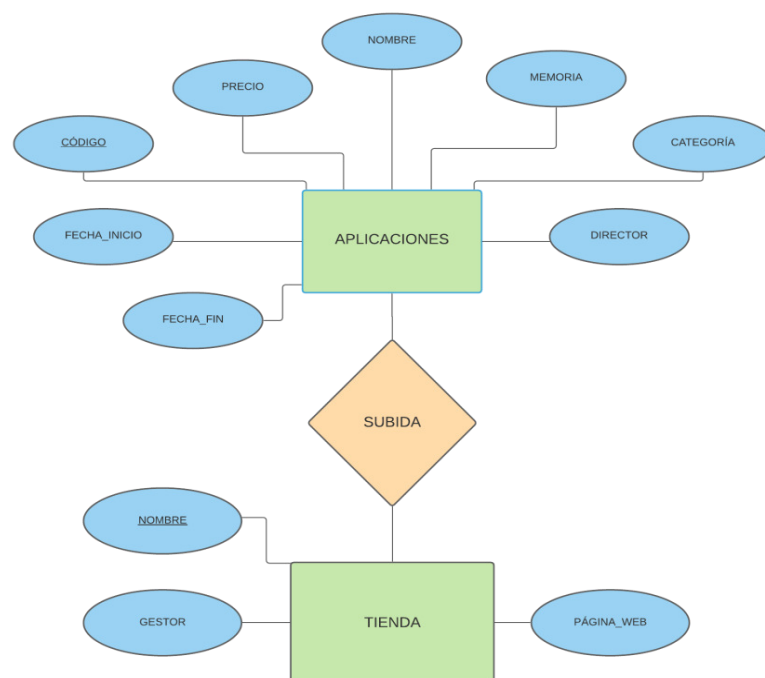


Imagen 1: Relación de subida entre las Apps y las tiendas

Recordamos que los atributos subrayados son las claves primarias de las entidades, es decir, aquellos atributos que sirven para diferenciar unas instancias de otras.

A continuación, el texto nos hace saber que existe una relación de contratación entre las empresas y sus empleados, informándonos de las propiedades de cada una de las entidades. Es importante también destacar que un empleado puede haber trabajado para distintas empresas, e incluso haber trabajado para la misma en diferentes periodos de tiempo. A pesar de que nos volvemos a encontrar con una relación del tipo  $N:N$ , el dato anterior nos impide que podamos escribir como claves primarias de la relación de contratación únicamente las claves primarias de las entidades empresa y empleado. Esto se debe a que una empresa y un empleado no definen una relación de contratación única, puesto que el empleado puede haber terminado un contrato con la empresa y haber vuelto a firmar uno nuevo después de un cierto tiempo. Es por esto por lo que definimos como clave primaria de la relación un código asociado a cada contrato, dejando el código de la empresa y el código del empleado como atributos generales.

Además, como se nos pide el historial de cada empleado, deberíamos definir una fecha de inicio y una fecha de fin de cada contrato. La posibilidad de que muchos de los contratos no hayan terminado o sean indefinidos nos dejaría muchos *NULL* en la columna reservada al atributo "*fecha\_fin*". Esto podríamos evitarlo distinguiendo entre dos relaciones de contratación diferente entre los empleados y las empresas, una relación para los contratos que sigan vigentes, que no dispondría del atributo "*fecha\_fin*", y otra relación para los contratos que ya hayan expirado. De este modo nuestro modelo E-R para este caso concreto quedaría del siguiente modo:

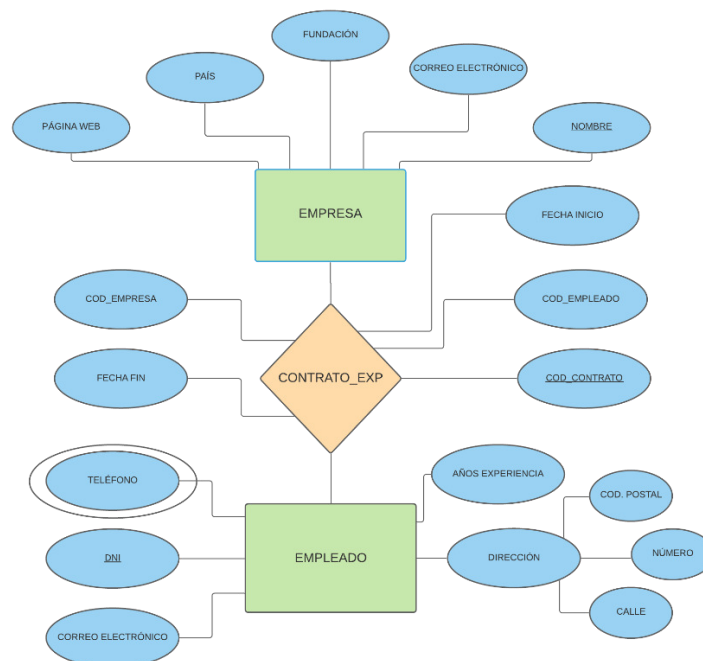


Imagen 2: Relación de contratación entre empleados y empresas (contratos expirados)

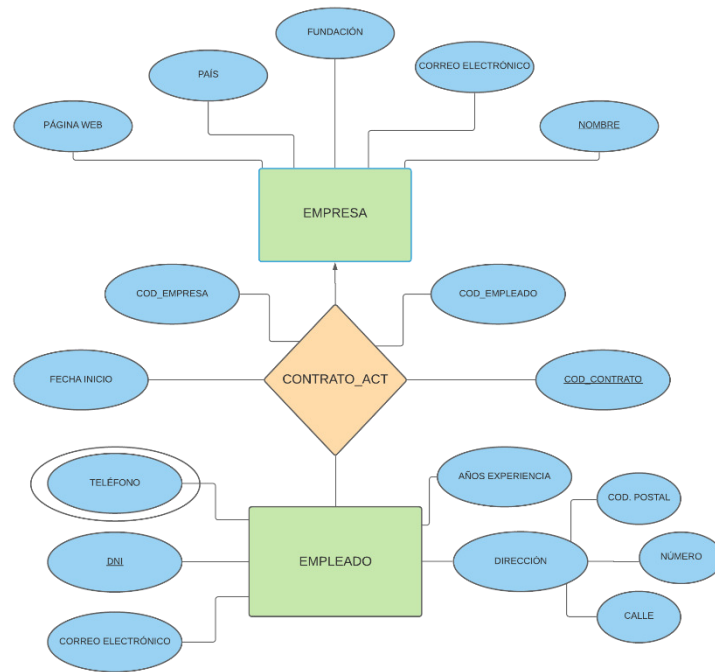


Imagen 3: Relación de contratación entre empleados y empresas (contratos vigentes)

Cabe destacar también que el atributo *teléfono* aparece incluido dentro de un doble contorno ovalado. Esto hace referencia a que se trata de un atributo multivalorado, es decir, que puede contener varios valores simultáneamente. De este modo estamos asumiendo que un empleado puede tener más de un número de teléfono. También señalamos que en la entidad *empleado* aparece el único atributo compuesto del modelo, es decir, el único atributo constituido por otros atributos. Se trata de *dirección*, formado por *cod.postal*, *número* y *calle*.

Posteriormente nos dan la información de que las empresas desarrollan las aplicaciones, de las que conocemos su nombre, el código de la aplicación, la fecha en la que se comenzó a realizar, la fecha en la que se terminó, el precio, la memoria que ocupa y la categoría. Como nos especifican que el nombre de la aplicación es único, elegimos este atributo como clave primaria, aunque el código también debería valer.

Consideramos también que la relación es de tipo 1: *N*, es decir, que una empresa puede hacer varias aplicaciones, pero que una aplicación es desarrollada única y exclusivamente por una empresa. Con todo esto, el esquema perteneciente al modelo E-R referente a esta parte quedaría del siguiente modo:

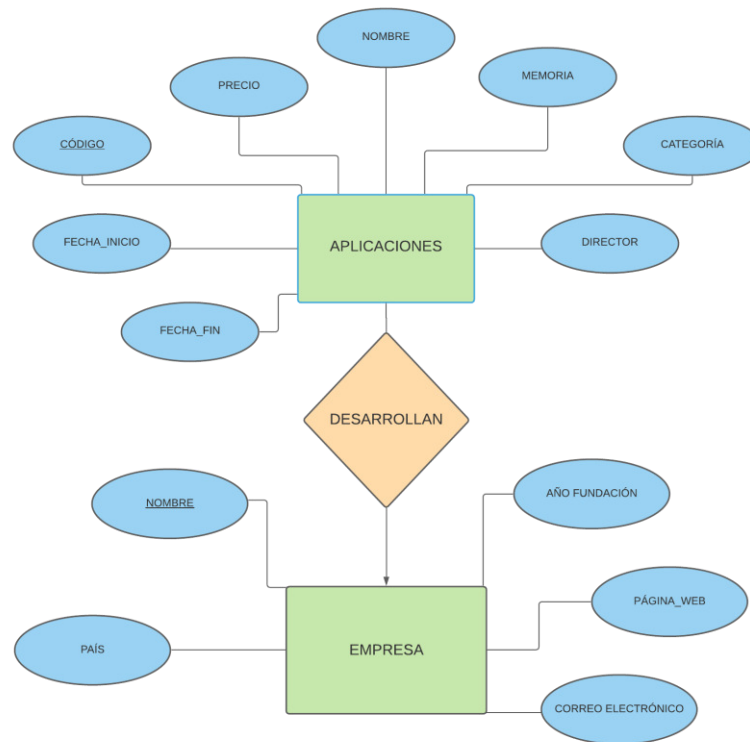


Imagen 4: Relación de desarrollo entre Apps y empresas

Continuamos leyendo y nos damos cuenta de que una aplicación puede ser subida a varias tiendas y que una tienda, evidentemente, puede tener varias aplicaciones. De las tiendas también encontramos las propiedades que tienen, que son: el nombre, el gestor y su página web. Además, se nos dice que una aplicación está hecha por un grupo de empleados y dirigida por un solo empleado únicamente.

El párrafo anterior, entonces, nos da pie a generar tres nuevas relaciones. Primero, una relación “subida” de tipo  $N:N$  entre las tiendas y las aplicaciones. Segundo, una relación “hace” también de tipo  $N:N$  entre los empleados y las aplicaciones y, finalmente, una relación “dirige” de tipo  $1:N$  entre empleados y aplicaciones. Esto es así porque consideramos que un empleado puede dirigir varias aplicaciones simultáneamente. Los esquemas quedarían del siguiente modo:

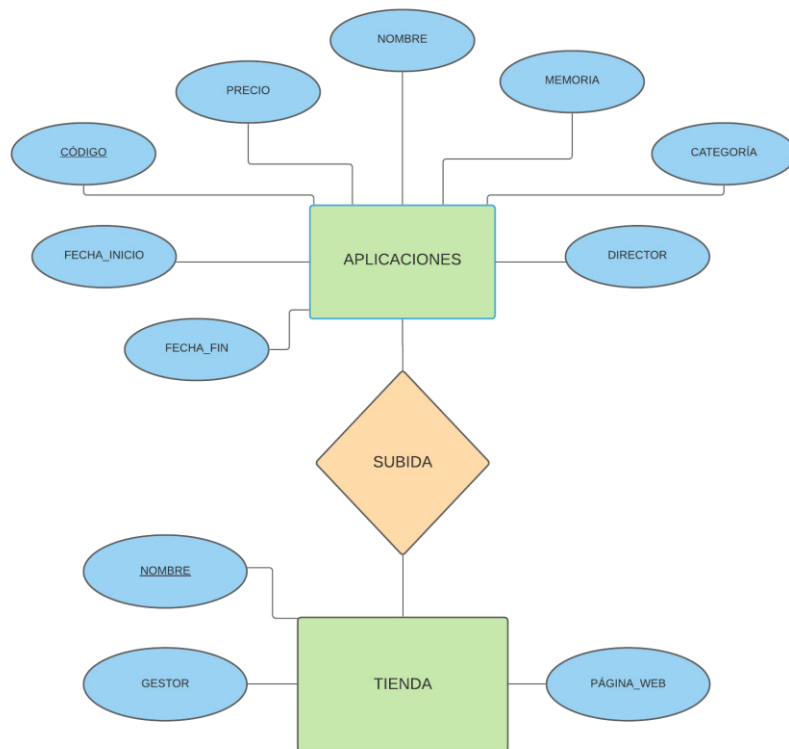


Imagen 5: Relación de subida entre las aplicaciones y las tiendas

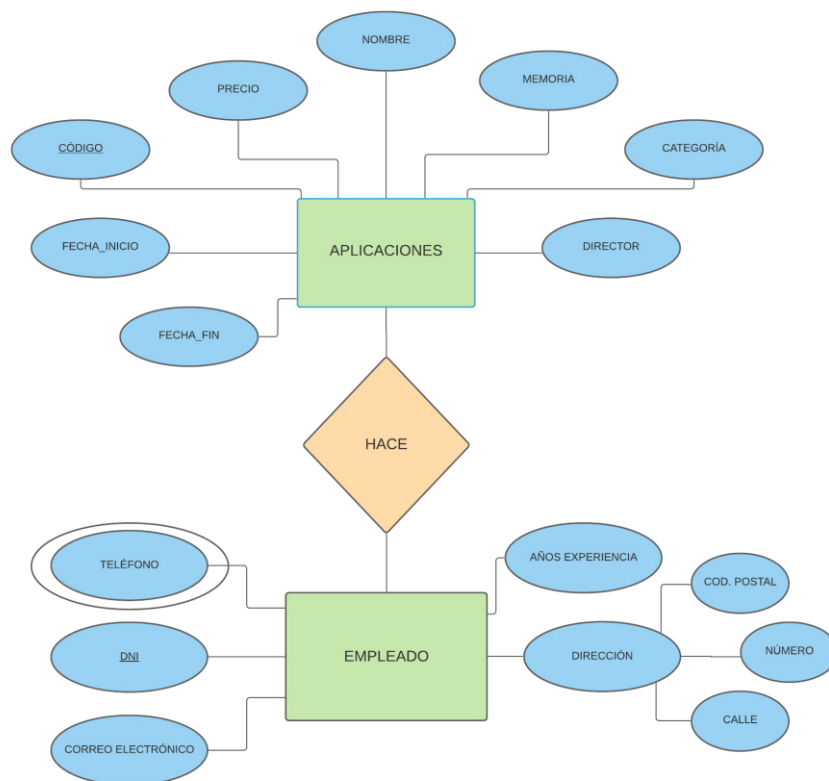


Imagen 6: Relación entre las aplicaciones y los empleados que las hacen

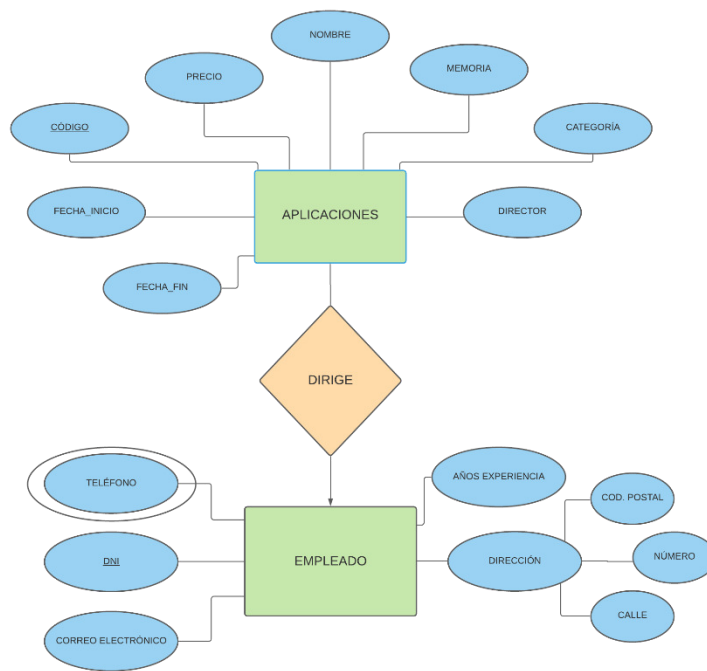


Imagen 7: Relación entre las aplicaciones y los empleados que las dirigen

Cabe destacar que no encontramos ningún atributo derivado, es decir, ningún atributo que pueda obtenerse a partir de otro atributo. Tampoco encontramos ninguna entidad débil, o entidad cuya existencia dependa de otra entidad.

Para finalizar incluimos una imagen con el modelo en su totalidad, donde se muestran las entidades y las relaciones entre estas. Eliminamos los atributos para mejorar la claridad del esquema e introducimos las relaciones de coordinación entre las diferentes entidades.

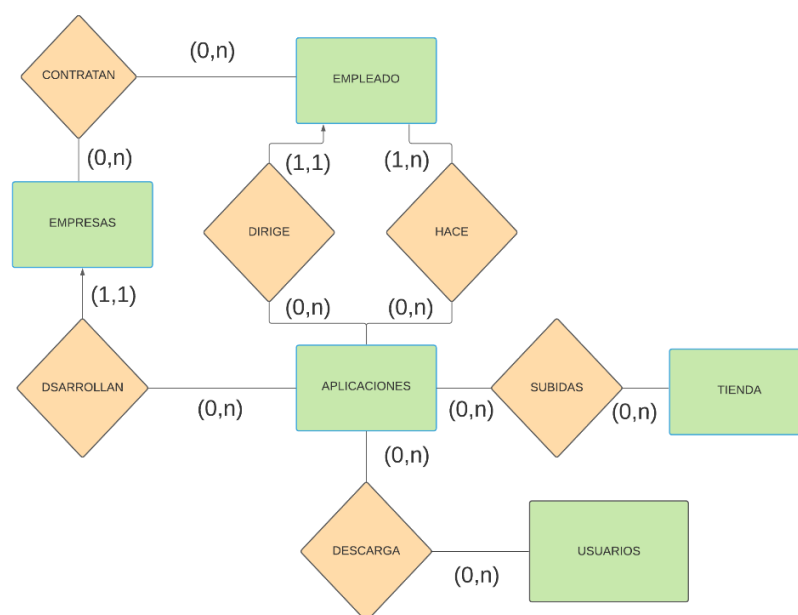


Imagen 7: Modelo E-R (sin atributos)

En el esquema superior, por sencillez, no hemos incluido la diferenciación que hemos hecho anteriormente entre los dos tipos de contrato que podemos considerar. Es útil observar también cómo, efectivamente, se cumple el requisito que nos imponía el enunciado con respecto a la inexistencia de una relación directa entre las empresas y las tiendas.

Una vez hemos diseñado y descrito nuestro diagrama entidad-relación, continuamos con su transformación a las tablas que definirán el modelo relacional. Dividimos entre el paso a tabla de las entidades y el paso a tabla de las relaciones:

### 1) Entidades:

- EMPRESA (Nombre, Correo, Fundado, País, Página\_Web)
- EMPLEADO (DNI, Tlfn, Correo, Calle, Número, Cod\_Postal)
- APLICACIONES (Nombre, Código, Fecha\_inicio, Fecha\_fin, Precio, Memoria, Nombre\_empresa, Director)
- USUARIOS (Num-Cuenta , Nombre, Dirección, Tlfn)
- TIENDA(Nombre, Gestor, Página\_web)

### 2) Relaciones:

- HACE(Nombre-App, DNI-empleado)
- SUBIDAS(Nombre-App, Nombre-Tienda)
- DESCARGA(Nombre-App, Cuenta-usuario, Comentario, Num-móvil, Nota)
- CONTRATO\_EXP(Código-contrato, DNI-empleado, Nombre-empresa, Inicio, Fin)
- CONTRATO\_ACT(Código-contrato, DNI-empleado, Nombre-empresa, Inicio)

Es importante remarcar que en la tabla “*APLICACIONES*” encontramos los atributos “*Nombre-empresa*” y “*Director*”. Estos son las claves primarias de las entidades “*EMPRESA*” y “*EMPLEADO*”, que se incluyen para reflejar las relaciones de tipo 1: *N* “*DIRIGE*” y “*DESARROLLAN*”



Por tanto, en el apartado (2) sólo encontraríamos representadas las relaciones de tipo  $N:N$ , que requieren, por norma general, la creación de una tabla propia que contenga como clave primaria las claves primarias de las entidades involucradas.

Una excepción a esto último son las relaciones “*CONTRATO*”, en las que hemos decidido establecer como clave primaria el código de cada contrato en lugar de las claves primarias de las entidades involucradas. Esto es así porque un mismo empleado puede haber firmado varios contratos con la misma empresa en momentos diferentes de tiempo, por lo que el DNI del empleado y el nombre de la empresa no definen instancias únicas dentro de la relación.

Podríamos pensar que lo mismo ocurre con la relación “*DESCARGA*”, puesto que un usuario en principio podría descargar varias veces la misma aplicación. Sin embargo, esta posibilidad se niega en el enunciado del ejercicio. Es por esto por lo que mantiene la estructura general de las relaciones tipo  $N:N$ .

Por último, destacamos que el atributo “*categoría*”, en principio, debería aparecer en dentro de los atributos de la entidad “*APLICACIONES*” y no lo hace. El motivo es que se trata de un atributo multivalorado, es decir, un atributo que puede tomar diferentes valores para una misma instancia. Es por esto por lo que crearemos una tabla a parte que lo considere. Lo hacemos del siguiente modo:

-CATEGORÍA(Nombre-App, Categoría)

Con esto damos por finalizada la primera parte del proyecto, donde hemos analizado y diseñado nuestra base de datos a través del modelo E-R y su posterior paso a tablas. Nos queda, por tanto, su implementación.

Esta parte final se adjunta en un documento con extensión “.sql” que hemos generado a través de “*MySQL Workbench*”.