

‘Creacion de una Base de Datos’  
Proyecto final Modulo 1



Autor:

Christopher Rodríguez Príncipe

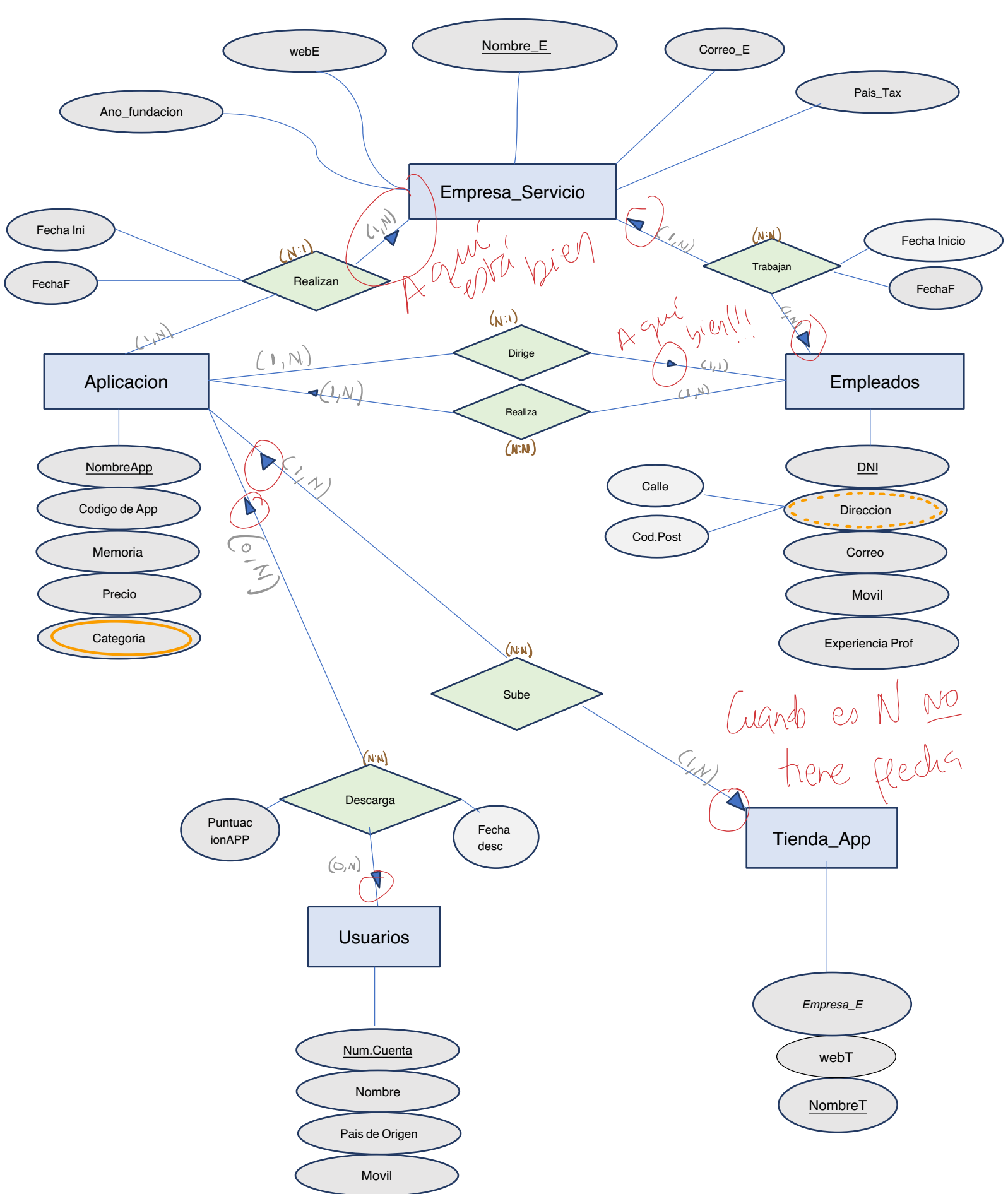
Universidad Complutense de Madrid

## II. Modelo E-R Descripción

El Modelo de Entidad-Relación esta formado por 5 principales entidades compuestas por :

Empresa\_Servicio, Aplicación , Empleados, Usuarios y la tienda donde se vende la aplicación.

Las entidades se conectan a traves de cinco relaciones, en donde tres de las relaciones son compuestas y dan más detalle sobre la relación entre entidades . Las supociones de las cardinalidades y relaciones se ven reflejadas en el diagrama posterior y estan creadas basado en la interpretación del Enunciado. A continuación especificare más detalle sobre las relaciones y las tablas creadas en este proyecto.



### III. Diseño Lógico

1. Empresa\_servicio ( Nombre\_E, Ano\_fundacion , Pais\_Tax, Web, Correo)

2. Aplicaciones( NombreApp, Codigo , Memoria, Precio, Categoria)

3. Empleados( DNI, Nombre\_E, Correo, Movil, Calle, Codigo Postal)

4. Usuario( Num\_Cuenta, Nombre, Pais, Correo)

5. Tienda\_App ( NombreT, Nombre\_E, Web)

6. Realizan ( Nombre\_E, NombreApp, Fecha Inial, Fecha Final)

7. Trabajan( DNI, Nombre\_E, Fecha inicial, Fecha final)

8. Descarga( Num\_cuenta, NombreApp, Puntuacion, Fecha Descarga, Pais)

*Handwritten notes:*

- Solo con DNI es suficiente* (pointing to DNI in item 3)
- ya que no todo tiene fecha final, menos los que están todavía probando en una empresa* (pointing to Fecha Final in item 6)
- No, porque un usuario solo se puede descargar una vez una app determinada* (pointing to Fecha Descarga in item 8)

1. Empresa\_servicio ( Nombre\_E, Ano\_fundacion , Pais\_Tax, Web, Correo)

2. Aplicaciones( NombreApp,Codigo , Memoria, Precio, Categoria)

La tabla Aplicaciones tiene una relación (N:1), dado que muchas aplicaciones le pueden pertenecer a una misma empresa. Tiene cardinalidad (1,N) correspondiente a el minimo de aplicaciones que crea una empresa es 1 y el maximo es N.

3. Empleados( DNI, Nombre\_E , Correo, Movil, Calle, Codigo Postal)

La tabla Aplicaciones tiene una relación (N:1) donde Multiples empleados le pertenecen a una empresa. Con cardinalidad (1,N) dado que la empresa necesita como minimo un empleado para su operación y un máximo de N. DNI es la clave primaria de la tabla, *Nombre\_E* es la clave foranea.

4. Usuario( Num\_Cuenta, Nombre, Pais, Correo)

Tabla Usuario tiene relación (N:M) dode multiples usuarios le pueden pertenecer a una aplicación y multiples aplicaciones le pueden pertenecer a un usuario. Tiene una Cardinalidad de (0,N) ya que el minimo de personas que pueden bajar la aplicación es 0, suponiendo que la premisa de que usuario puede instalar la app o no. Num\_cuenta es la clave primaria de esta tabla.

5. Tienda\_App ( NombreT, *Nombre\_E*, Web)

En una aplicación puede ser subida a varias tiendas, en una tienda existen varias aplicaciones distintas, cuya relación es (N:N) . Cardinalidad de (1,N), una tienda debe tener minimo una aplicación como para ser un negocio rentable. NombreT es clave primaria mientras que hace referencia a la tabla de la empresa(*Nombre\_E*).

9. Realizan ( *Nombre\_E*, *NombreApp*, Fecha Inial, Fecha Final)

Esta tabla es una relacion entre La empresa(*Nombre\_E*) y la tabla de la aplicación(*NombreApp*).

La tabla sirve para evitar repetición de datos entre ambas tablas mencionadas anteriormente. Tiene relación (N:1) dado que múltiples aplicaciones distintas le pueden pertenecer a la misma empresa. Esta tabla tiene como implicación que los empleados que trabajan en la empresa son los que realizan el labor de desarrollo de la aplicación.

6. Trabajan(DNI, Nombre\_E, Fecha inicial, Fecha final)

Esta tabla relaciona a la empresa y al empleado. Es útil para conocer el historial de empleo de cada empleado. Tiene relación de (N:N) por que múltiples empleados trabajan en una empresa y un empleado puede trabajar en muchas empresas ( en distintas etapas de su vida). Los datos tienen la suposición que solamente un empleado en la muestra tiene historial de empleo con otra empresa, mientras que los demás han trabajado en la misma desde el inicio. Los datos suponen que cada empresa ha desarrollado como mínimo una aplicación.

7. Descarga( Num\_cuenta, NombreApp, Puntuacion, Fecha Descarga, País )

La Tabla relaciona a los Usuarios y a la aplicación que descargan. Tiene tipo de relación (N:N) dado que una aplicación puede ser instalada por múltiples usuarios. Mientras que un usuario puede instalar múltiples aplicaciones en su Móvil. Los datos hacen la suposición de que todos los usuarios dejan una puntuación en la aplicación mientras que ningún usuario en la muestra de usuarios dejó un comentario.

#### IV. Consultas

Los datos importados a las tablas fueron inicialmente importados a la base de datos mediante Import Wizard. Dado el proyecto fue completado en una MAC la importación de la data mediante Código no fue posible con el código provisto en la documentación de este curso. Por tal razón incluí los datos mediante un 'Insert into' en adición al import wizard. Los datos CSV serán incluidos junto a la entrega del script y el pdf.

Respecto los datos, es importante aclarar que su selección fue limitada para facilitar la comprensión de lectura y el proceso de desarrollo. Las consultas seleccionadas demuestran una variedad de conceptos discutidos en el módulo, aunque parezcan similar cada consulta contiene un concepto que la diferencia de las demás. En adición al contenido del módulo, se utilizó la búsqueda en internet de nuevos conceptos de SQL más avanzados.

```
--
-----
-----
-- #1. Incluye Todas las tablas y todos sus valores
--
-----
-----
select * FROM TiendaAplicaciones.Empresa_Servicio;
Select * from TiendaAplicaciones.aplicaciones;
select * from TiendaAplicaciones.Empleados;
select * from TiendaAplicaciones.Usuario;
select * from TiendaAplicaciones.Tienda_App;
select * from TiendaAplicaciones.Descarga;
select * from TiendaAplicaciones.Realizan;
select * from TiendaAplicaciones.Trabajan;
```

```

--
-----
-----
-- #2. Las tres empresas fundadas más reciente
--
-----
-----
Select
    Nombre_E as 'Nombre de la Empresa',
    Ano_fundacion as 'Año de fundacion'
From Empresa_Servicio
order by Ano_fundacion desc
limit 3 ;

--
-----
-----
-- #3 Puntuación Promedio de Cada Aplicación.
--
-----
-----

SELECT NombreApp, AVG(Puntuacion) AS 'Puntuacion
Promedio'
FROM Descarga
GROUP BY NombreApp;

--
-----
-----
-- #4 Ordenar Lista de nombre de clientes en orden
descendiente.
--
-----
-----

select Nombre as 'Lista de nombres descendiente'
from Usuario
order by Nombre desc;

--
-----
-----
-- #5 Precio Incrementado de Aplicaciones cuyo precio es
mayor de 1 euro

```



```
--
-----
-----
```

```
select NombreApp,Precio
, (case when (Precio>1) then round(Precio * 1.05,2)
    else round(precio * 1.00,2) end ) as 'Precio
Incrementado'
from Aplicaciones;
```

*Así no se incrementa, esto  
es solo para ver que pasaria  
para incrementar  
es necesario hacer  
un update*

```
--
```

```
-- #6
```

*Enunciado?*

```
select NombreApp
from Aplicaciones --
```

```
-----
where Precio>0 and Precio<4 Aplicaciones con un precio
mayor Aplicaciones
order by Precio asc; -- con un
precio mayor que 0 y menor que 4.
```

```
-----
select NombreApp
from Aplicaciones --
```

```
-----
where Precio = 0 --
Aplicaciones que son Gratis.
order by NombreApp asc; --
```

```
--
```

```
-- #7
```

*Enunciado?*

```
-----
select D.NombreApp as 'Nombre de App' , A.Precio as
'Precio'
from Descarga as D inner join Aplicaciones as A
```

```

on A.NombreApp = D.NombreApp
where Precio in(
-----
select Precio
Aplicaciones Descargadas con Precio > 0 -
from Aplicaciones
-----
where Precio > 0 );

```

```

select distinct(pais)
from Descarga as D inner join Aplicaciones as A
on A.NombreApp = D.NombreApp
-----
where Precio in(
-- Países en donde se han vendido Aplicaciones
select Precio
-- cuyo Precio > 0
from Aplicaciones
-----
where Precio > 0);

```

```

--
-----
-- #8 País donde los usuarios más aplicaciones se han
descargado
--
-----

```

```

SELECT Pais, COUNT(*) as 'Total Descargas'
FROM Descarga
GROUP BY Num Cuenta, Pais
HAVING COUNT(*) > 1;

```

Así calculas el país y el nº de descargas en ese país. País está en usuario así pues descargas lo tienes que unir con usuario.

```

--
-----
-- #9 Nombre de los Clientes y las Aplicaciones que
Instalaron.
--
-----
select U.Nombre as 'Nombre del Usuario' ,D.NombreApp as
'Nombre de aplicación'

```

```
from Usuario as U Inner Join Descarga as D Inner Join
Aplicaciones as A
    on U.Num_Cuenta = D.Num_Cuenta and D.NombreApp =
A.NombreApp;
```

```
--
```

```
-----
```

```
-----
```

```
-- #10 empleados que comenzaron a trabajar en empresas
durante el 2004 y 2010.
```

```
--
```

```
-----
```

```
-----
```

```
select distinct T.DNI as 'DNI de empleado',T.Nombre_E as
'Empresa' ,Fecha_Ini as 'Fecha de Inicio'
from Empleados as E inner join Trabajan as T
where E.DNI=T.DNI and Fecha_Ini between '2004' and
'2010'
order by Fecha_Ini asc;
```

```
--
```

```
-----
```

```
-----
```

```
-- #11 Categoria de la Aplicacion Wework. Utilizando %.
```

```
--
```

```
-----
```

```
-----
```

```
select A.NombreApp,Category
from Aplicaciones as A inner join Realizan as R
    on R.NombreApp = A.NombreApp
where A.NombreApp like '%Wework%';
```

```
--
```

```
-----
```

```
-----
```

```
-- #12 Desplega el DNI del EMpleado y la Empresa de
aquellos empleados que vivan en una calle que termina con
letra E.
```

```
--
```

```
-----
```

```
-----
```

```
select DNI,Nombre_E
from Trabajan
```

```
where DNI in (
    select DNI
    from Empleados
    where Calle like '%e'
);
```

--

```
-----
-----
-- #13 Empleados que han trabajado en más de una compañía
--
-----
```

```
-----
SELECT dni as "DNI del empleado"
FROM Trabajan
GROUP BY dni
HAVING COUNT(dni) >1;
```

--

```
-----
-----
-- #14 Determinando cuando dinero gasto cada usuario al
instalar las aplicaciones
--
-----
```

```
-----
-----
SELECT D.Num_Cuenta ,A.Precio*COUNT(D.Num_Cuenta) as
'Total Gastado'
FROM Descarga as D inner join Aplicaciones as A
    on D.NombreApp=A.NombreApp
group by D.Num_Cuenta, A.Precio
order by D.Num_Cuenta;
```

--

```
-----
-----
-- #15 Cantidad de Aplicaciones Instaladas
--
-----
```

```
-----
-----
SELECT NombreApp, COUNT(NombreApp) AS "Apps Instaladas"
FROM Descarga
GROUP BY NombreApp
```

*sum (precio)*

*Contar las apps descargadas, si quieres distintas*

*Select count(distinct nombreApp) from descarga*

```
UNION ALL
```

```
SELECT 'Total' NombreApp, COUNT(NombreApp)
FROM Descarga;
```

```
--
-----
-- # VISTAS - Crea una tabla enseña que las tiendas en
las que se venden las aplicaciones. Detalle: Hay
Aplicaciones exclusivas a ciertas tiendas.
--
-----
```

```
-----
drop view if exists AplicacionesEnTienda;
create view AplicacionesEnTienda
(NombreApp,NombreT,Nombre_E) as
select R.NombreApp,TA.NombreT,R.Nombre_E
from Tienda_App as TA inner join Realizan as R
on R.Nombre_E=TA.Nombre_E ;
```

```
--
-----
-- # Trigger que suma 10 euros al Precio del Aplicacion
que comienza con el codigo '001'.
--
-----
```

```
-----
use TiendaAplicaciones;
drop table if exists AplicacionesDescuento;
drop trigger if exists actualizPrecio_;
create table AplicacionesDescuento (
    NombreApp_ varchar(20) not null,
    Codigo_ char(5),
    Memoria_ varchar(10),
    Precio_ decimal(6,2),
    Category_ varchar(20)
);
delimiter //
create trigger actualizPrecio_ before update
on Aplicaciones for each row
begin
```

*¿Que sentido tiene?*  
*Esto se hace con un update a los un y particular para un trigger*

```
insert into AplicacionesDescuento values  
(old.NombreApp,old.Codigo,old.Memoria,new.Precio,old.Cate  
gory);  
end//
```

```
update Aplicaciones  
set Precio = Precio+10.00  
where Codigo = '001';
```

↳ 1/2 the completion.