

# PECL2 - BBDD

---

Miguel Ángel Losada Fernández – 53824672A

Sergio Sanz Sacristán – 03207618S

Juan Casado Ballesteros – 09108762A

**Profesor: Sergio Caro**

**Bases de Datos GII Laboratorio: 12:00 a 14:00**

---

# Índice

<b>PECL2 - BBDD .....</b>	<b>1</b>
<b>Índice .....</b>	<b>2</b>
<b>Cambios respecto al modelo Entidad-Relación .....</b>	<b>3</b>
<b>Modelo Relacional.....</b>	<b>5</b>
<b>Integridad de los datos .....</b>	<b>6</b>
<b>Por implementar (triggers) .....</b>	<b>8</b>
<b>Consultas .....</b>	<b>9</b>
<b>Archivos aportados .....</b>	<b>17</b>

## Cambios respecto al modelo Entidad-Relación

Para realizar esta práctica nos hemos juntado dos equipos que en la práctica anterior trabajamos por separado, uno de los equipos estaba formado por Miguel y Sergio y el otro por Juan luego para explicar las modificaciones realizadas sobre los diagramas Entidad-Relación haremos referencias a ambos modelos. Describimos a continuación cada una de las tablas creadas para en el modelo relacional y a partir de ellas expondremos las citadas modificaciones.

**Empleado:** hemos incluido dentro del empleado como atributos individuales los atributos que en el modelo Entidad-Relación formaban un atributo compuesto llamado dirección tal y como se indicaba en la guía para crear el modelo Relacional. Ya que hemos supuesto que cada empleado podría dar una cantidad distinta de teléfonos de contacto (más de uno) hemos creado una tabla llamada Teléfono en la que almacenarlos y en caso de que esto se diera que la BBDD lo pudiera soportar.

**Surtidor:** Cada surtidor tiene un número propio y único dentro de la BBDD que se corresponde con su número dentro de la gasolinera, dicho número es el utilizado para realizar la especialización en los distintos tipos de Surtidor que puede haber. Para que cada uno de esos tipos tenga su propia PK los tipos ya especializados tienen otro número que se correspondería con el número de serie de los surtidores, de modo que los tipos especializados tienen el Número\_de\_surtidor como PFK y el Número como PK. Adicionalmente la descripción del surtidor se ha hecho coincidir con su tipo en la especialización de modo que se simplifican en gran medida algunas consultas.

**Reposta:** Esta tabla enlaza los Clientes con los Surtidores en los que han repostado, en una de las implementaciones Entidad-Relación se incluía en esta relación el precio y en otra los litros. Sobre la tabla de modelo Relacional hemos incluido los puntos obtenidos al repostar y los litros repostados, adicionalmente como PK se incluye la fecha en la que se ha repostado pues si no se hiciera un cliente no podría repostar dos veces en el mismo surtidor cosa que claramente si debe poder ocurrir.

**Canjea:** sobre esta tabla del modelo Relacional incluimos los puntos que se canjean, así como las PFK de Nombre\_de\_usuario y Código\_de\_barras. Hemos tenido que incluir como PK una fecha para que los clientes pudieran canjear varias veces el mismo producto, de otra forma no podrían.

**Ticket:** dentro de esta tabla incluimos un atributo que puede ser nulo con la fecha en la que el ticket fue premiado, de este modo representamos cuando los tickets han sido premiados y cuando no. respetando la inclusión existente en el modelo Entidad-Relación entre sortea y emite ya que si un ticket no se a emitido no podrá ser premiado por el hecho de que no existe.

**Contiene:** hemos añadido una fecha como PK a esta tabla para que dos clientes puedan comprar tickets con los mismos artículos en la misma cantidad, de otra forma no se podría.

**Artículo:** hemos eliminado la redundancia entre PvP y precio sin iva dejando solo el PvP ya que entendemos el precio sin iva como un atributo calculable a partir del PvP.

**Cliente:** ninguna modificación realizada.

**Opinión:** ninguna modificación realizada.

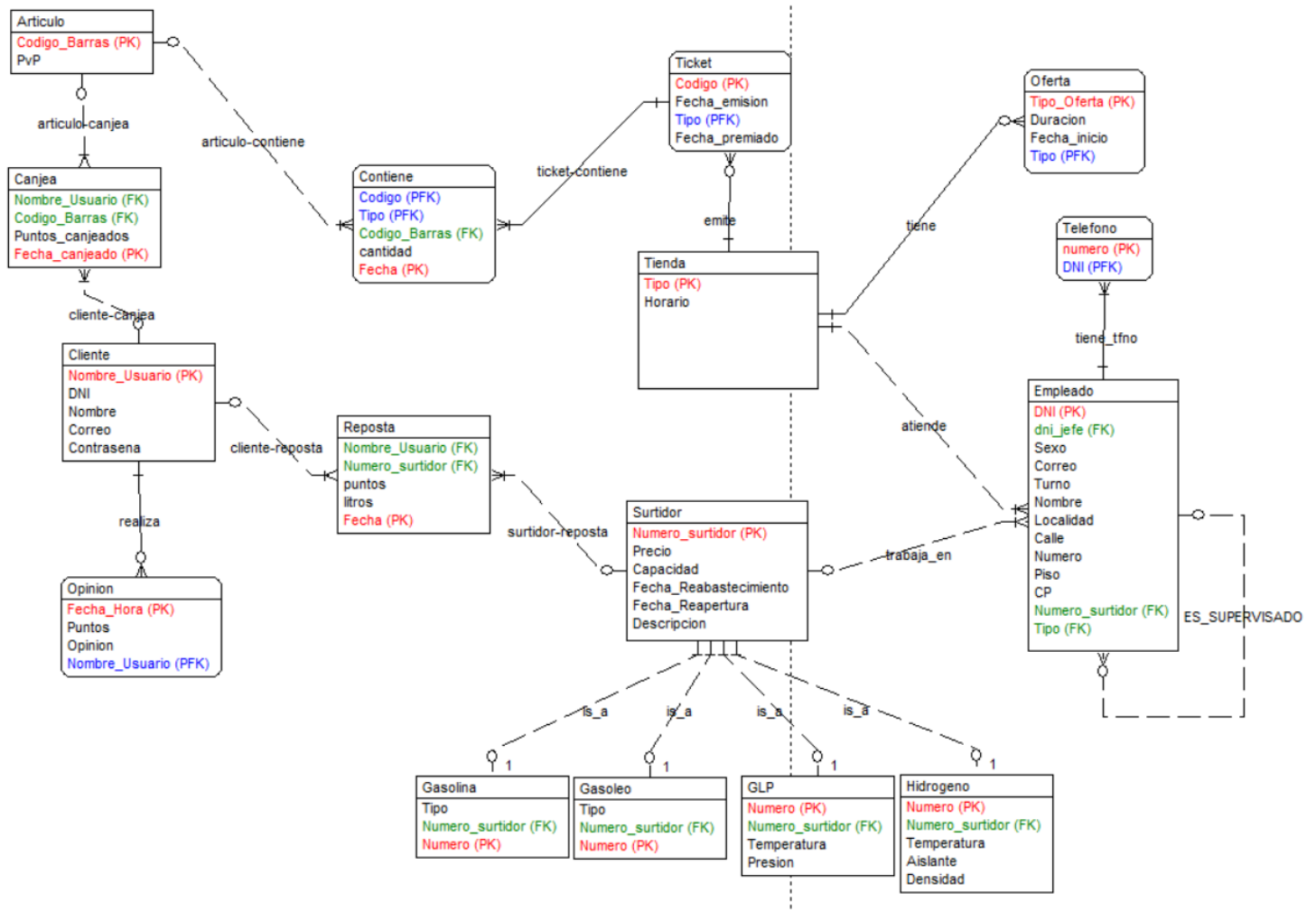
**Tienda:** ninguna modificación realizada.

**Oferta:** ninguna modificación realizada.

Se han eliminado redundancias respecto a lo forma de llevar el recuento de puntos, estos se ganan al repostar y son un atributo más de esa tabla y se pierden al canjear, son otro atributo de esa otra tabla.

También se han eliminado redundancias respecto a los precios, se guarda el precio de cada artículo como PvP, todos los demás precios se pueden calcular a partir de ese y con ayuda de la cantidad de artículos de cada tipo en los tickets.

## Modelo Relacional



Se incluye una foto del modelo relacional realizado con TOAD en la que se pueden ver todas las tablas creadas con sus atributos y los nombres dados a las restricciones de integridad.

Quedan resaltados sobre el resto de los atributos las PK en rojo, FK en verde y PFK en azul.

---

## Integridad de los datos

Procedemos ahora a explicar las relaciones de integridad existentes entre las tablas creadas. Para ello utilizaremos los nombres que les hemos dado en el diagrama Relacionar realizado en Toad mostrado con anterioridad.

**Es supervidado:** se relacionan los empleados unos con otros de modo 0:0...N de tal modo que los empleados tendrán como FK a su jefe pudiendo estos no tener jefe en caso de ser el dueño o no ser jefe de nadie si son un trabajador sin cargo. A la hora de insertar o actualizar los jefes lo haremos con la restricción de que el empleado que les ponemos como jefe exista. En caso de modificar un jefe modificaremos los empleados que le tengan como jefe en cascada y en caso de borrarlo de la BBDD la FK de estos se pondrá a nulo a la espera de que el gestor de la BBDD les asigne su nuevo jefe.

**Atiende:** la tienda se relaciona con el empleado de modo 1:1...N, en cada tienda es obligatorio que trabaje un empleado, aunque pueden hacerlo más. No podemos asignar empleados a tiendas que no existan y si modificamos la tienda modificamos al empleado en cascada, aunque en el caso de borrarla dejaremos al empleado con un valor de FK de la tienda como nulo a la espera de ser reasignado. El empleado tendrá como FK a la tienda.

**Tiene:** La tienda tiene ofertas de modo 1:0...N es decir todas las ofertas deben estar asignadas a una tienda y las tiendas pueden tener de ninguna a varias ofertas. No podremos poner ofertas a tiendas que no existen y en caso de borrar la tienda o modificarla afectaremos de igual modo a las ofertas. No nos interesa tener ofertas guardadas de una tienda que ya ha cerrado pues no se podrían aplicar. La oferta tendrá como PFK a la tienda.

**Tiene tlf:** relaciona empleado con teléfono de forma 1:1...N es decir todos los empleados deben de tener al menos un teléfono pudiendo tener varios y no se guardarán teléfonos no asignados a los empleados. Las inserciones o modificaciones de teléfonos están restringidas, debe existir el empleado y las modificaciones o el borrado de un empleado se aplica en cascada sobre los teléfonos. El teléfono tendrá como PKF al empleado.

**Emite:** relaciona de modo 1:0...N las tiendas con los tickets, todo ticket debe estar asignado a una tienda y las tiendas podrán tener cero o varios tickets. Los cambios o el borrado de las tiendas afectan a los tickets en cascada y la inserción o modificación de los tickets está restringido a la existencia de las mismas. El ticket tendrá como PFK a la tienda.

**IS A:** relaciona a Surtidor con cada uno de sus posibles tipos de modo 1:0...1 es decir cada surtidor podrá ser solo de un tipo. Las modificaciones o el borrado del surtidor afectan a los tipos en cascada y las inserciones o modificaciones de los tipos están restringidas a que el surtidor exista. Los tipos de surtidor tendrán como FK al surtidor.

**Realiza:** relaciona a los clientes con sus opiniones de modo 1:0...N es decir, un cliente puede tener de cero a varias opiniones pero una opinión no puede existir sin que un cliente la haya realizado. Las actualizaciones o el borrado de los clientes afectan a las opiniones en cascada y la inserción o modificación de las opiniones está restringida a la existencia del cliente. La opinión tendrá como PFK al cliente.

**Tique-contiene/Artículo-contiene, artículo-canjea/cliente-canjea y cliente-reposta/surtidor-reposta:** estos tres pares de relaciones forman cada par una relación N:M del modelo Entidad-Relación, en el modelo relaciona esto se hace de forma que una tabla se relaciona con otra mediante una tabla tercera intermediaria que tiene a ambas dos relaciones como sus FK. Todas estas relaciones son del tipo 0:1...N excepto ticket-contiene que es 1:1...N ya que no puede haber un ticket que no contenga artículos. El borrado se hará en cascada hacia la tabla intermedia en los casos de ticket-contiene, cliente-canjea y cliente-reposta pues consideramos que nos es útil saber que un cliente ha canjeado o repostado si este ya no existe o que un artículo estuvo en un ticket borrado, por el contrario en artículo-contiene, artículo-canjea y surtidor-reposta si consideramos que aunque el artículo o surtidor se borren debemos mantener la tabla intermedia pues esta contiene otros datos de interés como los puntos o la cantidad. Las modificaciones afectarán en cualquier caso en cascada. La creación de una nueva fila en la tabla intermedia debe estar restringida a que las filas de las tablas que quiera relacionar existan.

---

## Por implementar (triggers)

Algunos dominios podrían estar mejor restringidos, probablemente esto se realice mediante los triggers que veremos en futuras sesiones. Nos referimos a los horarios por ejemplo, en vez de ser un `char[]` que queden restringidos a MAÑANA, TARDE ,12h,24h.

Los empleados no deberían poder trabajar en un surtidor si ya lo hacen en una tienda y viceversa, probablemente con triggers esto pueda quedar reflejado en el esquema de la BBDD.

Controlar que un cliente no canjee artículos por más puntos que los que ha conseguido repostando es otra funcionalidad que pensamos que se puede añadir al esquema mediante los triggers.



## Consultas

1.	Mostrar los artículos presentes en la base de datos, mostrando su código y el precio de venta.		
SELECT "Codigo_Barras", "PvP" FROM "Articulo"		Codigo_Barras character(60)	PvP double precision
	1	jha-sfe-aedw	12
	2	Akn-3jn-NS9	20
	3	hsi-JS7-js7y	16
	4	hs8-pqw-H78	12
	5	iKl-ks7-Jh0e	5
	6	LM8-gsi-6tYd	16
	7	bnc-mbU-OP87	3
	8	bcy-JGH-98HJ	5
	9	FAT-B66-CFu7	9999
	10	Ldp-fFh-203m	6
	11	wef-rKM-dD93	1
	12	aso-LJk-08SS	4
	13	osk-iw9-IS8K	6
	14	kja-pkJ-232r	4
	15	gha-GTn-sl2r	4
	16	Zma-Vkz-dj8x	4

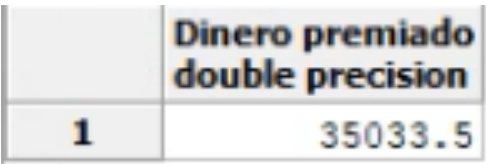
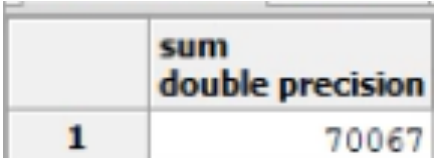
2.	Determinar el número total de surtidores que hay en la gasolinera.						
<pre>SELECT COUNT (*) FROM "Surtidor"</pre>		<table><tr><td></td><td>count bigint</td></tr><tr><td>1</td><td>7</td></tr></table>		count bigint	1	7	
	count bigint						
1	7						

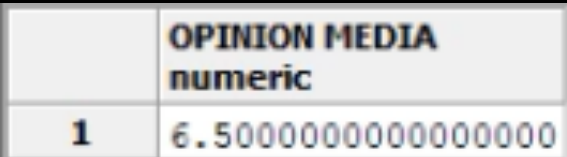
3.	<p>Obtener el total del dinero facturado por las tiendas de la gasolinera desde la implementación de la base de datos.</p> <p>En esta consulta no teníamos claro a lo que se refería, por ello, realizamos estas dos consultas.</p>																		
<pre>SELECT sum ("PvP"*"cantidad") as "Dinero" FROM "Contiene" INNER JOIN "Articulo" ON "Contiene"."Codigo_Barras" = "Articulo"."Codigo_Barras"</pre>	<table><tr><th></th><th>Dinero double precision</th></tr><tr><td>1</td><td>70308</td></tr></table>		Dinero double precision	1	70308														
	Dinero double precision																		
1	70308																		
<pre>SELECT "Tipo", sum ("PvP"*"cantidad") as "Dinero" FROM "Contiene" INNER JOIN "Articulo" ON "Contiene"."Codigo_Barras" = "Articulo"."Codigo_Barras" GROUP BY "Tipo"</pre>	<table><tr><th></th><th>Tipo character(50)</th><th>Dinero double precision</th></tr><tr><td>1</td><td>Ultramarinos</td><td>120</td></tr><tr><td>2</td><td>Panaderia</td><td>44</td></tr><tr><td>3</td><td>Libreria</td><td>5</td></tr><tr><td>4</td><td>Cafeteria</td><td>70026</td></tr><tr><td>5</td><td>Kiosko</td><td>113</td></tr></table>		Tipo character(50)	Dinero double precision	1	Ultramarinos	120	2	Panaderia	44	3	Libreria	5	4	Cafeteria	70026	5	Kiosko	113
	Tipo character(50)	Dinero double precision																	
1	Ultramarinos	120																	
2	Panaderia	44																	
3	Libreria	5																	
4	Cafeteria	70026																	
5	Kiosko	113																	

4.	Mostrar el nombre de los empleados de las tiendas que trabajan en turno de mañana.										
<pre>SELECT "Nombre"       FROM "Empleado"       WHERE "Empleado"."Turno" = 'MAÑANA' AND "Empleado"."Numero_surtidor" IS NULL</pre>	<table> <tr> <th></th><th>Nombre character(60)</th></tr> <tr> <td>1</td><td>Dugan Gallardo Barraza</td></tr> <tr> <td>2</td><td>Elisabeth Ochoa Vargas</td></tr> <tr> <td>3</td><td>Kim Dotcom</td></tr> <tr> <td>4</td><td>Olimpia Luevano Gaona</td></tr> </table>		Nombre character(60)	1	Dugan Gallardo Barraza	2	Elisabeth Ochoa Vargas	3	Kim Dotcom	4	Olimpia Luevano Gaona
	Nombre character(60)										
1	Dugan Gallardo Barraza										
2	Elisabeth Ochoa Vargas										
3	Kim Dotcom										
4	Olimpia Luevano Gaona										

5.	Mostrar los puntos totales canjeados por los socios de la gasolinera, junto con su identificador y nombre.																		
SELECT "Nombre_Usuario", SUM ("Puntos_canjeados")as "Puntos Totales" FROM "Canjea" GROUP BY "Nombre_Usuario"	<table><tr><th></th><th>Nombre_Usuario character(50)</th><th>Puntos Totales bigint</th></tr><tr><td>1</td><td>Jordi ENP</td><td>35</td></tr><tr><td>2</td><td>Mmmmm gasss</td><td>85</td></tr><tr><td>3</td><td>El chuches</td><td>20</td></tr><tr><td>4</td><td>Tu gasolinero</td><td>35</td></tr><tr><td>5</td><td>La gran manguera</td><td>50</td></tr></table>		Nombre_Usuario character(50)	Puntos Totales bigint	1	Jordi ENP	35	2	Mmmmm gasss	85	3	El chuches	20	4	Tu gasolinero	35	5	La gran manguera	50
	Nombre_Usuario character(50)	Puntos Totales bigint																	
1	Jordi ENP	35																	
2	Mmmmm gasss	85																	
3	El chuches	20																	
4	Tu gasolinero	35																	
5	La gran manguera	50																	

6.	Mostrar los 5 artículos más canjeados por los socios.			
<pre>SELECT "Codigo_Barras", COUNT (*) as "VECES" FROM "Canjea" GROUP BY "Codigo_Barras" ORDER BY "VECES" DESC LIMIT 5</pre>				
		Codigo_Barras character(60)	VECES bigint	
		1	iKl-ks7-Jh0e	4
		2	kja-pkJ-232r	3
		3	wef-rKM-dD93	3
		4	hs8-pqw-H78	2
		5	LM8-gsi-6tYd	2

7.	<p>Mostrar el importe total devuelto en los tickets en todos los sorteos realizados hasta la fecha.</p> <p>En esta consulta realizamos dos, ya que, no teníamos claro a lo que se refería el enunciado. En esta consulta realizamos dos, ya que, no teníamos claro a lo que se refería el enunciado.</p>				
<pre>SELECT SUM("cantidad"*"PvP")/2 AS "Dinero premiado"   FROM "Articulo"   INNER JOIN (SELECT "Codigo_Barras","cantidad"               FROM "Ticket"               INNER JOIN "Contiene" ON "Ticket"."Codigo" = "Contiene"."Codigo"               WHERE "Fecha_premiado" IS NOT NULL) AS "TABLA" ON "Articulo"."Codigo_Barras" = "TABLA"."Codigo_Barras"</pre>	 <table border="1"> <thead> <tr> <th></th> <th>Dinero premiado double precision</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>35033.5</td> </tr> </tbody> </table>		Dinero premiado double precision	1	35033.5
	Dinero premiado double precision				
1	35033.5				
<pre>SELECT SUM("cantidad"*"PvP") AS "Dinero premiado"   FROM "Articulo"   INNER JOIN (SELECT "Codigo_Barras","cantidad"               FROM "Ticket"               INNER JOIN "Contiene" ON "Ticket"."Codigo" = "Contiene"."Codigo"               WHERE "Fecha_premiado" IS NOT NULL) AS "TABLA" ON "Articulo"."Codigo_Barras" = "TABLA"."Codigo_Barras"</pre>	 <table border="1"> <thead> <tr> <th></th> <th>sum double precision</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>70067</td> </tr> </tbody> </table>		sum double precision	1	70067
	sum double precision				
1	70067				

8.	<p>Determinar el grado de satisfacción medio de las opiniones que los clientes han realizado por internet, mostrando la puntuación media.</p>				
<pre>SELECT AVG("Puntos") AS "OPINION MEDIA"   FROM "Opinion"</pre>	 <table border="1"> <thead> <tr> <th></th> <th>OPINION MEDIA numeric</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>6.5000000000000000</td> </tr> </tbody> </table>		OPINION MEDIA numeric	1	6.5000000000000000
	OPINION MEDIA numeric				
1	6.5000000000000000				

9.	Determinar el número de tickets que ha emitido cada tienda, mostrando el número de tickets y el nombre de la tienda. Ordenar la salida de mayor a menor.																		
SELECT "Tipo", COUNT("Codigo") AS "NUMERO DE TICKETS" FROM "Ticket" GROUP BY "Tipo" ORDER BY "NUMERO DE TICKETS" DESC	<table><tr><th></th><th>Tipo character(50)</th><th>NUMERO DE TICKETS bigint</th></tr><tr><td>1</td><td>Cafeteria</td><td>3</td></tr><tr><td>2</td><td>Kiosko</td><td>3</td></tr><tr><td>3</td><td>Ultramarinos</td><td>2</td></tr><tr><td>4</td><td>Panaderia</td><td>2</td></tr><tr><td>5</td><td>Libreria</td><td>1</td></tr></table>		Tipo character(50)	NUMERO DE TICKETS bigint	1	Cafeteria	3	2	Kiosko	3	3	Ultramarinos	2	4	Panaderia	2	5	Libreria	1
	Tipo character(50)	NUMERO DE TICKETS bigint																	
1	Cafeteria	3																	
2	Kiosko	3																	
3	Ultramarinos	2																	
4	Panaderia	2																	
5	Libreria	1																	

10.	Aunque el sorteo de tickets se hace aleatoriamente entre todos los emitidos en la semana, determinar cuál es la tienda en la que más tickets han sido premiados hasta la fecha.																		
<pre>SELECT "Tipo", COUNT(*) AS "TIQUETS PREMIADOS" FROM "Ticket" WHERE "Fecha_premiado" IS NOT NULL GROUP BY "Tipo" ORDER BY "TIQUETS PREMIADOS" DESC LIMIT 1</pre>																			
<table><tr><th colspan="3">Data Output</th><th>Explain</th><th>Messages</th><th>History</th></tr><tr><th></th><th>Tipo character(50)</th><th>TIQUETS PREMIADOS bigint</th><th colspan="3"></th></tr><tr><td>1</td><td>Cafeteria</td><td>2</td><td colspan="3"></td></tr></table>		Data Output			Explain	Messages	History		Tipo character(50)	TIQUETS PREMIADOS bigint				1	Cafeteria	2			
Data Output			Explain	Messages	History														
	Tipo character(50)	TIQUETS PREMIADOS bigint																	
1	Cafeteria	2																	

11.	De los repostajes que proporcionan puntos a los socios, determinar el total de litros repostados en cada surtidor, ordenando la salida de menor a mayor.																										
SELECT "Numero_surtidor", SUM ("litros") AS "Litros" FROM "Reposta" WHERE "puntos" IS NOT NULL GROUP BY "Numero_surtidor" ORDER BY "Litros" ASC		<table><tr><th></th><th>Numero_surtidor integer</th><th>Litros double precision</th></tr><tr><td>1</td><td>5</td><td>20</td></tr><tr><td>2</td><td>1</td><td>30</td></tr><tr><td>3</td><td>6</td><td>30</td></tr><tr><td>4</td><td>7</td><td>30</td></tr><tr><td>5</td><td>2</td><td>40</td></tr><tr><td>6</td><td>3</td><td>70</td></tr><tr><td>7</td><td>4</td><td>130</td></tr></table>			Numero_surtidor integer	Litros double precision	1	5	20	2	1	30	3	6	30	4	7	30	5	2	40	6	3	70	7	4	130
	Numero_surtidor integer	Litros double precision																									
1	5	20																									
2	1	30																									
3	6	30																									
4	7	30																									
5	2	40																									
6	3	70																									
7	4	130																									

12.	Mostrar el tamaño medio y el precio/litro de los surtidores de gasolina y gasoil.
<pre>(SELECT "Surtidor"."Descripcion" AS "TIPO", AVG ("Surtidor"."Precio") AS "PRECIO MEDIO", AVG ("Surtidor"."Capacidad") AS "CAPACIDAD MEDIA" FROM "Gasolina" INNER JOIN "Surtidor" ON "Gasolina"."Numero_surtidor" = "Surtidor"."Numero_surtidor" GROUP BY "TIPO") UNION (SELECT "Surtidor"."Descripcion" AS "TIPO", AVG ("Surtidor"."Precio") AS "PRECIO MEDIO", AVG ("Surtidor"."Capacidad") AS "CAPACIDAD MEDIA" FROM "Gasoleo" INNER JOIN "Surtidor" ON "Gasoleo"."Numero_surtidor" = "Surtidor"."Numero_surtidor" GROUP BY "TIPO")</pre>	

	TIPO character(100)	PRECIO MEDIO double precision	CAPACIDAD MEDIA double precision
1	GASOLEO	1.09533333333333	1033.33333333333
2	GASOLINA	1.339	1000

13.	Determinar el número trabajadores que tiene cada tienda y cada surtidor, ordenando la salida de menor a mayor.																																																				
SELECT "Tipo","Numero_surtidor", COUNT(*) AS "CANTIDAD" FROM "Empleado" GROUP BY "Tipo", "Numero_surtidor" ORDER BY "CANTIDAD" ASC	<table><tr><th></th><th>Tipo character(50)</th><th>Numero_surtidor integer</th><th>CANTIDAD bigint</th></tr><tr><td>1</td><td>Kiosko</td><td></td><td>1</td></tr><tr><td>2</td><td>Libreria</td><td></td><td>1</td></tr><tr><td>3</td><td>Panaderia</td><td></td><td>1</td></tr><tr><td>4</td><td></td><td>6</td><td>2</td></tr><tr><td>5</td><td></td><td>2</td><td>2</td></tr><tr><td>6</td><td>Cafeteria</td><td></td><td>2</td></tr><tr><td>7</td><td></td><td>3</td><td>2</td></tr><tr><td>8</td><td></td><td>5</td><td>2</td></tr><tr><td>9</td><td></td><td>1</td><td>2</td></tr><tr><td>10</td><td></td><td>7</td><td>2</td></tr><tr><td>11</td><td></td><td>4</td><td>2</td></tr><tr><td>12</td><td>Ultramarinos</td><td></td><td>2</td></tr></table>		Tipo character(50)	Numero_surtidor integer	CANTIDAD bigint	1	Kiosko		1	2	Libreria		1	3	Panaderia		1	4		6	2	5		2	2	6	Cafeteria		2	7		3	2	8		5	2	9		1	2	10		7	2	11		4	2	12	Ultramarinos		2
	Tipo character(50)	Numero_surtidor integer	CANTIDAD bigint																																																		
1	Kiosko		1																																																		
2	Libreria		1																																																		
3	Panaderia		1																																																		
4		6	2																																																		
5		2	2																																																		
6	Cafeteria		2																																																		
7		3	2																																																		
8		5	2																																																		
9		1	2																																																		
10		7	2																																																		
11		4	2																																																		
12	Ultramarinos		2																																																		

14.	Mostrar el número de surtidores de gasolina, gasoil, GLP e Hidrógeno que hay en la gasolinera, ordenado de mayor a menor.																	
SELECT "Descripcion", COUNT(*) AS "Cantidad" FROM "Surtidor" GROUP BY "Descripcion" ORDER BY "Cantidad" DESC		<table><tr><th></th><th>Descripcion character(100)</th><th>CANTIDAD bigint</th></tr><tr><td>1</td><td>GASOLEO</td><td>3</td></tr><tr><td>2</td><td>GASOLINA</td><td>2</td></tr><tr><td>3</td><td>GLP</td><td>1</td></tr><tr><td>4</td><td>HIDROGENO</td><td>1</td></tr></table>			Descripcion character(100)	CANTIDAD bigint	1	GASOLEO	3	2	GASOLINA	2	3	GLP	1	4	HIDROGENO	1
	Descripcion character(100)	CANTIDAD bigint																
1	GASOLEO	3																
2	GASOLINA	2																
3	GLP	1																
4	HIDROGENO	1																

15.	Mostrar el porcentaje de surtidores de gasolina, gasoil, GLP e Hidrógeno que existen en la gasolinera.																	
SELECT "Descripcion", COUNT(*)*100/ (SELECT COUNT(*) FROM "Surtidor")AS "Porcentaje" FROM "Surtidor" GROUP BY "Descripcion"		<table><tr><th></th><th>Descripcion character(100)</th><th>Porcentaje bigint</th></tr><tr><td>1</td><td>GASOLEO</td><td>42</td></tr><tr><td>2</td><td>GASOLINA</td><td>28</td></tr><tr><td>3</td><td>GLP</td><td>14</td></tr><tr><td>4</td><td>HIDROGENO</td><td>14</td></tr></table>			Descripcion character(100)	Porcentaje bigint	1	GASOLEO	42	2	GASOLINA	28	3	GLP	14	4	HIDROGENO	14
	Descripcion character(100)	Porcentaje bigint																
1	GASOLEO	42																
2	GASOLINA	28																
3	GLP	14																
4	HIDROGENO	14																

16.	Mostrar el valor total de todos los artículos canjeados por los socios en el último mes.				
<pre>SELECT SUM("PvP") AS "Dinero" FROM "Canjea" INNER JOIN "Articulo" ON "Canjea"."Codigo_Barras" = "Articulo"."Codigo_Barras";</pre>	<table> <tr> <th></th><th>Dinero double precision</th></tr> <tr> <td>1</td><td>131</td></tr> </table>		Dinero double precision	1	131
	Dinero double precision				
1	131				

17.	Mostrar las fechas en las que no se puede utilizar los surtidores que se encuentren averiados.								
SELECT "Numero_surtidor", "Fecha_Reapertura" FROM "Surtidor" WHERE "Fecha_Reapertura" IS NOT NULL		<table><tr><td></td><td>Numero_surtidor integer</td><td>Fecha_Reapertura timestamp with time zone</td></tr><tr><td>1</td><td>4</td><td>2098-06-18 13:13:13+02</td></tr></table>			Numero_surtidor integer	Fecha_Reapertura timestamp with time zone	1	4	2098-06-18 13:13:13+02
	Numero_surtidor integer	Fecha_Reapertura timestamp with time zone							
1	4	2098-06-18 13:13:13+02							

18.	Mostrar a los supervisores (DNI y nombre) de los empleados que trabajan en surtidores de GLP, en los que los socios hayan repostado y, posteriormente, canjeado artículos.						
<pre>SELECT "Empleado"."DNI", "Empleado"."Nombre"       FROM(SELECT DISTINCT "Empleado"."dni_jefe"       AS "DNI"             FROM "Reposta"             INNER JOIN "Canjea" ON "Reposta"."Nombre_Usuario" = "Canjea"."Nombre_Usuario" INNER JOIN "Surtidor" ON "Reposta"."Numero_surtidor" = "Surtidor"."Numero_surtidor" INNER JOIN "Empleado" ON "Reposta"."Numero_surtidor" = "Empleado"."Numero_surtidor"             WHERE "Surtidor"."Descripcion" = 'GLP')AS "JEFES"       INNER JOIN "Empleado" ON "JEFES"."DNI" = "Empleado"."DNI"</pre>	<table><tr><th></th><th>DNI character(20)</th><th>Nombre character(60)</th></tr><tr><td>1</td><td>82423423V</td><td>Segismundo Molina Medrano</td></tr></table>		DNI character(20)	Nombre character(60)	1	82423423V	Segismundo Molina Medrano
	DNI character(20)	Nombre character(60)					
1	82423423V	Segismundo Molina Medrano					

19.	Mostrar los artículos que han sido canjeados por los socios y que hayan sido comprados en las tiendas.																
<pre> SELECT DISTINCT "Canjea"."Codigo_Barras"   FROM "Canjea"       INNER JOIN "Contiene" ON "Canjea"."Codigo_Barras" = "Contiene"."Codigo_Barras" </pre>	<table> <tr> <th></th><th>Codigo_Barras character(60)</th></tr> <tr> <td>1</td><td>kja-pkJ-232r</td></tr> <tr> <td>2</td><td>Akn-3jn-NS9</td></tr> <tr> <td>3</td><td>hsi-JS7-js7y</td></tr> <tr> <td>4</td><td>wef-rKM-dD93</td></tr> <tr> <td>5</td><td>hs8-pqw-H78</td></tr> <tr> <td>6</td><td>LM8-gs1-6tYd</td></tr> <tr> <td>7</td><td>iKl-ks7-Jh0e</td></tr> </table>		Codigo_Barras character(60)	1	kja-pkJ-232r	2	Akn-3jn-NS9	3	hsi-JS7-js7y	4	wef-rKM-dD93	5	hs8-pqw-H78	6	LM8-gs1-6tYd	7	iKl-ks7-Jh0e
	Codigo_Barras character(60)																
1	kja-pkJ-232r																
2	Akn-3jn-NS9																
3	hsi-JS7-js7y																
4	wef-rKM-dD93																
5	hs8-pqw-H78																
6	LM8-gs1-6tYd																
7	iKl-ks7-Jh0e																

20.	Mostrar los artículos que ni hayan sido canjeados por los socios ni que hayan sido comprados en las tiendas.				
<pre> SELECT "Codigo_Barras"   FROM "Articulo"       WHERE "Codigo_Barras" NOT IN (       (SELECT "Codigo_Barras"       FROM "Contiene")       UNION       (SELECT "Codigo_Barras"       FROM "Canjea")) </pre>	<table> <tr> <th></th><th>Codigo_Barras character(60)</th></tr> <tr> <td>1</td><td>Zma-Vks-dj8x</td></tr> </table>		Codigo_Barras character(60)	1	Zma-Vks-dj8x
	Codigo_Barras character(60)				
1	Zma-Vks-dj8x				

21.	Mostrar los artículos que hayan sido canjeados por los socios y que hayan sido comprados en las tiendas, pero que no aparezcan en tickets premiados.												
<pre> SELECT DISTINCT "Canjea"."Codigo_Barras"   FROM "Canjea"   INNER JOIN "Contiene" ON "Canjea"."Codigo_Barras" = "Contiene"."Codigo_Barras"   WHERE "Canjea"."Codigo_Barras" NOT IN (SELECT "Codigo_Barras"   FROM "Contiene"   INNER JOIN "Ticket" ON "Contiene"."Codigo" = "Ticket"."Codigo"   WHERE "Ticket"."Fecha_premiado" IS NOT NULL) </pre>	<table> <tr> <th></th><th><b>Codigo_Barras character(60)</b></th></tr> <tr> <td><b>1</b></td><td>kja-pkJ-232r</td></tr> <tr> <td><b>2</b></td><td>LM8-gsi-6tYd</td></tr> <tr> <td><b>3</b></td><td>wef-rKM-dD93</td></tr> <tr> <td><b>4</b></td><td>Akn-3jn-NS9</td></tr> <tr> <td><b>5</b></td><td>iKl-ks7-Jh0e</td></tr> </table>		<b>Codigo_Barras character(60)</b>	<b>1</b>	kja-pkJ-232r	<b>2</b>	LM8-gsi-6tYd	<b>3</b>	wef-rKM-dD93	<b>4</b>	Akn-3jn-NS9	<b>5</b>	iKl-ks7-Jh0e
	<b>Codigo_Barras character(60)</b>												
<b>1</b>	kja-pkJ-232r												
<b>2</b>	LM8-gsi-6tYd												
<b>3</b>	wef-rKM-dD93												
<b>4</b>	Akn-3jn-NS9												
<b>5</b>	iKl-ks7-Jh0e												



---

## Archivos aportados

- Se incluye esta memoria donde se explica la práctica realizada y las decisiones tomadas para su creación.
- Un archivo de Excel con los datos incluidos en cada tabla y los resultados que deberían arrojar las consultas en función de ellos.
- Archivos .sql a partir de los cuales se generó la BBDD
- Un archivo .backup para volver a crear la BBDD.
- Archivos .sql donde están realizadas las consultas para ejecutar sobre la BBDD generada.
- Archivos .png donde se muestra el resultado de cada consulta.
- Un archivo del programa TOAD con el que poder ver el modelo Relacional creado.

Para ingresar los datos en la BBDD han sido guardados los archivos de la inserción en formato .txt, si se desea crear la BBDD de nuevo a partir de estos archivos se ha de seguir un orden específico, ya que las restricciones de parentesco no nos permiten seguir un orden aleatorio. El orden más adecuado (aunque puede haber unos pequeños cambios) sería: Surtidor, Tienda, Empleado (dividido en F1, F2, F3, ...), Teléfono, Oferta, Ticket, Artículo, Contiene, Cliente, Reposta, Opinión, Canjea, Gasóleo, Gasolina, GLP e Hidrógeno.

La inserción de los empleados debe seguir un orden específico el cual lo hemos dividido en fases, esto se debe a que los declarados como jefes no pueden ser declarados a la vez que los empleados a los que dirigen, por este motivo hay distintos archivos de empleados.