

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Laboratorio Sistema de Base de Datos 1 N

Manual Técnico

Practica 2

<i>Virginia Sarai Gutierrez Depaz</i>	<i>201504443</i>
<i>Maynor Octavio Piló Tuy</i>	<i>201531166</i>

06/03/2022

Descripción del sistema operativo

- Microsoft Windows
- procesador intel core i7
- Memoria: 12GB (64 bits)
- Espacio en disco: el espacio libre en disco de 10.8GB

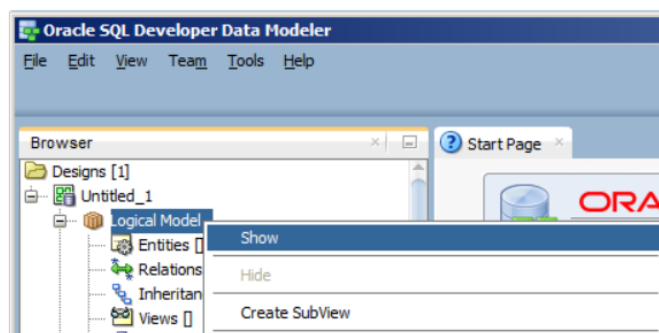
Descripción de la herramienta Data Modeler

Oracle SQL Developer Data Modeler es una herramienta gráfica gratuita que mejora la productividad y simplifica las tareas de modelado de datos.

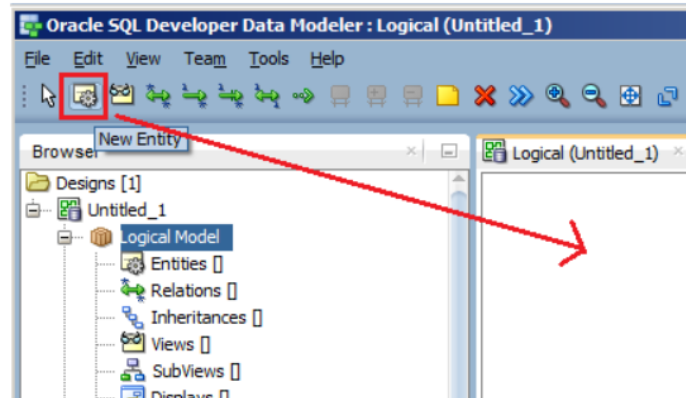
El Data Modeler proporciona capacidades de ingeniería directa e inversa y apoya el desarrollo colaborativo a través del control de código fuente integrado.

Uso de la herramienta Data Modeler

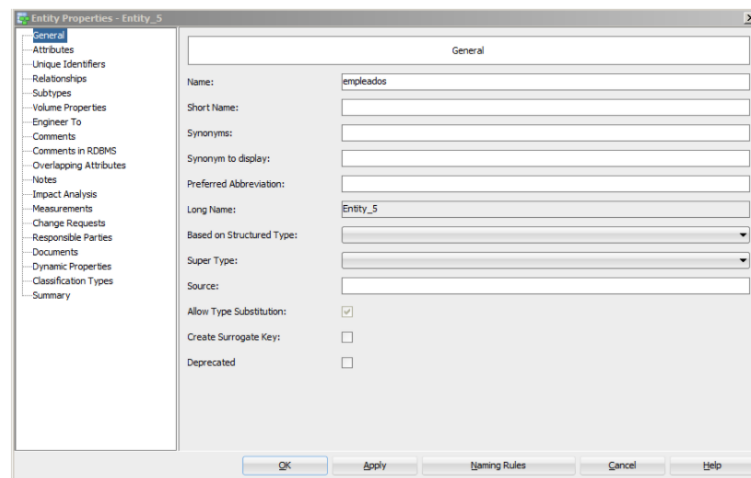
Abrimos el IDE y en la sección de **LOGICAL MODEL** hacemos click derecho y presionamos **SHOW**



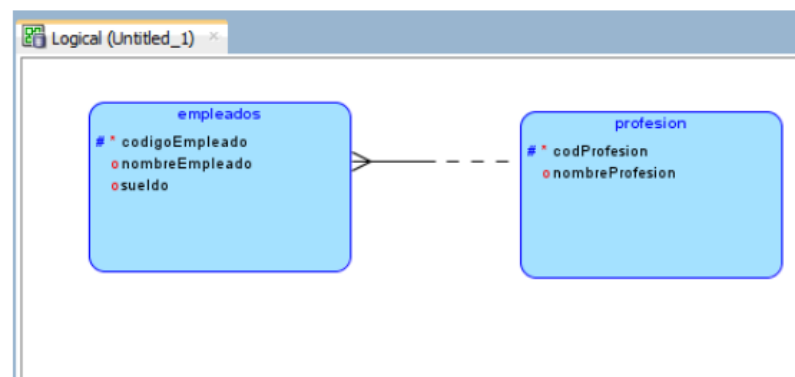
El editor de logical Model es visible, podemos agregar una nueva entidad haciendo click en el icono ENTITY y luego dando click en la superficie del diseñador logico



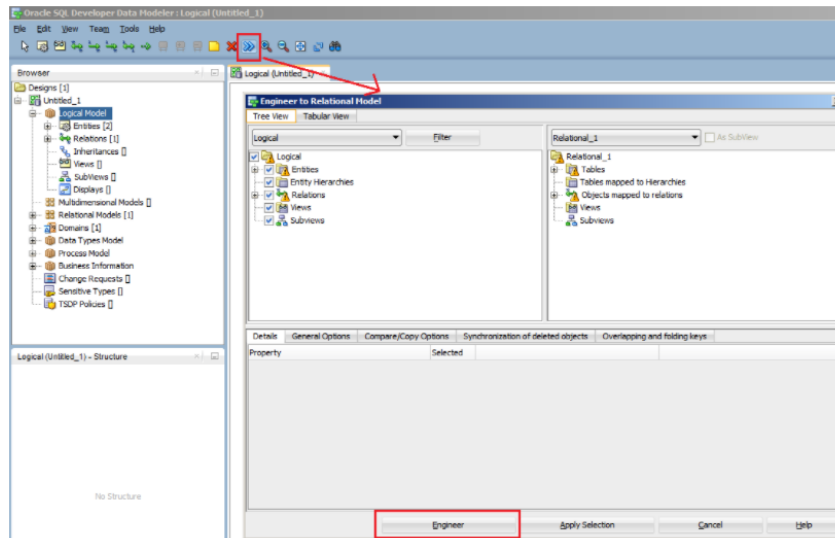
Despu s ingresamos el nombre de la entidad y sus atributos especificando la clave primaria.



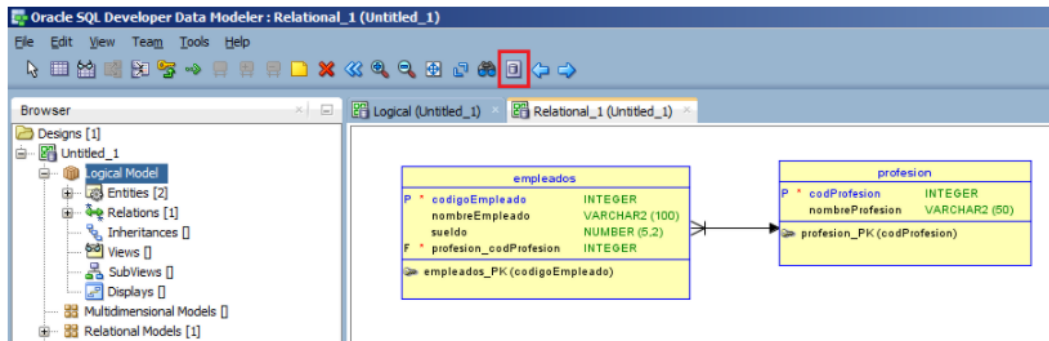
Para relacionar entidades presionar click en el boton de la relacion que corresponda entre dos entidades, automaticamene aparecer  una venta para ingresar el nombre de la relacion que acabamos de crear, la relacion ya es visible



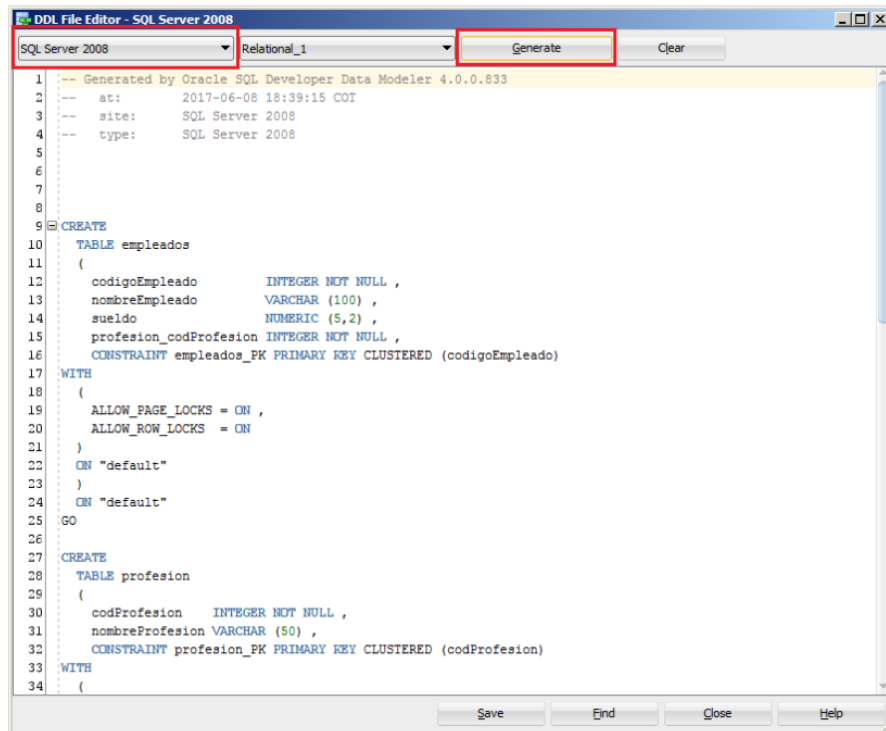
Un vez tenemos nuestro diseño conceptual toca generar el modelo logico, para ello presionamos click en el boton >> aparecerà una ventana para confirmar que entidades vamos a para al modelo logico y presionamos el boton **ENGINEER**



ya tenemos nuestro modelo logico, ahora toca diseñar el script de base de datos para eso presionamos click con el boton marcado en la siguiente imagen



En la siguiente ventana elegir la base de datos y presionamos **GENERATE** automáticamente se generará el script DDL para la base de datos seleccionada



Enunciado 1

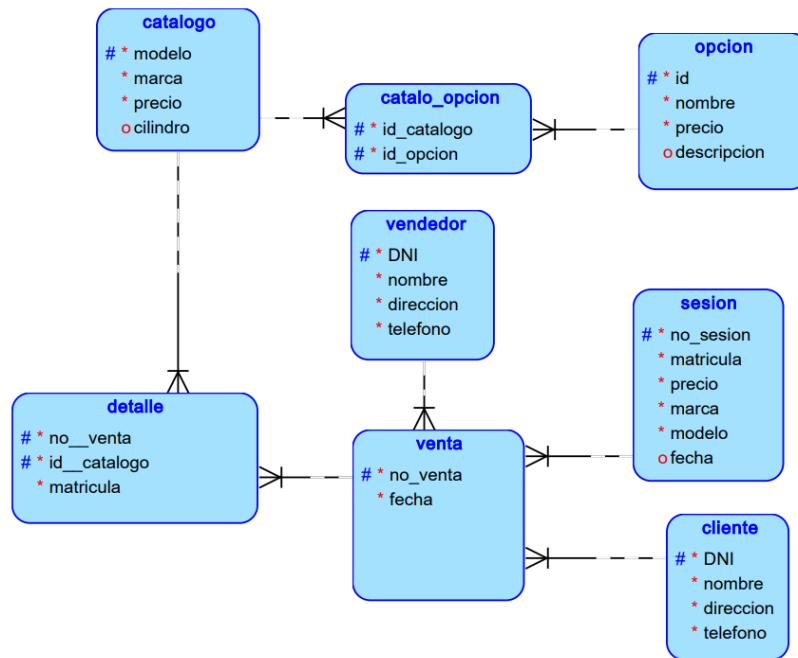
Un concesionario de automóviles desea informatizar su gestión de ventas de vehículos. En particular, se quiere tener almacenada la información referente a los clientes que compran en el concesionario, los vehículos vendidos, así como los vendedores que realizan las distintas ventas. Para ello se tendrá en cuenta que:

- El concesionario dispone de un catálogo de vehículos definidos por su marca, modelo, cilindrada y precio.
- Cada uno de los modelos dispondrá de unas opciones adicionales (aire acondicionado, pintura metalizada, etc.).

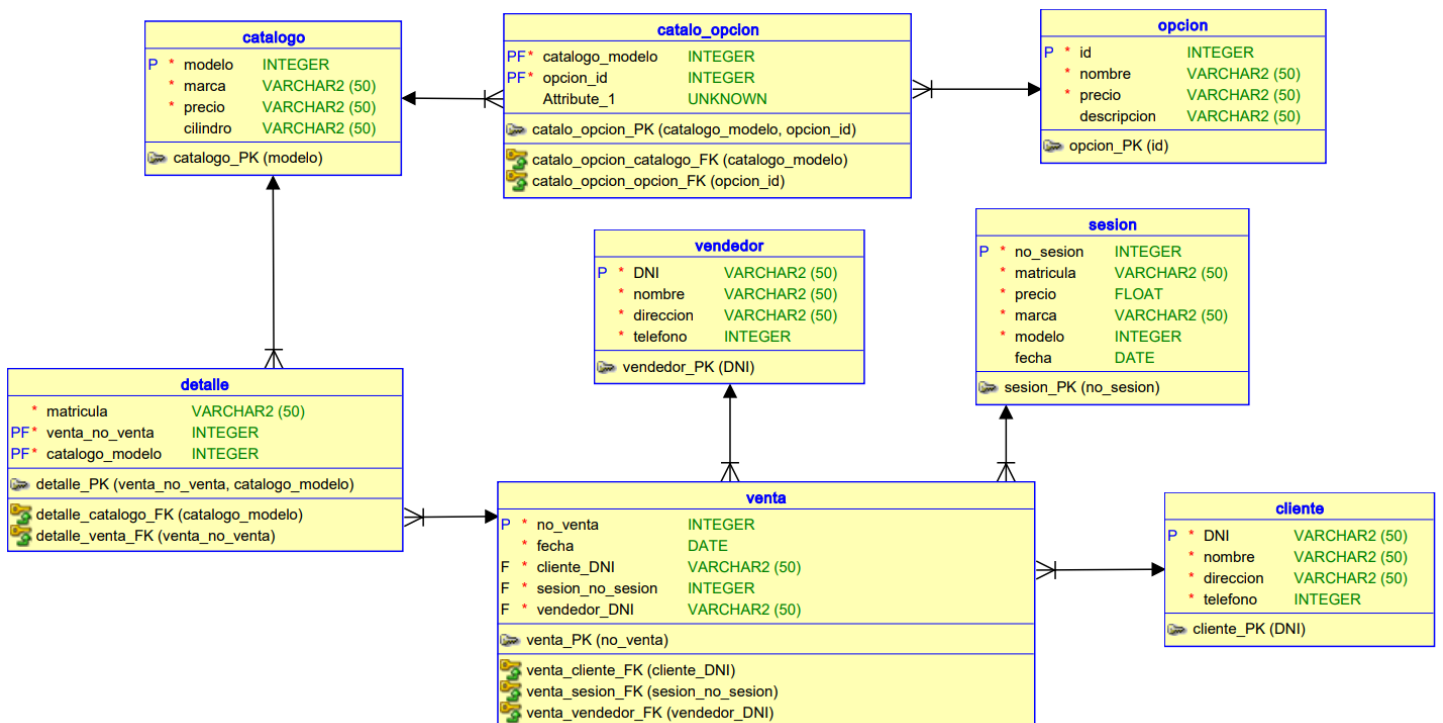
Las opciones vienen definidas por un nombre y una descripción. Hay que tener en cuenta que una opción puede ser común para varios modelos variando sólo el precio en cada caso.

- En cuanto a los clientes, la información de interés es el nombre, DNI, dirección y teléfono, lo mismo que para los vendedores.
- Los clientes pueden ceder su vehículo usado en el momento de comprar un vehículo nuevo. El vehículo usado vendrá definido por su marca, modelo, matrícula y precio de tasación. Es importante conocer la fecha en la que el cliente realiza esta cesión.
- Se desea saber qué vendedor ha vendido qué modelo a qué cliente. También la fecha de la venta y la matrícula del nuevo vehículo. Es importante así mismo saber las opciones que el cliente ha elegido para el modelo que compra.

Esquema Conceptual



Esquema Lógico



Modelo Físico

```
CREATE TABLE catalo_opcion (  
    catalogo_modelo INTEGER NOT NULL,  
    opcion_id      INTEGER NOT NULL,  
);
```

```
ALTER TABLE catalo_opcion ADD CONSTRAINT catalo_opcion_pk PRIMARY KEY (  
catalogo_modelo, opcion_id );
```

```
CREATE TABLE catalogo (  
    modelo  INTEGER NOT NULL,  
    marca   VARCHAR2(50) NOT NULL,  
    precio  VARCHAR2(50) NOT NULL,  
    cilindro VARCHAR2(50)  
);
```

```
ALTER TABLE catalogo ADD CONSTRAINT catalogo_pk PRIMARY KEY ( modelo );
```

```
CREATE TABLE cliente (  
    dni      VARCHAR2(50) NOT NULL,  
    nombre   VARCHAR2(50) NOT NULL,  
    direccion VARCHAR2(50) NOT NULL,  
    telefono INTEGER NOT NULL  
);
```

```
ALTER TABLE cliente ADD CONSTRAINT cliente_pk PRIMARY KEY ( dni );
```

```
CREATE TABLE detalle (  
    matricula   VARCHAR2(50) NOT NULL,  
    venta_no_venta INTEGER NOT NULL,  
    catalogo_modelo INTEGER NOT NULL  
);
```

```
ALTER TABLE detalle ADD CONSTRAINT detalle_pk PRIMARY KEY ( venta_no_venta,  
catalogo_modelo );
```

```
CREATE TABLE opcion (  
    id      INTEGER NOT NULL,  
    nombre  VARCHAR2(50) NOT NULL,  
    precio  VARCHAR2(50) NOT NULL,  
    descripcion VARCHAR2(50)  
);
```

```
ALTER TABLE opcion ADD CONSTRAINT opcion_pk PRIMARY KEY ( id );
```



```
CREATE TABLE sesion (
    no_sesion INTEGER NOT NULL,
    matricula VARCHAR2(50) NOT NULL,
    precio    FLOAT NOT NULL,
    marca     VARCHAR2(50) NOT NULL,
    modelo    INTEGER NOT NULL,
    fecha     DATE
);
```

```
ALTER TABLE sesion ADD CONSTRAINT sesion_pk PRIMARY KEY ( no_sesion );
```

```
CREATE TABLE vendedor (
    dni        VARCHAR2(50) NOT NULL,
    nombre     VARCHAR2(50) NOT NULL,
    direccion  VARCHAR2(50) NOT NULL,
    telefono   INTEGER NOT NULL
);
```

```
ALTER TABLE vendedor ADD CONSTRAINT vendedor_pk PRIMARY KEY ( dni );
```

```
CREATE TABLE venta (
    no_venta      INTEGER NOT NULL,
    fecha         DATE NOT NULL,
    cliente_dni   VARCHAR2(50) NOT NULL,
    sesion_no_sesion INTEGER NOT NULL,
    vendedor_dni  VARCHAR2(50) NOT NULL
);
```

```
ALTER TABLE venta ADD CONSTRAINT venta_pk PRIMARY KEY ( no_venta );
```

```
ALTER TABLE catalo_opcion
ADD CONSTRAINT catalo_opcion_catalogo_fk FOREIGN KEY ( catalogo_modelo )
REFERENCES catalogo ( modelo );
```

```
ALTER TABLE catalo_opcion
ADD CONSTRAINT catalo_opcion_opcion_fk FOREIGN KEY ( opcion_id )
REFERENCES opcion ( id );
```

```
ALTER TABLE detalle
ADD CONSTRAINT detalle_catalogo_fk FOREIGN KEY ( catalogo_modelo )
REFERENCES catalogo ( modelo );
```

```
ALTER TABLE detalle
ADD CONSTRAINT detalle_venta_fk FOREIGN KEY ( venta_no_venta )
REFERENCES venta ( no_venta );
```

```
ALTER TABLE venta
ADD CONSTRAINT venta_cliente_fk FOREIGN KEY ( cliente_dni )
REFERENCES cliente ( dni );
```

```
ALTER TABLE venta
ADD CONSTRAINT venta_sesion_fk FOREIGN KEY ( sesion_no_sesion )
REFERENCES sesion ( no_sesion );
```

```
ALTER TABLE venta
ADD CONSTRAINT venta_vendedor_fk FOREIGN KEY ( vendedor_dni )
REFERENCES vendedor ( dni );
```

Descripción de Tablas

Tabla catalogo

modelo	marca	cilindro	precio Q
Q	Infiniti	6	80,000
911	Porsche	4	85,000
catera	Cadillac	4	125,000
M3	BMW	3	100,000

Tabla opcion

id	nombre	descripcion	precio
1	nacarado	Requiere el uso de un pigmento especial que refleje la luz solar de manera diferente según su ubicación en la capa de color. Así es como aparece un brillo especial de nácar	10,000
2	Tercera fila	Asientos que permiten que se instale una fila extra de asientos en la parte trasera del vehículo.	10,000
3	Aro Aluminio:	Se debe al gran rendimiento entre el peso del vehículo y su resistencia. Es un material con fácil reparación para golpes leves y sin duda su bajo costo es una de sus ventajas más fuertes	5,000

Tabla cliente

DNI	nombre	direccion	telefono
43772-0023	Mureil Gore	96270 Armistice Crossing	(518)7067916
258161-155	Field Targetter	85 Lake View Way	(626) 9514075
30781-5421	Sara Sturch	92 Sauthoff Trail	(353) 4456340
52686-235	Kayley Ripon	20 McCormick Plaza	(276) 4569582

Tabla vendedor

DNI	nombre	direccion	telefono
55154-2709	Bentley Christophers	76 Tennyson Drive	(157) 5744326
68428-015	Clayborn Gilbertson	70082 Di Loreto Road	(159) 8721033
63304-736	Joy De Haven	92 Sauthoff Trail3092 Gale Center	(690) 5927027
52584-168	Ashley Tomkiss	099 Center Place	(970) 1098263

Tabla cesion

no_cesion	matricula	precio	marca	modelo	fecha
1	7792 VVA	16300	dni	dni	5/28/2021
2	6135 HPC	14085	Pontiac	CC	7/3/2021
3	2217 AJD	16390	Toyota	Leganza	7/22/2021
4	2471 MYL	30214	Nissan	E250	9/20/2021

Tabla venta

no_venta	id_cliente	id_vendedor	id_cesion	fecha
1	43772-002 3	55154-2709		5/19/2021
2	258161-15 5	68428-015		9/3/2021
3	30781-542 1	63304-736	3	7/22/2021
4	52686-235	52584-168	4	9/20/2021

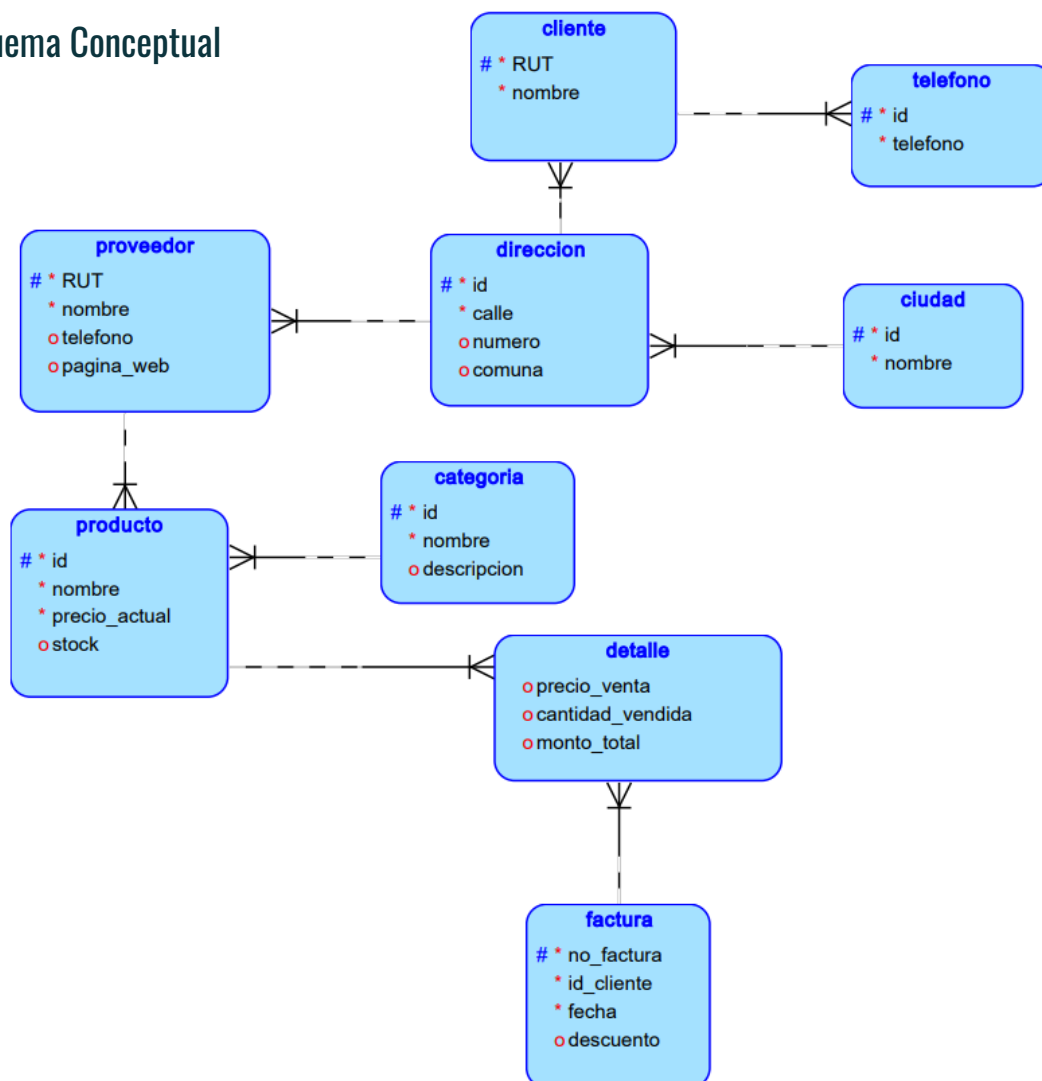
Tabla detalle

no_venta	id_catalogo	matricula	
1	Q	9072 QXV	
2	911	1361 TGR	
3	catera	6418 TQJ	
4	M3	4749 XIX	

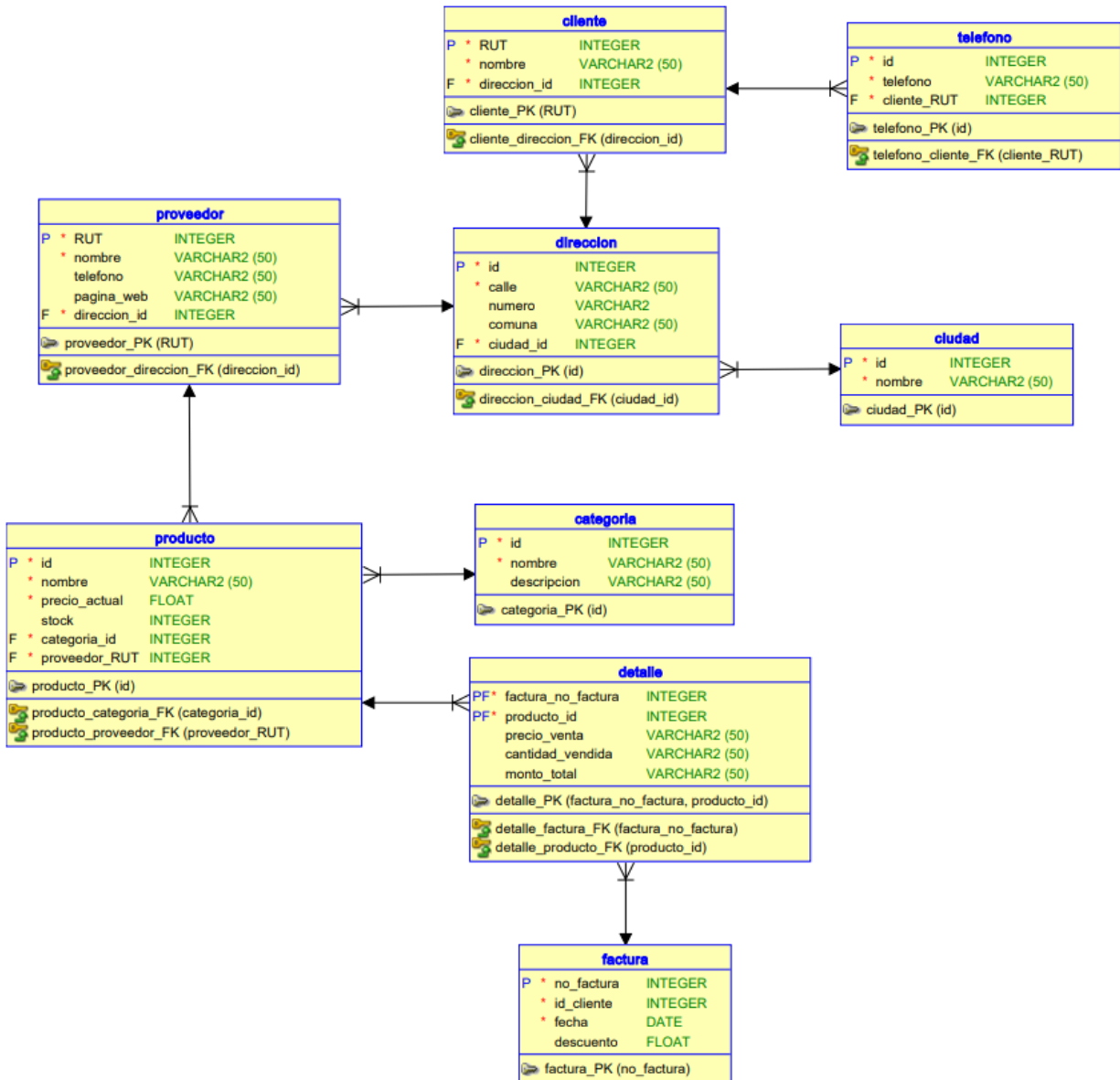
Enunciado 2

Le contratan para hacer una BD que permita apoyar la gestión de un sistema de ventas. La empresa necesita llevar un control de proveedores, clientes, productos y ventas. Un proveedor tiene un RUT, nombre, dirección, teléfono y página web. Un cliente también tiene RUT, nombre, dirección, pero puede tener varios teléfonos de contacto. La dirección se entiende por calle, número, comuna y ciudad. Un producto tiene un id único, nombre, precio actual, stock y nombre del proveedor. Además, se organizan en categorías, y cada producto va sólo en una categoría. Una categoría tiene id, nombre y descripción. Por razones de contabilidad, se debe registrar la información de cada venta con un id, fecha, cliente, descuento y monto final. Además, se debe guardar el precio al momento de la venta, la cantidad vendida y el monto total por el producto.

Esquema Conceptual



Esquema Lógico



Modelo Fisico

```
CREATE TABLE categoria (  
  id      INTEGER NOT NULL,  
  nombre  VARCHAR2(50) NOT NULL,  
  descripcion VARCHAR2(50)  
);
```

```
ALTER TABLE categoria ADD CONSTRAINT categoria_pk PRIMARY KEY ( id );
```

```
CREATE TABLE ciudad (  
  id      INTEGER NOT NULL,  
  nombre  VARCHAR2(50) NOT NULL  
);
```

```
ALTER TABLE ciudad ADD CONSTRAINT ciudad_pk PRIMARY KEY ( id );
```

```
CREATE TABLE cliente (  
  rut      INTEGER NOT NULL,  
  nombre   VARCHAR2(50) NOT NULL,  
  direccion_id INTEGER NOT NULL  
);
```

```
ALTER TABLE cliente ADD CONSTRAINT cliente_pk PRIMARY KEY ( rut );
```

```
CREATE TABLE detalle (  
  factura_no_factura INTEGER NOT NULL,  
  producto_id        INTEGER NOT NULL,  
  precio_venta        VARCHAR2(50),  
  cantidad_vendida    VARCHAR2(50),  
  monto_total         VARCHAR2(50)  
);
```

```
ALTER TABLE detalle ADD CONSTRAINT detalle_pk PRIMARY KEY ( factura_no_factura,  
 producto_id );
```

```
CREATE TABLE direccion (  
  id      INTEGER NOT NULL,  
  calle   VARCHAR2(50) NOT NULL,  
  numero  VARCHAR2,  
  comuna  VARCHAR2(50),  
  ciudad_id INTEGER NOT NULL  
);
```

```
ALTER TABLE direccion ADD CONSTRAINT direccion_pk PRIMARY KEY ( id );
```

```
CREATE TABLE factura (  
    no_factura INTEGER NOT NULL,  
    id_cliente INTEGER NOT NULL,  
    fecha     DATE NOT NULL,  
    descuento FLOAT  
);
```

```
ALTER TABLE factura ADD CONSTRAINT factura_pk PRIMARY KEY ( no_factura );
```

```
CREATE TABLE producto (  
    id          INTEGER NOT NULL,  
    nombre      VARCHAR2(50) NOT NULL,  
    precio_actual FLOAT NOT NULL,  
    stock       INTEGER,  
    categoria_id INTEGER NOT NULL,  
    proveedor_rut INTEGER NOT NULL  
);
```

```
ALTER TABLE producto ADD CONSTRAINT producto_pk PRIMARY KEY ( id );
```

```
CREATE TABLE proveedor (  
    rut          INTEGER NOT NULL,  
    nombre       VARCHAR2(50) NOT NULL,  
    telefono     VARCHAR2(50),  
    pagina_web   VARCHAR2(50),  
    direccion_id INTEGER NOT NULL  
);
```

```
ALTER TABLE proveedor ADD CONSTRAINT proveedor_pk PRIMARY KEY ( rut );
```

```
CREATE TABLE telefono (  
    id          INTEGER NOT NULL,  
    telefono    VARCHAR2(50) NOT NULL,  
    cliente_rut INTEGER NOT NULL  
);
```

```
ALTER TABLE telefono ADD CONSTRAINT telefono_pk PRIMARY KEY ( id );
```

```
ALTER TABLE cliente  
    ADD CONSTRAINT cliente_direccion_fk FOREIGN KEY ( direccion_id )  
        REFERENCES direccion ( id );
```

```
ALTER TABLE detalle  
    ADD CONSTRAINT detalle_factura_fk FOREIGN KEY ( factura_no_factura )  
        REFERENCES factura ( no_factura );
```



```

ALTER TABLE detalle
  ADD CONSTRAINT detalle_producto_fk FOREIGN KEY ( producto_id )
    REFERENCES producto ( id );

ALTER TABLE direccion
  ADD CONSTRAINT direccion_ciudad_fk FOREIGN KEY ( ciudad_id )
    REFERENCES ciudad ( id );

ALTER TABLE producto
  ADD CONSTRAINT producto_categoria_fk FOREIGN KEY ( categoria_id )
    REFERENCES categoria ( id );

ALTER TABLE producto
  ADD CONSTRAINT producto_proveedor_fk FOREIGN KEY ( proveedor_rut )
    REFERENCES proveedor ( rut );

ALTER TABLE proveedor
  ADD CONSTRAINT proveedor_direccion_fk FOREIGN KEY ( direccion_id )
    REFERENCES direccion ( id );

ALTER TABLE telefono
  ADD CONSTRAINT telefono_cliente_fk FOREIGN KEY ( cliente_rut )
    REFERENCES cliente ( rut );

```

Descripción de Tablas

Tabla ciudad

id	nombre
1	Alegrete
2	Laixi
3	Jaruco
4	Semongkat

Tabla direccion

id_direccion	calle	numero	comuna	id_ciudad
1	Grim	90	Elgar	1
2	Cascade	13	Colorado	2
3	Melody	64	Clemons	2
4	Mesta	47	Drewry	4

Tabla cliente

RUT	nombre	id_direccion
43772	Mureil Gore	1
25816	Field Targetter	2
30781	Sara Sturch	3
52686	Kayley Ripon	4

Tabla teléfono

id	telefono	id_cliente
1	814-637-7906	1
2	757-570-0804	2
3	477-752-6862	2
4	988-595-5771	4

Tabla proveedor

rut	nombre	id_direccion	telefono	pagina_web
1	Chuck Lording	1	217-293-6611	http://patch.com/ut.js?phasellus=a&sit=nibh
2	Cammie Claye	2	610-136-0757	https://t-online.de/duis/bibendum/morbi/non/quam/nec.js?feugiat=suspendisse&et=potenti&eros=cras
3	Eddy Oaker	2	629-401-1951	http://gnu.org/id/ornare.jpg?et=in&tempus=congue
4	Isabella L'Episcopi	4	691-804-0435	https://virginia.edu/in.js?elementum=dui&nullam=luctus&varius=rutrum&nulla=nulla

Tabla categoria

id	nombre	descripcion
1	Galletas	galletas can can
2	refrescos	refrescos en lata
3	agua	agua en botellas de 1 litro
4	articulos de limpieza	detergentes de 1kilo

Tabla producto

id	nombre	precio_actual	stock	id_proveedor	id_categoria
1	Can Can	1	10	1	1
2	Rica roja	2.5	20	2	2
3	Salvadidas	5	15	3	3
4	Ace	3.5	3	4	4

Tabla factura

no_factura	id_cliente	fecha	descuento	id_proveedor	id_categoria
123456	1	4/5/2021	10%	1	1
256312	2	9/17/2021	5%	2	2
378923	3	10/23/2021	1%	3	3
454646	4	1/7/2022	3%	4	4

Tabla detalle

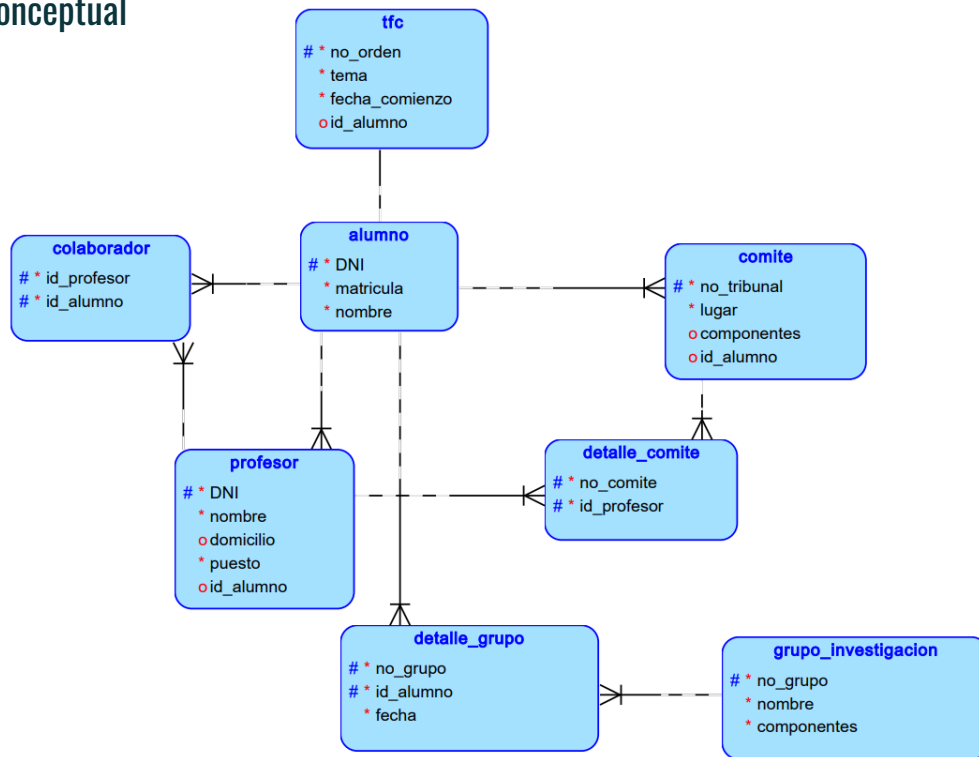
no_factura	id_producto	precio_actual	cantidad_	monto_total
123456	1	1	3	3
256312	2	2.5	5	12.5
378923	3	5	2	10
454646	4	3.5	5	17.5

Enunciado 3

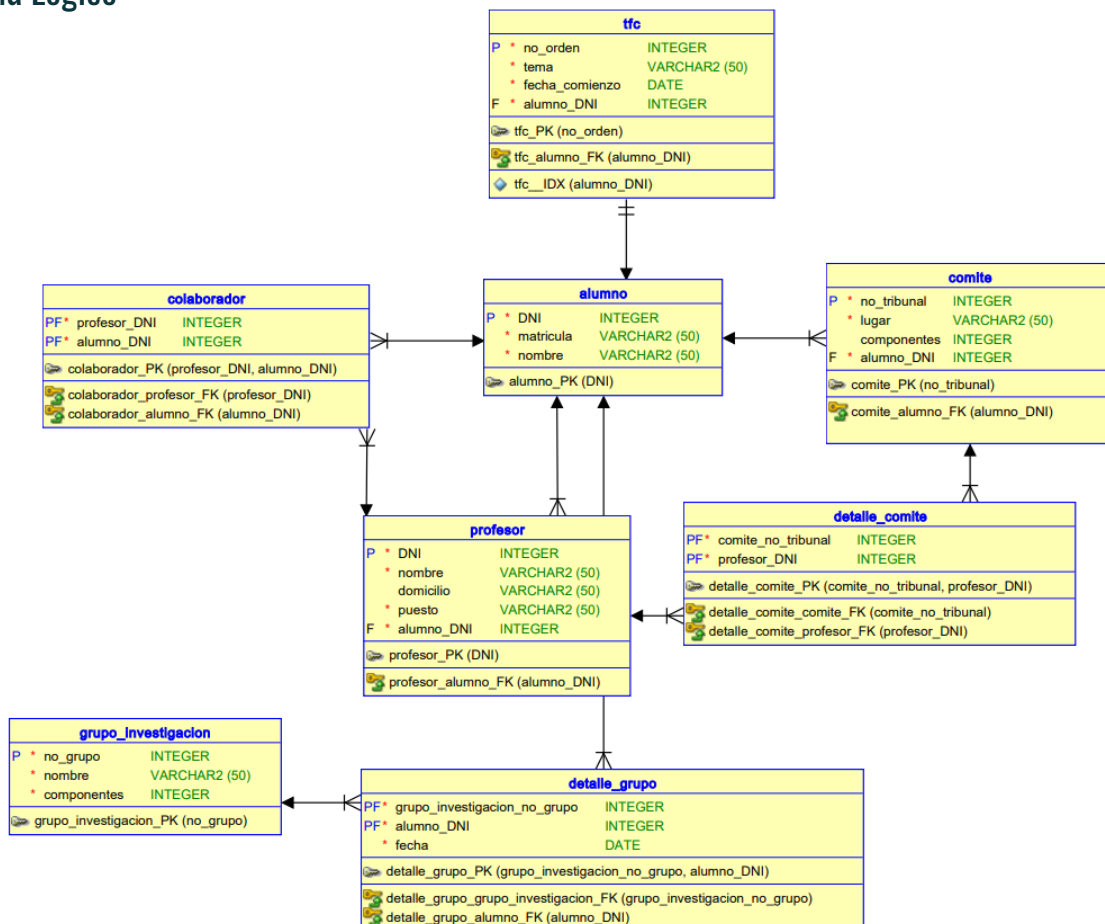
Una Escuela de Computación quiere generar un sistema para tener controlado en una base de datos todo lo referente a los Trabajos Fin de Carrera (TFC): alumnos que los realizan, profesores que los dirigen, temas de los que tratan y tribunales que los corrigen. Por tanto, es de interés:

- Que los alumnos se definan por su número de matrícula, DNI y nombre. Un alumno realiza, evidentemente, sólo un T.F.C.
- Que los T.F.C. se definen por su tema, por un número de orden y por la fecha de comienzo. Un T.F.C. determinado, no puede ser realizado por varios alumnos.
- Que un profesor se define por su DNI, nombre y domicilio; y puesto que los T.F.C. son del área en el que trabaja, NO interesa conocer el T.F.C. que dirige sino a qué alumno se lo dirige.
- Que un Comité está formado por varios profesores y los profesores pueden formar parte de varios Comités. Por otra parte, sí es de interés para el comité conocer qué alumno es el que se presenta, con qué T.F.C. y en qué fecha lo ha defendido. El comité se define por un número de tribunal, lugar de examen y por el número de componentes.
- Al margen de esto, un alumno puede haber pertenecido a algún grupo de investigación del que haya surgido la idea del T.F.C. Dichos grupos se identifican por un número de grupo, su nombre y por su número de componentes. Un alumno no puede pertenecer a más de un grupo y no es de interés saber si el grupo tiene algo que ver o no con el T.F.C. del alumno; sí siendo de interés la fecha de incorporación a dicho grupo.
- Por otra parte, un profesor, al margen de dirigir el T.F.C. de algunos alumnos, puede haber colaborado con otros en la realización de dicho T.F.C. pero siendo otro profesor el que lo dirige. En este caso, sólo es interesante conocer qué profesor ha ayudado a qué alumno (a un alumno le pueden ayudar varios profesores).

Esquema Conceptual



Esquema Lógico



Modelo Fisico

```
CREATE TABLE alumno (  
    dni    INTEGER NOT NULL,  
    matricula VARCHAR2(50) NOT NULL,  
    nombre  VARCHAR2(50) NOT NULL  
);
```

```
ALTER TABLE alumno ADD CONSTRAINT alumno_pk PRIMARY KEY ( dni );
```

```
CREATE TABLE colaborador (  
    profesor_dni INTEGER NOT NULL,  
    alumno_dni  INTEGER NOT NULL  
);
```

```
ALTER TABLE colaborador ADD CONSTRAINT colaborador_pk PRIMARY KEY (  
profesor_dni,alumno_dni );
```

```
CREATE TABLE comite (  
    no_tribunal INTEGER NOT NULL,  
    lugar      VARCHAR2(50) NOT NULL,  
    componentes INTEGER,  
    alumno_dni  INTEGER NOT NULL  
);
```

```
ALTER TABLE comite ADD CONSTRAINT comite_pk PRIMARY KEY ( no_tribunal );
```

```
CREATE TABLE detalle_comite (  
    comite_no_tribunal INTEGER NOT NULL,  
    profesor_dni      INTEGER NOT NULL  
);
```

```
ALTER TABLE detalle_comite ADD CONSTRAINT detalle_comite_pk PRIMARY KEY (  
comite_no_tribunal, profesor_dni );
```

```
CREATE TABLE detalle_grupo (  
    grupo_investigacion_no_grupo INTEGER NOT NULL,  
    alumno_dni                  INTEGER NOT NULL,  
    fecha                      DATE NOT NULL  
);
```

```
ALTER TABLE detalle_grupo ADD CONSTRAINT detalle_grupo_pk PRIMARY KEY (  
grupo_investigacion_no_grupo, alumno_dni );
```

```
CREATE TABLE grupo_investigacion (
    no_grupo INTEGER NOT NULL,
    nombre VARCHAR2(50) NOT NULL,
    componentes INTEGER NOT NULL
);
```

```
ALTER TABLE grupo_investigacion ADD CONSTRAINT grupo_investigacion_pk PRIMARY
KEY ( no_grupo );
```

```
CREATE TABLE profesor (
    dni INTEGER NOT NULL,
    nombre VARCHAR2(50) NOT NULL,
    domicilio VARCHAR2(50),
    puesto VARCHAR2(50) NOT NULL,
    alumno_dni INTEGER NOT NULL
);
```

```
ALTER TABLE profesor ADD CONSTRAINT profesor_pk PRIMARY KEY ( dni );
```

```
CREATE TABLE tfc (
    no_orden INTEGER NOT NULL,
    tema VARCHAR2(50) NOT NULL,
    fecha_comienzo DATE NOT NULL,
    alumno_dni INTEGER NOT NULL
);
```

```
CREATE UNIQUE INDEX tfc__idx ON
tfc (
    alumno_dni
ASC );
```

```
ALTER TABLE tfc ADD CONSTRAINT tfc_pk PRIMARY KEY ( no_orden );
```

```
ALTER TABLE colaborador
ADD CONSTRAINT colaborador_alumno_fk FOREIGN KEY ( alumno_dni )
REFERENCES alumno ( dni );
```

```
ALTER TABLE colaborador
ADD CONSTRAINT colaborador_profesor_fk FOREIGN KEY ( profesor_dni )
REFERENCES profesor ( dni );
```

```
ALTER TABLE comite
ADD CONSTRAINT comite_alumno_fk FOREIGN KEY ( alumno_dni )
REFERENCES alumno ( dni );
```

```
ALTER TABLE detalle_comite
ADD CONSTRAINT detalle_comite_comite_fk FOREIGN KEY ( comite_no_tribunal )
REFERENCES comite ( no_tribunal );
```



```
ALTER TABLE detalle_comite
  ADD CONSTRAINT detalle_comite_profesor_fk FOREIGN KEY ( profesor_dni )
    REFERENCES profesor ( dni );
```

```
ALTER TABLE detalle_grupo
  ADD CONSTRAINT detalle_grupo_alumno_fk FOREIGN KEY ( alumno_dni )
    REFERENCES alumno ( dni );
```

```
-- ERROR: FK name length exceeds maximum allowed length(30)
ALTER TABLE detalle_grupo
  ADD CONSTRAINT detalle_grupo_grupo_investigacion_fk FOREIGN KEY (
    grupo_investigacion_no_grupo )
    REFERENCES grupo_investigacion ( no_grupo );
```

```
ALTER TABLE profesor
  ADD CONSTRAINT profesor_alumno_fk FOREIGN KEY ( alumno_dni )
    REFERENCES alumno ( dni );
```

```
ALTER TABLE tfc
  ADD CONSTRAINT tfc_alumno_fk FOREIGN KEY ( alumno_dni )
    REFERENCES alumno ( dni );
```

Descripción de Tablas

Tabla alumno

DNI	matricula	nombre
56002193E	1453MGL	Ezra St. Queintain
48297242W	3884LJK	Christyna Hews
46662535C	4724MFH	Olenolin Hewlings
23936105M	0731JDB	Andie Kieran

Tabla tfc

no_orden	tema	fecha_comienzo	id_alumno
1	Major Banks	6/19/2021	56002193E
2	Natural Gas Distribution	8/22/2021	48297242W
3	Computer Software: Prepackaged Software	2/2/2022	46662535C
4	Electromedical & Electrotherapeutic Apparatus	2/21/2022	23936105M

Tabla profesor

DNI	nombre	domicilio	puesto	id_alumno
59259903N	Jonatan Villa Marti	Camino De España, 28	Supervisor	56002193E
11310253A	Hortensia Olmedo Mateo	Estrada Horno, 51	Supervisor	48297242W
63780554J	Izan Hidalgo Carbonell	Passatge Pedralbes, 93	Supervisor	46662535C
41222898K	Lara Cordoba Pacheco	Camí Mayor, 14	Supervisor	23936105M

Tabla comite

no_tribunal	lugar	componente	id_alumno
592	Campo Horno, 70	2	56002193E
113	Parque Madrid, 8	3	48297242W
637	Urbanización Madrid, 79	4	46662535C
412	Travesía Iglesia, 68	5	23936105M

Tabla detalle_comite

no_comite	id_profesor
592	59259903N
113	11310253A
592	63780554J
113	59259903N

Tabla colaborador

id_alumno	id_profesor
56002193E	59259903N
48297242W	11310253A
46662535C	63780554J
23936105M	59259903N

Tabla grupo_investigacion

no_grupo	nombre	componente
59	Fintone	2
11	Veribet	6
52	Sonair	2
13	Voyatouch	5

Tabla detalle_grupo

no_grupo	id_alumno	fecha
59	56002193E	10/22/2021
11	48297242W	7/30/2021
52	46662535C	10/15/2021
13	23936105M	1/14/2022

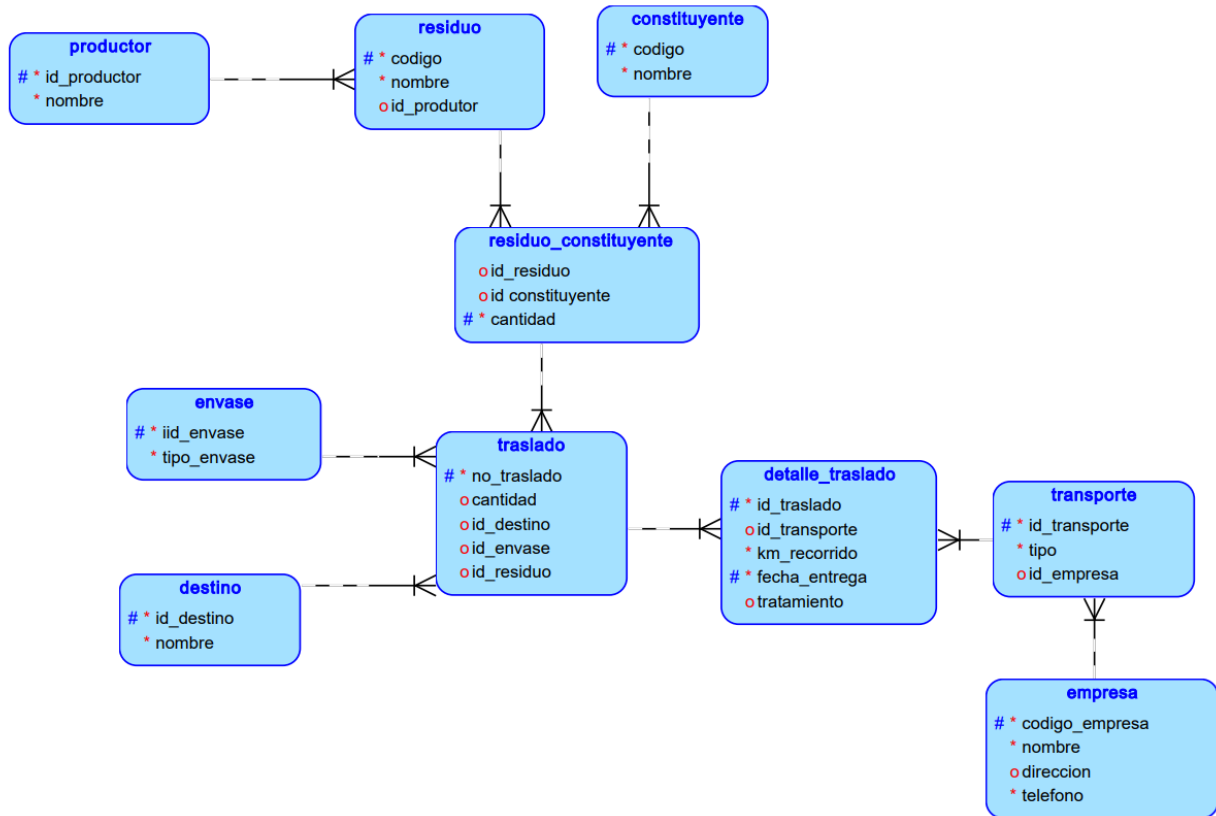
Enunciado 4

Se desea construir un sistema de información para la gestión de los residuos sólidos y peligrosos producidos en una ciudad. La información para contemplar es la que corresponde desde que el residuo es producido hasta que llega a un lugar seguro y recibe un tratamiento especial. Esto incluye información de los residuos, sus productores, las empresas que los transportan a los lugares seguros y los traslados de los residuos. Las reglas de gestión para tener en cuenta son las siguientes:

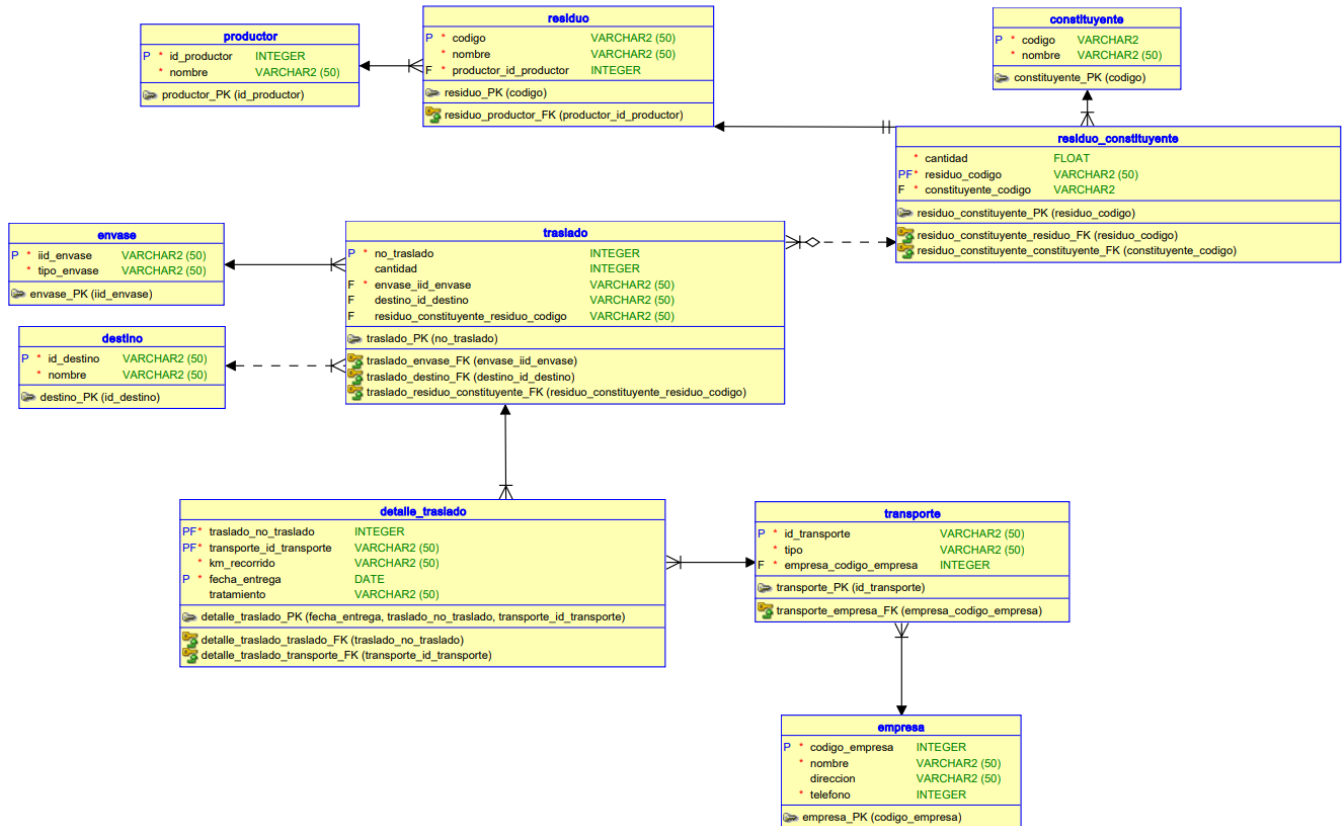
Un productor (empresa u organización) produce un número variable de residuos constituidos, a su vez, por un número variable de constituyentes químicos.

- Más de un productor puede producir residuos con iguales constituyentes químicos y con las mismas o distintas cantidades de dichos constituyentes.
- Los productores asignan un código único a los residuos que producen, lo que les permite diferenciar distintas producciones de los mismos productos.
- Además, más de un productor puede asignar el mismo código a los residuos que produce.
- Los residuos pueden ser trasladados en su totalidad (cantidad total del mismo) o en partes, o no ser trasladados nunca.
- En cada traslado de residuos la cantidad que se traslada de los mismos es enviada a un único destino.
- En cada traslado puede intervenir más de una empresa transportista usando el mismo o distinto transporte, por lo que resulta interesante conocer tanto el medio de transporte utilizado como los kilómetros realizados, así como el coste del trabajo.
- El residuo se traslada en un tipo de envase determinado por el productor y que no varía a lo largo de su traslado.
- Interesa conocer la fecha de llegada a destino y el tratamiento posterior a que se someten los residuos.
- Por seguridad, se considera que en un traslado sólo puede trasladarse un residuo de un productor.

Esquema Conceptual



Esquema Lógico



Modelo Fisico

```
CREATE TABLE constituyente (  
    codigo VARCHAR2  
    nombre VARCHAR2(50) NOT NULL  
);
```

```
ALTER TABLE constituyente ADD CONSTRAINT constituyente_pk PRIMARY KEY ( codigo );
```

```
CREATE TABLE destino (  
    id_destino VARCHAR2(50) NOT NULL,  
    nombre    VARCHAR2(50) NOT NULL  
);
```

```
ALTER TABLE destino ADD CONSTRAINT destino_pk PRIMARY KEY ( id_destino );
```

```
CREATE TABLE detalle_traslado (  
    traslado_no_traslado    INTEGER NOT NULL,  
    transporte_id_transporte VARCHAR2(50) NOT NULL,  
    km_recorrido            VARCHAR2(50) NOT NULL,  
    fecha_entrega           DATE NOT NULL,  
    tratamiento             VARCHAR2(50)  
);
```

```
ALTER TABLE detalle_traslado  
    ADD CONSTRAINT detalle_traslado_pk PRIMARY KEY (  
fecha_entrega,traslado_no_traslado,transporte_id_transporte );
```

```
CREATE TABLE empresa (  
    codigo_empresa INTEGER NOT NULL,  
    nombre          VARCHAR2(50) NOT NULL,  
    direccion       VARCHAR2(50),  
    telefono        INTEGER NOT NULL  
);
```

```
ALTER TABLE empresa ADD CONSTRAINT empresa_pk PRIMARY KEY ( codigo_empresa );
```

```
CREATE TABLE envase (  
    iid_envase VARCHAR2(50) NOT NULL,  
    tipo_envase VARCHAR2(50) NOT NULL  
);
```

```
ALTER TABLE envase ADD CONSTRAINT envase_pk PRIMARY KEY ( iid_envase );
```



```
CREATE TABLE productor (
    id_productor INTEGER NOT NULL,
    nombre VARCHAR2(50) NOT NULL
);
```

```
ALTER TABLE productor ADD CONSTRAINT productor_pk PRIMARY KEY ( id_productor );
```

```
CREATE TABLE residuo (
    codigo VARCHAR2(50) NOT NULL,
    nombre VARCHAR2(50) NOT NULL,
    productor_id_productor INTEGER NOT NULL
);
```

```
ALTER TABLE residuo ADD CONSTRAINT residuo_pk PRIMARY KEY ( codigo );
```

```
CREATE TABLE residuo_constituyente (
    cantidad FLOAT NOT NULL,
    residuo_codigo VARCHAR2(50) NOT NULL,
    constituyente_codigo VARCHAR2
);
```

```
ALTER TABLE residuo_constituyente ADD CONSTRAINT residuo_constituyente_pk PRIMARY KEY ( residuo_codigo );
```

```
CREATE TABLE transporte (
    id_transporte VARCHAR2(50) NOT NULL,
    tipo VARCHAR2(50) NOT NULL,
    empresa_codigo_empresa INTEGER NOT NULL
);
```

```
ALTER TABLE transporte ADD CONSTRAINT transporte_pk PRIMARY KEY ( id_transporte );
```

```
CREATE TABLE traslado (
    no_traslado INTEGER NOT NULL,
    cantidad INTEGER,
    envase_iid_envase VARCHAR2(50) NOT NULL,
    destino_id_destino VARCHAR2(50),
    residuo_constituyente_residuo_codigo VARCHAR2(50)
);
```

```
ALTER TABLE traslado ADD CONSTRAINT traslado_pk PRIMARY KEY ( no_traslado );
```

```
ALTER TABLE detalle_traslado
    ADD CONSTRAINT detalle_traslado_transporte_fk FOREIGN KEY (
transporte_id_transporte )
    REFERENCES transporte ( id_transporte );
```

```
ALTER TABLE detalle_traslado
  ADD CONSTRAINT detalle_traslado_traslado_fk FOREIGN KEY ( traslado_no_traslado )
    REFERENCES traslado ( no_traslado );
```

-- ERROR: FK name length exceeds maximum allowed length(30)

```
ALTER TABLE residuo_constituyente
  ADD CONSTRAINT residuo_constituyente_constituyente_fk FOREIGN KEY (
constituyente_codigo )
    REFERENCES constituyente ( codigo );
```

-- ERROR: FK name length exceeds maximum allowed length(30)

```
ALTER TABLE residuo_constituyente
  ADD CONSTRAINT residuo_constituyente_residuo_fk FOREIGN KEY ( residuo_codigo )
    REFERENCES residuo ( codigo );
```

```
ALTER TABLE residuo
  ADD CONSTRAINT residuo_productor_fk FOREIGN KEY ( productor_id_productor )
    REFERENCES productor ( id_productor );
```

```
ALTER TABLE transporte
  ADD CONSTRAINT transporte_empresa_fk FOREIGN KEY ( empresa_codigo_empresa )
    REFERENCES empresa ( codigo_empresa );
```

```
ALTER TABLE traslado
  ADD CONSTRAINT traslado_destino_fk FOREIGN KEY ( destino_id_destino )
    REFERENCES destino ( id_destino );
```

```
ALTER TABLE traslado
  ADD CONSTRAINT traslado_envase_fk FOREIGN KEY ( envase_iid_envase )
    REFERENCES envase ( iid_envase );
```

Descripción de Tablas

Tabla productor

id_productor	nombre
182	Atomize Sl
340	Telegue Sa
562	Gauleck Sa
262	Alteboss Sl

Tabla residuo

codigo	nombre	id_productor
R8312291A	Residuo de metales	182
R6480513H	Lodos de galvanización	340
B53857348	cenizas de cables de cobre	562
H56003742	Soluciones electrolíticas	262

Tabla constituyente

codigo	nombre
A1020	Antimonio
A1030	Arsénico
A1050	Berilio
A1060	Cadmio

Tabla residuo_constituyente

id residuo	id_constituyente	cantidad
R8312291A	A1020	20
R6480513H	A1030	30
B53857348	A1050	50
H56003742	A1060	60

Tabla residuo_constituyente

id_residuo	id_constituyente	cantidad
R8312291A	A1020	20
R6480513H	A1030	30
B53857348	A1050	50
H56003742	A1060	60

Tabla envase

id_envase	tipo_envase
R83	Envase de plástico debidamente etiquetado
R64	Envase de vidrio debidamente etiquetado

Tabla destino

id_destino	nombre
1	Rúa Pedralbes, 35
2	Avinguda Pedralbes, 42
3	Avenida Real, 83
4	Jardins De España, 45

Tabla traslado

no_traslado	cantidad	id_destino	id_envase	id_residuo
1	35	1	R83	R8312291A
2	42	2	R64	R6480513H
3	83	3	R83	B53857348
4	45	4	R64	H56003742

Tabla empresa

codigo	nombre	direccion	telefono
1568	Vitawastre Sa	Jardins De España, 45	956817292
2752	Generes Sa	Kalea Madrid, 53	958935999
1814	Campigue Sa	Avinguda De España, 26	942726682
2529	Ratiodeus Sa	Kalea Madrid, 35	929117220

Tabla transporte

id_transporte	tipo	id_empresa
1568	Vehículo RSU	1568
2752	Semirremolques	2752
1814	Carga Trasera	1814
2529	Carga Trasera norba	2529

Tabla traslado

id_traslado	id_transporte	km	fecha_entrega	tratamiento
15	1568	10	6/22/2021	Preparación para su reutilización
27	2752	2	11/14/2021	Compostaje
18	1814	18	8/11/2021	Biometanización
25	2529	25	1/19/2022	Biometanización