

# Segona Activitat d'Avaluació

**GEI**

**Grau en Enginyeria Informàtica**

2n Curs GEI  
Grup 2

10/01/2021

Ton Lluçà Senserrich  
47125160T

Eduard Sales Jové  
49539818A

Ian Ribalta Gené  
47983753A

Gerard Llubes Cano  
49381025W

Una empresa, ÒPTIQUES-LLEIDATANES S.A, es dedicada a la venda de material òptic. Aquesta empresa ens ha contractat perquè li dissenyem la base de dades per tal d'automatitzar els seus processos. La raó principal d'aquesta decisió és millorar l'eficiència i reduir els costos de l'empresa.

Els **productes** que comercialitza l'empresa són **lents de contacte** o **ulleres**. De cada **producte** es vol conèixer el **codi de producte** (que és únic), el **seu nom**, el **seu preu de venda** i la **seva marca**. A més, de les **lents de contacte** interessa conèixer la seva **tipologia** (toves, rígides, etc.), el **seu color** i la **graduació**. Per a les **ulleres**, es desitja també emmagatzemar el **model**, **material de fabricació** i la **graduació**.

Tant per a les **ulleres** com per a les **lents de contacte** es disposaran d'una sèrie de **tècnica de neteja** òptimes requerides per a la seva cura i durabilitat. D'aquestes tècniques de neteja és interessant emmagatzemar el seu **nom** i **descripció**. A més, aquestes tècniques de neteja tindran una **categoria associada** (tècniques aquoses, tècniques productes líquids, tècniques productes humits, etc.). Com que, tant les ulleres com les lents, tenen una graduació, és interessant saber el **problema ocular associat** (**miopia**, **astigmatisme**, etc.).

Els **productes** són subministrats per una sèrie de **proveïdors**. Quan es necessita, es realitza una **comanda** de **productes** a un proveïdor amb una **quantitat associada**. En aquest cas, l'empresa necessita emmagatzemar la **història** de tots les **comandes** realitzades a un proveïdor per a cada producte indicant la **quantitat servida** i el **preu de compra**. A més, de cada **proveïdor** es vol emmagatzemar el seu **NIF**, **nom**, **descripció** i **persona de contacte**.

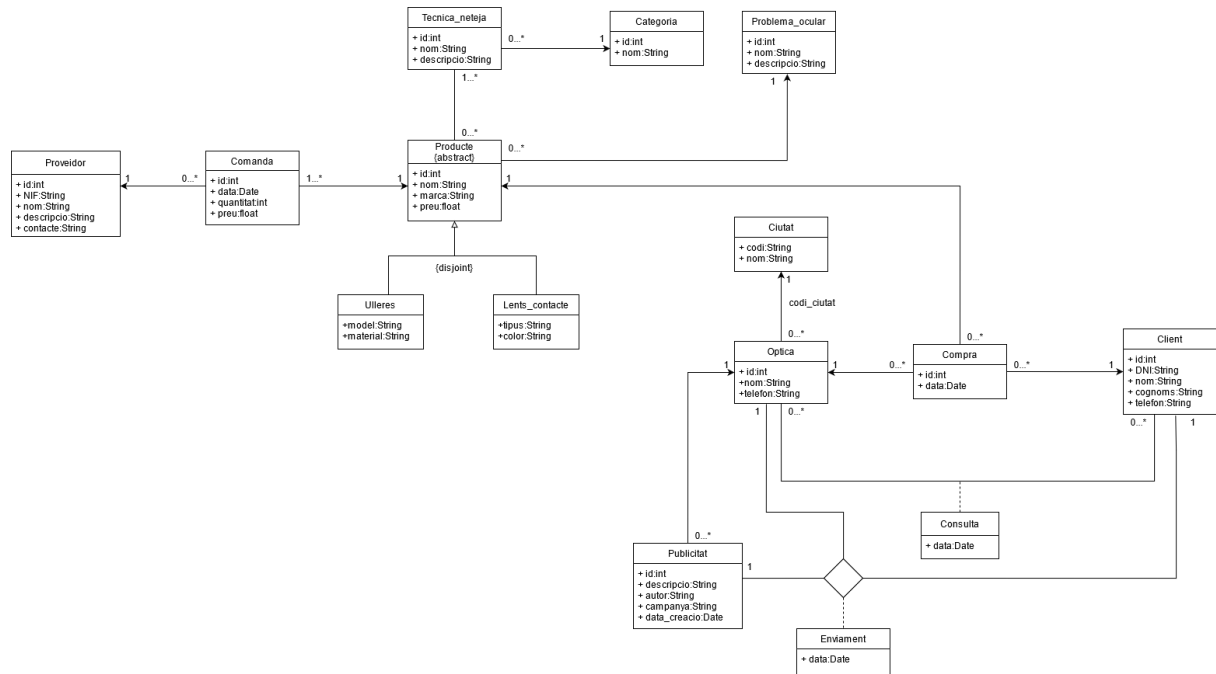
D'altra banda, l'empresa serveix els productes a una sèrie d'**òptiques** en què diferents **clients** compren els seus productes. Aquestes òptiques estan ubicades en diferents **ciutats**. A més, com en una ciutat pot haver moltes òptiques, les **òptiques** queden identificades per un **codi associat a la ciutat** a la qual pertany.

Per tenir controlada la garantia, l'empresa desitja emmagatzemar quins **clients** compren quins productes en què **òptiques**, així com la **data** de compra de cada producte. De les **òptiques**, emmagatzemarem el seu **número d'identificació**, el **seu nom**, **adreça** i **telèfon**. Dels **clients** s'emmagatzemarà el **DNI**, **nom**, **cognoms** i **telèfon**.

ÒPTIQUES-LLEIDATANES S.A també desitja emmagatzemar la informació sobre la **propaganda** que una **òptica** envia als seus clients, tenint en compte que pot haver clients que mai hagin rebut propaganda i que cada client rebrà propaganda d'una sola **òptica**. A més, es vol saber, per al cas dels clients que reben propaganda, la **data de l'últim enviament** per no atosigar-los amb anuncis de l'empresa.

Finalment, l'empresa manté informació de quins **clients** acudeixen a quines **òptiques** per informar-se sobre els seus **productes**. Els **clients** no van necessàriament a les **òptiques** de les que reben **propaganda**. Un **client** pot acudir a diverses **òptiques** interessat a conèixer informació dels **productes**. Per a això, es desitja saber la **data de realització d'aquestes consultes**.

# Disseny del Diagrama UML



## Traducció dels diagrames al model relacional

Proveidor(#id,NIF,nom,descripcio,contacte)

Comanda(#id,idProducte,idProveidor,data,quantitat,preu)

foreign key (idProducte) references Producte(id)

foreign key (idProveidor) references Proveidor(id)

constraint notNullProducte not null(idProducte)

constraint notNullProveidor not null(idProveidor)

Categoria(#id,nom)

Tecnica\_neteja(#id,idCategoria,nom,descripcio)

foreign key (idCategoria) references Categoria(id)

constraint notNullCategoria not null(idCategoria)

Tecnica\_producte(#idProducte,#idTecnica)

foreign key (idProducte) references Producte(id)

foreign key (idTecnica) references Tecnica\_neteja(id)

create assertion totsCheck (not exists (select \* from Producte p where p.id not in(select tp.idProducte from Tecnica\_producte tp)))

Problema\_ocular(#id,nom,descripcio)

Ulleres(#id,idProblema,nom,marca,preu,model,material)  
foreign key (idProblema) references Problema\_ocular(id)  
constraint notNullProblema not null(idProblema)

Lents\_contacte(#id,idProblema,nom,marca,preu,tipus,color)  
constraint notNullProblema not null(idProblema)  
foreign key (idProblema) references Problema\_ocular(id)

create view Producte as  
select id,idProblema,nom,marca,preu  
from Ulleres  
union  
select id,nom,marca,preu  
from Lents\_contacte;

Ciutat(#codi,nom)

Optica(#id,codi\_ciutat,nom,telefon)  
foreign key (codi\_ciutat) references Ciutat(codi)  
constraint notNullCiutat not null(codi\_ciutat)

Publicitat(#id,idOptica,descripcio,autor,campanya,data\_creacio)  
foreign key (idOptica) references Optica(id)  
constraint notNullCiutat not null(idOptica)

Compra(#id,idProducte,idOptica,idClient,data)  
foreign key (idProducte) references Producte(id)  
foreign key (idOptica) references Optica(id)  
foreign key (idClien) references Client(id)  
constraint notNullProducte not null(idProducte)  
constraint notNullOptica not null(idOptica)  
constraint notNullClient not null(idClien)

Client(#id,DNI,nom,cognoms,telefon)

Consulta(#idOptica,#idClient,data)  
foreign key (idOptica) references Optica(id)  
foreign key (idClient) references Client(id)

```

Enviament(#idPublicitat,#idOptica,#idClient,data)
foreign key (idOptica) references Optica(id)
foreign key (idClient) references Client(id)
foreign key (idPublicitat) references Publicitat(id)
create or replace function uniqueOptica() returns trigger as $$
declare
    optiques int;
begin
    select into optiques count(idOptica) from Enviament where idOptica!=new.idOptica and
        idClient=new.idClient;
    if optiques > 0 then
        raise exception 'aquest client ja te una optica que li envia publicitat';
        return old;
    end if;
    return new;
end;
$$ language 'plpgsql';

create trigger myTrigger after insertion on Enviament
for each row procedure uniqueOptica();

```