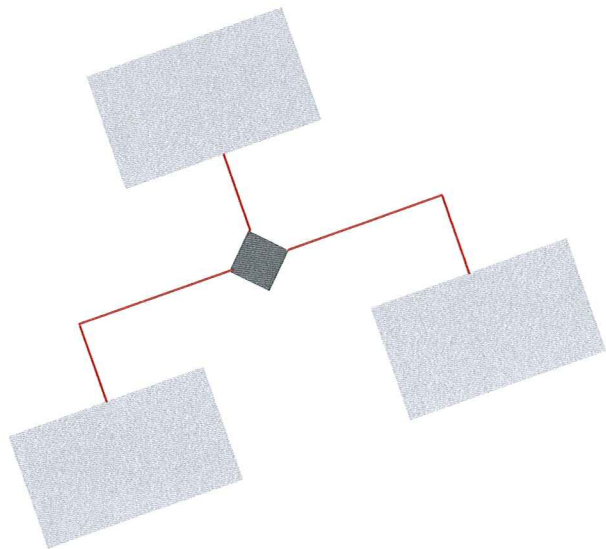


Capítol V

Cas d'Estudi



Índex del capítol

1. IMPLEMENTACIÓ DEL CAS D'ESTUDI EN ORACLE9I.....	104
2. APLICACIÓ DEL PROCÉS D'ENGINYERIA INVERSA.....	108
2.1. APLICACIÓ REGLA1	109
2.2. APLICACIÓ REGLA2.1 I REGLA3	110
2.3. APLICACIÓ REGLA2.2 I REGLA3	110
2.4. APLICACIÓ REGLA2.4.....	111
2.5. APLICACIÓ REGLA2.5.....	111
2.6. APLICACIÓ REGLA2.7.....	111
2.7. APLICACIÓ REGLA2.9.....	112
2.8. APLICACIÓ REGLA2.10.....	112
2.9. APLICACIÓ REGLA4.....	113
2.10. APLICACIÓ REGLA6.....	114
2.11. APLICACIÓ REGLA7.....	114
2.12. APLICACIÓ REGLA8.....	114
2.13. APLICACIÓ REGLA9.....	114
2.14. APLICACIÓ REGLA10.....	115
2.15. APLICACIÓ REGLA11.....	116
3. ESQUEMA CONCEPTUAL OBTINGUT.....	117

Una premissa evident d'aquest projecte és que, abans d'aplicar el procés d'enginyeria inversa sobre una Base de Dades, és imprescindible disposar de la Base de Dades en qüestió.

En un entorn real en el qual es volgués aplicar la funcionalitat d'aquest projecte, la Base de Dades ja estaria definida prèviament. En l'àmbit d'aquest projecte, però, la construcció de la Base de Dades constitueix un pas previ al procés que és objecte del projecte.

En aquest capítol es presentarà un cas d'estudi complet, descrivint amb detall totes les fases del procés:

1. Implementació del cas d'estudi en Oracle. Com hem dit, és un procés previ necessari en el context d'aquest projecte.
2. Aplicació del procés d'enginyeria inversa sobre el cas d'estudi anterior. Es veurà quines regles s'apliquen i quina transformació produeixen.
3. Esquema Conceptual obtingut. Es comentarà l'Esquema Conceptual obtingut com a resultat del procés anterior.

1. Implementació del cas d'estudi en Oracle9i

Aquesta pre-fase consisteix en crear la Base de Dades sobre la qual s'aplicarà el procés d'enginyeria inversa. Cal destacar que no és necessari que la Base de Dades contingui dades, ja que per determinar l'Esquema Conceptual associat a una Base de Dades n'hi ha prou amb disposar d'informació sobre l'estructura de les dades, i no de les dades en sí.

La Base de Dades es crearà invocant sentències SQL contra un Gestor de la Base de Dades de tipus Objecte-Relacional (*Oracle*). L'Esquema de la Base de Dades creada consistirà en un conjunt de tipus, taules, operacions, etc. sense cap instanciació de dades.

A continuació es mostren les sentències SQL necessàries per crear la Base de Dades associada al cas d'estudi, que s'ha anomenat "Centre Excursionista" i s'ha dissenyat per a l'ocasió. Cal destacar que, tot i que s'han provat altres casos d'estudi dissenyats pel propi *Oracle*, en aquest capítol es presenta el cas d'estudi del "Centre Excursionista" perquè aquest, a diferència dels casos d'estudi d'*Oracle*, inclou molts dels elements més representatius d'un Esquema Conceptual.

```
-- Esquema de Base de Dades d'exemple: CENTRE EXCURSIONISTA
```

```
-----  
----- TIPUS DE DADES -----  
-----
```

```
-- Tipus Direccio
CREATE TYPE Direccio AS OBJECT(
  carrer      VARCHAR2(40),
  num         INTEGER,
  pis         VARCHAR(10),
  poblacio   VARCHAR(30),
  codi_postal INTEGER
);
/

-- Tipus Temps
CREATE TYPE Temps AS OBJECT(
  hores      INTEGER,
  minuts     INTEGER
);
/

-- Tipus Punt_Geografic
CREATE TYPE Punt_Geografic AS OBJECT(
  coordX     NUMBER(2),
  coordY     NUMBER(2),
  MEMBER FUNCTION distancia_XY RETURN NUMBER
);
/

-- Tipus Persona
CREATE TYPE Persona AS OBJECT(
  id         INTEGER,
  dni        VARCHAR2(9),
  nom        VARCHAR2(30),
  edat       INTEGER,
  dir        DIRECCIO,
  mail       VARCHAR2(30),
  MEMBER PROCEDURE set_Mail (nouMail VARCHAR2),
  MEMBER PROCEDURE incr_Edat
) NOT INSTANTIABLE NOT FINAL;
/

-- Array de 2 telefons
CREATE TYPE llista_telefons IS VARRAY(2) OF INTEGER;
/

-- Declaració tipus Control
CREATE TYPE Control;
/
```

```
-- Tipus Organitzador
CREATE TYPE Organitzador UNDER PERSONA(
  telefon llista_telefons,
  supervisa REF Control
);
/

-- Tipus referència a Organitzador
CREATE TYPE Organitzador_REF_VARRAY AS OBJECT(
  ref_organitzador REF Organitzador
);
/

-- Array de 6 Organitzadors
CREATE TYPE Organitzador_REFS_VARRAY IS VARRAY(6) OF
Organitzador_REF_VARRAY;
/

-- Declaració tipus Etapa
CREATE TYPE ETAPA;
/

-- Tipus Control
CREATE OR REPLACE TYPE Control AS OBJECT(
  id          INTEGER,
  descripcio  VARCHAR2(100),
  id_excursio INTEGER,
  supervisat_per Organitzador_REFS_VARRAY,
  pertany_a   REF Etapa,
  situat_a    REF Punt_Geografic,
  MEMBER PROCEDURE modificar_dades (id INTEGER, descr VARCHAR)
);
/

-- Declaració del tipus Excursio
CREATE TYPE Excursio;
/

-- Tipus Etapa
CREATE OR REPLACE TYPE Etapa AS OBJECT(
  id          INTEGER,
  km_etapa    NUMBER(2),
  comensa_a   REF Punt_Geografic,
  acaba_a     REF Punt_Geografic,
  forma_part_de REF Excursio
);
```

```

/

-- Taula niuada d'Etapes
CREATE TYPE Etapa_NT AS TABLE OF ETAPA;
/

-- Tipus Excursio
CREATE OR REPLACE TYPE Excursio AS OBJECT(
  id          INTEGER,
  dia         DATE,
  km_excursio NUMBER(2),
  consta_de   Etapa_NT,
  MEMBER FUNCTION get_num_etapes RETURN INTEGER
);
/

-- Tipus Excursionista
CREATE TYPE Excursionista UNDER PERSONA(
  anys_experiencia INTEGER,
  num_excursions   INTEGER
);
/

-- Tipus Participacio
CREATE TYPE Participacio AS OBJECT(
  id          INTEGER,
  marca       Temps,
  ref_excursio REF Excursio,
  ref_excursionista REF Excursionista
);
/

-----
----- TAULES OBJECTE -----
-----

-- Taula Punt_Geografic
CREATE TABLE Punt_Geografic_OT OF Punt_Geografic(
  coordX NOT NULL,
  coordY NOT NULL
);

-- Taula Persona
CREATE TABLE Persona_OT OF Persona(
  id PRIMARY KEY NOT NULL,

```

```
nom UNIQUE,  
CONSTRAINT check_edat CHECK (edat > 10)  
) OBJECT ID PRIMARY KEY;  
  
-- Taula Organitzador  
CREATE TABLE Organitzador_OT OF Organitzador;  
  
-- Taula Control  
CREATE TABLE Control_OT OF Control(  
  id          PRIMARY KEY,  
  descripcio NOT NULL  
) OBJECT ID PRIMARY KEY;  
  
-- Taula Etapa  
CREATE TABLE Etapa_OT OF Etapa(  
  ID PRIMARY KEY  
) OBJECT ID PRIMARY KEY;  
  
-- Taula Excursio  
CREATE TABLE Excursio_OT OF Excursio(  
  ID PRIMARY KEY  
  CONSTRAINT check_id CHECK (id > 0)  
)  
NESTED TABLE consta_de STORE AS nt_etapes;  
  
-- Taula Excursionista  
CREATE TABLE Excursionista_OT OF Excursionista;  
  
-- Taula Participacio  
CREATE TABLE Participacio_OT OF Participacio(  
  id          PRIMARY KEY,  
  marca NOT NULL  
) OBJECT ID PRIMARY KEY;
```

2. Aplicació del procés d'enginyeria inversa

El procés de transformació d'un Esquema de Base de Dades a un Esquema Conceptual no és una tasca senzilla. Aquest procés consisteix en aplicar les regles de transformació, definides al capítol IV, sobre cadascun dels elements de l'Esquema de la Base de Dades.

Per transmetre una estimació de la complexitat d'aquest procés per al cas d'estudi que s'acaba d'implementar, a continuació es mostren algunes dades numèriques:

Nombre de vegades que s'apliquen les regles	Per convertir l'Esquema de la BD del cas d'estudi en un Esquema Conceptual, s'han aplicat les regles un total de 113 vegades.
Nombre de conflictes	Durant el procés anterior, s'han produït 6 conflictes que han requerit la intervenció directa de l'usuari.
Temps	El procés de transformació, sense comptar la latència entre que es produeix el conflicte i l'usuari contesta a la pregunta, ha trigat menys d'un minut.

Per tal de generar l'Esquema Conceptual, les regles de transformació s'apliquen en ordre, començant per la *Regla1* fins la *Regla11*. Amb l'objectiu d'il·lustrar aquest procés pas per pas, en els següents apartats es descriu de quina manera intervenen les regles en la creació de fragments de l'Esquema Conceptual i, finalment, es mostra l'Esquema Conceptual obtingut.

2.1. Aplicació Regla1

Aquesta regla consulta els UDTs que conté la Base de Dades i crea Classes o Tipus de Dades de l'Esquema Conceptual, segons convingui.

L'aplicació d'aquesta regla sobre el cas d'estudi provoca la creació del següent Esquema Conceptual:

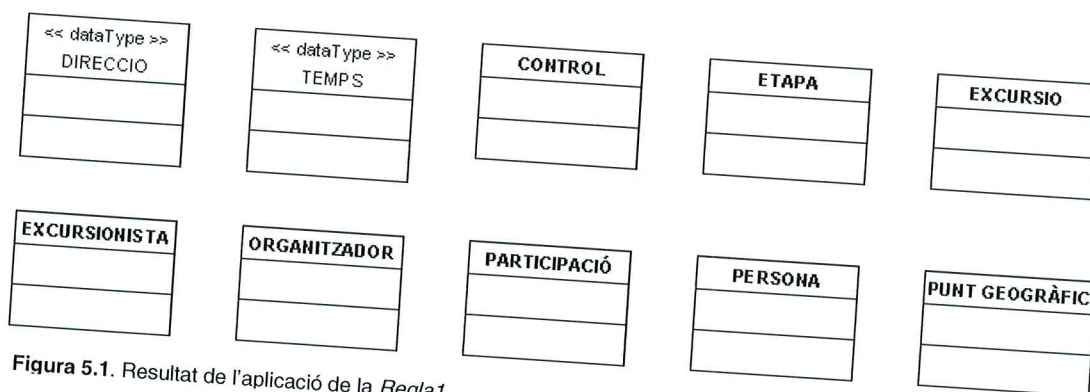


Figura 5.1. Resultat de l'aplicació de la *Regla1*.

2.2. Aplicació Regla2.1 i Regla3

La Regla2.1 transforma els atributs de tipus simple de la Base de Dades en atributs simples dels tipus de dades o classes de l'Esquema Conceptual. La Regla3, per la seva part, determina la correspondència entre els tipus bàsics d'Oracle i els tipus d'UML.

L'aplicació d'aquesta regla sobre una de les classes del cas d'estudi provoca la creació del següent Esquema Conceptual:

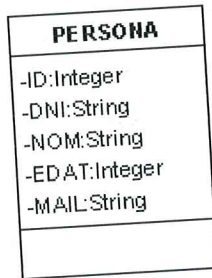


Figura 5.2. Resultat de l'aplicació de la Regla2.1 i Regla3.

Per a la resta de classes i tipus de dades, la transformació dels atributs de tipus simple és equivalent.

2.3. Aplicació Regla2.2 i Regla3

La Regla2.2 transforma els atributs de tipus array de tipus simple de la Base de Dades en atributs simples amb multiplicitat fixa dels tipus de dades o classes de l'Esquema Conceptual. La Regla3, igual que en cas anterior, determina la correspondència entre els tipus bàsics d'Oracle i els tipus d'UML.

L'aplicació d'aquesta regla sobre la classe *Organitzador* del cas d'estudi provoca la creació de l'atribut *telefon*, de tipus *Integer*, amb multiplicitat màxima 2, tal com es veu a continuació:

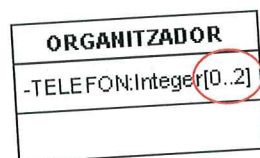


Figura 5.3. Resultat de l'aplicació de la Regla2.2 i Regla3.

2.4. Aplicació Regla2.4

La Regla2.4 transforma els atributs de tipus *REF* de la Base de Dades en associacions de multiplicitat màxima 1 entre els elements de l'Esquema Conceptual.

L'aplicació d'aquesta regla sobre la classe *Control* del cas d'estudi provoca la creació de l'associació *pertany_a* entre *Control* i *Etapa*, tal com es veu a continuació:

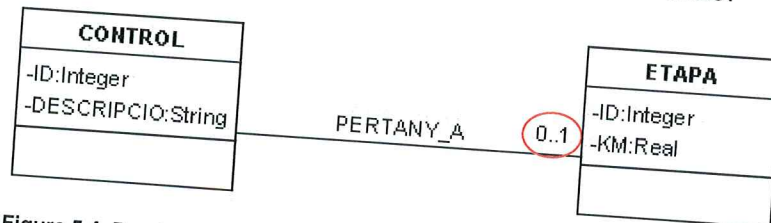


Figura 5.4. Resultat de l'aplicació de la Regla2.4.

2.5. Aplicació Regla2.5

La Regla2.4 transforma els atributs de tipus *array* de referències de la Base de Dades en associacions de multiplicitat màxima limitada entre els elements de l'Esquema Conceptual.

L'aplicació d'aquesta regla sobre la classe *Control* del cas d'estudi provoca la creació de l'associació *supervisat_per* entre *Control* i *Organitzador*, tal com es veu a continuació:

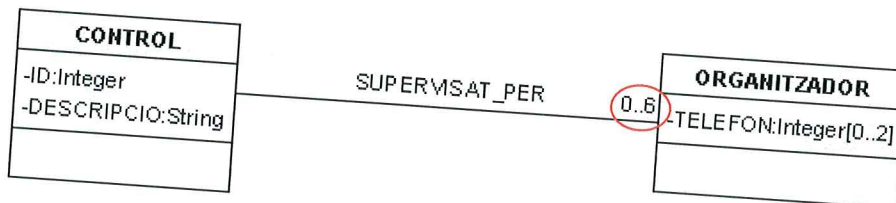


Figura 5.5. Resultat de l'aplicació de la Regla2.5.

2.6. Aplicació Regla2.7

La Regla2.7 transforma els atributs de tipus definit per l'usuari de la Base de Dades en atributs de tipus definits per l'usuari o bé composicions de multiplicitat màxima 1 de l'Esquema Conceptual.

L'aplicació d'aquesta regla sobre l'atribut *DIR* classe *Persona* del cas d'estudi provoca la creació d'un atribut (*DIR*) que és d'un dels tipus definit per l'usuari (*DIRECCIO*), tal com es veu a continuació:



Figura 5.6. Resultat de l'aplicació de la Regla2.7.

2.7. Aplicació Regla2.9

La Regla2.9 transforma els atributs de tipus *taula niuada* d'UDT definits a la Base de Dades en composicions de multiplicitat màxima il·limitada de l'Esquema Conceptual.

L'aplicació d'aquesta regla sobre l'atribut *consta_de* de la classe *Excursio* del cas d'estudi provoca la creació d'una composició de multiplicitat màxima il·limitada entre les classes *Excursio* i *Etapa*, tal com es veu a continuació:

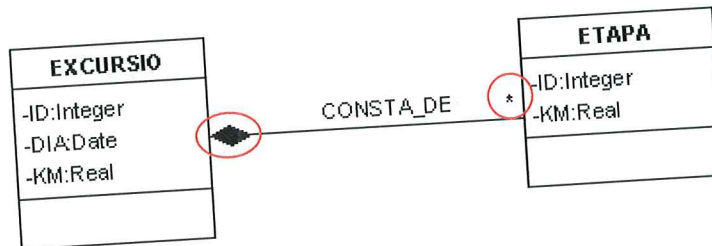


Figura 5.7. Resultat de l'aplicació de la Regla2.9.

2.8. Aplicació Regla2.10

L'aplicació de la Regla2.4 sobre les classes *Excursionista*, *Excursio* i *Participació* genera un Esquema Conceptual amb dues associacions com les següents:

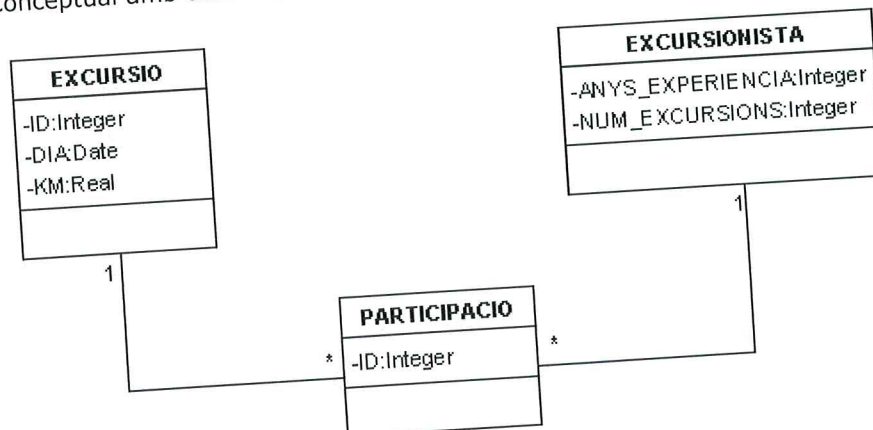


Figura 5.8. Patró d'una possible classe associativa.

Quan, posteriorment, s'aplica la *Regla2.10*, es determina que el patró anterior és un candidat a format una classe associativa, ja que compleix les condicions del patró que determina una classe associativa.

L'aplicació no pot decidir per ella mateixa si es tracta d'una classe associativa o no, per això és imprescindible la intervenció directa de l'usuari. Si aquest respon afirmativament, s'esborraran les dues relacions anteriors de l'Esquema Conceptual i es crearà una nova relació com la següent:

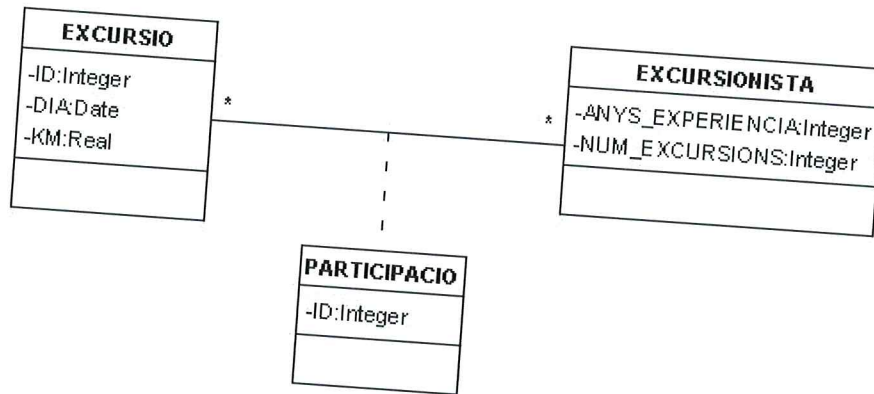


Figura 5.9. Resultat de l'aplicació de la *Regla2.10*.

2.9. Aplicació Regla4

La *Regla4* transforma els mètodes i procediments dels UDTs de la Base de Dades en mètodes i procediments de les classes de l'Esquema Conceptual.

L'aplicació d'aquesta regla sobre els mètodes de l'UDT *Persona* de la Base de Dades provoca la creació de les dues operacions que es mostren a continuació:

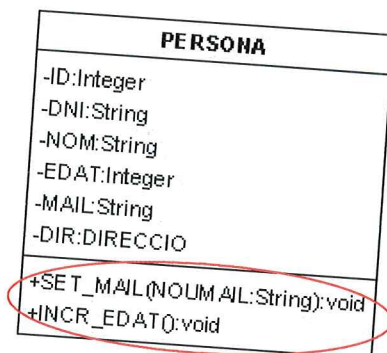


Figura 5.10. Resultat de l'aplicació de la *Regla4*.

2.10. Aplicació Regla6

La *Regla6* converteix les restriccions CHECK de la Base de Dades en Restriccions d'Integritat de l'Esquema Conceptual.

L'aplicació d'aquesta regla sobre la taula objecte associada a l'UDT *Persona* de la Base de Dades provoca la creació d'una Restricció d'Integritat com la següent:

```
context Persona inv: self.edat > 10
```

2.11. Aplicació Regla7

La *Regla7* converteix les restriccions de tipus UNIQUE de la Base de Dades en Restriccions d'Integritat de l'Esquema Conceptual.

L'aplicació d'aquesta regla sobre la taula objecte associada a l'UDT *Persona* provoca la creació d'una Restricció d'Integritat com la següent:

```
context Persona inv: Persona.allInstances()->isUnique(dni)
```

2.12. Aplicació Regla8

La *Regla8* converteix les restriccions de tipus PRIMARY KEY de la Base de Dades en Restriccions de clau primària de l'Esquema Conceptual.

L'aplicació d'aquesta regla sobre la taula objecte associada a l'UDT *Persona* provoca la creació d'una Restricció d'Integritat com la següent:

```
context Persona inv: Persona.allInstances()->isUnique(id)
```

2.13. Aplicació Regla9

La *Regla9* converteix la característica UNDER de la Base de Dades en una generalització de l'Esquema Conceptual.

L'aplicació d'aquesta regla sobre les taules objecte associades a l'UDT *Excursionista* i *Organitzador* de la Base de Dades genera la següent generalització a l'Esquema Conceptual:

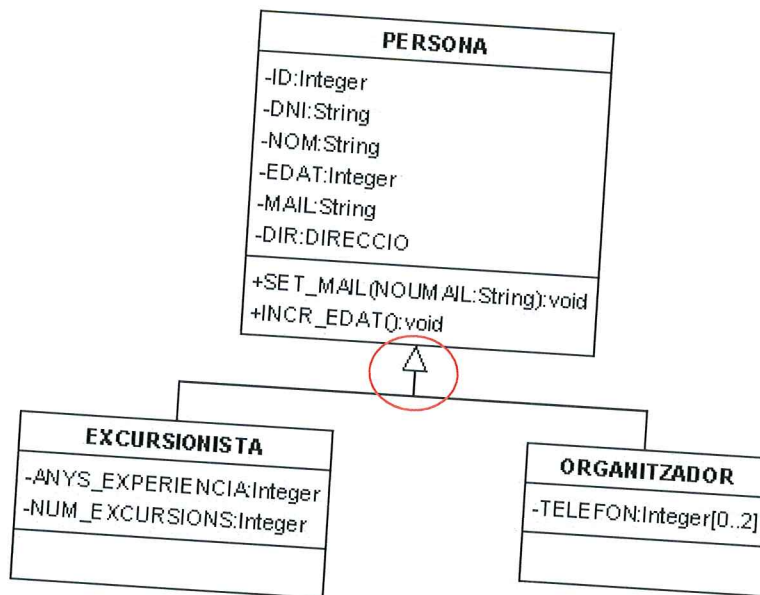


Figura 5.11. Resultat de l'aplicació de la Regla9.

2.14. Aplicació Regla10

La Regla10 converteix la característica NOT INSTANTIABLE de la Base de Dades en la característica *abstracte* d'una classe de l'Esquema Conceptual.

L'aplicació d'aquesta regla sobre l'UDT *Persona* produeix la següent modificació sobre l'Esquema Conceptual:

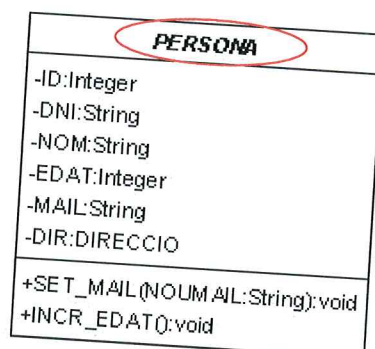


Figura 5.12. Resultat de l'aplicació de la Regla10.

2.15. Aplicació Regla11

La *Regla11* transforma la conjunció de les característiques UNDER i NOT INSTANTIABLE de la Base de Dades en la característica *complete* d'un conjunt de generalitzacions de l'Esquema Conceptual.

L'aplicació d'aquesta regla sobre l'Esquema Conceptual de la figura 5.11 produeix la següent modificació:

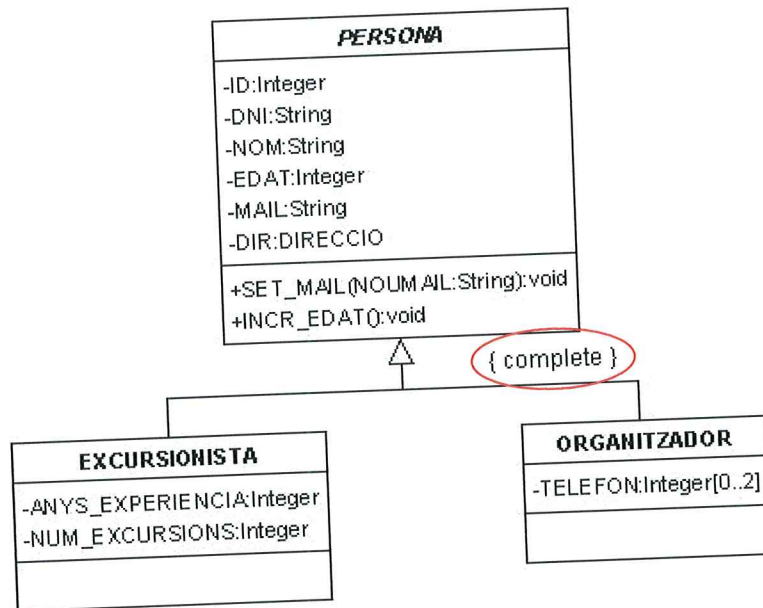


Figura 5.13. Resultat de l'aplicació de la *Regla11*.

3. Esquema Conceptual obtingut

A continuació es mostra l'Esquema Conceptual sencer obtingut com a resultat del procés anterior.

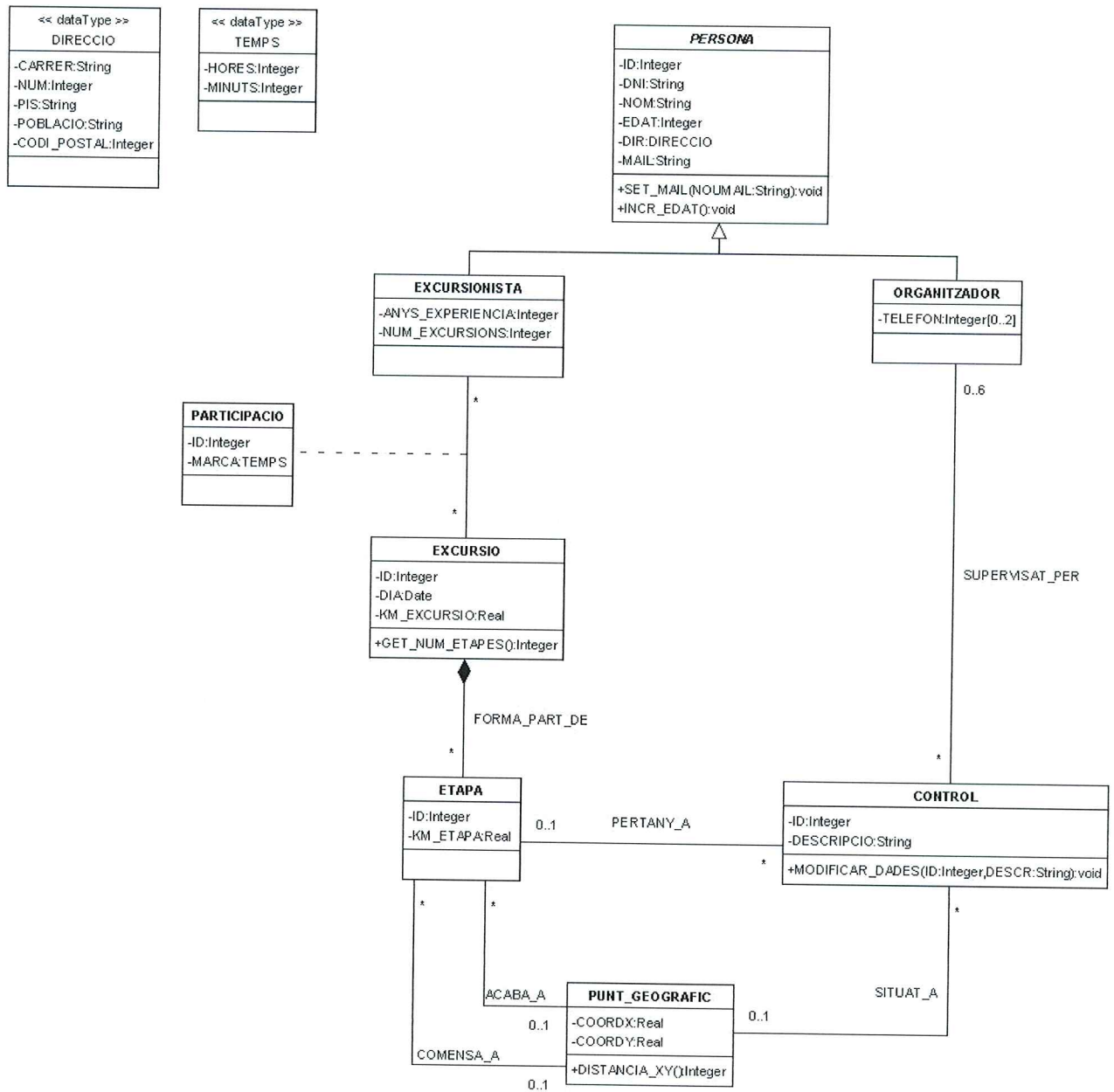


Figura 5.15. Esquema Conceptual complet.

Restriccions d'Integritat associades a l'Esquema:

```
Context PARTICIPACIO inv: PARTICIPACIO.allInstances->isUnique(ID)
Context EXCURSIO inv: EXCURSIO.allInstances->isUnique(ID)
Context EXCURSIO inv: self.id > 0
Context ETAPA inv: ETAPA.allInstances->isUnique(ID)
Context CONTROL inv: CONTROL.allInstances->isUnique(ID)
Context CONTROL inv: self.DESCRIPCIO->size() <= 100
Context PERSONA inv: PERSONA.allInstances->isUnique(ID)
Context PERSONA inv: self.edat > 10
Context PERSONA inv: PERSONA.allInstances->isUnique(NOM)
Context PERSONA inv: self.MAIL->size() <= 30
Context PERSONA inv: self.NOM->size() <= 30
Context PERSONA inv: self.DNI->size() <= 9
Context DIRECCIO inv: self.POBLACIO->size() <= 30
Context DIRECCIO inv: self.PIS->size() <= 10
Context DIRECCIO inv: self.CARRER->size() <= 40
```

Observem que l'Esquema Conceptual obtingut, que tracta sobre l'organització d'excursions per part d'un Centre Excursionista, conté els principals elements de tots els Esquemes Conceptuals que s'han tractat en aquest projecte: Classes, Tipus de Dades, Atributs, Associacions de diverses multiplicitats, Composicions, Classes Associatives, Operacions, Atributs derivats, Generalitzacions, Restriccions d'Integritat, etc.