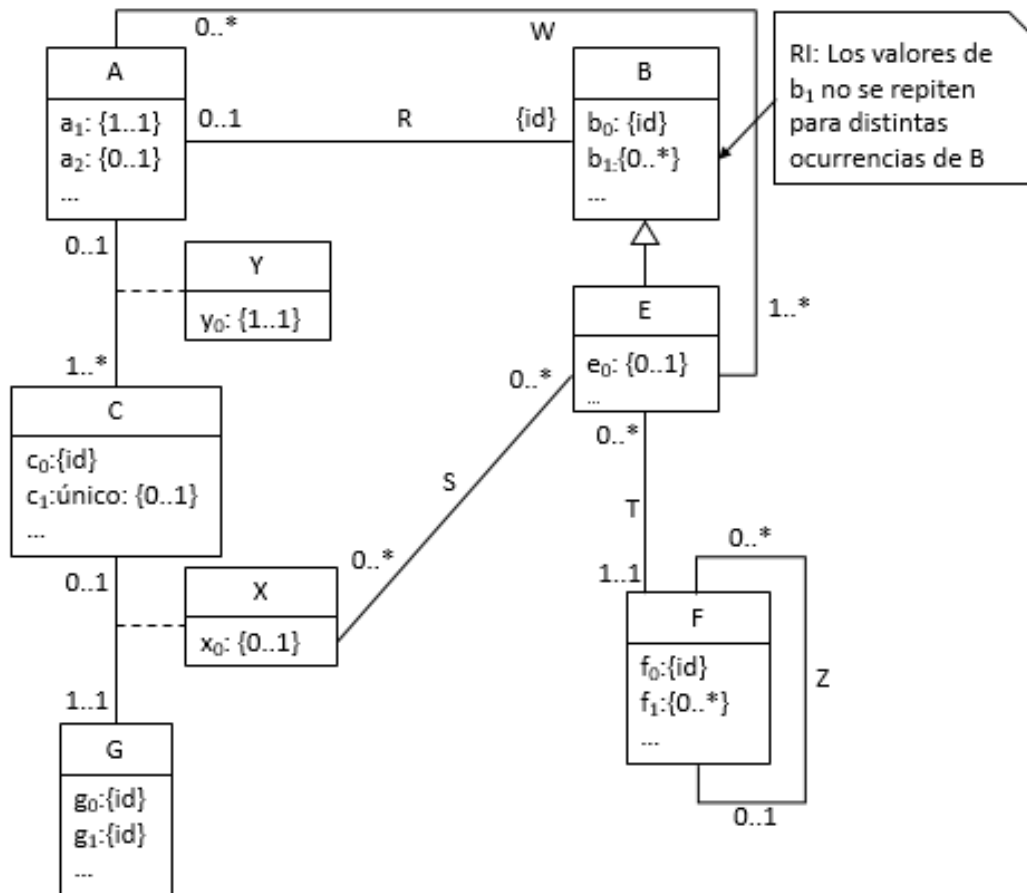


1. Realice el diseño lógico del siguiente diagrama de clases en UML para obtener un conjunto equivalente de relaciones del modelo relacional. Las restricciones que no pueda expresar en el esquema relacional, escríbalas en lenguaje natural. **En el diagrama no se han especificado los tipos de datos, así que no es necesario que los incluya en el esquema lógico. (1'5 puntos).**



2. Sea el siguiente esquema de relación:

R(A: tipo_A, B: tipo_B, C: tipo_C, D: tipo_D, E: tipo_E)

CP: {A, B}

VNN: {C, D, E}

tipo_A: char(10);

tipo_B: entero;

tipo_E: registro de {F: char(10), G: char(10)}

tipo_C: conjunto de enteros;

tipo_D: char(10);

A partir de las dependencias que aparecen a continuación, transforme la relación a un conjunto de relaciones en tercera forma normal **(0'5 puntos)**.

{B} → {D} {F} → {G}

3. Diseñe un diagrama de clases en UML para el sistema de información que se describe a continuación. Las restricciones que no se puedan expresar gráficamente, escribálas en lenguaje natural. **(1'5 puntos)**

La sección de Árboles Monumentales de la ciudad desea diseñar una base de datos para gestionar la poda y el mantenimiento de dichos árboles, a continuación, se describe el sistema de información.

Los árboles monumentales están codificados de forma única y a algunos de ellos se les conoce por un nombre que también es único; es imprescindible conocer su antigüedad (aunque sea aproximada), su ubicación y la especie a la que pertenecen. Las especies también están codificadas y se debe conocer su nombre científico, que es único, y el nombre común si es que lo tienen; será necesario disponer de información respecto a cómo se deben cuidar los árboles de la especie.

La poda de los árboles se planifica en sesiones de un día en las que se incluyen un conjunto de árboles (al menos uno). De cada poda se es imprescindible saber el código, que es único, el día en que se realizará y la cuadrilla de jardineros que va a realizar el trabajo.

Las cuadrillas están numeradas de forma única, tienen un nivel y constan de entre 1 y 5 jardineros de los cuales uno es el capataz (un jardinero pertenece como mucho a una cuadrilla). De cada jardinero se conoce siempre su número de trabajador, que es único, su nombre, su número de cuenta corriente, su dirección y su teléfono. Las cuadrillas no pueden estar asignadas a dos podas el mismo día.

El material de que dispone la sección está codificado, tiene un nombre (cestas elevadoras, sierras mecánicas, camiones para recoger desechos, ...) y es importante saber cómo funcionan y cuántas unidades de cada material hay. Una poda puede reservar material siendo necesario saber qué cantidad reserva de cada material. Para un día concreto, de un material no se podrán reservar más unidades de las que hay de ese material.

Además de las podas programadas, los árboles pueden sufrir incidencias (caídas de ramas, incendios, ...) que requieran de una intervención no programada. De cada incidencia, que se identifica por un número que es único para cada día, se quiere conocer el árbol o árboles implicados y una descripción del problema. Si es necesario, alguno de los jardineros que no forman parte de ninguna cuadrilla realizará un informe de alguno de estos árboles. En el informe debe quedar claro el árbol investigado, la incidencia que se estudia, la fecha y el resultado del análisis visual. Es posible que de este análisis se deriven posibles compensaciones económicas para algún ciudadano que se haya visto perjudicado por la incidencia. Esta información se debe almacenar, además de cada compensación se quiere saber su código, que la identifica, la cantidad asignada y el ciudadano que la va a recibir (DNI, nombre y email).

4. Dado el siguiente esquema de una base de datos

```
CREATE TABLE Monitor (
DNI CHAR(10) CONSTRAINT pk_monitor PRIMARY KEY DEFERRABLE,
nombre VARCHAR(50) NOT NULL,
telefono CHAR(10) NOT NULL,
num INTEGER DEFAULT 0 NOT NULL)

CREATE TABLE Viaje (
id char(5) CONSTRAINT pk_viaje PRIMARY KEY DEFERRABLE,
DNI CHAR(10) CONSTRAINT fk_viaje_monitor REFERENCES monitor(DNI) DEFERRABLE,
fecha DATE NOT NULL)
```

Donde:

- La relación **Monitor** almacena los datos del monitor: *DNI*, *nombre*, *teléfono* y *num*, el número de viajes que tiene asignado el monitor.
- La relación **Viaje** almacena la información de un viaje: *id* del viaje, *DNI* del monitor asignado al viaje y *fecha* del viaje.

Implementar un disparador (*trigger*) en PL-SQL de Oracle para que el valor del atributo *num* sea coherente con el significado del atributo para la operación de **insertar** un nuevo **monitor**. **(0'5 puntos)**

SOLUCIONES

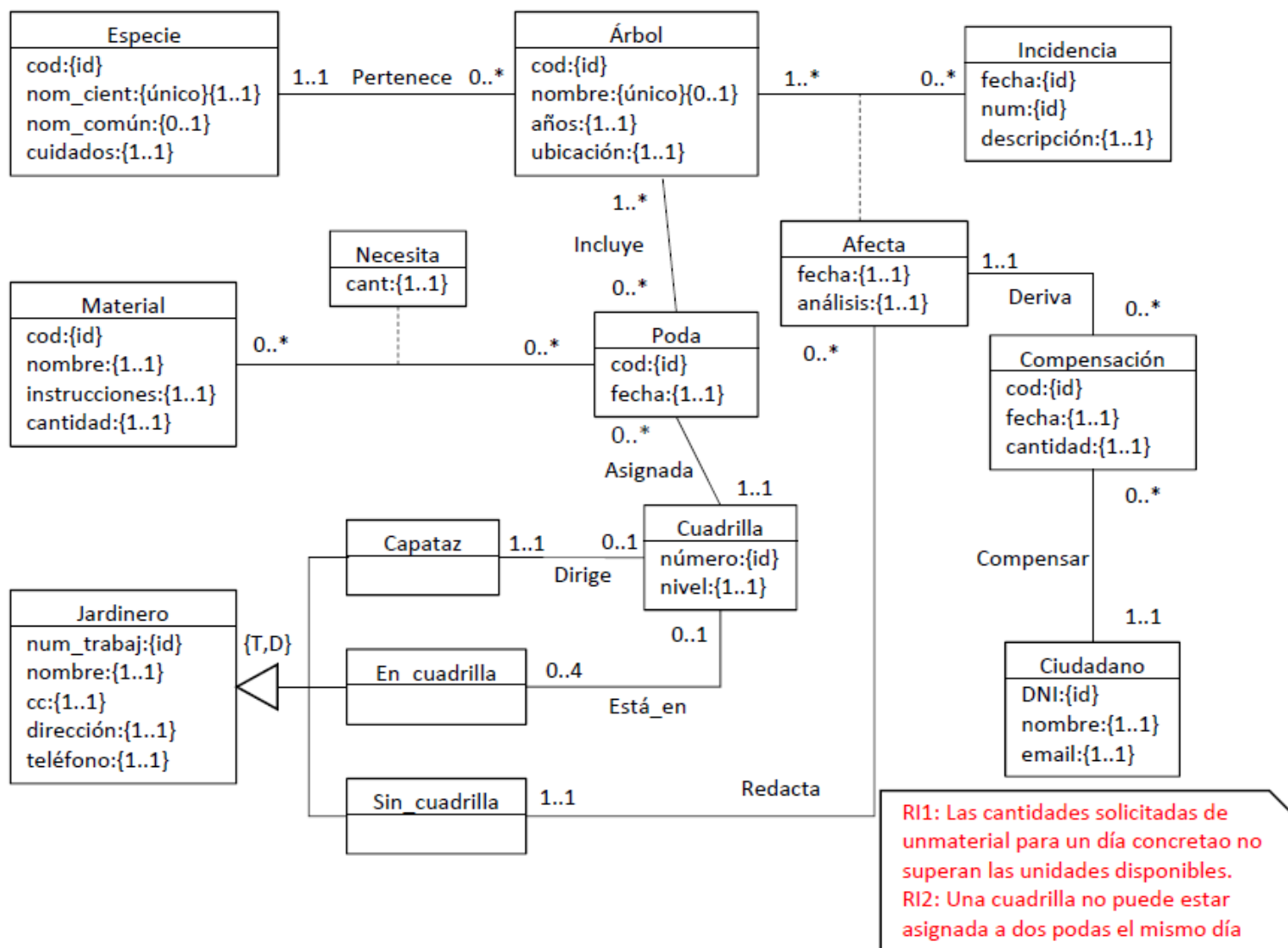
1.-

$B(b_0, \dots)$ $CP: \{b_0\}$ $B-B1(b_0, b_1)$ $CP: \{b_1\}$ $CAj: \{b_0\} \rightarrow B(b_0)$ $VNN: \{b_0\}$	$C(c_0, c_1, \dots, g_0, g_1, x_0)$ $CP: \{c_0\}$ $\text{Único: } \{c_1\}$ $CAj: \{g_0, g_1\} \rightarrow G(g_0, g_1)$ $VNN: \{g_0, g_1\}$ $\text{Único: } \{g_0, g_1\}$ $G(g_0, g_1, \dots)$ $CP: \{g_0, g_1\}$
$A(b_0, a_1, a_2, \dots)$ $CP: \{b_0\}$ $CAj: \{b_0\} \rightarrow B(b_0)$ $VNN: \{a_1\}$ $Y(b_0, c_0, y_0)$ $CP: \{c_0\}$ $VNN: \{b_0\}$ $CAj: \{b_0\} \rightarrow A(b_0)$ $CAj: \{c_0\} \rightarrow C(c_0)$ $VNN: \{y_0\}$ $RI_{\min_A_en\ Y}$: Todos los valores de b_0 que aparezcan en A deben aparecer en en el b_0 de Y.	$E(b_0, e_0, \dots, f_0)$ $CP: \{b_0\}$ $CAj: \{b_0\} \rightarrow B(b_0)$ $CAj: \{f_0\} \rightarrow F(f_0)$ $VNN: \{f_0\}$ $S(b_0, c_0)$ $CP: \{b_0, c_0\}$ $CAj: \{b_0\} \rightarrow E(b_0)$ $CAj: \{c_0\} \rightarrow C(c_0)$
$F(f_0, \dots, f_{0_Z})$ $CP: \{f_0\}$ $CAj: \{f_{0_Z}\} \rightarrow F(f_0)$ $F-F1(f_0, f_1)$ $CP: \{f_0, f_1\}$ $CAj: \{f_0\} \rightarrow F(f_0)$	$W(b_0, b_{0_A})$ $CP: \{b_0, b_{0_A}\}$ $CAj: \{b_0\} \rightarrow E(b_0)$ $CAj: \{b_{0_A}\} \rightarrow A(b_0)$ $RI_{\min_A_en\ W}$: Todos los valores de b_0 que aparezcan en A deben aparecer en en el b_0 de W.

2.-

$R(A: \text{tipo_A}, B: \text{tipo_B}, F: \text{tipo_F})$ $CP: \{A, B\}$ $VNN: \{F\}$ $CAj: \{B\} \rightarrow R-D(B)$ $CAj: \{F\} \rightarrow R-G(F)$	$R-D(B: \text{tipo_B}, D: \text{tipo_D})$ $CP: \{B\}$ $VNN: \{D\}$
$R-G(F: \text{tipo_F}, G: \text{tipo_G})$ $CP: \{F\}$ $VNN: \{G\}$	$R-C(A: \text{tipo_A}, B: \text{tipo_B}, C: \text{entero})$ $CP: \{A, B, C\}$ $CAj: \{A, B\} \rightarrow R(A, B)$
RI : <ul style="list-style-type: none"> • Todo par de valores (A, B) que aparecen en R deben aparecer en R-C • Todo valor de B que aparece en R-D debe aparecer en R • Todo valor de B que aparece en R-F debe aparecer en R 	

3.-



5. Hay dos soluciones correctas.

/*Esta puede ser BEFORE o AFTER/*

```
CREATE OR REPLACE TRIGGER T_ins_monitor
BEFORE INSERT ON monitor
FOR EACH ROW
WHEN new.num <> 0
BEGIN
    RAISE_APPLICATION_ERROR(...);
END;
```

/*Esta sólo puede ser BEFORE/*

```
CREATE OR REPLACE TRIGGER T_ins_monitor
BEFORE INSERT ON monitor
FOR EACH ROW
WHEN new.num <> 0
BEGIN
    :new.num := 0;
END;
```