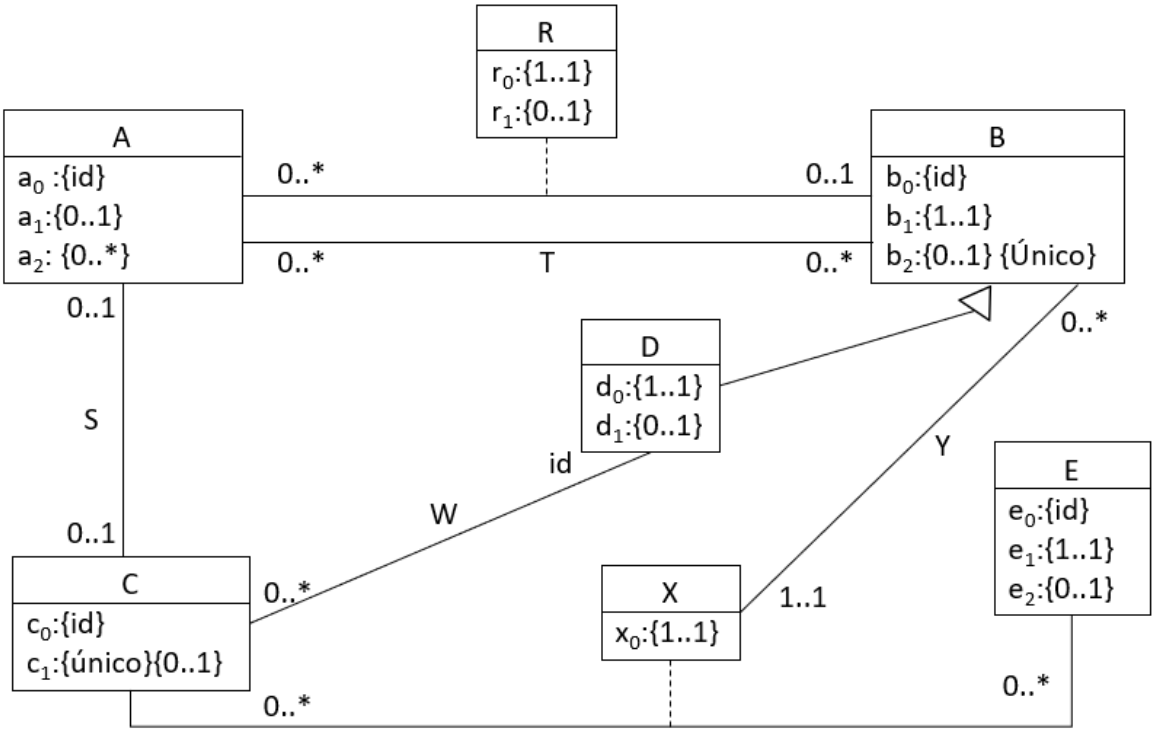
EXAMEN DE BASES DE DATOS Y SISTEMAS DE INFORMACIÓN 10/01/2020

PARCIAL 4: UNIDADES DIDÁCTICAS 4

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 **(1'5 puntos).**



2. **(0'5 puntos)** Sea el siguiente esquema de relación:
   R (A: char, B: char, C: conjunto de H:int, D: char, E: char, F: char, G: char)
   CP: {A, B} VNN: {C, D, E, F, G}

   A partir de las dependencias que aparecen a continuación, transforme la relación a un conjunto de relaciones en tercera forma normal.
   {H} → {A, B}
   {B}→ {D}
   {E} →{F}
   {F} →{G}

3. **(0'25 puntos)** Enumera las propiedades del correcto procesamiento de transacciones. Define dos de ellas (sólo dos).

4. 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, escríbalas en lenguaje natural. **(1'75 puntos)**

La universidad ha cambiado de empresa *vending* para el próximo año siendo la nueva adjudicataria la empresa *Benivending*. Dado el volumen de negocio, esta nueva empresa ha decidido diseñar una base de datos para la gestión específica de las máquinas ubicadas en el campus. A continuación, se describe el sistema de información.

Los productos ofertados en las máquinas son comprados por *Benivending* a proveedores externos de los que se quieren almacenar los siguientes datos: código interno (que los identifica), nombre, email, teléfono y nombre del contacto. Todos estos datos son obligatorios.

De cada producto que adquiere la empresa para ofertarlo en las máquinas se quiere saber: código interno en *Benivending* que sirve para identificarlo, nombre, precio al que se vende, y el tipo de producto. También hay que saber, para cada proveedor que suministra el producto, el código que éste tiene en el proveedor y el precio al que ese proveedor lo vende a *Benivending.* Todos estos datos son obligatorios.

De cada una de las máquinas se debe conocer el código interno que las identifica, el modelo, las dimensiones (alto, ancho y profundo), el edificio del campus donde está situada (indicando el piso), la fecha de adquisición y la fecha de la siguiente revisión. Hay que tener en cuenta que en el mismo piso de un edificio no se puede ubicar más de una máquina. Cada máquina tiene asignada una lista de productos que se ofertan en ella indicando, en cada caso, la cantidad que se suele poner de ese producto en la máquina. Esta lista es a modo orientativo, es decir no impide que, en un momento determinado, una máquina oferte algo que no está en la lista.

Los edificios de la universidad están codificados con un número y una letra, tienen un nombre, una ubicación en el campus y un teléfono de contacto.

Hay distintos modelos de máquinas, de cada modelo se quiere saber el código (que lo identifica), el nombre, las características y el empleado o empleados (al menos uno) especializados en las reparaciones de las máquinas de este modelo. De cada empleado se debe conocer su DNI, que lo identifica, el nombre y el número de teléfono móvil.

Para la gestión del *stock* de la empresa es importante almacenar información de todas las recargas que se les hacen a las máquinas (una máquina es recargada como mucho una vez al día). De cada recarga hay que saber: qué empleado la ha hecho, en qué máquina se ha hecho, en qué día, y la cantidad de cada producto que se haya recargado. También es importante poder incluir comentarios en la recarga para el caso en que haya alguna información relevante que almacenar.

Soluciones:

1.

A(a0, a1, …) CP:{a0} **A2**(a2, a0) CP:{a0, a2} CAj:{a0} → A	**B**(b0, b1, b2, …, c0, b0x, e0) CP:{b0} VNN{b1} Uni:{b2} CAj:{ c0, b0x, e0} → X(c0, b0, e0) VNN:{ c0, b0x, e0}
C(c0, c1, …, b0, a0) CP{c0, b0} Uni:{c1} CAj:{b0} → D CAj:{a0} → A (Se puede implementar en A) Uni:{a0}	**D**(b0, d0, d1, …) CP:{b0} CAj:{b0} → B VNN:{d0}
E(e0, e1, e2, …) CP:{e0} VNN{e1}	**R**(a0, b0, r0, r1) CP{a0} CAj:{a0} → A CAj:{b0} → B VNN:{b0} VNN:{r0}
T(a0, b0) CP{a0, b0} CAj:{a0} → A CAj:{b0} → B	**X**(c0, b0, e0, x0) CP:{c0, b0, e0} CAj:{c0, b0} → C CAj:{e0} → E VNN:{x0}

2.

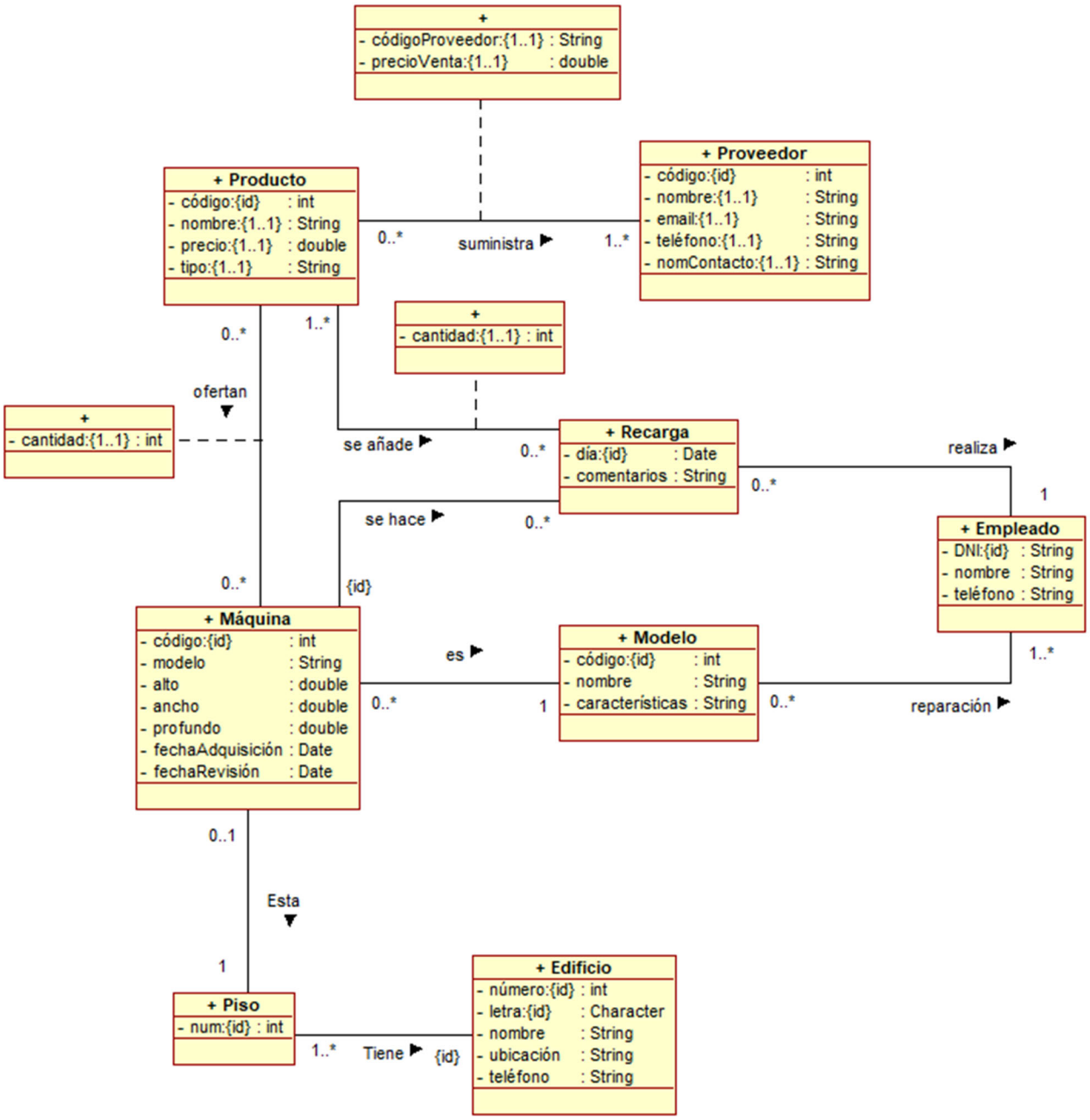
R(A: char, B: char, E: char) CP:{A,B} VNN:{E} CA:{E} →R2 CA:{B} →R3 Todo valor de (A,B) de R debe existir en R1	R1 (A: char, B: char, H: int) CP:{H} CA: {A,B} →R VNN:{A,B}
R2(E: char, F:char) CP:{E} CA:{F} →R4 VNN:{F}	R3(B: char, D:char) CP:{B} VNN:{D}
R4(F: char, G: char) CP:{F} VNN:{G}	

3.

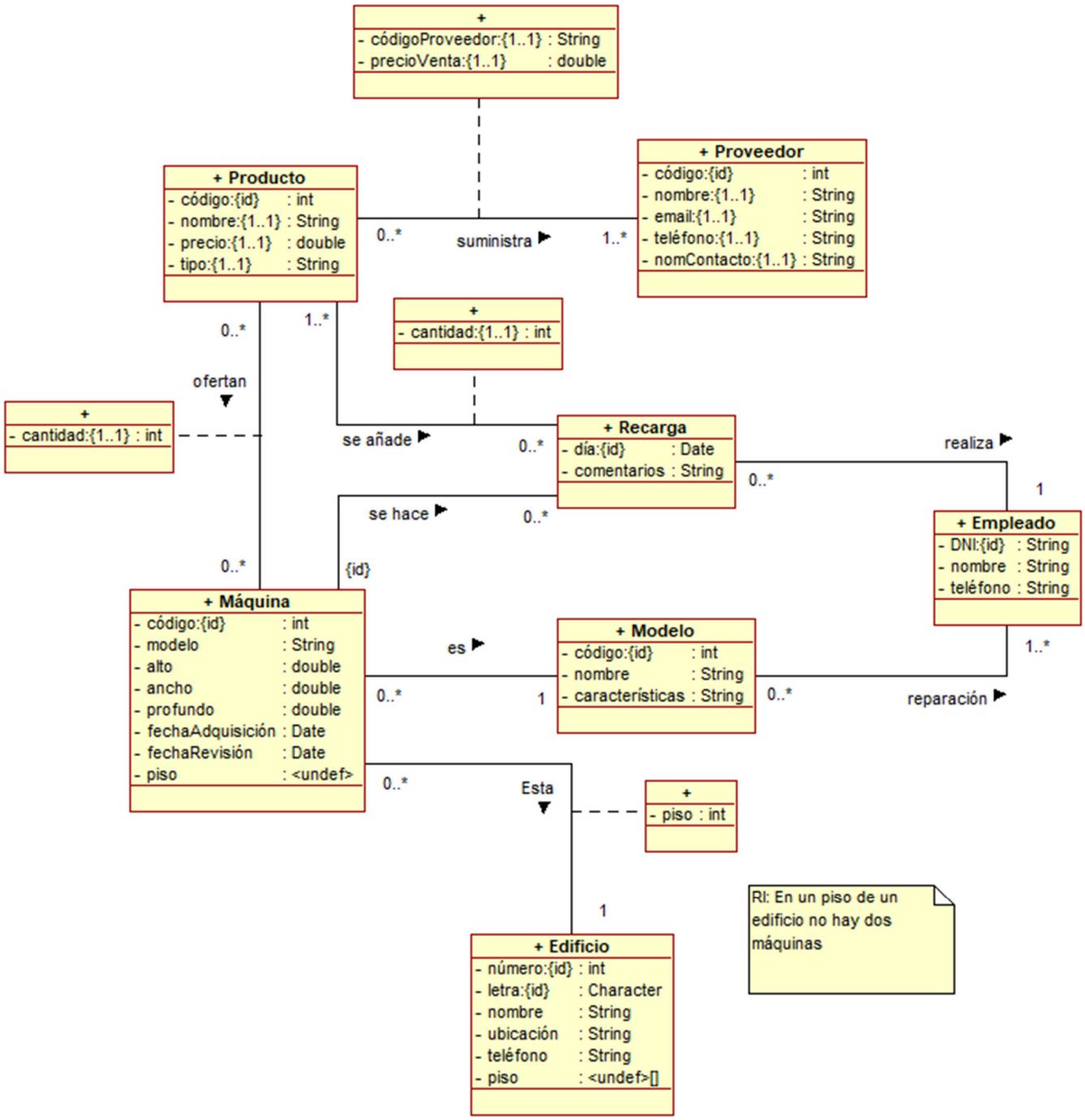
- **Atomicidad:** una transacción es una unidad atómica de ejecución (o se ejecutan todas sus operaciones o ninguna)
- **Consistencia:** la transacción debe dar lugar a un estado de la base de datos consistente (se cumplen todas las restricciones de integridad)
- **Aislamiento:** las modificaciones introducidas por una transacción no confirmada no son visibles al resto de transacciones
- **Persistencia**: la confirmación implica la grabación de los cambios introducidos en la base de datos, de forma que no se puedan perder por fallo del sistema o de otras transacciones

4.

Solución 1



Solución 2


+
- códigoProveedor:{1..1} : String
- precioVenta:{1..1} : double
+ Producto
- código:{id} : int
- nombre:{1..1} : String
- precio:{1..1} : double
- tipo:{1..1} : String
0..*
suministra
1..*
+ Proveedor
- código:{id} : int
- nombre:{1..1} : String
- email:{1..1} : String
- teléfono:{1..1} : String
- nomContacto:{1..1} : String
0..*
1..*
+
- cantidad:{1..1} : int
ofertan
+
- cantidad:{1..1} : int
se añade
0..*
+ Recarga
- día:{id} : Date
- comentarios : String
realiza
0..*
1
se hace
0..*
+ Empleado
- DNI:{id} : String
- nombre : String
- teléfono : String
0..*
{id}
+ Máquina
- código:{id} : int
- modelo : String
- alto : double
- ancho : double
- profundo : double
- fechaAdquisición : Date
- fechaRevisión : Date
- piso : <undef>
es
+ Modelo
- código:{id} : int
- nombre : String
- características : String
1..*
0..*
1
0..*
reparación
0..*
Esta
+
- piso : int
1
+ Edificio
- número:{id} : int
- letra:{id} : Character
- nombre : String
- ubicación : String
- teléfono : String
- piso : <undef>[]
RI: En un piso de un edificio no hay dos máquinas