

Aparcamientos de Madrid

Práctica ADBD 2019-2020

Grupo 20:

Hernando Brecht, Rebeca,
Herruzo Herrero, Juan,
Jáñez Lagüéns, Guillermo,
Llorente Pérez, Mario.

Índice

Índice	1
Diagrama Entidad Relación.	2
Aclaraciones.	2
Clase abono.	2
Clase Aparcamiento.	2
Renovaciones.	2
Clase conReserva.	2
Restricciones	2
Imagen.	4
Esquema Relacional.	5
Aclaraciones.	5
Herencia feedback.	5
Herencia aparcamiento.	5
Herencia titular.	5
Herencia Abono y Solicitud.	5
Herencia Vehículo y Vehículo abonado.	5
Imagen.	6
Consultas.	7

Diagrama Entidad Relación.

Aclaraciones.

Clase abono.

- El atributo *fechaCaducidad* refleja el último día que el abono estará en vigor.
- El atributo *fechaFinalizacion* guarda la fecha en la que el abono fué cancelado si esta es igual o anterior a la *fechaCaducidad*. Mientras este atributo sea null el abono estará en vigor.
- El atributo *cicloDePago* guarda la fecha del siguiente pago, si esta fecha es anterior a la fecha actual significa que el pago está atrasado.

Clase Aparcamiento.

- Un aparcamiento es mixto si es del tipo rotacional y residencial a la vez.

Renovaciones.

Decidimos no crear una clase renovación ya que estas se resuelven al instante, es decir, cuando se genera una solicitud de renovación si se acepta se amplía la *fechaCaducidad* del abono ya existente y si se rechaza se genera una nueva solicitud para un abono del mismo tipo que el anterior.

Clase conReserva.

Esta clase es una representación de que solo los abonos con reserva van asociados a una plaza, en el modelo relacional desaparece.

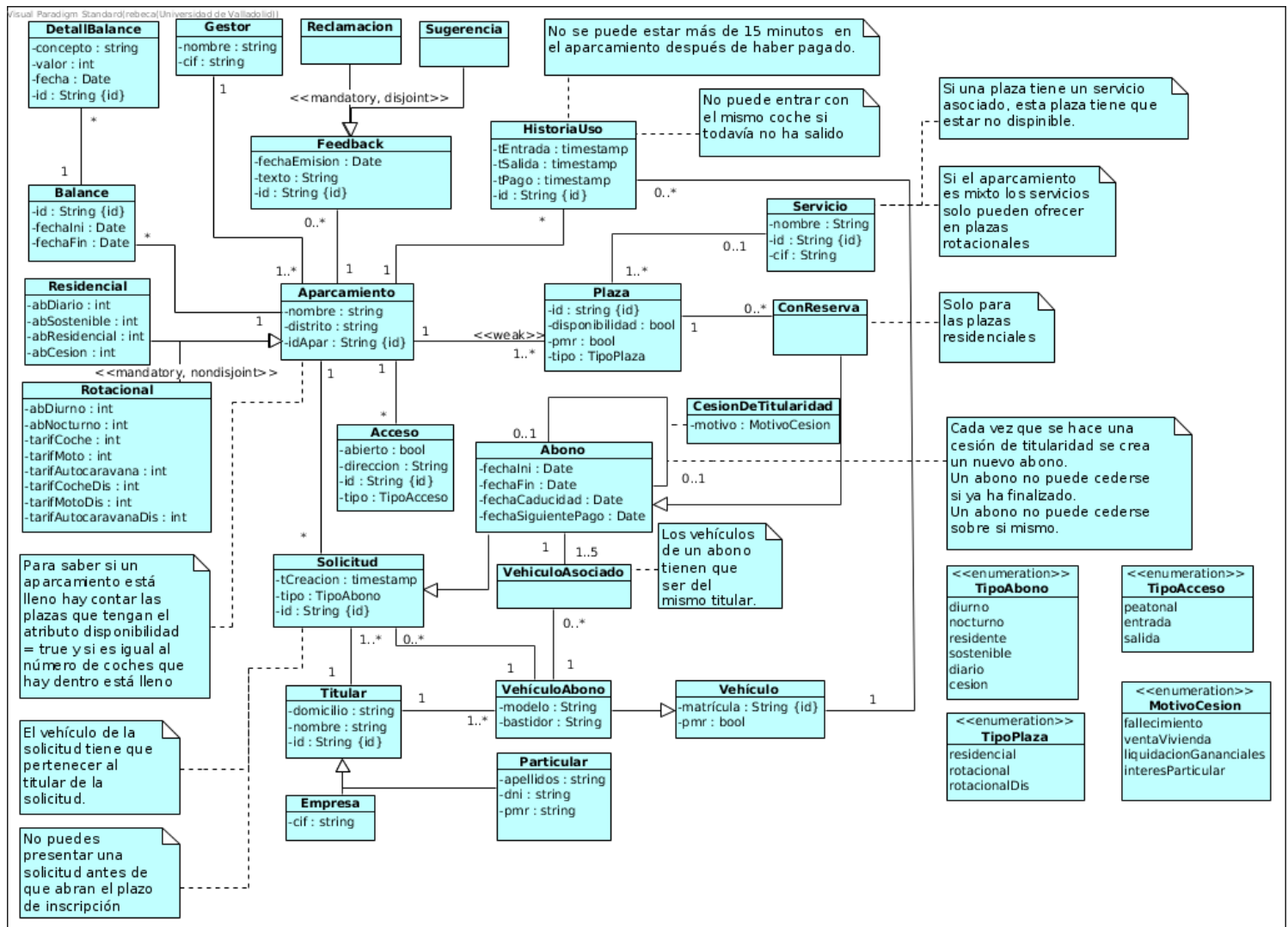
Restricciones

A continuación especificamos unas restricciones que no aparecen en el diagrama ER aunque si en el script.

- Se ha de comprobar que haya plazas libres antes de dejar entrar un coche a un aparcamiento mixto o rotacional.
- No puede haber dos balances en el mismo periodo de tiempo en el mismo aparcamiento.
- La fecha del detalle de balance debe estar comprendida entre la fecha inicial y la fecha final del balance al que pertenece.
- Si el CIF es null implica que un particular tiene como id la id de este titular.

- Si el CIF no es null implica que no existe un particular con una id como la de este titular.
- El tipo de solicitud ha de ser compatible con el tipo de aparcamiento al que se pide.
- La fecha fin del abono que se cessiona es menor que la fecha inicial del abono que se crea.
- La fecha final del abono a de ser mayor que la fecha inicial.
- Un abono cedido tiene la misma plaza del abono que se cede y la misma fecha de caducidad.

Imagen.



Esquema Relacional.

Aclaraciones.

Herencia feedback.

Las clases, *feedback*, *reclamación* y *sugerencia* se combinan en una única tabla *ReclamacionesYSugerencias* ya que se componen de los mismos atributos.

Herencia aparcamiento.

Como cada aparcamiento tiene sus precios diferentes y puede ser rotacional, residencial o mixto consideramos que es deberían ser 3 tablas diferentes ya que es la forma más sencilla de diferenciarlos y de saber su tipo según a qué tabla referencie.

Herencia titular.

Como hay dos tipos de titular, unos siendo empresas y los otros particulares, hacemos dos tablas, una con los datos comunes de los dos tipos de titular y otra con los datos específicos de los particulares.

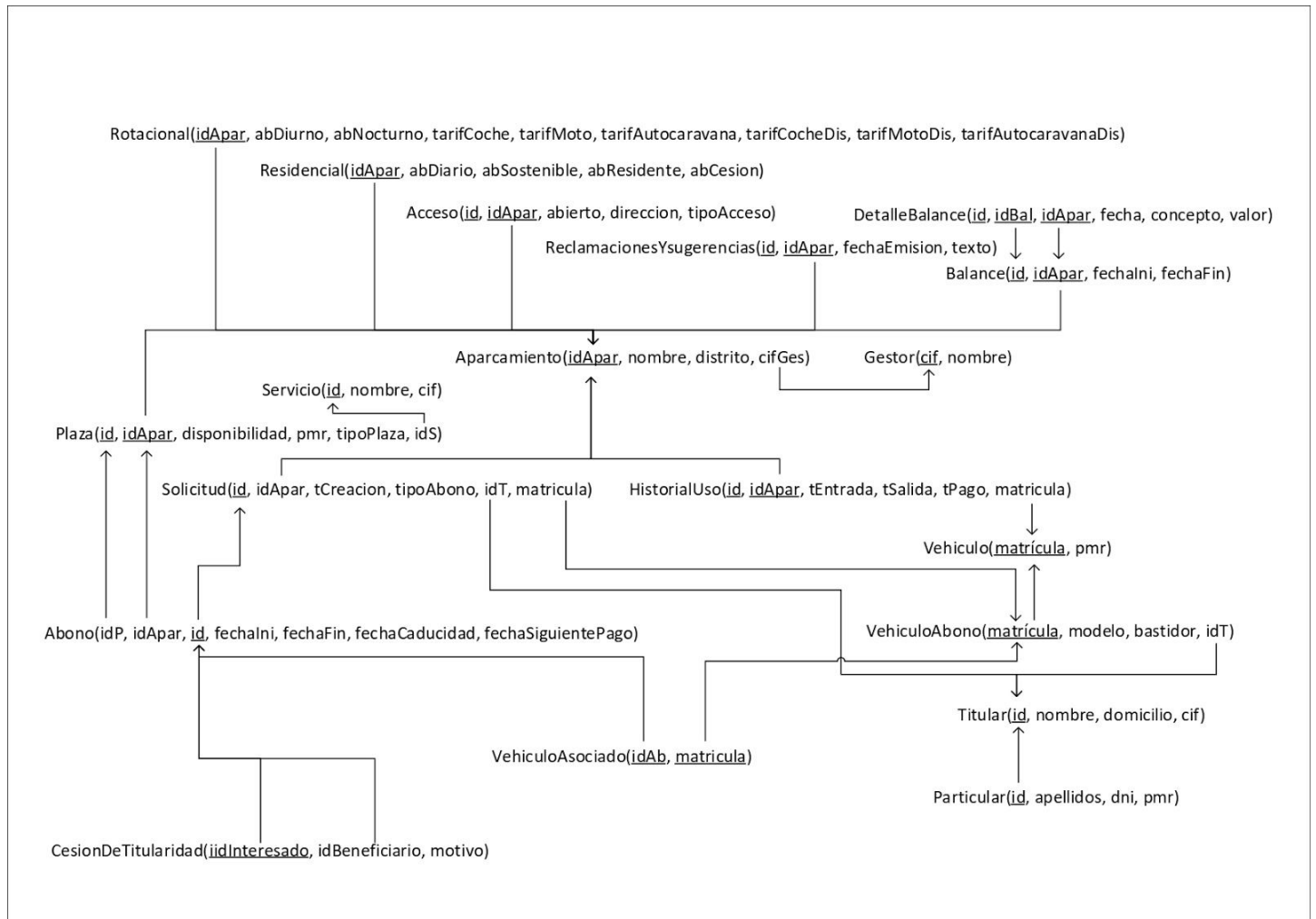
Herencia Abono y Solicitud.

Como todos los abonos provienen de una solicitud, pero no todas las solicitudes se convierten en abonos, hemos decidido representarlos en dos tablas.

Herencia Vehículo y Vehículo abonado.

De un vehículo que entra a un aparcamiento rotacional solo podemos conocer la matrícula y si es de movilidad reducida, mientras que de los vehículos en un abono conocemos más información, los representamos en dos tablas.

Imagen.



Consultas.

- Número de aparcamientos que hay en el sistema:

```
select count(Ap.idApar)
from Aparcamiento Ap;
-- resultado 3 = correcto
```
- Numero de plazas que hay en cada aparcamiento:

```
select Ap.idApar, count(*)
from Aparcamiento Ap,Plaza P
where Ap.idApar = P.idApar
group by Ap.idApar;
-- resultado 4 por aparcamiento = correcto
```
- El número de plazas rotacionales en cda aparcamiento:

```
select Ap.idApar, count(*)
from Aparcamiento Ap,Plaza P
where Ap.idApar = P.idApar and P.tipo ='rotacional'
group by Ap.idApar;
-- resultado 4 en el aparcamiento 2 y 2 en el aparcamiento 3 = correcto
```