

Trabajo final - Bases de Datos
Grado Ingeniería Informática

BICIVAL



Curso: 3AL2

Participantes.

- Ángel Carmona Martí
- Daniel Sánchez Tardy
- Javier Diéguez Bens
- Pablo Gonzálbez Cabo

Índice:

1. Introducción	Página 3
Explicación del trabajo y sus objetivos.	
2. Análisis	Página 4
Descripción del sistema de información, requisitos de información y de procesos del caso proporcionado por los profesores ampliado.	
3. Diseño conceptual	Página 6
Diagrama de clases UML y restricciones.	
4. Diseño lógico	Página 7
Esquema relacional (traducción del diagrama de clases del UML) y esquema relacional normalizado, junto con las restricciones.	
5. Implementación	Página 12
Esquema gráfico relacional en oracle.	



1. INTRODUCCIÓN

En este trabajo buscamos resolver un problema planteado mediante las bases de datos. Nuestro proyecto consiste en realizar una base de datos para BiciVal. Todo esto ha sido descrito en esta memoria y entregado en diferentes fases.

En la primera fase analizamos el sistema de información realizando una ampliación y aclaración del resultado, además realizamos el diseño conceptual realizando un diagrama de clases con LucidChart.

Posteriormente, realizamos el diseño lógico, un diagrama Oracle de la base de datos generado con el Data Modeler y la normalización de los datos.

Y por último en la tercera fase lo primero fue llevar a cabo la carga de datos, con el objetivo de optimizar el tiempo, por ello, hemos utilizado ‘Mockaroo’, una plataforma que permite generar datos de manera masiva.



2. ANÁLISIS DEL SI

La empresa BiciVal desea diseñar una base de datos para la gestión del sistema de transporte en bicicleta. Las bicicletas tienen un código único y una fecha de adquisición. Se dispone de un sistema que permite conocer en todo momento en qué estado se encuentra cada bicicleta: en uso, aparcada, si se encuentra en reparación o en traslado.

Por otra parte, la empresa dispondrá de otro vehículo, las motocicletas eléctricas. Estas se podrán utilizar únicamente si el usuario tiene carnet adjuntado en su cuenta de socio. En ella se deberá conocer el estado en el que se encuentra, en caso de no tener batería, ésta no podrá ser alquilada, además se especificará el código de la moto que será único y la cilindrada será de 50 cc o 125cc (según el tipo de carnet que poseamos AM, B(50cc) y A1, B después de 3 años (125cc). R1

Si el vehículo está aparcado, se debe conocer en qué aparcamiento está. Todos los aparcamientos tienen un código único, una dirección y una capacidad.

En el caso en que esté en uso, se debe guardar la siguiente información: fecha, hora y aparcamiento de recogida, así como el usuario que la está utilizando. Para ser usuario del servicio, hay que registrarse como socio de BiciVal, pagando una cuota anual, y se debe conocer el DNI, que es único, nombre, fecha de nacimiento, dirección completa, un número de cuenta bancaria y carnet de conducir.

Cuando un usuario finaliza el viaje se guarda en la base de datos que el vehículo ya está aparcado. Los datos del viaje realizado se registran en un histórico de viajes, en el que se debe guardar, el vehículo que se ha usado, el usuario que la ha utilizado, el día y hora de recogida, y el día y hora de entrega, teniendo en cuenta que en un día y a una hora un vehículo no puede hacer dos viajes. Esta información se utilizará para facturar a final de mes al socio aquellos viajes que han excedido de un tiempo superior a la media hora.

Los vehículos también pueden estar en reparación, en este caso se debe conocer el taller donde está reparándose, indicando el código del taller, que es único, la dirección completa, el teléfono, y el transportista que la ha llevado. De los transportistas se conoce su DNI, que es único, su nombre, dirección y teléfono.

En todo momento, se puede consultar en la base de datos la ocupación de cada aparcamiento. Esta información se usa para hacer traslados de vehículos de unos aparcamientos a otros. Cuando se quiere realizar un traslado, se comunica por teléfono al transportista cuántas unidades ha de trasladar de un aparcamiento a otro. Cuando el transportista recoge una bicicleta para trasladarla, ésta pasa al estado “en traslado”, y cuando la deposita en el aparcamiento destino, se almacena que está en estado aparcada. Finalizado el traslado, el transportista comunica que el traslado ya está hecho y se guarda en la base de datos los datos siguientes: código del traslado (que es único), fecha y hora de finalización del traslado, aparcamiento de origen, aparcamiento de destino, cantidad de bicicletas trasladadas y el transportista que lo ha realizado.



RESTRICCIONES

R1: Solo podremos utilizar motos de 50 cc si poseemos el carnet B o AM

R2: Solo podremos utilizar motos de 125 cc si poseemos A1 o B con más de 3 años

R3: La ocupación del aparcamiento debe ser menor o igual a la capacidad.

R4: Si la motocicleta tiene la batería a 0 no podrá alquilarse.

R5: Aquellos viajes que han excedido de un tiempo superior a la media hora serán facturados.

LISTA DE REQUISITOS DE PROCESO Y/O CONSULTA

Se desea diseñar una base de datos para la gestión de un servicio municipal de transporte en bicicleta y moto eléctrica para las personas de la ciudad. En el análisis del sistema se han identificado los requerimientos que van a hacerse con más frecuencia, que son:

USUARIO:

- Se podrán hacer consultas sobre el histórico de los viajes de los distintos vehículos así como de las características de los mismos.
- Obtener un listado de los aparcamientos más cercanos que estén disponibles.
- El usuario puede solicitar una factura parcial de lo que lleva gastado ese mes.
- El usuario podrá actualizar sus datos de licencia de conducción.

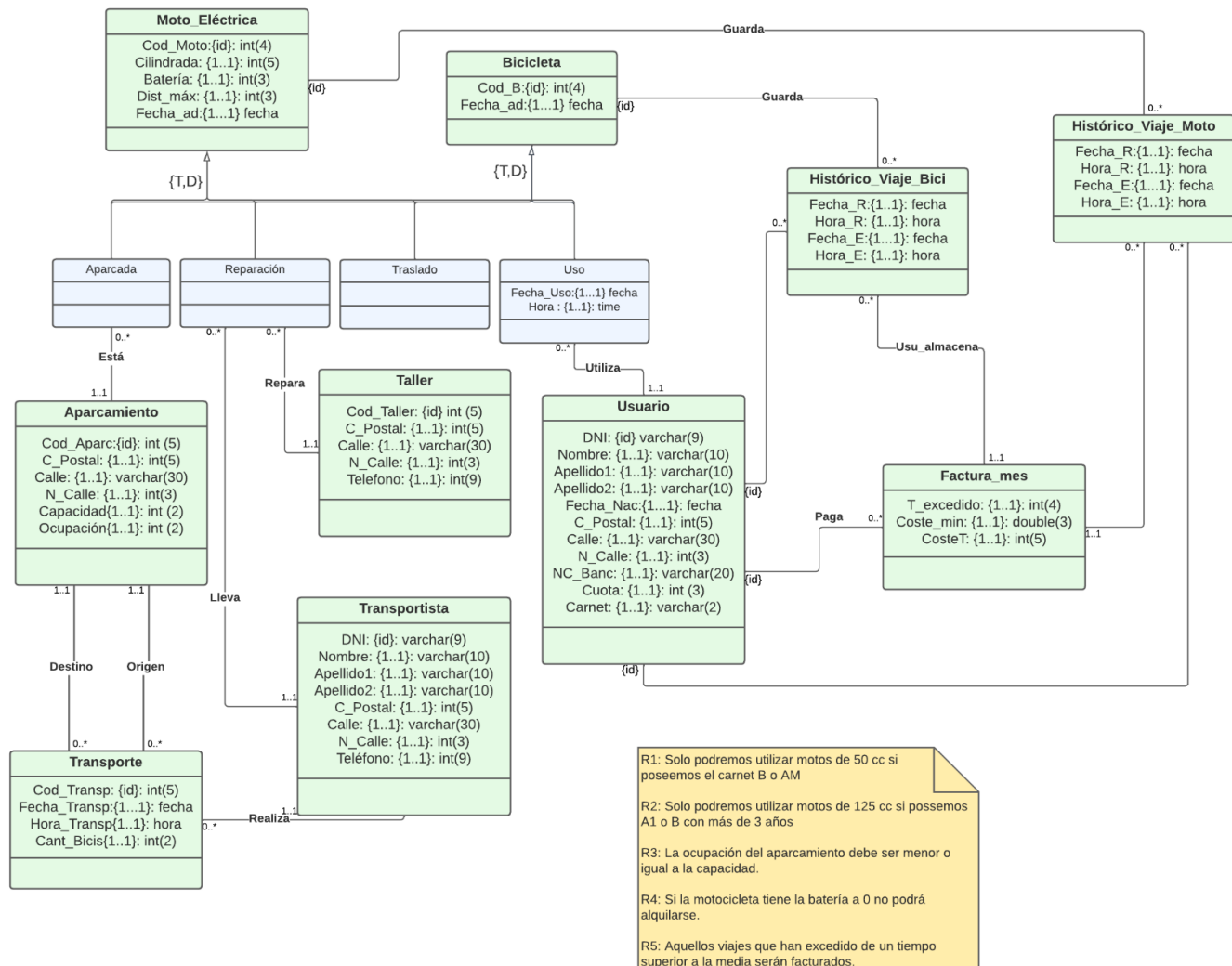
TRANSPORTISTA

- El transportista podrá obtener un listado de las ubicaciones de los talleres o aparcamientos para cada vehículo.
- El transportista podrá visualizar los aparcamientos en función de su ocupación, para garantizar que el aparcamiento tenga suficiente capacidad para el traslado.
- El transportista podrá visualizar la batería de los vehículos eléctricos que tengan menos de un 20% de capacidad para proceder a recargar su batería en el sitio reservado que tienen los aparcamientos (este está hecho con tal de que ninguna otra persona pueda hacer uso de éste).



3. DISEÑO CONCEPTUAL

DIAGRAMA DE CLASES UML:



Insertamos el enlace por si la imagen no es clara: [BiciVal: Lucidchart](#)



4. DISEÑO LÓGICO

El diseño lógico consiste en la transformación del esquema conceptual, que se encuentra descrito con un cierto modelo de datos, en estructuras descritas en términos del modelo de datos en el cual se basa el sistema de gestión de bases de datos que se vaya a utilizar. Así pues, el diseño lógico consistirá en transformar un diagrama de clases (esquema conceptual) en un esquema relacional (esquema lógico).

Bicicleta (Cod_B: int(4), Fecha_ad: Date)

CP: {Cod_B}

VNN: {Fecha_ad}

Histórico_Viaje_Bici (Fecha_R: Date, Hora_R: float(23), Fecha_E: Date, Hora_E: float(23),

Cod_B: int(4), DNI: varchar(9))

CP: {DNI, Cod_B}

VNN: {Fecha_R, Hora_R, Fecha_E, Hora_E}

CAj: {Cod_B} → Bicicleta (Cod_B)

MODIFICACIÓN CASCADA

CAj: {DNI} → Usuario (DNI)

MODIFICACIÓN CASCADA

Histórico_Viaje_Moto (Fecha_R: Date, Hora_R: float(23), Fecha_E: Date, Hora_E: float(23),

Cod_Moto: int(4), DNI: varchar(9))

CP: {DNI, Cod_Moto}

VNN: {Fecha_R, Hora_R, Fecha_E, Hora_E}

CAj: {Cod_Moto} → Moto_Electrica (Cod_Moto)

MODIFICACIÓN CASCADA

CAj: {DNI} → Usuario (DNI)

MODIFICACIÓN CASCADA

Traslado (Cod_B: int(4), Cod_Moto: int(4))

CP: {Cod_B}, {Cod_Moto}

CAj: {Cod_B} → Bicicleta(Cod_B)

CAj: {Cod_Moto} → Moto_Electrica(Cod_Moto)

BORRADO CASCADA

MODIFICACIÓN CASCADA

Uso (Fecha_Uso: Date, Hora: Time, DNI: varchar(9), Cod_B: int(4), Cod_Moto: int(4))

CP: {Cod_B}, {Cod_Moto}

VNN: {Fecha_Uso, Hora, DNI}

CAj: {Cod_B} → Bicicleta(Cod_B)

CAj: {Cod_Moto} → Moto_Electrica(Cod_Moto)

BORRADO CASCADA

MODIFICACIÓN CASCADA

CAj: {DNI} → Usuario (DNI)

MODIFICACIÓN CASCADA



Aparcada (Cod_B: int(4), Cod_Aparc: int(5), Cod_Moto: int(4))

CP: {Cod_B},{Cod_Moto}

VNN: {Cod_Aparc}

CAj: {Cod_B} → Bicicleta(Cod_B)

CAj: {Cod_Moto} → Moto_Electrica(Cod_Moto)

BORRADO CASCADA

MODIFICACIÓN CASCADA

CAj: {Cod_Aparc} → Parking(Cod_Aparc)

MODIFICACIÓN CASCADA

Reparación (Cod_B: int(4), DNI: varchar(9), Cod_Taller: int(5), Cod_Moto: int(4))

CP: {Cod_B},{Cod_Moto}

VNN: {DNI, Cod_Taller}

CAj: {Cod_B} → Bicicleta(Cod_B)

CAj: {Cod_Moto} → Moto_Electrica(Cod_Moto)

BORRADO CASCADA

MODIFICACIÓN CASCADA

CAj: {DNI} → Transportista (DNI)

MODIFICACIÓN CASCADA

CAj: {Cod_Taller} → Taller (Cod_Taller)

MODIFICACIÓN CASCADA

Usuario (DNI: varchar (9), Nombre: varchar(10) , Fecha_Nac: Date, Dir_Usu: varchar(30),

NC_Banc: varchar(18), Cuota: int(3), Carnet: varchar(2))

CP: {DNI}

VNN: {Nombre, Fecha_Nac, Dir_Usu, NC_Banc, Cuota}

Aparcamiento (Cod_Aparc: int(5), Dir_Aparc: varchar (30), Capacidad: int(2), Ocupación: int(2))

CP: {Cod_Aparc}

VNN: {Dir_Aparc, Capacidad, Ocupación}

Transporte (Cod_Transp: int(5), Fecha_Transp: Date, Hora_Transp: float(23), Cant_Bicis: int(2),

Cod_Aparc1: int(5), Cod_Aparc2: int(5), DNI_Transp: varchar(9))

CP: {Cod_Transp}

VNN: {Fecha_Transp, Hora_Transp, Cant_Bicis, Cod_Aparc1, Cod_Aparc2, Cod_Transp} 5

CAj: {Cod_Aparc1} → Parking(Cod_Aparc1)

MODIFICACIÓN CASCADA

CAj: {Cod_Aparc2} → Parking(Cod_Aparc2)

MODIFICACIÓN CASCADA

CAj: {DNI_Transp} → Transportista (DNI)

MODIFICACIÓN CASCADA

Transportista (DNI: varchar (9), Nombre: varchar (50), Dir_Trans: varchar (30), Teléfono: int(9))

CP: {DNI}

VNN: {Nombre, Dir_Trans, Teléfono}



Taller (Cod_taller: int(5), Dir_Taller: varchar(30), Teléfono: int(9))

CP: {Cod_taller}

VNN: {Dir_Taller, Teléfono}

Moto_Eléctrica (Cod_Moto: int(4), Cilindrada: int(5), Bateria: int(3), Dist_max: int(3), Fecha_ad: Date)

CP: {Cod_Moto}

VNN: {Cilindrada, Bateria, Dist_max, Fecha_ad}

Factura_mes(T_excedido: int(4), Coste_min: double(3), Coste: int(5), DNI: int(9))

CP: {DNI}

VNN {T_excedido, Coste_min, Coste}

CAj: {DNI} → Usuario(DNI)

RESTRICCIONES DE INTEGRIDAD

Las restricciones de integridad que no se han podido representar en el diagrama son la siguientes:

RI_{Total}:

Todo valor que aparezca en el atributo Cod_B de Bicicleta debe aparecer en el atributo Cod_B de Uso, de Traslado, de Reparación o de Aparcada.

RI_{Disjunta}:

No puede existir un valor idéntico en el atributo:

- Cod_B de Traslado y en el Cod_B de Uso
- Cod_B de Traslado y en el Cod_B de Aparcada
- Cod_B de Traslado y en el Cod_B de Reparación
- Cod_B de Uso y en el Cod_B de Aparcada
- Cod_B de Uso y en el Cod_B de Reparación
- Cod_B de Aparcada y en el Cod_B de Reparación

Es decir, una bicicleta no podrá estar en diferentes estados a la vez.

TEORÍA DE LA NORMALIZACIÓN:

Para comprobar que nuestro esquema relacional se encuentra adecuadamente diseñado utilizamos los conceptos de la teoría de la normalización, así nos aseguramos de que no existe ningún problema de manipulación.

Respecto a la 1ª Forma Normal (**1FN**), observamos que tenemos conjuntos de valores que no deben existir debido a que las relaciones en esta primera forma sólo pueden tomar valores simples e indivisibles, es decir, no deben existir conjuntos ni tuplas de valores.



En este caso observamos que la Dir_Aparc, Dir_Usu, Dir_Taller tienen la dirección compuesta por Nombre de Calle, Número de Calle, por lo que cada una de las variables que contengan dirección se descompondrá en los mencionados anteriormente.

Además el usuario como hemos supuesto que Nombre contiene nombre y apellido, hemos descompuesto también dicha variable en Nombre, Apellido1, Apellido2.

Respecto a la 2a Forma Normal (**2FN**), se puede afirmar que todas las relaciones, excepto HistóricoViaje_Bici, HistóricoViaje_Moto, cumplen **2FN** puesto que estas relaciones solo contienen una clave primaria mientras que las dos mencionadas tienen más de una clave primaria. Esto implica que todos los atributos no-primos deben depender funcionalmente de forma completa de las claves. Si se eliminara cualquier clave principal, la dependencia no existiría ya que son necesarias para determinar el día y hora de recogida y el código de usuario que ha realizado el viaje y el día y hora de entrega de la bicicleta o la moto eléctrica. Dado todo lo expuesto podemos afirmar que con las modificaciones pertinentes hemos cumplido con la (**2FN**).

Por último, si el esquema está en 3a Forma Normal (**3FN**) deberemos comprobar que no existen dependencias funcionales entre los atributos no-primos de una relación. En nuestro esquema podemos observar que todas cumplen dicha condición.

MODIFICACIONES EN EL ESQUEMA RELACIONAL:

Aparcamiento (Cod_Aparc: int(5), C_Postal: int(5), Calle: varchar(30), N_Calle: int(3), Capacidad: int(2), Ocupación: int(2))

CP: {Cod_Aparc}

VNN: {Capacidad, Ocupación, C_Postal, Calle, N_Calle}

Usuario (DNI: varchar (9), Nombre: varchar(10) , Apellido1: varchar(10), Apellido2: varchar(10), Fecha_Nac: Date, C_Postal: int(5), Calle: varchar(30), N_Calle: int(3), NC_Banc: varchar(18), Cuota: int(3), Carnet: varchar(2))

CP: {DNI}

VNN: {Nombre, Nombre, Apellido1, Apellido2, Fecha_Nac, NC_Banc, Cuota, C_Postal, Calle, N_Calle}

Taller(Cod_taller: int(5), C_Postal: int(5), Calle: varchar(30), N_Calle: int(3), Teléfono:int(9))

CP: {Cod_taller}

VNN: {C_Postal, Calle, N_Calle, Teléfono}

Transportista (DNI: varchar(9), Nombre: varchar(10), Apellido1: varchar(10), Apellido2: varchar(10), C_Postal: int(5), Calle: varchar(30), N_Calle: int(3), Teléfono: int(9))

CP: {DNI}

VNN: {Nombre, Apellido1, Apellido2, C_Postal, Calle, N_Calle, Teléfono}



5. IMPLEMENTACIÓN

