	Apellidos:		
	Nombre:		
	DNI:	Num. mat.:	Grupo:

Normativa de examen

- No está permitido el uso de dispositivos móviles ni otros dispositivos electrónicos, así como libros ni apuntes.
- Durante el examen, los profesores podrán solicitar acreditar la identidad de los participantes en el mismo. Deberá tener en todo momento su Documento Nacional de Identidad y/o Carné de la UPM visible sobre la mesa.
- Deberá escribir su nombre, con bolígrafo, en todas las hojas de las que consta el examen.
- No se permite abandonar el aula de examen durante los primeros 15 minutos. Transcurrido este tiempo, no se permitirá entrar al examen.
- El examen tiene una duración máxima de **2.5 horas**.
- Justifique sus respuestas lo mejor posible indicando, si fuese necesario, los pasos realizados.
- Las calificaciones provisionales serán publicadas en el Moodle de la asignatura a los 15 días hábiles desde la fecha de realización del examen.
- La fecha para la revisión del examen se anunciará en el Moodle de la asignatura.

BLOQUE 1. (2½ Puntos) **Modelado**

Realizar un modelo conceptual de datos mediante la técnica del **modelo Entidad-Relación de Chen** teniendo en cuenta la siguiente descripción:

El **Moto Club Quemados** cuenta con varios centenares de socios. Su actividad principal es organizar quedadas de moteros a lo largo de toda España. Estas quedadas pueden durar varios días y su actividad principal es organizar rutas para los socios que se apunten a estos eventos. Al principio, la organización de las quedadas era informal y se podía hacer a mano; sin embargo, con el incremento de asociados, los organizadores pierden mucho tiempo cuadrando todo e informando a sus socios. Por esta razón, nos encargan realizar una aplicación mediante la cual, el moto club pueda tener controlado todo lo referente a las quedadas: Los socios participantes, las rutas, las motos, los premios y los hoteles usados durante un evento. De esta forma, los socios podrán consultar, también, a través del móvil los detalles que necesiten.

Los datos de los socios son: DNI, nombre y apellidos, dirección, teléfono, correo y fecha de alta en el moto club. Los socios tienen moto. Los datos de las motos son: Matrícula, marca, modelo y cilindrada. Cada moto es, obligatoriamente, de un único tipo (Scooter, Custom, Supermotard, Trail, etc.). De los tipos se guarda el tipo en sí y una descripción. Un tipo de moto puede aparecer asociado a varias motos. Cada moto es de un único socio y no se guardan datos históricos al respecto.

Cuando se organiza un evento o quedada, se decide un nombre para el mismo, una descripción y una fecha, así como la localidad donde se produce el evento. El evento tendrá asociada, al menos, una ruta que realizarán los socios que se apunten al evento. Las rutas vienen definidas por una descripción, kilómetros, hora de salida y de llegada, y un campo para almacenar un track GPX para los GPS. Las rutas están diseñadas específicamente para cada evento (no se reutilizan en otros eventos) y para cada tipo de moto. Puede ocurrir, sin embargo, que una ruta sea buena para varios tipos de moto. En cualquier caso, los socios quieren poder consultar cual es la ruta recomendada para su moto.

Los eventos de fin de semana completo tendrán uno o varios hoteles disponibles para alojar a los participantes. De los hoteles se almacenará el nombre, la dirección completa el teléfono y el correo. De los hoteles, sólo uno de ellos hará de cuartel general del evento. Es necesario almacenar esta información. Los socios participantes en el evento tendrán reserva en los distintos hoteles, si así lo desean, indicando la fecha de la reserva. Los

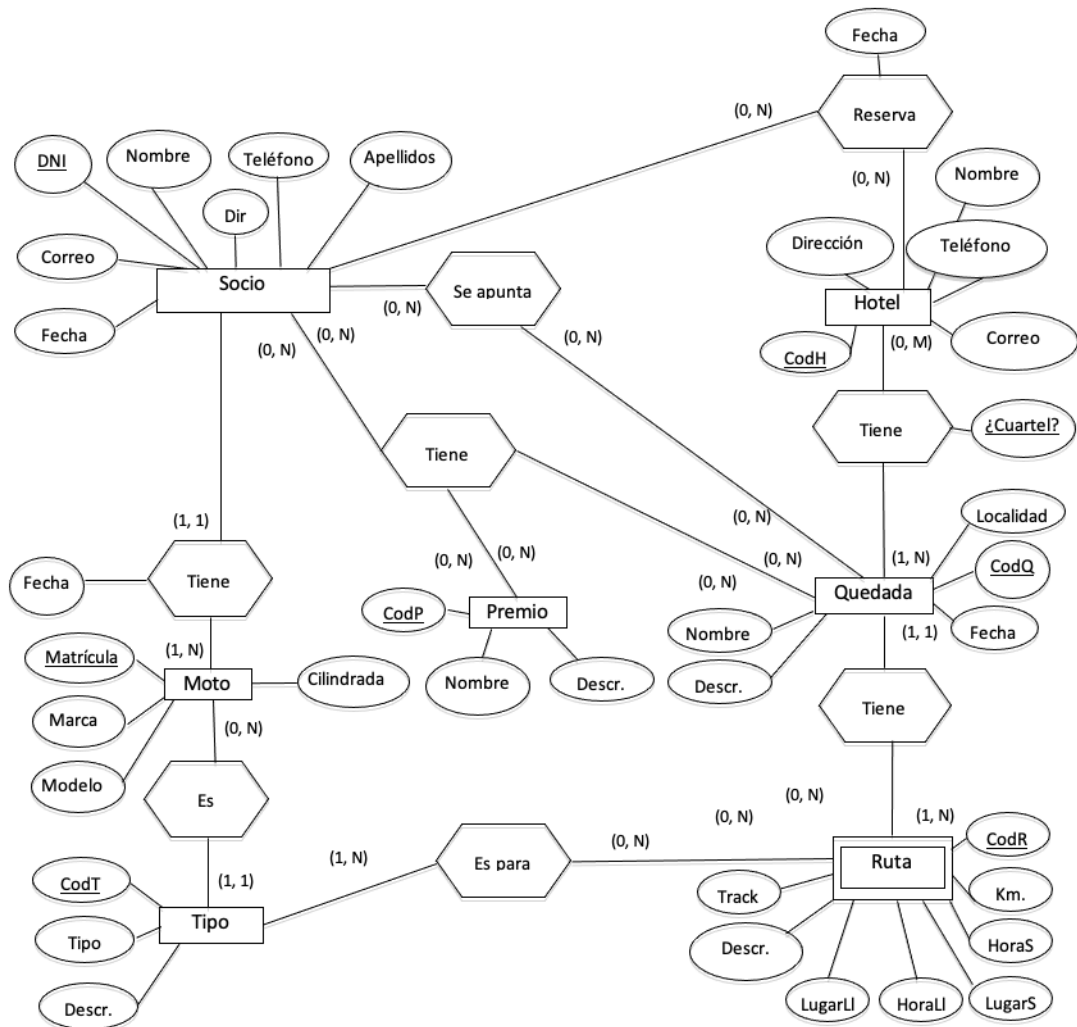
hoteles podrán participar en varios eventos.


Por último, se otorgaran distintos premios a los socios en cada quedada. De los premios se almacenará un nombre y una descripción. Se guardará un histórico de quién ha recibido cada premio en cada evento. En un evento en particular, un socio sólo puede tener un premio y cada premio es para un único socio, pero se pueden dar los mismos premios en distintas ocasiones.

Se pide: Se pide realizar el Modelo Entidad-Relación del problema descrito anteriormente empleando la notación de Chen, la utilización de cualquier otro formato será penalizada.

IMPORTANTE: Se penalizará la inclusión de elementos no necesarios o redundantes.

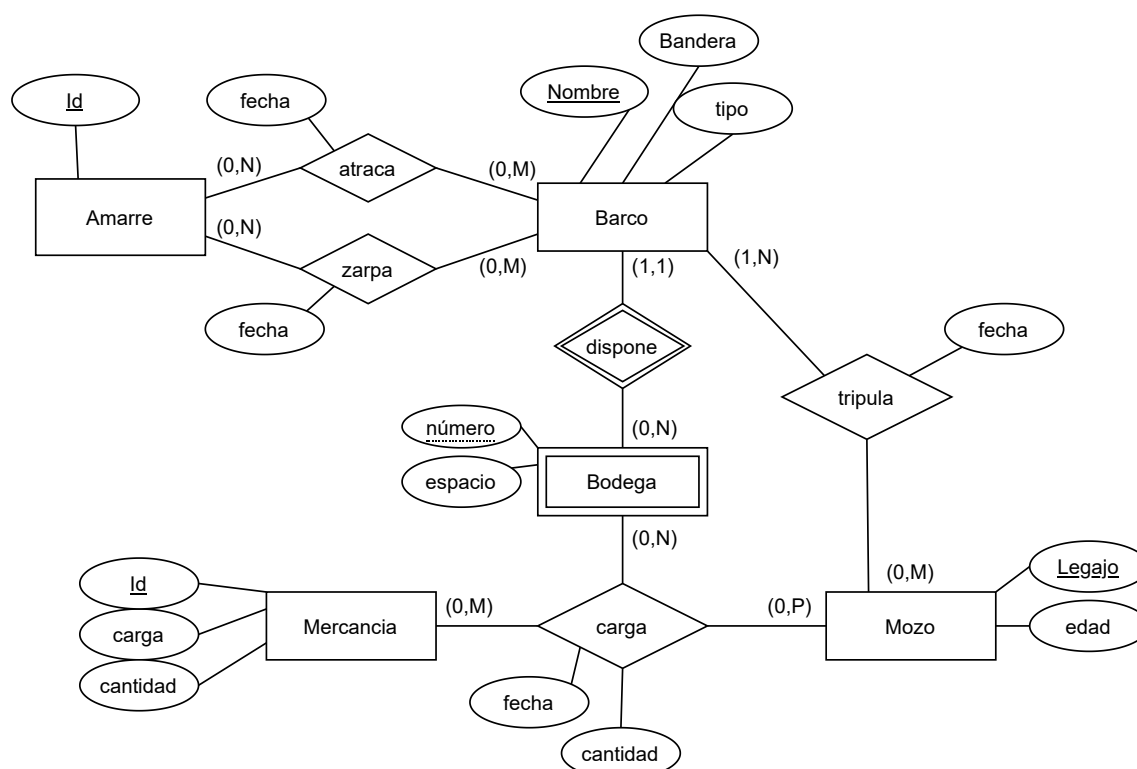
Solución propuesta:



	Apellidos:		
	Nombre:		
	DNI:	Num. mat.:	Grupo:

BLOQUE 2. (3½ puntos) Consultas

Chen ha vuelto a hacer de las suyas y se ha pasado por el puerto de *Citra* para vender su modelo a los comerciantes locales. *Zoltan Chivay* ha quedado prendado y ha decidido implementarlo para las mercancías de joyas que llegan en barco a los diferentes amarres del puerto. A pesar de tener ya el modelo Entidad/Relación finalizado, queda aún trabajo por realizar antes de implementar el sistema completo. Resuelva las preguntas del **Bloque 2** y **Bloque 3** utilizando el modelo relacional resultante del modelo Entidad-Relación detallado a continuación:



(a) **Álgebra relacional:**

- I. ($\frac{1}{2}$ Punto) Obtener el **#nombre** de los barcos que tengan como tripulación todos los “*Mozos*” con 30 años.

Solución propuesta:

$$\frac{\Pi_{\#nombre, \#legajo} (MOZO_TRIPULA)}{\Pi_{\#legajo} (\sigma_{edad=30} (Mozo))}$$

- II. ($\frac{1}{2}$ Punto) Obtener las diferentes edades de los mozos que nunca han descargado mercancía de ningún barco y pertenecen a la tripulación de un barco con bandera española.

Solución propuesta:

$$\frac{\Pi_{edad} [MOZO \bowtie \Pi_{\#legajo} (MOZO_TRIPULA \bowtie \sigma_{Bandera="Española"} (BARCO))]}{\Pi_{\#legajo} (CARGA)}$$

- III. ($\frac{1}{2}$ Punto) Obtener el *Nombre* y *tipo* de aquellos barcos que hayan atracado y zarpado en el mismo día de "07/02/2041".

Solución propuesta:

$$\Pi_{\#nombre, tipo} (\sigma_{fecha="07/02/2041"} ATRACA) \cap \Pi_{\#nombre, tipo} (\sigma_{fecha="07/02/2041"} ZARPA)$$

(b) **SQL:**

- I. (1/2 Punto) Generar las sentencias SQL necesarias para la creación de las tablas que se generan de la relación: “Carga”. Se deben incluir las claves primarias y ajenas correspondientes.

Solución propuesta:

```
CREATE TABLE Carga (  
  Id INTEGER NOT NULL,  
  Nombre VARCHAR(60) NOT NULL,  
  numero INTEGER NOT NULL,  
  Legajo VARCHAR(60) NOT NULL,  
  fecha DATE NOT NULL,  
  cantidad INTEGER NOT NULL,  
  PRIMARY KEY (nombre,numero,Id,Legajo,fecha),  
  CONSTRAINT  
    FOREIGN KEY (Nombre,numero)  
    REFERENCES Barco (Nombre,numero),  
  CONSTRAINT  
    FOREIGN KEY (Id)  
    REFERENCES Mercancia(id),  
  CONSTRAINT  
    FOREIGN KEY (Legajo)  
    REFERENCES Mozo(Legajo)  
);
```

- II. (1/2 Punto) Obtener el Nombre y el número de las bodegas que nunca han transportado mercancía de “carga = Anillos de Nilfgaard”.

Solución propuesta:

```
SELECT nombre, numero  
FROM bodega INNER JOIN mozo_tripula  
  ON mozo.legajo = mozo_tripula.legajo  
  INNER JOIN barco  
  ON barco.nombre = mozo_tripula.nombre  
WHERE bodega.nombre NOT IN (SELECT nombre  
  from carga inner join mercancia  
  on carga.Id = Mercancia.ID  
  where carga = 'Anillos de Nilfgaard')  
AND bodega.numero NOT IN (SELECT numero  
  from carga inner join mercancia  
  on carga.Id = mercancia.ID  
  where carga = 'Anillos de Nilfgaard')
```

- III. (1/2 Punto) Obtener el nombre de los barcos que tengan como tripulación algún “Mozo” con edades comprendidas entre 15 años y 30 años, y que además hayan atracado en todos los amarres.


Solución propuesta:

```
SELECT barco.nombre
FROM barco INNER JOIN tripula
            ON barco.nombre = tripula.nombre
INNER JOIN mozo
            ON mozo.legajo. = tripula.legajo
INNER JOIN atraca
            ON barco.nombre = atraca.nombre
WHERE mozo.edad BETWEEN 15 AND 30
GROUP BY barco.nombre
HAVING COUNT(DISTINCT atraca.id) = (SELECT COUNT(*)
                                   FROM amarre)
```

- IV. (1/2 Punto) Borrar todos aquellos mozos que nunca han tripulado ningún barco con bandera Española y que hayan cargado un total de mercancías mas bajo de 10.

Solución propuesta:

```
DELETE
FROM mozo
WHERE mozo.legajo NOT IN (SELECT legajo
                          from tripula INNER JOIN barco
                          ON barco.nombre = tripula.nombre)
AND mozo.legajo IN (SELECT legajo
                   FROM carga
                   GROUP BY legajo
                   HAVING SUM(cantidad) < 10)
)
```


	Apellidos:		
	Nombre:		
	DNI:	Num. mat.:	Grupo:

BLOQUE 3. (2 puntos) Gestión

- (a) (1 Punto) Las autoridades el puerto nos piden una forma fácil para rastrear los mozos que son sospechos de haber robado. Para ello, desarrolle un procedimiento para obtener los mozos que han tripulado un barco y han cargado en él mercancía de un determinado tipo y en un periodo de tiempo entre dos fechas distintas.

Solución propuesta:

```
DELIMITER $$
CREATE PROCEDURE pillar_ladrones (IN FECHA_INICIO DATETIME,
                                IN FECHA_FINAL DATETIME,
                                IN nombre_carga VARCHAR(60))
BEGIN
    SELECT distinct mozo_tripula.legajo
    FROM mozo_tripula JOIN carga_bodega
        ON mozo_tripula.nombre = carga_bodega.nombre
        JOIN mercancia ON mercancia.id=carga_bodega.id
    WHERE carga_bodega.legajo = mozo_tripula.legajo AND
        mercancia.carga = nombre_carga AND
        carga_bodega.fecha > FECHA_INICIO AND
        carga_bodega.fecha < FECHA_FINAL;

END$$
DELIMITER ;
```


- (b) (1 Punto) Indica la sentencia SQL necesaria para añadir un atributo fuerza a la tabla de los Mozos. A continuación, implementa un trigger para que únicamente aquellos mozos con fuerza mayor que 20 puedan cargar mercancía en bodegas de los supercargueros (barcos en los que alguna de sus bodegas tenga valor “Grande” en la columna *espacio*). Utiliza un cursor para recorrer las bodegas.

Solución propuesta:

```
ALTER TABLE mozo ADD fuerza INT;

DELIMITER $$
CREATE TRIGGER FUERZA_SUFICIENTE BEFORE INSERT ON carga_bodega
FOR EACH ROW
BEGIN
    DECLARE done INT DEFAULT FALSE;
    DECLARE espacio_bodega VARCHAR(20);
    DECLARE cur1 CURSOR FOR SELECT espacio FROM bodega WHERE nombre=NEW.nombre;
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
    OPEN cur1;
    read_loop: LOOP
        FETCH cur1 INTO espacio_bodega;
        IF espacio_bodega = "Grande" THEN
            IF EXISTS (SELECT legajo FROM mozo
                       WHERE mozo.legajo=NEW.legajo and fuerza < 20)
            THEN SIGNAL SQLSTATE '45000'
                 SET MESSAGE_TEXT = 'Este mozo no puede cargar porque
                                     no tiene suficiente fuerza';
            END IF;
        END IF;
        IF done THEN LEAVE read_loop;
    END IF;
END LOOP;
CLOSE cur1;

END$$
DELIMITER ;
```

	Apellidos:		
	Nombre:		
	DNI:	Num. mat.:	Grupo:

BLOQUE 4. (2 puntos) Ficheros(a) (1 1/2 Puntos) Dado el siguiente fichero `inventario.xml`:

```
<?xml version="1.0" encoding="UTF-8"?>
<inventario>
  <juego codigo-prod="J001">
    <titulo>Zelda Breath Of The Wild</titulo>
    <año>2017</año>
    <genero>aventuras</genero>
    <genero>accion</genero>
    <plataforma referencia="C001">Nintendo Switch</plataforma>
    <precio>69.99</precio>
    <descuento>5</descuento>
  </juego>
  <juego codigo-prod="J002">
    <titulo>Super Mario Odyssey</titulo>
    <año>2017</año>
    <genero>aventuras</genero>
    <genero>plataformas</genero>
    <plataforma referencia="C001">Nintendo Switch</plataforma>
    <precio>59.99</precio>
    <descuento>10</descuento>
  </juego>
  <juego codigo-prod="J003">
    <titulo>Fortnite La Ultima Risa</titulo>
    <año>2020</año>
    <genero>shooter</genero>
    <genero>accion</genero>
    <plataforma referencia="C001">Nintendo Switch</plataforma>
    <precio>54.90</precio>
    <descuento>15</descuento>
  </juego>
  <consola codigo-prod="C001">
    <nombre>Nintendo Switch</nombre>
    <modelo>Estandar</modelo>
    <precio>300</precio>
    <descuento>10</descuento>
  </consola>
</inventario>
```

1. Diseña una consulta en XPath para obtener el título de los juegos de tipo shooter. El resultado de la consulta debe ser el siguiente:

```
<titulo>Fortnite La Ultima Risa</titulo>
```

Solución propuesta:

```
/inventario/juego[genero='shooter']/titulo
```

- II. Diseña una consulta en XPath para obtener el número de juegos que tiene el fichero. El resultado de la consulta debe ser el siguiente: 3

Solución propuesta:

```
count(/inventario/juego)
```

- III. Diseña una consulta en XQuery para obtener el precio más alto de todos los juegos de aventuras. El resultado de la consulta debe ser el siguiente:

```
<precio_max>69.99</precio_max>
```

Solución propuesta:

```
for $a in /inventario
let $m := max($a//juego[genero='aventuras']/precio)
return <precio_max>{$m}</precio_max>
␣
for $a in /inventario
where $a/juego/genero='aventuras'
return <precio_max>{max($a//juego/precio)}</precio_max>
```

- (b) (1/2 Punto) A partir del fichero `inventario.xml`, pase TODA la información relativa a los tres juegos a un fichero de tipo CSV.

Solución propuesta:

```
codigo-prod,titulo,año,genero1,genero2,plataforma,referencia,precio,descuento
J001,Zelda Breath Of The Wild,2017,aventuras,accion,Nintendo Switch,C001,69.99,5
J002,Super Mario Odyssey,2017,aventuras,plataformas,Nintendo Switch,C001,59.99,10
J003,Fortnite La Ultima Risa,2020,shooter,accion,Nintendo Switch,C001,54.90,15
```