

NOMBRE ALUMNO:

PEC 2: RECOPIACIÓN PREPARATORIA PARA EL EXAMEN

Presentación

Esta prueba consta de 5 ejercicios enfocados a determinar el grado de comprensión de los módulos didácticos después de su estudio:

- Modelo conceptual
- Modelo lógico
- Normalización
- Árboles B+
- Preguntas teóricas



NOMBRE ALUMNO:

Descripción de la PEC

EJERCICIO 1: modelo conceptual

El gobierno de las Islas Caimán quiere diseñar parte de una base de datos que permita almacenar información relacionada con las becas universitarias que pueden recibir sus estudiantes.

Las universidades tienen un código, que las identifica, un nombre y unas iniciales. Cada una de ellas tiene diferentes facultades. Una facultad sólo pertenece a una única universidad.

Las facultades tienen un código, identificador dentro de la universidad a la que pertenecen, es decir, que dos facultades de universidades diferentes, pueden tener el mismo código. También tienen un nombre y la dirección de donde están ubicadas.

Cada facultad ofrece diferentes estudios en un año determinado. Un estudio se ofrece en una facultad varios años. Cada año se puede ofrecer el mismo estudio en diferentes facultades.

Los estudios tienen un código identificador, un nombre, la duración en años y, en algunos casos, una descripción de su contenido. Cada estudio es de una rama de conocimiento.

Las ramas de conocimiento tienen un identificador numérico y una descripción. Cada una incluye diferentes estudios.

Los estudiantes cursan estudios. Cada estudiante puede cursar varios estudios y un mismo estudio lo cursan diferentes estudiantes.

Los estudiantes, que se identifican por un código numérico, tienen un nombre, la fecha de nacimiento y una o más nacionalidades.

Los países de las nacionalidades de los estudiantes tienen una abreviación, que los identifica, un nombre, que es único, y se quiere saber si entre su gobierno y el de las Islas Caimán hay algún acuerdo de colaboración en el ámbito de la enseñanza universitaria.

**NOMBRE ALUMNO:**

Un estudiante que cursa unos estudios, puede recibir becas. Concretamente, por cada estudio que haga un estudiante puede recibir varias becas, y una misma beca se puede dar a diferentes estudiantes que hagan un estudio.

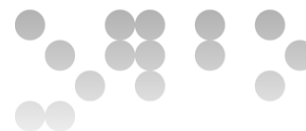
Las becas tienen un código numérico, que las identifica, un nombre, un importe base y diferentes condiciones a cumplir en el momento de la asignación.

Las condiciones tienen un código identificador, una definición, de qué tipo son (económico, de salud, familiar u otras) y cuántos puntos valen (entre 1 y 10).

Hay dos tipos diferentes de becas: aquellas que subvenciona el ministerio y aquellas que subvenciona cada universidad. De las primeras se quiere saber qué departamento gubernamental las organiza (sólo puede ser uno). Un departamento del gobierno puede organizar más de una beca ministerial. De las segundas se quiere saber a qué universidad pertenecen (sólo puede ser a una). Cada universidad puede ofrecer diferentes becas universitarias.

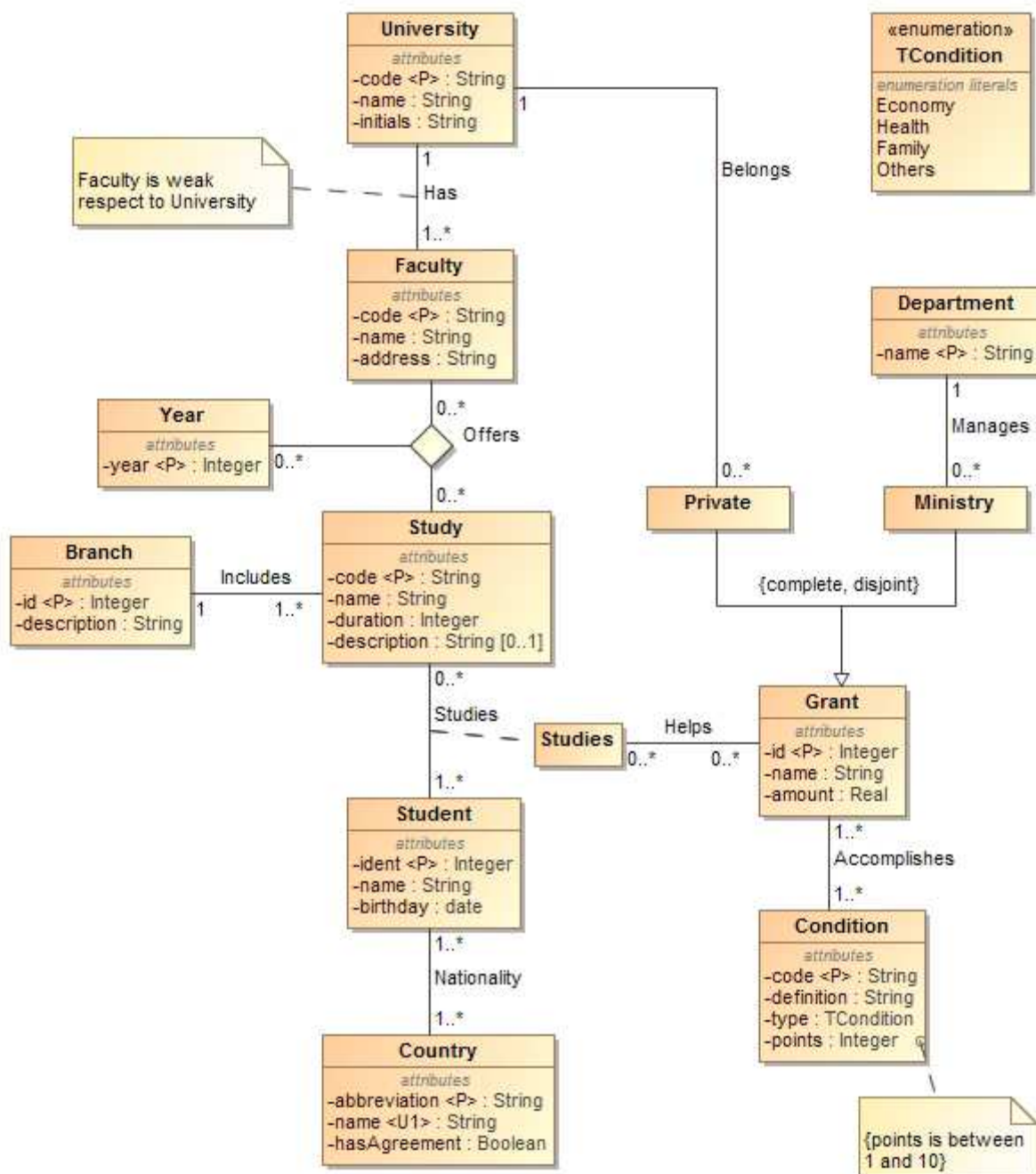
Los departamentos del ministerio tienen un nombre, que los identifica.

Se pide: realizar el diseño conceptual, mediante un diagrama de clases UML, que recoja toda la semántica del enunciado. Hay que indicar los atributos de los tipos de entidades, las cardinalidades de los tipos de relaciones, así como los requisitos que no han quedado reflejados en el esquema propuesto. Si se ha realizado alguna suposición semántica adicional también se tiene que indicar.



NOMBRE ALUMNO:

SOLUCIÓN:

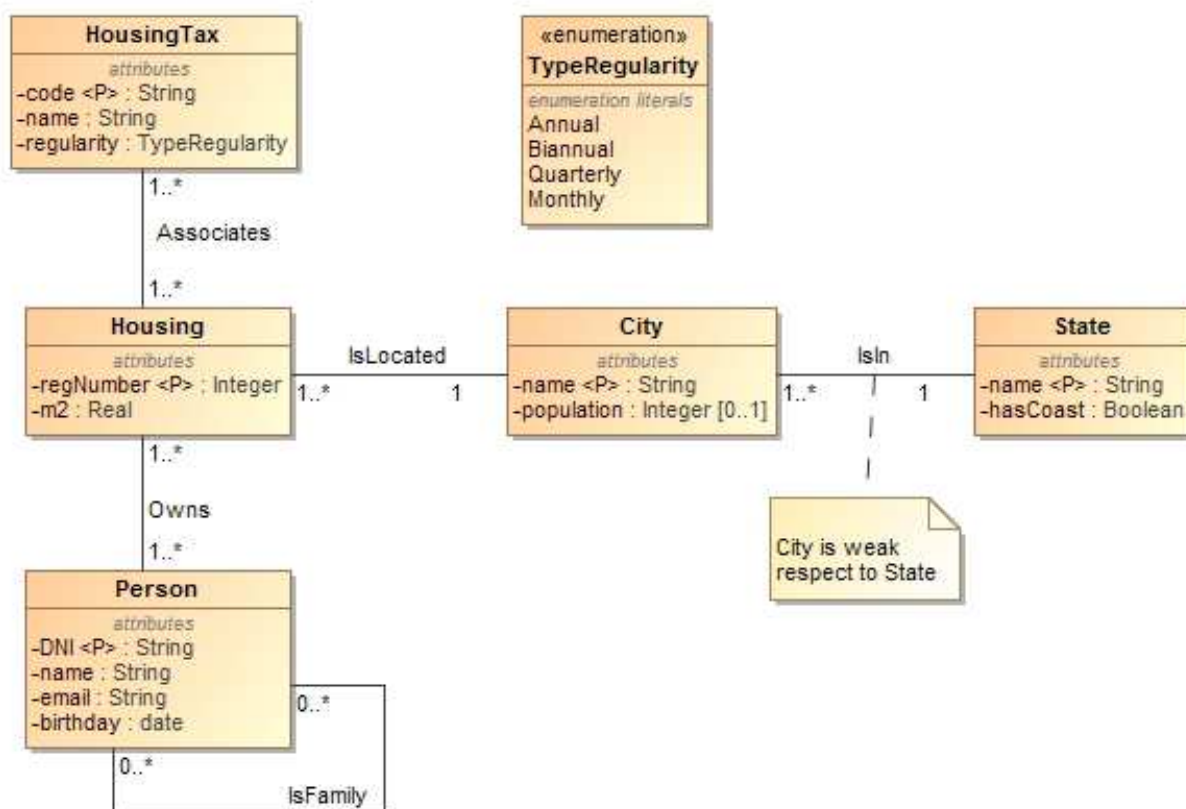




NOMBRE ALUMNO:

EJERCICIO 2: modelo lógico

Dado el modelo conceptual siguiente:



Se pide: obtener el modelo lógico relacional correspondiente, indicando si, debido a la transformación, aparecen atributos que pueden tomar valor NULL.



NOMBRE ALUMNO:

SOLUCIÓN:

State (name, **hasCoast**)

City (cityName, stateName, population)

{stateName} is foreign key to State

HousingTax (code, **name**, **regularity**)

Housing (regNumber, **m2**, **cityName**, **stateName**)

{cityName, stateName} is foreing key to City

Associates (taxCode, regNumber)

{taxCode} is foreign key to HousingTax

{regNumber} is foreign key to Housing

Person (DNI, **name**, **email**, **birthday**)

IsFamily (person1, person2)

{person1} is foreign key to Person

{person2} is foreign key to Person

Owns (DNI, regNumber)

{DNI} is foreign key to Person

{regNumber} is foreign key to Housing

Nota: los atributos que no son clave y no pueden ser nulo están en negrita.



NOMBRE ALUMNO:

EJERCICIO 3: normalización

Nos proporcionan la siguiente relación en modelo relacional:

Student (name, address, phone, category, profession)

No nos proporcionan ninguna otra información sobre los determinantes de los atributos.

Contestad las siguientes preguntas y justificad todas las respuestas:

- 1) ¿Está la relación en 1FN?
- 2) ¿Está la relación en 2FN?
- 3) ¿Está la relación en 3FN?
- 4) Explicad qué dependencias deberíamos tener en esta relación para asegurar que estuviese en FNBC.

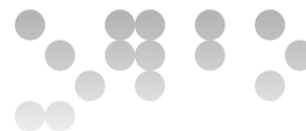
SOLUCIÓN:

- 1) Podemos asegurar que la relación está en 1FN. Todos los atributos deberían ser atómicos, cosa que de hecho ya nos asegura el modelo relacional.
- 2) También podemos asegurar que la relación está en 2FN. Esto es así porque la clave primaria consta de un solo atributo, y no tenemos ninguna otra clave candidata. Por tanto, no sería posible encontrar ningún atributo que no perteneciese a ninguna clave candidata y que solo dependiese de una parte de alguna clave candidata.
- 3) Con la información que tenemos no lo podemos saber. Podría ser que alguno de los atributos que no pertenece a ninguna clave candidata dependiese de algún atributo que no fuese *name*. Al no disponer de esta información, no lo podemos confirmar.
- 4) Para tener la relación en FNBC, los atributos que no forman parte de ninguna clave candidata tienen que depender al completo de todas las claves candidatas de la relación. Además, la FNBC también debe cumplir



NOMBRE ALUMNO:

que no pueden haber dependencias entre claves candidatas. En este caso solo tenemos una clave candidata (*name*), con lo cual ya se cumple esto último. Por tanto, para asegurar que la relación estuviese en FNBC, el resto de atributos (*address*, *phone*, *category*, *profession*) tendrían que depender de (y solo de) el atributo *name*.

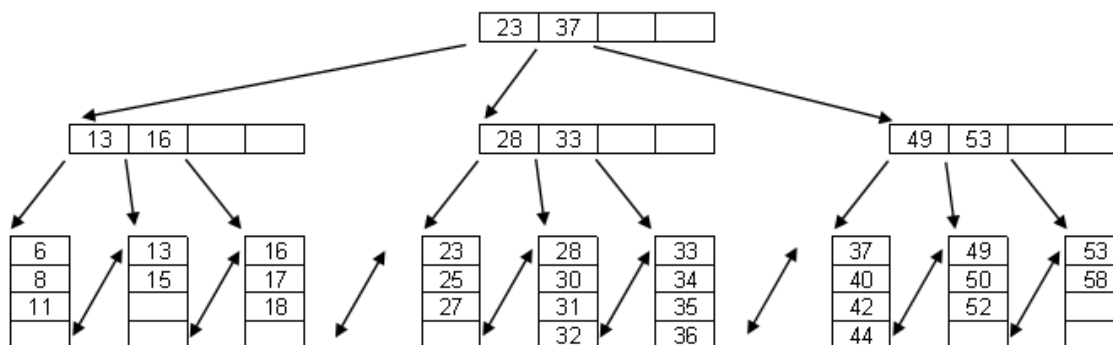


NOMBRE ALUMNO:

EJERCICIO 4: árboles B+

APARTADO A

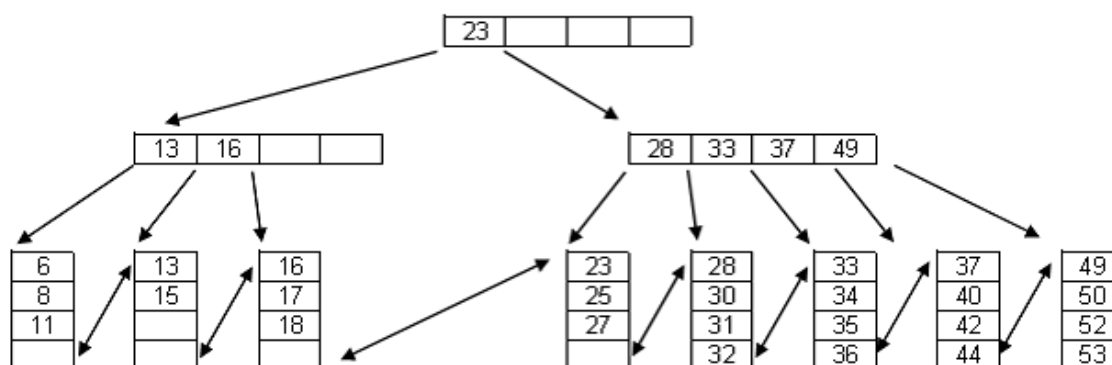
En la siguiente figura tenemos representado un árbol B+ de orden 2:



Mostrad cómo quedaría el árbol después de eliminar el valor 58.

SOLUCIÓN:

El nodo queda sólo con un valor (53), por tanto se ha de fusionar su hermano de la izquierda, ya que no tiene hermano a la derecha. El mismo valor se pierde en el nodo padre, quedando solo una posición ocupada, por tanto también se ha de fusionar con el nodo padre hermano de la izquierda.

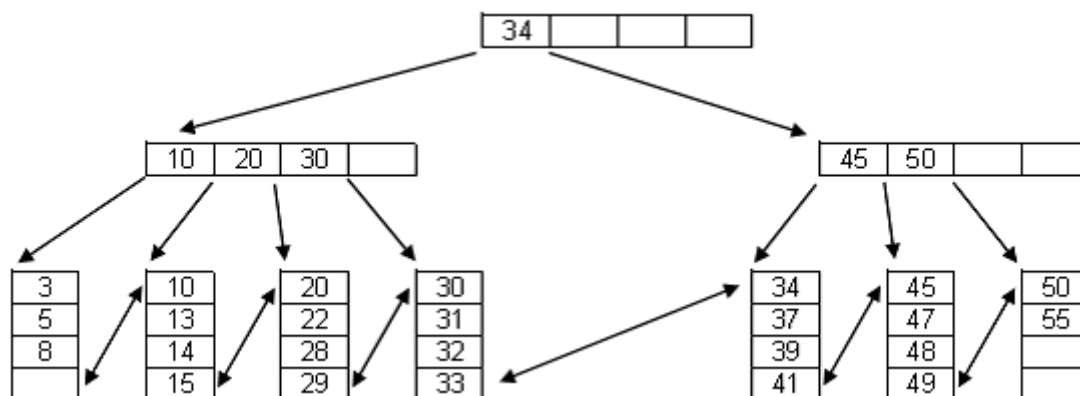




NOMBRE ALUMNO:

APARTADO B

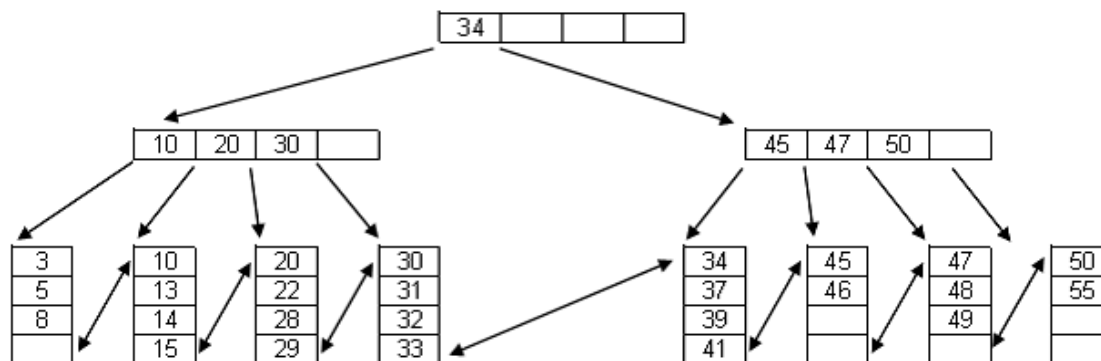
En la siguiente figura tenemos representado un árbol B+ de orden 2:



Mostrad cómo quedaría el árbol después de insertar el valor 46.

SOLUCIÓN:

Se ha de dividir la hoja, los primeros $d(2)$ valores (45 y 46) irán al nodo de la izquierda y el resto (47, 48 y 49) al de la derecha. También se debe poner un nuevo valor en el nodo padre (47).





NOMBRE ALUMNO:

EJERCICIO 5: preguntas teóricas

Contestad si cada una de las siguientes cuestiones es CIERTA o FALSA y argumentad vuestra respuesta (60 palabras como máximo).

- 1) Un inconveniente de una vista no materializada es que ocupa más recursos de espacio en disco que una vista materializada.
- 2) Cuando los datos que muestra una vista se actualizan muy a menudo en las tablas que los contienen y necesitamos verlos actualizados en todo momento, es recomendable que la vista sea materializada con la opción REFRESH ON COMMIT.
- 3) Las vistas, independientemente de si son materializadas o no, nos permiten hacer cálculos complejos sobre los datos de las tablas que consultan.
- 4) Nos proporcionan un valor que debemos buscar en una columna determinada de una tabla y, a partir de él, listar los siguientes n valores de la tabla, secuencialmente. En estos casos, un índice de tipo *hash* nos puede ser especialmente útil.
- 5) Los índices nos permiten un acceso muy rápido a los datos. Por tanto, deberíamos tener índices en todas las columnas de una tabla por si en algún momento tenemos que hacer búsquedas por aquellas columnas.

SOLUCIÓN:

- 5) FALSO. Es justo al contrario. Cuando una vista se materializa, se guardan los resultados en disco y por tanto ocupa un espacio que una vista no materializada no ocuparía.
- 6) FALSO. La materialización de la vista implica escribir el resultado en disco. Si los datos se actualizan muy a menudo, la opción REFRESH ON COMMIT hará que constantemente se estén escribiendo los resultados en disco, cosa que sería contraproducente e innecesaria. Sería mucho mejor tener una



NOMBRE ALUMNO:

vista no materializada y ejecutarla cada n segundos, en función de nuestras necesidades.

- 7) CIERTO. Cualquier vista nos permite efectuar este tipo de cálculos y de hecho es una de sus principales utilidades. El hecho de que sea materializada o no solo implica que los resultados se escriban o no en disco, respectivamente.
- 8) FALSO. Los índices de tipo *hash* nos permiten accesos por valor, pero no secuenciales. Sería mucho más eficiente un índice de tipo *B+*, ya que nos permiten los dos tipos de acceso que necesitamos para esta funcionalidad: acceder por valor al valor proporcionado y, a partir de él, acceder secuencialmente a los n siguientes valores.
- 9) FALSO. Aunque los índices son una herramienta muy potente para acceder a los datos, se debe tener cuidado en definir solo aquellos que realmente necesitamos, ya que tienen un coste. Los índices ocupan espacio en disco y se tienen que mantener (insertando/eliminando/moviendo valores). Por tanto, todos los índices que definamos y no sean realmente utilizados implican un sobre coste innecesario.



NOMBRE ALUMNO:

Recursos

Para solucionar esta PEC es necesario utilizar los contenidos de los módulos 1 a 5 del material docente.

Criterios de valoración

Se valoraran los ejercicios según los criterios anotados en la PEC1 y en las Prácticas, por lo que se deberán seguir las indicaciones se hicieron constar en las mismas, tales como; en el diseño conceptual es necesario anotar las PK en todas las entidades e indicar las cardinalidades, se debe utilizar el tipo Camel y Pascal en el diseño lógico relacional, se deben argumentar las respuestas al indicar las formas normales y al responder las preguntas teóricas, etcétera.

El peso de los ejercicios en la nota total de la PEC es el siguiente:

- Ejercicio 1: 25%
- Ejercicio 2: 20%
- Ejercicio 3: 15%
- Ejercicio 4: 20%
- Ejercicio 5: 20%

Esta PEC tiene un peso del 50% en la nota de evaluación continuada.

Formato y fecha de entrega

El formato del fichero tiene que ser Word u OpenOffice, y se tiene que entregar una versión del mismo fichero en PDF. Haced envíos independientes para la versión doc/odt y el fichero PDF.

El nombre del fichero tendrá el formato siguiente:

PEC2_Apellido1_Apellido2_Nombre.extensión

IMPORTANTE: El nombre y los apellidos del estudiante también tienen que aparecer en la portada del documento de la solución.

Es responsabilidad del estudiante asegurarse que los documentos entregados se han subido correctamente y corresponden a la actividad que hay que presentar.

La fecha límite para entregar la PEC 2 es el lunes 29 de mayo.



NOMBRE ALUMNO:

Nota: Propiedad intelectual

A menudo es inevitable, al producir una obra multimedia, hacer uso de recursos creados por terceras personas. Es por lo tanto comprensible hacerlo en el marco de una práctica de los estudios del Grado de Informática, siempre y cuando se documente claramente y no suponga plagio en la práctica.

Por lo tanto, al presentar una práctica que haga uso de recursos ajenos, se tiene que presentar junto con ella un documento en el que se detallen todos ellos, especificando el nombre de cada recurso, su autor, el sitio de donde se obtuvo y su estatus legal: si la obra está protegida por el copyright o se acoge a alguna otra licencia de uso (Creative Commons, licencia GNU, GPL...). El estudiante tendrá que asegurarse que la licencia que sea no impide específicamente su uso en el marco de la práctica. En caso de no encontrar la información correspondiente tendrá que asumir que la obra está protegida por el copyright.

Además, de deberán adjuntar los ficheros originales cuando las obras utilizadas sean digitales, y su código fuente si corresponde.

Otro punto a considerar es que cualquier práctica que haga uso de recursos protegidos por el copyright no podrá en ningún caso publicarse en Mosaico, la revista del Graduado en Multimedia a la UOC, a no ser que los propietarios de los derechos intelectuales den su autorización explícita.