

Conceptos básicos y modelos NoSQL

Propuesta de solución PEC1

Ejercicio 1 (20%)

A partir de la lectura de los apuntes (transcripciones de los vídeos) de los primeros temas del curso se pide responder con tus propias palabras a las siguientes preguntas de manera concisa (una página y media en total), **justificando la respuesta** mediante lo que has leído en los materiales. En **cada justificación deberá indicarse la cita de los apuntes, vídeo o libros en la que se sustenta**. No se considerarán válidas las respuestas que no se justifiquen o no estén convenientemente citadas.

1. Explica las implicaciones de que el esquema sea *schemaless* en las bases de datos NoSQL con respecto a la independencia de los datos
2. Explica qué limitaciones presenta la estrategia *one size fits all* respecto: 1) a las consultas, 2) las propiedades ACID y 3) el mantenimiento del esquema de la base de datos.
3. Explica 3 situaciones genéricas dónde es más útil el modelo de datos en grafo que el modelo relacional.
4. Explica 3 situaciones donde el uso del modelo de agregación es ventajoso respecto a otras alternativas.

Los criterios de evaluación para evaluar este ejercicio son los siguientes:

No logrado (C-)	Mínimamente logrado (C+)	Logrado (B)	Logrado de forma sobresaliente (A)
No se responde a la pregunta planteada o la respuesta es incompleta.	La respuesta es correcta y es mayoritariamente completa, pero no está justificada o la justificación no es coherente con los contenidos vistos en clase. Indica las referencias de los apuntes en los que se basa.	La respuesta es correcta, completa y está correctamente justificada. Indica las referencias de los apuntes en los que se basa.	La respuesta es correcta, está correctamente justificada y cumple los requisitos formales (concreción - ocupa menos de una página y media). La argumentación es excelente, aborda los aspectos relevantes e indica las referencias de los apuntes en los que se basa.

Solución

1. Explica las implicaciones de que el esquema sea *schemaless* en las bases de datos NoSQL con respecto a la independencia de los datos

Tal y como se indica en las páginas 9 y 11 del documento B1_T1_IntroducciónNoSQL, la característica "schemaless" que presentan las bases de datos NoSQL implica que la interpretación de los datos se realiza explícitamente en el código de los programas que acceden a la base de datos, lo cual compromete la independencia de los datos, porque su estructura y organización están estrechamente ligadas a la implementación específica de la aplicación que accede a ellos. Si se cambia la forma en que se interpreta el esquema en el código de la aplicación, podría afectar la manera en que se acceden y manipulan los datos, lo que podría dificultar la interoperabilidad entre diferentes sistemas o versiones de la misma aplicación. De esta forma se compromete la independencia de los datos, lo que puede llevar a problemas de mantenimiento y gestión a medida que evolucionan las necesidades del sistema y se agregan nuevos tipos de datos.

2. Explica qué limitaciones presenta la estrategia *one size fits all* respecto: 1) a las consultas, 2) las propiedades ACID y 3) el mantenimiento del esquema de la base de datos.

Tal y como se indica en la página 11 del documento B1_T2_PersistenciaPoliglota. La estrategia "one size fits all" presenta limitaciones en los aspectos indicados. Respecto a las consultas, debido a su orientación hacia datos estructurados, se dificulta la manipulación de datos semi-estructurados o complejos, limitando así la flexibilidad y la capacidad de análisis de la información. En segundo lugar, las propiedades ACID asociadas con este enfoque afectan al rendimiento en entornos donde la velocidad y la disponibilidad son críticas, ya que garantizar estas propiedades puede requerir recursos adicionales y afectar la escalabilidad. Por último, el mantenimiento del esquema de la base de datos puede ser complicado y lento de adaptar a las necesidades cambiantes del negocio o del mercado, lo que limita la capacidad de la estrategia "one size fits all" para mantenerse ágil y responder eficazmente a las demandas emergentes.

3. Explica 3 situaciones genéricas donde es más útil el modelo de datos en grafo que el modelo relacional.

Tal y como se indica en la página 8 del documento B2_T4_1_ModelosEnGrafo, el modelo de datos en grafo supera al modelo relacional en escenarios con múltiples relaciones interconectadas, donde el enfoque de grafo facilita la representación eficiente de estas complejas interrelaciones. Además, cuando la importancia principal

recae en las conexiones y relaciones entre los datos, el modelo de grafo destaca al enfocarse en estas interacciones directas, a diferencia del modelo relacional que se centra más en la información individual y sus relaciones a través de claves foráneas. Por último, en ámbitos específicos como redes sociales, logística, análisis geoespacial, aplicaciones semánticas y biología computacional, donde las estructuras relacionales son fundamentales para el análisis y la toma de decisiones, el modelo de datos en grafo sobresale por su capacidad para representar y manejar eficazmente estas complejas relaciones.

4. Explica 4 situaciones donde el uso del modelo de agregación es ventajoso respecto a otras alternativas.

Tal y como se indica en la página 10 del documento B2_T3_2_ModelosAgregaciónCaracterísticas, en primer lugar el modelo de agregación es una opción ventajosa cuando las funcionalidades de la aplicación están claramente definidas y estables, evitando así la necesidad de cambios drásticos en el diseño de los agregados. Además, su utilidad se destaca cuando las funcionalidades presentan poco solapamiento en cuanto a necesidades de información, lo que permite minimizar repeticiones de datos entre agregados de diferente tipo. Asimismo, el modelo resulta idóneo en escenarios donde no existen interrelaciones complejas en el dominio de la aplicación, simplificando la estructura de la base de datos. Por último, su eficacia se manifiesta cuando los datos experimentan pocos cambios, facilitando operaciones de inserción y consulta sin necesidad de actualizaciones frecuentes de los agregados, lo que conlleva a un rendimiento más predecible y consistente en la aplicación.

Ejercicio 2 (20%)

A partir de la lectura de los apuntes de los temas I y II indica si te parecen ciertas o falsas las siguientes afirmaciones.

Para cada una de las afirmaciones indica si es cierta o falsa, **justificando la respuesta** mediante lo que has leído en los materiales. En **cada justificación deberá indicarse la cita de los apuntes, vídeo o libros en la que se sustenta. No serán válidas las respuestas que no se justifiquen o no tengan cita**. Se valorará la concisión (una página y media para las 5 afirmaciones como máximo).

Afirmación 1

Las bases de datos NoSQL tienen un esquema bien definido y estructurado, lo que facilita la gestión de la base de datos y la implementación de medidas de seguridad.

Afirmación 2

Las propiedades ACID encajan mejor con la estrategia PUSH.

Afirmación 3

En el diseño de bases de datos en grafo, no es necesario identificar las relaciones entre los datos ni asignar nombres a ellas, ya que el sistema puede inferir automáticamente la estructura de la base de datos a partir de los datos ingresados inicialmente.

Afirmación 4

Los modelos de agregación garantizan la integridad de los datos. Es decir, si tenemos los mismos datos (por ejemplo los datos de un cliente) en distintos agregados de la base de datos, el sistema gestor de bases de datos se encargará de garantizar que las distintas ocurrencias de los mismos datos (del mismo cliente) siempre tengan sus valores actualizados y consistentes.

Afirmación 5

Algunas bases de datos basadas en el modelo clave-valor ofrecen flexibilidad al permitir definir una estructura mínima para los agregados y en algunos casos permiten establecer relaciones entre ellos, lo que lo hace adecuado para una variedad de casos de uso en diferentes dominios.

Los criterios de evaluación para evaluar este ejercicio son los siguientes:

No logrado (C-)	Mínimamente logrado (C+)	Logrado (B)	Logrado de forma sobresaliente (A)
La respuesta es incorrecta o no está justificada/referenciada.	La respuesta es incorrecta, pero la justificación es correcta, demuestra conocimiento sobre el tema y está referenciada.	La respuesta es correcta y está convenientemente justificada/referenciada.	La respuesta es correcta, está correctamente justificada/referenciada y es concisa (ocupa menos de una página y media).

Solución

Afirmación 1

La afirmación es **falsa**. Tal y como se indica en la página 10 del documento B1_T2_PersistenciaPoliglota: “El esquema de una base de datos NoSQL puede ser complejo y, en general, no está definido. Un sistema gestor de bases de datos relacional guarda información sobre el esquema de la base de datos. En el caso de una base de datos NoSQL el esquema está, casi siempre, gestionado por las

aplicaciones. Este hecho puede dificultar su modificación y la gestión de otros aspectos, por ejemplo, los ligados a la seguridad (derechos de acceso de los usuarios a la base de datos).

Afirmación 2

La afirmación es **falsa**, tal y como se indica en la página 15 del documento B1_T2_PersistenciaPoliglota: “Para finalizar, cabe recordar que en los sistemas que trabajan con flujos de datos, la eficiencia y el rendimiento priman sobre la consistencia, por lo que no es necesario cumplir con las propiedades ACID”

Afirmación 3

La afirmación es **falsa**, tal y como se indica en la página 24 del documento B2_T4_1_ModelosEnGrafo: “Al no tener un esquema explícito, el diseño de la base de datos es mucho más sencillo. No es necesario definir un modelo conceptual del dominio de aplicación, sino identificar los conceptos más importantes del mismo. Una vez hecho, es importante identificar las distintas relaciones que pueden existir entre los datos. Una vez identificadas, se debe establecer qué nombre se utilizará para cada relación. Es importante no utilizar distintos nombres para una misma relación ya que eso dificultaría enormemente la consulta de los datos. Recordad que el nombre asociado a una relación indica su tipo”.

Afirmación 4

La afirmación es **falsa** tal y como se indica en la página 5 del documento B2_T3_2_ModelosAgregaciónCaracterísticas: “Los modelos de agregación pueden introducir redundancias en los datos. Esto implica que datos referidos a unos mismos objetos del mundo real pueden quedar repetidos en diferentes agregados, como consecuencia de una decisión de diseño... Si ante cambios en esos objetos, el dominio de aplicación requiere que se mantenga la coherencia en el valor de los objetos, garantizar esa coherencia es responsabilidad del programa (y en consecuencia de los personas que intervienen en su creación) y no del sistema gestor de la base de datos”

Afirmación 5

La afirmación es **cierta**, tal y como se indica en la página 4 del documento B2_T3_3_ModelosAgregaciónTipos: “El sistema gestor de la base de datos desconoce la estructura interna asociada al agregado (el elemento valor del par (clave, valor)). En definitiva, el valor asociado al agregado es almacenado como un blob (binary large object) en la base de datos. Esto no significa necesariamente que el agregado no tenga estructura, sino que ésta sólo será comprendida por los programas que manipulan los agregados. A pesar de ello, existen bases de datos clave-valor, como es el caso de Riak, que permiten definir una estructuración mínima del agregado y relaciones entre agregados”.

Ejercicio 3 (30%)

Considerar una base de datos de agregación documental de una empresa de ventas de vino donde se almacena información sobre los vinos vendidos, los clientes que han realizado los pedidos y los ingresos de las bodegas. Se desea diseñar los tipos de documentos más eficientes para las siguientes dos consultas:

Consulta 1: Para cada mes, año y tipo de vino (tinto, blanco, rosado), se desea obtener el número total de envíos realizados de ese tipo de vino, el volumen promedio de cada envío de ese tipo de vino y una lista detallada de los envíos agrupados por región de origen del vino. Para cada región de origen, se quiere conocer el nombre de la región, el número de envíos que contienen vinos de esa región, el volumen total de vino enviado desde esa región y una lista detallada de cada envío desde esa región. Para cada envío, se requiere el número de seguimiento, la fecha de envío, el destino del envío, el volumen del paquete de vino y el precio unitario del vino.

Consulta 2: A nivel anual (año de venta), para cada bodega de vinos y cada añada (año de la cosecha) vendida, se desea obtener el total de ingresos generados por la venta de vinos de esa bodega en esa añada en el año de venta considerado, y para cada tipo de vino vendido mostrar el nombre del tipo de vino, una lista con los nombres de los clientes que compraron ese vino y el número de botellas compradas por cliente, la puntuación media otorgada por los críticos y el precio unitario por botella.

Cada consulta requiere un tipo concreto de documento. Para cada pregunta, se pide indicar:

- Una representación gráfica del documento propuesto utilizando un diagrama de cajas anidadas como el que se explica en los apuntes sobre "Diseño de agregados" proporcionado en el aula virtual.
- Una breve explicación de la estructura del documento y una justificación del porqué de la estructura propuesta.
- Un documento JSON que ejemplifique el resultado de una potencial consulta a cada agregado.

Los criterios de evaluación para evaluar este ejercicio son los siguientes:

No logrado (C-)	Mínimamente logrado (C+)	Logrado (B)	Logrado de forma sobresaliente (A)
El agregado propuesto es incorrecto. Entendemos que un agregado no es correcto cuando no permite almacenar la información solicitada.	El agregado propuesto es correcto pero la justificación de su estructura es insuficiente y no existe un ejemplo de documento JSON.	El agregado propuesto es correcto, su estructura está correctamente justificada y se ejemplifica mediante un documento JSON.	El agregado propuesto es correcto, tiene una estructura eficiente y está correctamente justificado y ejemplificado.

Solución

Consulta 1

De acuerdo a la descripción de la consulta, para poder realizarla eficientemente se propone un tipo de documento estructurado en tres niveles de anidamiento. En el nivel superior, se encuentra la información correspondiente al mes y año para los cuales se proporciona la información. A continuación se presenta información específica para cada tipo de vino: el número total de envíos realizados, el volumen promedio de cada envío e información sobre los envíos por cada región de origen del vino. Para cada región de origen, se detalla el nombre de la región, el número de envíos que contienen vinos de esa región, el volumen total de vino enviado desde esa región, y una lista con información específica sobre cada envío: el número de seguimiento, la fecha de envío, el destino del envío, el volumen del paquete de vino y el precio unitario del vino. El agregado propuesto corresponde al 100% a la petición realizada y por tanto proporcionará toda la información relevante en una sola lectura de la base de datos. En el siguiente diagrama se muestra el diseño planteado:

Mes:

Año:

Tipo_de_Vino:

Tipo:

Numero_total_envios:

Volumen_promedio_envio:

Envios_por_region_origen:

Nombre_region:

Numero_envios:

Volumen_total_enviado:

Detalle_envios:

Numero_seguimiento:

Fecha_envio:

Destino_envio:

Volumen_paquete_vino:

Precio_unitario_vino:

.....

Numero_seguimiento:

Fecha_envio:

Destino_envio:

Volumen_paquete_vino:

Precio_unitario_vino:

.....

.....

En formato JSON, un ejemplo de la consulta 1 quedaría de la siguiente manera:

```
{
  "mes": "enero",
  "año": 2024,
  "tipos_de_vino": [
    {
      "tipo": "tinto",
```



```

    "numero_total_envios": 50,
    "volumen_promedio_envio": 750,
    "envios_por_region_origen": [
    {
    "nombre_region": "Rioja",
    "numero_envios": 20,
    "volumen_total_enviado": 15000,
    "detalle_envios": [
        {
            "numero_seguimiento": "ABC123",
            "fecha_envio": "2024-01-10",
            "destino_envio": "Madrid",
            "volumen_paquete_vino": 750,
            "precio_unitario_vino": 15
        },
        // Otros detalles de envíos...
    ]
    },
    // Otros regiones de origen...
    ]
    },
    // Otros tipos de vino...
    ]
}

```

Consulta 2

De acuerdo a la descripción de la consulta, para poder realizarla eficientemente se propone un tipo de documento que tendrá como campos más externos el número del año y un campo bodegas que contendrá una lista de subdocumentos asociados a cada bodega. Dentro de cada subdocumento, se detallan el nombre de la bodega y una lista con información de las añadas: año, total de ingresos generados por las ventas de esa bodega en esa añada y una lista con información de los vinos vendidos de ese añada en ese año de venta. Para cada tipo de vino se incluye la información acerca del nombre del tipo de vino, una lista con los nombre de los clientes que compraron ese vino y el número de botellas compradas por cliente, la puntuación media otorgada por los críticos y el precio unitario por botella. Dicho agregado corresponde al 100% a la petición realizada y por tanto proporcionará toda la información relevante en una sola lectura de la base de datos.

En el siguiente diagrama se muestra el diseño planteado:



En formato JSON, un ejemplo de la consulta 3 quedaría de la siguiente manera:

```
{
  "año": 2023,
  "bodegas": [
    {
      "nombre": "Bodega Pedro Lorenzo",
      "añadas": [
        {
          "año": 2019,
          "ingresos_totales": 7500,
          "tipo_vinos": [
            {
```

```

    "nombre_vino": "Tinto",
    "clientes": [
        {"nombre": "Ana García", "número":20},
        {"nombre":"Carrefour","número":500}
    ],
    "precio_unitario_vino": 25,
    "puntuacion_media_criticos": 92
},
// Otras tipos de vino...
]
},
// Otras añadas de la bodega...
]
},
// Otras bodegas...
]
}

```

Ejercicio 4 (30%)

Se desea desarrollar un sistema para el monitoreo y gestión de la producción agrícola sostenible en respuesta a las estrictas normativas europeas. Para ello, el sistema dispone de un conjunto de sensores distribuidos en los campos para recopilar datos sobre parámetros clave como humedad del suelo, temperatura, calidad del agua y presencia de plagas. Aunque los datos se puedan almacenar de forma distribuida, serán procesados en tiempo real por una plataforma de análisis, que emplea algoritmos avanzados para identificar patrones y tendencias, permitiendo así la toma de decisiones informadas para optimizar las prácticas agrícolas. Observar que al haber tantos datos por sensor (lecturas por segundo como mínimo), leer datos desactualizados no tiene un impacto importante en las inferencias del sistema y, por lo tanto, la consistencia no es un factor relevante en este caso. Por otra parte, dado que los sensores pueden tener distintas estructuras de datos según el fabricante, el sistema se adapta para integrar y procesar esta información de manera coherente. Además, se implementan medidas específicas según las necesidades de cada zona agrícola, lo que permite aplicar estrategias personalizadas de riego, fertilización y control de plagas, contribuyendo así a una producción agrícola más eficiente y sostenible.

En primer lugar, se pide realizar la lectura del artículo: “Data management in cloud environments: NoSQL and NewSQL data stores” que se puede encontrar en la siguiente dirección web: <https://journalofcloudcomputing.springeropen.com/articles/10.1186/2192-113X-2-22>

A continuación, tomando como base la lectura del artículo, así como los apuntes de los temas vistos, se pide argumentar brevemente y de forma razonada la certeza o falsedad de las siguientes afirmaciones:

1. Una base de datos relacional es la mejor solución para las necesidades de almacenamiento y procesamiento descritas en el supuesto.
2. La utilización de un esquema explícito para los datos a almacenar sería mejor que usar un esquema implícito. Además, en caso de no usar un modelo relacional, la mejor solución sería utilizar una base de datos NewSQL.
3. De acuerdo a los requisitos planteados, la solución debe ser fácilmente escalable en un ambiente distribuido.
4. Respecto a la replicación, los datos de interés no permiten una fragmentación ni una replicación eficiente. Por lo tanto, sería mejor que no haya réplicas y exista una única copia en un ambiente centralizado.
5. Si se considera que la mejor solución es una base de datos NoSQL, entonces el modelo óptimo sería el orientado a grafos.

Los criterios de evaluación para evaluar este ejercicio son los siguientes:

Pregunta	No logrado (C-)	Mínimamente logrado (C+)	Logrado (B)	Logrado de forma sobresaliente (A)
Todas	La respuesta es incorrecta.	La respuesta es correcta pero no se justifica o se justifica incorrectamente.	La respuesta es correcta, pero la justificación no está suficientemente argumentada.	La respuesta es correcta y se justifica suficientemente con argumentos.

Solución

1. Una base de datos relacional es la mejor solución para las necesidades de almacenamiento y procesamiento descritas en el supuesto.

Es falso. Una base de datos relacional no es la mejor opción debido a varios factores. Primero, la estructura de los datos recopilados de los sensores distribuidos en los campos puede ser heterogénea debido a las diferentes estructuras de datos de los sensores de diferentes fabricantes. Esto hace que sea más difícil representar eficientemente estos datos en un modelo relacional, que generalmente requiere una estructura de tabla predefinida y uniforme. Segundo, el procesamiento en tiempo real y el análisis de grandes volúmenes de datos para identificar patrones y tendencias pueden ser más eficientes con tecnologías de bases de datos no relacionales, como bases de datos NoSQL, que pueden manejar datos no estructurados o semiestructurados de manera más flexible y escalable. Finalmente, la necesidad de implementar medidas específicas según las necesidades de cada zona agrícola, como estrategias personalizadas de riego, fertilización y control de plagas, podría requerir una capacidad de consulta y actualización más dinámica y adaptable que la que ofrece una base de datos relacional.

2. La utilización de un esquema explícito para los datos a almacenar sería mejor que usar un esquema implícito. Además, en caso de no usar un modelo relacional, la mejor solución sería utilizar una base de datos NewSQL.

Es falso. La utilización de un esquema explícito puede ser útil en ciertos casos, ya que proporciona una estructura clara y definida para los datos, lo que facilita su comprensión y gestión. Sin embargo, en situaciones donde los datos son heterogéneos o cambian con frecuencia, un esquema explícito puede resultar restrictivo y difícil de mantener. Además, el sistema requiere flexibilidad en el esquema de datos dado que en la estructura de los datos recopilados de los sensores agrícolas (los atributos y la forma de los datos pueden variar ampliamente entre diferentes tipos de sensores o incluso con el tiempo). En este sentido, los datos recopilados de los sensores agrícolas podrían representarse fácilmente como documentos JSON u otros formatos similares, donde cada documento puede contener una combinación única de atributos según el tipo de sensor. Una base de datos

NoSQL proporciona la flexibilidad necesaria para manejar estos datos de manera eficiente y escalable (permiten almacenar datos sin un esquema fijo predefinido, lo que facilita la incorporación de nuevos tipos de datos y cambios en la estructura de los mismos), siendo más beneficioso que intentar adaptarlos a un esquema relacional o NewSQL.

3. De acuerdo a los requisitos planteados, la solución debe ser fácilmente escalable en un ambiente distribuido.

Es cierto. Dado que el sistema debe manejar un conjunto de sensores distribuidos en campos agrícolas y procesar datos en tiempo real, es fundamental que la solución sea fácilmente escalable en un entorno distribuido. La distribución de sensores implica una gran cantidad de datos que deben ser recopilados y procesados de manera eficiente, y la capacidad de escalar el sistema para manejar un mayor volumen de datos y una mayor carga de trabajo es esencial para garantizar un rendimiento óptimo. Además, la escalabilidad también es importante para adaptarse a futuros crecimientos en la infraestructura y en la cantidad de datos generados por los sensores. Por lo tanto, la solución debe estar diseñada para escalar horizontalmente y distribuir la carga de manera equitativa entre los diferentes componentes del sistema.

4. Respecto a la distribución y replicación, lo mejor es que la información se encuentre centralizada en un único lugar, y no haya réplicas de la misma pues la disponibilidad no es un factor importante.

Es falso. En el contexto de un sistema de monitoreo y gestión de la producción agrícola sostenible, donde la información proviene de sensores distribuidos en diferentes ubicaciones, la distribución y replicación de datos son aspectos críticos para garantizar la disponibilidad, la tolerancia a fallos y el rendimiento del sistema. Al tener una única ubicación centralizada para toda la información, se aumenta el riesgo de puntos únicos de fallo y se reduce la disponibilidad en caso de problemas en esa ubicación central. La distribución y replicación de datos permiten mejorar la disponibilidad al tener múltiples copias de la información en diferentes ubicaciones, lo que garantiza que el sistema siga funcionando incluso si una ubicación o servidor falla. Además, la replicación de datos puede mejorar el rendimiento al permitir el acceso a datos locales en lugar de depender de una conexión de red para acceder a un servidor centralizado. Por lo tanto, en este contexto, es beneficioso distribuir y replicar la información en lugar de mantenerla centralizada en un único lugar.

5. Si se considera que la mejor solución es una base de datos NoSQL, entonces el modelo óptimo sería el orientado a grafos.

Es falso. La elección del modelo de base de datos NoSQL depende de los requisitos específicos del sistema y de la naturaleza de los datos que se manejan. En el caso de un sistema de monitoreo y gestión de la producción agrícola sostenible, donde se recopilan datos de sensores distribuidos y se necesita un procesamiento en tiempo real, es más probable que un modelo de datos basado en documentos o columnas sea más adecuado que un modelo orientado a grafos. Los

modelos de documentos y columnas son más flexibles y escalables para manejar grandes volúmenes de datos no estructurados o semiestructurados, como los datos recopilados de sensores agrícolas. Además, estos modelos permiten realizar consultas eficientes y escalables sobre grandes conjuntos de datos, lo que es crucial para analizar y tomar decisiones basadas en los datos recopilados. Además, algunos de estos modelos están orientados para proveer una alta eficiencia en la escritura de datos, como por ejemplo Cassandra - como veremos más adelante en la asignatura-. Por lo tanto, en este contexto, un modelo de datos orientado a documentos o columnas sería más apropiado que un modelo orientado a grafos.

Criterios de valoración

Los ejercicios tienen un peso de 20%, 20%, 30%, 30% respectivamente. Se valorará, para cada apartado, la validez de la solución y la claridad de la argumentación de acuerdo a la rúbrica facilitada en cada ejercicio. Muchos de los ejercicios planteados no tienen una única respuesta, por lo que es importante justificar la respuesta propuesta adecuadamente. Cualquier solución no justificada se considerará incompleta.

Formato y fecha de entrega

Tenéis que enviar la PEC a través del enlace “Entrega de la PEC1” que se encuentra en el apartado Actividades disponible en el aula. El formato del archivo que contiene vuestra solución puede ser .pdf, .odt, .doc y .docx. Para otras opciones, por favor, contactar previamente con vuestro profesor colaborador. El nombre del fichero debe contener el código de la asignatura, vuestro apellido y vuestro nombre, así como el número de actividad (PEC1). Por ejemplo apellido1_nombre_nosql_pec1.docx. La fecha límite para entregar la PEC1 es el **2 de abril**.

Propiedad intelectual

Al presentar una práctica o PEC que haga uso de recursos ajenos, se tiene que presentar junto con ella un documento en que se detallen todos ellos, especificando el nombre de cada recurso, su autor, el lugar 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 etc.). El estudiante tendrá que asegurarse que la licencia que sea no impide específicamente su uso en el marco de la práctica o PEC. En caso de no encontrar la información correspondiente tendrá que asumir que la obra está protegida por el copyright.

Será necesario, además, adjuntar los ficheros originales cuando las obras utilizadas sean digitales, y su código fuente, si así corresponde.

