

# Distribución de datos y *Map Reduce*

## PEC2

### Ejercicio 1 (25%)

*Healthy Soils* es una empresa multinacional especializada en la optimización de explotaciones agrarias a través de Internet of Things (IoT). Entre las múltiples soluciones que ofrece a sus clientes destaca un dispositivo, alimentado por una batería propia que, montado sobre una placa Arduino Feather M0 equipada con una tarjeta de comunicaciones LoRaWAN (un protocolo de transmisión de datos asíncrono), que incluye los siguientes sensores:

- GS3: este sensor, que se introduce bajo tierra, ofrece una medida de la temperatura, otra de la conductividad eléctrica, y otra de la permeabilidad.
- SHT31: este sensor obtiene la humedad y la temperatura del aire.
- TSL2591: este sensor ofrece tres medidas independientes sobre el grado de luminosidad: la infrarroja, la del espectro entero (full-spectrum) y de la luz visible por el ojo humano.

Se debe tener en cuenta que en el futuro es posible que se incorporen nuevos tipos de sensores a la placa, incrementando el número y variedad de los datos generados. Se desea que esos datos también sean incorporados a la misma base de datos.

*Healthy Soils* se ha propuesto implantar este dispositivo en miles de explotaciones por todo el mundo. Cada placa enviará los datos de sus sensores conjuntamente, en intervalos de 10 minutos.

Además, para la explotación de los datos almacenados, la empresa creará una API que será consumida por una aplicación web que muestre datos y gráficas de la evolución temporal de los parámetros medidos por los sensores. Esta API también será expuesta, bajo autenticación y autorización, a cada uno de los clientes contratados que deseen utilizarla para visualizar y explotar sus propios datos, por lo que se espera que el número de consultas sea enorme y que éstas se produzcan de forma concurrente desde cualquier parte del mundo.

#### Se pide:

Estudiar el caso propuesto y responder a las siguientes preguntas:

1. Razonar cuál sería el modelo de almacenamiento de datos más adecuado, mencionando sus ventajas sobre el resto de modelos tratados en el curso.

Para el caso de Healthy Soils, es importante considerar que se manejará una gran cantidad de datos de sensores IoT no estructurados y heterogéneos, los cuales se enviarán en intervalos de 10 minutos y pueden incluir una variedad de tipos de sensores. Además, se espera que los clientes utilicen una API para acceder a sus propios datos de manera concurrente desde cualquier parte del mundo. Por lo tanto, se requiere una base de datos escalable y robusta capaz de manejar grandes volúmenes de datos en tiempo real y que soporte una elevada concurrencia de acceso a los datos.

Los requisitos anteriores hacen que una base de datos relacional no sea adecuada debido a su ineficiencia para manejar grandes cantidades de datos y por el alto costo que supondría añadir nuevos sensores debido a la rigidez de la estructura de datos. Además, las operaciones de lectura y escritura en una base de datos relacional pueden generar bloqueos y cuellos de botella que afecten negativamente a la escalabilidad y rendimiento de la aplicación, especialmente cuando se espera un alto nivel de concurrencia de los datos como en el caso de Healthy Soils.

Por lo tanto, es recomendable utilizar bases de datos NoSQL. El uso de un modelo de grafos no es necesario ya que la relación entre los datos no es de gran importancia. Además, el uso de un modelo clave-valor no facilitaría el filtrado o la separación por valores de ciertos campos. Por ello, se sugiere que el modelo de datos más adecuado para Healthy Soils sería un modelo orientado a documentos, como puede ser MongoDB. Este tipo de modelo permite una estructura de datos flexible y escalable, lo que facilita la adaptación a la variedad de tipos de sensores que se puedan incorporar en el futuro. Además, su capacidad para distribuir datos horizontalmente y tolerar fallos lo hace ideal para una empresa que espera manejar una gran cantidad de datos de sensores en tiempo real.

Finalmente, la API que se creará para los clientes requiere de una alta disponibilidad y una respuesta rápida a las consultas. Un modelo orientado a documentos como MongoDB también es adecuado en este caso, ya que puede soportar niveles elevados de concurrencia, permitiendo que los clientes puedan acceder y analizar sus datos en tiempo real.

## 2. Explicar qué estrategia de fragmentación sería la óptima.

La fragmentación horizontal permitiría mejorar la disponibilidad del sistema al distribuir los datos en varios nodos, lo que aseguraría que los datos estén siempre disponibles incluso si uno de los nodos falla. Además, permitiría una fácil escalabilidad al agregar más nodos según sea necesario. La distribución de carga también se mejoraría con la fragmentación horizontal ya que cada nodo tendría una carga de procesamiento y almacenamiento similar (consistent hashing), evitando cuellos de botella en el sistema, por ejemplo, se podría distribuir los datos usando como clave la ubicación geográfica de los sensores. Por último, la localidad de los datos se mejoraría ya que cada fragmento de datos estaría almacenado en el nodo más cercano a su origen, lo que minimizaría la latencia y mejoraría el rendimiento del sistema.

### 3. Explicar cuál sería la mejor estrategia de replicación de datos.

En el caso de Healthy Soils, la mejor estrategia de replicación de datos sería replicar los datos en nodos geográficamente cercanos con el objetivo de garantizar la disponibilidad del servicio en caso de caídas de los nodos. Esto se debe tener en cuenta en conjunto con la arquitectura de fragmentación geolocalizada propuesta para la distribución de carga. Al replicar los datos en nodos cercanos, se minimiza el tráfico de datos en la red debido a la proximidad geográfica de los nodos.

En cuanto al tipo de replicación, se propone utilizar un esquema asíncrono de tipo maestro-esclavo, ya que proporciona un alto rendimiento en escenarios con un alto volumen de solicitudes de lectura y escritura a la base de datos. Esto se debe a que el nodo maestro es responsable de recibir y procesar las solicitudes de escritura, mientras que los nodos esclavos se encargan de replicar los datos del nodo maestro en tiempo real. Con esta estrategia de replicación, se garantiza la escalabilidad de la base de datos y se minimizan los cuellos de botella y la latencia en el acceso a los datos.

### 4. El modelo transaccional más adecuado para los requisitos de la aplicación descrita.

Para la aplicación de Healthy Soils se propone utilizar un modelo transaccional BASE, el cual prioriza la disponibilidad (A) de los datos por encima de la consistencia (C) de los mismos. Dado que la aplicación requiere alta disponibilidad y velocidad de acceso a los datos, esta elección se ajusta bien a sus requerimientos. Con este modelo, es posible que haya algunas pérdidas temporales en la consistencia de la información, lo que podría impedir la visualización de algunos valores debidamente actualizados o hacer que se muestren posteriormente. En resumen, el modelo BASE es el más adecuado para la aplicación Healthy Soils debido a su alta disponibilidad y velocidad de acceso, lo que permitiría una experiencia de usuario óptima.

Pregunta	No logrado (D/C-)	Mínimamente logrado (C+)	Logrado (B)	Logrado de forma sobresaliente (A)
<b>Pregunta 1</b>	La respuesta	La respuesta es	La respuesta es	La respuesta es

	es incorrecta o no está justificada.	correcta y está justificada, pero sólo analiza el modelo de datos ganador.	correcta, analiza todos los modelos de datos y está justificada. La justificación en algún caso es incompleta (obvia algunos aspectos relevantes) o discutible.	correcta e indica, por cada tipo de base de datos, su adecuación y su justificación. Los argumentos planteados son completos y adecuados.
<b>Pregunta 2</b>	La respuesta es incorrecta o no está justificada.	La respuesta es correcta y está bien justificada en la mayoría de las características del sistema (disponibilidad, escalabilidad, distribución de carga y localidad de datos).	La respuesta es correcta y está bien justificada en casi todas las características del sistema (disponibilidad, escalabilidad, distribución de carga y localidad de datos).	La respuesta es correcta y está bien justificada para todas las características del sistema (disponibilidad, escalabilidad, distribución de carga y localidad de datos).
<b>Pregunta 3</b>	La respuesta es incorrecta o no está justificada.	La respuesta es correcta y está mínimamente justificada.	La respuesta es correcta, está bien justificada y aborda aspectos geográficos y de localidad de datos.	La respuesta es correcta, está justificada y aborda aspectos geográficos, de localidad de datos y propone una arquitectura de gestión de réplicas adecuada.
<b>Pregunta 4</b>	La respuesta es incorrecta o no está justificada.	La respuesta es correcta y está mínimamente justificada.	La respuesta es correcta, considera los distintos modelos disponibles y está bien justificada.	La respuesta es correcta, considera los distintos modelos disponibles, está bien justificada y justifica de acuerdo al enunciado la decisión tomada (consistencia vs disponibilidad).

## Ejercicio 2 (25%)

A partir de la lectura de los apuntes del curso expone, para cada una de las afirmaciones, si creéis que es cierta o falsa indicando una breve argumentación que justifique vuestra respuesta.

**Las afirmaciones en las que no se indique si es cierta o falsa, o bien carezcan de argumentación, se considerarán no válidas.** Se **valorará la concisión** (una página y media para las 5 afirmaciones como máximo) y el **uso de referencias** (sección/página del libro de referencia o apuntes del curso) para justificar las respuestas.

### *Afirmación 1*

La fragmentación en una base de datos relacional implica que un conjunto de datos sólo esté disponible en un fragmento y, por lo tanto, el nodo que almacene ese fragmento será el único responsable de sus datos.

En una base de datos relacional fragmentada, los datos se dividen en fragmentos y se distribuyen en diferentes nodos. Sin embargo, cada nodo no es el único responsable de los datos que almacena. En cambio, los fragmentos se replican en múltiples nodos para garantizar la disponibilidad y la redundancia de los datos. Por lo tanto, aunque un nodo pueda ser el principal responsable del almacenamiento y el mantenimiento de un fragmento específico, no es el único nodo que almacena esa información, y otros nodos también pueden tener copias de ese fragmento en caso de que ocurra una falla del nodo principal. En resumen, la fragmentación en una base de datos relacional no implica que un conjunto de datos esté disponible en un solo fragmento y que un solo nodo sea el único responsable de sus datos.

[BBDD Diseño pagina 7]

### *Afirmación 2*

La principal ventaja del modelo de replicación maestro-esclavo es la consistencia. Nunca es posible que diferentes clientes lean diferentes nodos secundarios y los datos sean diferentes.

Una de las principales ventajas del modelo de replicación maestro-esclavo es la consistencia de los datos. Esto se debe a que todos los datos se escriben primero en el nodo maestro y luego se replican a los nodos esclavos, asegurando así que todos los nodos tengan una copia idéntica de los datos en todo momento. Además, al tener un nodo maestro que es responsable de todas las escrituras de datos, se evita la posibilidad de conflictos de escritura que podrían surgir en un modelo de replicación entre pares (peer-to-peer).

En teoría, al usar el modelo de replicación maestro-esclavo, al realizar la misma consulta en diferentes nodos esclavos, se debería obtener el mismo

resultado ya que los datos en los nodos esclavos son una copia exacta de los datos en el nodo maestro. Sin embargo, en la práctica, puede haber situaciones en las que los datos no estén completamente sincronizados debido a un retraso en la replicación o a otros factores, lo que puede llevar a resultados ligeramente diferentes en diferentes nodos esclavos.

Por lo tanto, la afirmación es Verdadera.

[BBDD Base pagina 7]

### *Afirmación 3*

En el modelo MapReduce, todas las funciones de reducción son combinables.

Sí, en el modelo MapReduce todas las funciones de reducción son combinables. Esto se debe a que cada función de reducción toma un conjunto de valores y produce un valor combinado único como resultado. Por lo tanto, es posible aplicar varias funciones de reducción en secuencia para producir un valor combinado final a partir de múltiples conjuntos de valores.

Significa que se pueden aplicar varias funciones de reducción consecutivas en una secuencia de pasos de MapReduce. En otras palabras, la salida de una función de reducción se puede usar como entrada para otra función de reducción. Esto es posible debido a que las funciones de reducción en MapReduce son independientes entre sí y no tienen efectos secundarios.

Por lo tanto, la afirmación es Verdadera.

[El framework MapReduce pagina 6]

### *Afirmación 4*

Las bases de datos orientadas a documentos sólo admiten replicación, resultando imposible las técnicas de distribución como sharding.

Falsa,

Además de la replicación, las bases de datos orientadas a documentos también pueden utilizar técnicas de distribución como el sharding (fragmentación horizontal) para distribuir los datos en diferentes servidores o nodos. Al igual que con la replicación, el sharding también puede mejorar la disponibilidad y la tolerancia a fallos al permitir que la base de datos continúe funcionando incluso si uno de los nodos falla. Sin embargo, al igual que con cualquier técnica de distribución, el sharding también puede aumentar la complejidad y el costo de administración de la base de datos.

[BBDD Diseño pagina 14]

### *Afirmación 5*

El teorema CAP ha permitido un cambio de paradigma en la programación de la persistencia de información, ya que implica una relajación de las propiedades de transacciones ACID.

El teorema CAP establece que es imposible para un sistema de bases de datos distribuidas garantizar simultáneamente tres propiedades: consistencias, disponibilidad y tolerancia a particiones de red. Por lo tanto, cuando se construye un sistema de bases de datos distribuidos se deben hacer compromisos en cuanto a que propiedades son más importantes para el sistema y se deben seleccionar los algoritmos de replicación y distribución adecuados para cumplir con esas propiedades. Es decir, que el teorema implica una relajación de las propiedades de transacciones ACID en favor de sistemas más tolerantes a fallos y escalables horizontalmente. La afirmación es Verdadera.

[BBDD Base pagina 9]

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

No logrado (D/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).

## Ejercicio 3 (30%)

Considerar los datos de los atletas que participan en carreras de media y larga distancia en España.

Por un lado, se dispone de información personal de cada atleta, identificado de forma única por el ID del chip con el que participan en las competiciones.



chip_id	nombre	apellido1	apellido2	residente_en
JGK309	Alejandro	García	Rodríguez	Madrid
QNR874	María	Fernández	Pérez	Barcelona
BDM752	Pablo	González	Sánchez	Valencia
TLF931	Laura	López	Martínez	Sevilla
WXP684	Sergio	Torres	Gómez	Bilbao
HVF460	Ana	Ruiz	Jiménez	Barcelona
YTK639	Luis	Moreno	Álvarez	Murcia
EBN206	Carmen	Castro	Romero	Barcelona
IJM125	Javier	Morales	Gutiérrez	Malaga
UAO572	Marta	Navarro	Vázquez	Cordoba

Por otro lado, se cuenta con el registro de las inscripciones de las diferentes carreras oficiales de media/larga distancia en España.

chip_id	tipo_carrera	ciudad_carrera	fecha	tiempo
JGK309	Maratón	Madrid	12-05-2020	02h:43m:18s
QNR874	10Km	Barcelona	23-09-2021	01h:15m:59s
BDM752	5Km	Valencia	04-07-2021	00h:28m:27s
TLF931	Maratón	Sevilla	19-02-2021	03h:26m:45s
WXP684	10Km	Bilbao	29-11-2022	01h:02m:13s
HVF460	Maratón	Zaragoza	08-08-2022	03h:59m:38s
YTK639	Maraton	Murcia	27-01-2020	03h:21m:49s
EBN206	Maraton	Alicante	05-10-2022	03h:48m:02s
IJM125	5Km	Malaga	17-06-2021	00h:22m:56s
ABC123	Maratón	Cordoba	01-04-2021	02h:37m:04s

**Se desea identificar el atleta de los residentes en Barcelona con el mejor tiempo de maratón en 2022. El output tendrá que indicar el nombre y apellidos del atleta y el mejor tiempo.**

Se aconseja usar dos pasadas map-reduce. Para ello se pide explicar en cada pasada lo que ocurre en cada fase (map, shuffle, reduce).

Concretamente para cada fase:

- Mostrar todos los datos de entrada (input)
- Explicar qué acciones se producen



- Mostrar el resultado producido (output) por esas acciones en todos los datos de entrada.

Es imprescindible explicar claramente la lógica de las acciones realizadas, enseñando en detalle el input y el output de cada fase. Se valorarán la lógica y la eficiencia del algoritmo y las explicaciones, y no (pseudo)código.

Para explicar la solución se va a considerar el siguiente ejemplo, que representa la lista de las tuplas de la tablas:

```
atletasInformation =
[('JGK309', 'Alejandro', 'García', 'Rodriguez', 'Madrid'),
('QNR874', 'Maria', 'Fernandez', 'Perez', 'Barcelona'),
('BDM752', 'Pablo', 'Gonzalez', 'Sanchez', 'Valencia'),
('TLF931', 'Laura', 'Lopez', 'Martinez', 'Sevilla'),
('WXP684', 'Sergio', 'Torres', 'Gomez', 'Bilbao'),
('HVF460', 'Ana', 'Ruiz', 'Jimenez', 'Barcelona'),
('YTK639', 'Luis', 'Moreno', 'Alvarez', 'Murcia'),
('EBN206', 'Carmen', 'Castro', 'Romero', 'Barcelona'),
('IJM125', 'Javier', 'Morales', 'Gutierrez', 'Malaga'),
('UA0572', 'Marta', 'Navarro', 'Vazquez', 'Cordoba'),]
```

```
incricionResults = [('JGK309', 'Maraton', 'Madrid', '12-05-2020', '02h:43m:18s'),
('QNR874', '10km', 'Barcelona', '23-09-2021', '01h:15m:59s'),
('BDM752', '5km', 'Valencia', '04-07-2021', '00h:28m:27s'),
('TLF931', 'Maraton', 'Sevilla', '19-02-2021', '03h:26m:45s'),
('WXP684', '10km', 'Bilbao', '29-11-2022', '01h:02m:13s'),
('HVF460', 'Maraton', 'Zaragoza', '08-08-2022', '03h:59m:38s'),
('YTK639', 'Maraton', 'Murcia', '27-01-2020', '03h:21m:49s'),
('EBN206', 'Maraton', 'Alicante', '05-10-2022', '03h:48m:02s'),
('IJM125', '5km', 'Malaga', '17-06-2021', '00h:22m:56s'),
('UA0572', 'Maraton', 'Cordoba', '01-04-2021', '02h:37m:04s'),]
```

La operación Map deberá procesar las dos listas que tomará como entrada. El resultado de procesar una tupla de las listas será que la función Map devuelve tuplas clave-valor, con el campo ID del chip como clave y la tupla como el valor:

```
('JGK309', ('JGK309', 'Alejandro', 'García', 'Rodriguez', 'Madrid'))
('QNR874', ('QNR874', 'Maria', 'Fernandez', 'Perez', 'Barcelona'))
('BDM752', ('BDM752', 'Pablo', 'Gonzalez', 'Sanchez', 'Valencia'))
('TLF931', ('TLF931', 'Laura', 'Lopez', 'Martinez', 'Sevilla'))
('WXP684', ('WXP684', 'Sergio', 'Torres', 'Gomez', 'Bilbao'))
('HVF460', ('HVF460', 'Ana', 'Ruiz', 'Jimenez', 'Barcelona'))
('YTK639', ('YTK639', 'Luis', 'Moreno', 'Alvarez', 'Murcia'))
('EBN206', ('EBN206', 'Carmen', 'Castro', 'Romero', 'Barcelona'))
('IJM125', ('IJM125', 'Javier', 'Morales', 'Gutierrez', 'Malaga'))
('UA0572', ('UA0572', 'Marta', 'Navarro', 'Vazquez', 'Cordoba'))
```

('JGK309', ('JGK309', 'Maraton', 'Madrid', '12-05-2020', '02h:43m:18s'))  
 ('QNR874', ('QNR874', '10km', 'Barcelona', '23-09-2021', '01h:15m:59s'))  
 ('BDM752', ('BDM752', '5km', 'Valencia', '04-07-2021', '00h:28m:27s'))  
 ('TLF931', ('TLF931', 'Maraton', 'Sevilla', '19-02-2021', '03h:26m:45s'))  
 ('WXP684', ('WXP684', '10km', 'Bilbao', '29-11-2022', '01h:02m:13s'))  
 ('HVF460', ('HVF460', 'Maraton', 'Zaragoza', '08-08-2022', '03h:59m:38s'))  
 ('YTK639', ('YTK639', 'Maraton', 'Murcia', '27-01-2020', '03h:21m:49s'))  
 ('EBN206', ('EBN206', 'Maraton', 'Alicante', '05-10-2022', '03h:48m:02s'))  
 ('IJM125', ('IJM125', '5km', 'Malaga', '17-06-2021', '00h:22m:56s'))  
 ('UAO572', ('UAO572', 'Maraton', 'Cordoba', '01-04-2021', '02h:37m:04s'))

Antes de pasar la lista de tuplas a la operación Reduce, se agrupan los valores por clave (función Shuffle) y se obtiene:

('JGK309', [('JGK309', 'Alejandro', 'García', 'Rodriguez', 'Madrid'), ('JGK309', 'Maraton', 'Madrid', '12-05-2020', '02h:43m:18s')])  
 ('QNR874', [('QNR874', 'Maria', 'Fernandez', 'Perez', 'Barcelona'), ('QNR874', '10km', 'Barcelona', '23-09-2021', '01h:15m:59s')])  
 ('BDM752', [('BDM752', 'Pablo', 'Gonzalez', 'Sanchez', 'Valencia'), ('BDM752', '5km', 'Valencia', '04-07-2021', '00h:28m:27s')])  
 ('TLF931', [('TLF931', 'Laura', 'Lopez', 'Martinez', 'Sevilla'), ('TLF931', 'Maraton', 'Sevilla', '19-02-2021', '03h:26m:45s')])  
 ('WXP684', [('WXP684', 'Sergio', 'Torres', 'Gomez', 'Bilbao'), ('WXP684', '10km', 'Bilbao', '29-11-2022', '01h:02m:13s')])  
 ('HVF460', [('HVF460', 'Ana', 'Ruiz', 'Jimenez', 'Barcelona'), ('HVF460', 'Maraton', 'Zaragoza', '08-08-2022', '03h:59m:38s')])  
 ('YTK639', [('YTK639', 'Luis', 'Moreno', 'Alvarez', 'Murcia'), ('YTK639', 'Maraton', 'Murcia', '27-01-2020', '03h:21m:49s')])  
 ('EBN206', [('EBN206', 'Carmen', 'Castro', 'Romero', 'Barcelona'), ('EBN206', 'Maraton', 'Alicante', '05-10-2022', '03h:48m:02s')])  
 ('IJM125', [('IJM125', 'Javier', 'Morales', 'Gutierrez', 'Malaga'), ('IJM125', '5km', 'Malaga', '17-06-2021', '00h:22m:56s')])  
 ('UAO572', [('UAO572', 'Marta', 'Navarro', 'Vazquez', 'Cordoba'), ('UAO572', 'Maraton', 'Cordoba', '01-04-2021', '02h:37m:04s')])

Operación Reduce: Recibe como argumentos una clave formada por el ID del chip y una lista de las tuplas que comparten como valor del campo ID del chip, el valor de la clave. En la operación Reduce se filtran las tuplas que se reciben de acuerdo con las condiciones dadas.

('HVF460', [('HVF460', 'Ana', 'Ruiz', 'Jimenez', 'Barcelona'), ('HVF460', 'Maraton', 'Zaragoza', '08-08-2022', '03h:59m:38s')])  
 ('EBN206', [('EBN206', 'Carmen', 'Castro', 'Romero', 'Barcelona'), ('EBN206', 'Maraton', 'Alicante', '05-10-2022', '03h:48m:02s')])

## Ejercicio 4 (20%)

Considerar los siguientes sistemas:

1. Un sistema de sensores de la calidad del aire de una ciudad (consistencia final en el tiempo, Escrituras más frecuentes que lecturas)
2. Un periódico online que es visitado por lectores de diferentes lugares de un país (consistencia final en el tiempo, Lecturas más frecuentes que escrituras)
3. Un sistema de control aéreo (consistencia fuerte, Escrituras y lecturas de la misma importancia)
4. Un sistema de banca online (consistencia fuerte, Lecturas más frecuentes que escrituras)

Se pide argumentar qué configuraciones de los valores W, R y N encajan mejor con cada caso con respecto al tipo de consistencia (fuerte/final en el tiempo) y al tipo de lecturas y escrituras que son necesarias realizar (Lecturas más frecuentes que escrituras, Escrituras más frecuentes que lecturas o Escrituras y lecturas de la misma importancia).

C1: N=3, W=1, R=1

C2: N=3, W=4, R=2

C3: N=5, W=1, R=3

C4: N=5, W=3, R=1

C5: N=7, W=4, R=5

C6: N=7, W=5, R=4

C7: N=5, W=5, R=1

Para resolverlo:

- Primero identificar y razonar tipo de consistencia, y tipo de lecturas y escrituras.
- A continuación, analizar cuál de las configuraciones dadas encaja con el sistema propuesto.

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

No logrado (D/C-)	Mínimamente logrado (C+)	Logrado (B)	Logrado de forma sobresaliente (A)

La respuesta es incorrecta o no está justificada.	La respuesta es incorrecta en algún aspecto, pero la justificación es correcta, demuestra conocimiento sobre el tema y está referenciada.	La respuesta es correcta y está convenientemente justificada, pero sólo justifica la configuración ganadora o bien el análisis de alguna de las configuraciones es incorrecta.	La respuesta es correcta, está correctamente justificada y aborda todas las configuraciones posibles, indicando para cada una de ellas, su adecuación al problema planteado y el porqué.
---	---	--	--