Organización de Datos 75.06. Primer Cuatrimestre de 2019. Examen parcial, primera oportunidad: Resolución

1. Posible solución.
    a. Map por ID_BASE en ambos RDD.
    b. Filter en RDD bases por PCIA = Buenos Aires.
    c. Filter en RDD mediciones por TIMESTAMP entre 01-12-2017 y 31-12-2018.
    d. Inner join entre ambos RDD.
    e. MAP por ID_BASE, YEAR(TIMESTAMP), MONTH(TIMESTAMP).
    f. ReduceByKey para calcular suma y contador.
    g. MAP para cambiar la clave a ID_BASE y aprovechar para hacer el calculo del promedio. En los values debe quedar el promedio, el nombre de la base y el mes/año.
    h. GroupByKey
    i. Filter para comparar que la tasa de un mes a otro haya variado en 30%.
    j. Map para quedarse con ID_BASE y nombre.

   Map innecesarios, descuento de 3 puntos. Filter tardíos, descuento de 3 puntos. Joins con RDD sin clave descuento de al menos 5 puntos. GroupByKey cuando en realidad se debería realizar un reduceByKey descuento de al menos 5 puntos.

2. La estrategia de resolución se puede plantear de la siguiente forma. Por un lado se debe crear con la información del archivo de información transaccional se debe crear la información para las estadísticas internas (prefijo int_). Las estadísticas se pueden calcular mediante groupby sobre vehicle_model_id usando las funciones de agg de pandas para mean y std y luego llevando el dataframe al formato (vehicle_model_id, int_mean_price, int_std_price). De la misma forma se puede trabajar con el otro dataframe calculando (vehicle_model_id, ext_mean_price, ext_std_price).

   El punto clave del ejercicio es indicar cual es la forma en la cual se va a realizar el join. Esencialmente por lo que se plantea en el enunciado, y considerando que se hace el join con left el dataframe de información externa y right el de información interna, el 'how' a utilizar es left dado que queremos quedarnos con todos los modelos que tenemos disponibles en la información scrapeada, dado que sabemos que los modelos comercializados están incluidos en ellos por el enunciado. Realizar algún otro tipo de join (inner, outer) descuenta 3 puntos. Aplicar 'left' cuando hay que aplicar 'right' o viceversa descuenta 3 puntos.

3.
    a. Verdadero. Independiente de SHA-256, la construcción de Merkle-Damgard es resistente a colisiones si la función de compresión es resistente a colisiones. La demostración se puede realizar por contraposición, es decir, se puede asumir que la construcción de MD no es resistente a colisiones y llegar a un absurdo. Entonces, si H(X) es la salida de MD, suponemos que podemos encontrar x e y (distintos) tales que H(X)=H(X). Si eso ocurre, entonces tenemos los siguientes escenarios que pueden ocurrir:
        i. La última función de compresión recibió dos entradas distintas y generó dos salidas iguales. Es decir, encontramos una colisión en la última función de compresión. Esto significa que entonces la función de compresión no es resistente a colisiones.
        ii. La última función de compresión recibió dos entradas iguales, con lo cual las salidas que generará son iguales. Para que esto ocurra, entonces la anteúltima función de compresión tuvo que producir la misma salida para dos inputs diferentes. Esto significa que la anteúltima función de compresión (o cualquier anterior) no es resistente a colisiones.

    Dicho lo anterior, se llega a un absurdo porque se parte de funciones de compresión que sí son resistentes a colisiones.
    b. Verdadero. Dado que $|U| > (n-1)m$, entonces U contiene como mínimo (n-1)m+1 claves. Asumiendo que U tiene esta cardinalidad, en el mejor de los casos, si todas las claves se pueden hashear uniformemente, cada uno de los m slots tendrá (n-1) claves excepto por uno de ellos que deberá tener (n-1)+1 claves, es decir n claves (pigeonhole principle).
    c. Verdadero. Esto lo podemos ver en el grafo de colisiones: si tenemos 2 funciones de hash nos queda un grafo bipartito, si tenemos 3, uno tripartito, etc. Los caminos van a ser más cortos en promedio y podemos poner más datos sin que se degrade la performance. Otra forma de verlo es que *n* claves con dos funciones de hash solo pueden estar en un máximo de 2^n configuraciones, si son 3 funciones de hash en 3^n, etc. Se puede usar mejor el espacio teniendo más funciones de hash. Con tres funciones, se puede aumentar el load factor a 91%.
    d. Falso. El lema JL da una cota genérica, es decir, para cualquier set de puntos. Podemos tener configuraciones de puntos que sean más amigables y que se transformen a una dimensión menor con pequeña o nula distorsión.

4. La distancia angular es el ángulo entre los vectores. Graficando los puntos se puede notar fácilmente, sin hacer cuentas, que d(A,B) = 90º; d(A,C) = 180º y d(B,C) = 90º. Por definición, una familia del estilo H(d1,d2,p1,p2) indica que puntos que se encuentran a distancia d1 o menor tienen que tener probabilidad de colisión de al menos p1, mientras que puntos a distancia d2 o mayor tienen probabilidad de colisión de a lo sumo p2. El enunciado entonces nos sugiere armar la familia H(90º, 135º, 0.75, 0.4375). Sin embargo, recordando la familia para distancia angular, lo que se obtiene del enunciado es la familia H(90º, 135º, ½, ¼), y esto es porque p1 = 1 - 90º / 180º = ½ , mientras que p2 = 1 -135º / 180º = ¼. Lo que se debe realizar entonces es una amplificación con ***b*** y con ***r***. Si se toma, por ejemplo, b=2 con r=1, se obtiene que p1 = 1-(1-½)^2 = 0.75, mientras que p2=1-(1-¼)^2=0.4375 que es lo pedido con el enunciado.
   Para poder encontrar los hiperplanos que pide en el enunciado, nuevamente se pueden encontrar dichos hiperplanos en el gráfico. Con los hiperplanos r1=[1,1] y r2=[-1,1], se obtiene la siguiente tabla:

	A	B	C	Q
r1	1	1	-1	-1
r2	-1	1	1	1

De la primer banda se puede verificar que (A,B) serán candidatos. De la segunda banda se puede verificar que B y C serán candidatos. Se puede verificar que A y C no son candidatos en ninguna banda. Esto cumple con lo pedido en el enunciado.
El punto Q será candidato a ser similar con el punto B (porque coinciden en la segunda banda) y con el punto C (porque coincide en ambas bandas).

5. Dado que se trata de un flujo de datos que se recibe en forma continua el método a emplear debe ser dinámico. No se puede pensar en realizar dos pasadas. Sin embargo, si es posible pre-entrenar el modelo en función de lo que sabemos del formato de los datos. Se espera que se analice el formato del archivo a comprimir, y en base a eso se decida que método implementar. Se podrían comprimir distintos elementos de distinta forma, siempre que este clara como se va a realizar la sincronización o salto entre métodos y se pueda descomprimir. Con respecto a los datos, las fechas tendrán pocas variaciones (el año será constante durante el año, el mes cambiará cada 30 días de entrada, y hasta para un mismo día mantendremos muchas entradas, por lo que deberían pensar en algún esquema que permita contemplar eso. Para las horas tenemos valores limitados, aunque con mayor variación, y equiprobables. Para las temperaturas, los valores serán acotados y seguro se puede tomando una muestra analizar la distribución de valores para poder aprovechar las probabilidades de los mismos. Por último, para el registro de variaciones muy probablemente predominen los valores bajos, 0, 1 o -1, y rara vez existan variaciones mayores de temperatura en los 5 segundos que transcurren entre medición y medición. Es válido si toman los 5 segundos como delta de timestamp y calculan el timestamp en base al registro anterior. También es válido si guardan solo la variación de temperatura y calculan la misma en base a la medición anterior. De esta forma, solo podrían ser necesarias las variaciones y el resto de los datos sería calculable.
   Si no evalúan los datos vale cero, independientemente del método que propongan, porque no tiene justificación. Si piensan en cualquier método estático vale cero, conceptualmente no tiene sentido implementarlo. Si no se puede descomprimir vale cero también.

6. a) Deben usar concatenación de términos para el léxico. Si usan front-coding u otro método, descuento de 2 puntos. Para las distancias deberían utilizar 2 bits para cada una, ya que son solo 4 documentos. Si usan más, descuento de 1 punto. Índices con registro de longitud variable o sin orden, vale 0 puntos.
   b) Deben hacer la búsqueda binaria para cada término. Van a obtener que solo el documento 1 posee ambos términos. Por último, deben indicar que se debe revisar el documento 1 si realmente posee esta frase para devolverlo en la consulta. Si no realiza este chequeo, descuento de 1 punto.
   c) Deben agregar frecuencia de aparición del término en cada documento y las posiciones.
   d) Deben contar los accesos al índice, los accesos al léxico, los accesos a los punteros de documentos y el acceso al documento 1 para la revisión de la frase. Son 8 accesos para cada término. Aunque los 3 primeros accesos se repiten en ambos, se podrían obviar. Si no cuentan el acceso al documento 1, descuento de 1 punto.
7. Hay que evaluar la resolución pero, debe contemplar el componente tiempo, y permitir ver la progresión de los valores a lo largo del año. Podrían hacerlo mes a mes, pero tambien se podria manejar en función del timestamp. La visualización deberia tener como centro de análisis la evolución de los datos internos de la empresa, y permitir ver en contraste los datos externos para poder comparar el volumen. Se descuentan 5 puntos si falta el título, títulos de los ejes, referencias.