

Conceptos y aplicaciones en Big Data

Trabajo Práctico 3 – Modalidad de cursada presencial

Spark streaming

Pautas generales

- La entrega consiste en la implementación de un script con Spark streaming, resolviendo la consigna presente en este enunciado. Se deberá entregar el código fuente implementado.
- Los alumnos pueden conformar grupo de no más de dos integrantes y hacer una única entrega grupal.
- La entrega se realiza por la mensajería del curso en IDEAS.
- La fecha límite de entrega es el 14 de diciembre de 2025.

Enunciado

Basado en el problema presentado en el TP1 y TP2, implemente una solución en Spark streaming que permita actualizar la información almacenada por un sistema recomendador de jugadores a retar.

En este trabajo solo se pide el script que mantiene actualizado el “conocimiento” del sistema recomendador. No se pide implementar el módulo que hace la recomendación.

Para mantener el sistema actualizado el servidor genera un stream de tuplas $\langle id_retador, id_retado, tiempo_combate \rangle$. Se ofrecen 50 archivos con tuplas de combates jugados para poder simular el stream de datos.

Cada uno de los posibles jugadores a retar (id_retado) inicia con un *score* igual a cero. Y la actualización del *score* de cada posible jugador a retar se realiza en todas las ventanas temporales. Por lo tanto en cada ventana temporal deberá:

- 1) Acumular, para cada id_retado , el tiempo promedio de los combates entre todos los combates (de la misma ventana) que retaron al mismo jugador. El tiempo promedio resultante será acumulado al *score* de la del jugador id_retado .
- 2) Además, en cada ventana temporal procesada el sistema deberá ir descontando del *score* de todos los id_retado un valor de antigüedad A preestablecido.

Ejemplo 1: si un id_retado con un *score* de 30, y $A = 0.5$, en una ventana temporal no fue retado entonces su nuevo *score* será $30 - 0.5 = 29.5$.

Ejemplo 2: si un id_retado con un *score* de 30, y $A = 1$, en una ventana temporal fue retado durante 10 segundos en promedio, entonces su nuevo *score* será $30 + 10 - 1 = 39$.

Notar que la antigüedad se descuenta en todas las ventanas temporales, sin importar si un jugador (id_retado) fue retado o no.

Aclaración

Suponga que la cantidad de jugadores es Big Data.