* Diagrama de casos d’us
  + Part d’avaluació posar-ho a dins de recomanació
* Descripció del model
  + Explicar patró de les funcions de distància (no sé si era Strategy o Template o algo així)
* Estructures de dades
  + Explicar com s’implementa el TreeMap (BST)
  + Explicar més pseudocodi pels algoritmes per fer-ho menys ambigu
* Codi
  + Utiltizar més el keyword final amb Java
  + Declarar Strings com a constants, especialment a conjunt d’item, el mateix per alguns numeros (a Item) definir valor per defecte constant i utilitazar-lo
  + Canviar noms variables lletres simples
  + Definir excepcions de domini nosaltres (TaulaCSV modificat)
    - No canviar la semàntica de les excepcions
  + Eliminar codi comentat
  + Sintaxi consistent
    - Guió baix
    - Espai en els iguals
  + Implementar TipusAtribut com un map (tornar-ho a parlar)
  + Mètodes de KMeans
    - Funcions pures
    - Hi ha funcions que modifiquen l’estat intern de la classe
    - Passar-ho com arguments i treure el que calgui
    - Principalment funcions privades
    - És més per temes de tests
    - Millor no utilitzar moltes funcions sense arguments ni retorns
    - Com que canvia l’estat intern i depen de l’estat intern, amb mateixos inputs pot donar diferents outputs
  + Partir obteRecomanacions
  + FiltreTest
    - Fer funcions per treure-ho
  + KMeansTest, LectorCSVTest, SlopeOneTest afegir més tests
* Podem entregar nous tests i ell ho valora