

Documentació Projecte E.D. - Fase II	
Autors	Laia Alcalde - 1642329 Elena Gutiérrez - 1703828 Laia Cámara - 1710505
Qualificació "grade"	Hem anat polint el codi amb el test de proves proporcionat a caronte. 203
Canvis introduïts respecte a la primera part	<ul style="list-style-type: none"> + Classe VideoData: <ul style="list-style-type: none"> - Ara utilitza la classe ElementData per guardar les metadades encapsulades en lloc d'utilitzar representacions simples. Això permet una manipulació més robusta i modular de les dades. - Les dades dels vídeos s'organitzen en un graf mitjançant la classe GrafHash, on els nodes són vídeos identificats pel seu UUID i les arestes representen relacions seqüencials de reproducció entre vídeos. + Classe GrafHash: <ul style="list-style-type: none"> - Ara inclou mètodes com insert_vertex, get, __contains__, __getitem__, i __delitem__ per suportar la manipulació eficient de dades en el graf. + Classe ElementData: <ul style="list-style-type: none"> - Funciona com un contenidor per les metadades dels vídeos, incloent atributs com títol, artista, gènere, i altres. Tots els atributs són properties i la classe és hashable per garantir una gestió única i eficaç dins del graf. + Classe Metadata: <ul style="list-style-type: none"> - Es fa servir una llista que conté els UUIDs dels vídeos a buscar. + Classe VideoFiles: <ul style="list-style-type: none"> - self.current_files conté els

	<p>apths dels fitxers MP4</p> <ul style="list-style-type: none"> + Classe VideoID: <ul style="list-style-type: none"> - self.file_uuid_map és un diccionari que mapeja el camí absolut d'un fitxer MP4 al seu UUID corresponent. - self.used_uuids és un conjunt que emmagatzema tots els UUIDs que ja han estat generats i assignats a fitxers. + Classe VideoPlayer: <ul style="list-style-type: none"> - self.video_data conté informació dels vídeos, com els seus paths, metadades (títol, artista, duració, etc.) i UUIDs. + Classe PlayList: <ul style="list-style-type: none"> - Ara no permet que un vídeo aparegui més d'una vegada dins d'una llista, evitant així bucles dins el graf. - Ara pot llegir llistes de reproducció no només des d'un arxiu M3U sinó també des de llistes Python de UUIDs mitjançant el mètode <code>read_list</code>. + Classe SearchMetadata: <ul style="list-style-type: none"> - S'ha implementat <code>get_similar</code>, que calcula la semblança entre vídeos basant-se en un algorisme específic que té en compte distàncies dins el graf i el ranking dels vídeos. - La funció <code>get_auto_play</code> genera llistes de reproducció automatitzades basant-se en les connexions entre vídeos i els seus rankings.
Anàlisis del rendiment	<p>Durant la implementació i prova de la generació automàtica de reproduccions de vídeos, s'han identificat els següents problemes de rendiment:</p> <p>Càlcul de semblances entre vídeos:</p> <ul style="list-style-type: none"> • L'algorisme per calcular la semblança entre vídeos requereix múltiples recorreguts del graf per obtenir distàncies i rankings. Això

	<p>pot ser costós computacionalment en col·leccions grans, especialment si el graf conté molts nodes i arestes.</p> <p>Iteracions repetides:</p> <ul style="list-style-type: none">• A l'hora de calcular recomanacions per a múltiples vídeos, es produeixen càlculs redundants de distàncies i rankings. <p>A més del temps d'escriptura i lectura dels fitxers.</p>
--	---