

| | |
|--|--|
| Documentació Projecte E.D. - Fase II | |
| Autors | Adrià Muro Gómez (1665191), David Morillo Massagué (1666540) Grup 81-7 |
| Qualificació "grade" | 236 / 250 [26.1] FAIL: SearchMetadata get_similar() <u>result</u> |
| Canvis introduïts respecte a la primera part | No hem implementat canvis que afectin a la funcionalitat de les funcions respecte la primera part, a part de l'implementació de les funcionalitats demanades. |
| Anàlisi del rendiment | Temps d'execució del main propi: ~3.7 segons Temps d'execució del main de caronte: ~62 segons |
| Implementació utilitzada per a les estructures | MusicFiles: <ul style="list-style-type: none"> - _files: set - _files_added: set - _files_removed: set MusicID: <ul style="list-style-type: none"> - _uuids: dict[file: str] = uuid: uuid.UUID ElementData: <ul style="list-style-type: none"> - _title: str - _artist: str - _album: str - _genre: str - _title: int - _filename: str Vertex: <ul style="list-style-type: none"> - _key: uuid.UUID - _value: ElementData GrafHash: <ul style="list-style-type: none"> - _nodes: dict[uuid: uuid.UUID] = vertex: Vertex - _out: dict[uuid: uuid.UUID] = arestes: dict - _out: dict[uuid: uuid.UUID] = arestes: dict MusicData: <ul style="list-style-type: none"> - _songs: GrafHash MusicPlayer: <ul style="list-style-type: none"> - _songs_data: MusicData - _player: vlc.MediaPlayer PlayList: <ul style="list-style-type: none"> - _uuids: MusicID |

| | |
|---------------|---|
| | <ul style="list-style-type: none"> - <code>_player</code>: MusicPlayer - <code>_playlist</code>: list <p>SearchMetadata:</p> <ul style="list-style-type: none"> - <code>songs</code>: MusicData |
| Proves i test | <p>Hem construït el nostre programa <i>main</i> de manera que comprovi totes les funcionalitats implementades en aquesta segona part del projecte. A continuació, es destaquen les funcionalitats que es posen a prova:</p> <p>Inicialització i Actualització:</p> <ul style="list-style-type: none"> - El programa inicialitza diverses classes (MusicFiles, MusicID, MusicData, MusicPlayer, SearchMetadata, PlayList) amb la ruta root obtinguda amb <code>cfg.get_root()</code> - La funció update_from(path) crea aquestes classes, establint relacions entre elles, i fent servir únicament les funcionalitats de <code>__init__</code> per a crear-les i carregar-les. <p>Verificació de Classes:</p> <ul style="list-style-type: none"> - La funció <code>comprovar_prints</code> verifica i imprimeix informació sobre cada classe. - Per a cada classe, comprova si s'han implementat els mètodes <code>__len__</code>, <code>__iter__</code> i <code>__getitem__</code> i imprimeix informació rellevant. - També verifica la representació de la classe i itera sobre els seus elements. <p>Creació i Lectura de Llistes de Reproducció:</p> <ul style="list-style-type: none"> - Es creen un nombre d'instàncies de la classe PlayList, cadascuna carrega un arxiu de playlist (donades al caronte). - Les llistes de reproducció es llegeixen mitjançant el mètode <code>MusicData.read_playlist()</code>, <p>Top 5 de Cançons:</p> <ul style="list-style-type: none"> - A partir de la lectura de playlists mencionada, el programa obté el top 5 de cançons amb el mètode <code>search.get_topfive()</code>. Fem servir aquest ja que implica una execució en cadena de les funcions: <ul style="list-style-type: none"> - <code>SearchMetadata.get_similar()</code> - <code>SearchMetadata.song_similarity()</code> (funció auxiliar pròpia) - <code>MusicData.get_song_distance()</code> - <code>GrafHash.camiMesCurt()</code> - <code>GrafHash.dijkstraModif()</code> - <code>GrafHash.minDistance()</code> - <code>MusicData.get_song_rank()</code> <p>Busca per títol (1a part, opcional):</p> <ul style="list-style-type: none"> - Permet cercar cançons per el títol amb la classe <code>SearchMetadata</code>. Amb aquesta funció comprovem la funcionalitat de la primera part no ha canviat. <p>Mesura del Temps d'Execució:</p> <ul style="list-style-type: none"> - Es mesura el temps total d'execució del programa amb el mòdul <code>time</code>. |

| | |
|------------|--|
| Anotacions | <ul style="list-style-type: none"> - Per una major simplificació i millora en el rendiment, veiem que forçant el tipus str en comptes de uuid.UUID en els identificadors, observem que la execució del programa (3.68 s) resulta tenir un 250% de <i>speedup</i> (2.5 vegades més ràpid) respecta a l'execució original (9 s) . Suposem que és degut a que és més lleugera la comparació entre dos strings que dos classes UUID del mòdul uuid. Això ho hem fet canviant la classe MusicID i afegint la línia uuid = str(uuid) en la generació d'identificadors. <p>Comprovem que aquest canvi no afecta al funcionament o resultats del programa en cap aspecte. Afecta només al seu rendiment.</p> <p>Per un millor enteniment, no hem inclòs aquest canvi en el diagrama ni en l'implementació de les estructures mencionada.</p> <p>Anotacions del diagrama:</p> <ul style="list-style-type: none"> - Nodes → Classes implementades (no necessàriament creades per l'usuari, com GrafHash o ElementData) - Arestes → Referències entre classes i estructures: <ul style="list-style-type: none"> - Cap de la fletxa → Classe o estructura referenciada / apuntada - Cua de la fletxa → Objecte dins d'una classe que apunta a una classe o estructura - Estructures verdes → Estructures originals a memòria, sense apuntar. - Contenedor blau → Classe que conté una altra com un atribut directe. - En una execució normal, totes les estructures de llistes i diccionaris mantenen la mateixa longitud (la del total de cançons diferents dins del path), a excepció de la classe Playlist, que per cada llista, serà la quantitat de cançons en l'arxiu m3u (assumint que aquestes existeixen en la col·lecció). |
| Diagrama | <p>L'únic canvi funcional ha sigut el d'implementar les classes GrafHash, Vertex i ElementData:</p> <ul style="list-style-type: none"> - GrafHash i Vertex conté diccionaris amb claus referents a UUIDs (valors del diccionari a MusicID). - ElementData elimina la tupla que contenia el valor del diccionari de MusicData originalment, i ara conté la informació de la cançó i el nom de l'arxiu en els seus atributs. - Ara MusicData té un atribut GrafHash, que és on guarda tota la informació de la col·lecció, tant les dades de les cançons com les relacions entre si (graf). <p>*Diagrama a la pàgina següent:</p> |

Diagrama de dades per classes

