|  |  |
| --- | --- |
| Documentació Projecte E.D. - Fase I | |
| Autors | Carlos Oyardo Gutierrez (1569369) i Joel Tapia Salvador (1638962) |
| Corpus MP3 proves | Stock si / Altres no, 140 cançons i 10 directoris |
| Qualificació “grade” | 66 |
| Implementació utilitzada per a les estructures | MusicFiles:  Utilitzem llistes per guardar els paths actuals, els afegit i els  eliminats, ja que no necessitem accedir a ells en temps O(1)  a través del path, ja que recorrerem tots per iniciar el uuids i  guardar-los a MusicData.  MusicID:  Guardem un en un diccionari els uuid generats amb el path  com a clau per poder accedir en temps O(1) cada cop que  es crida el mètode get\_uuid(). També guardem en un  diccionari si el uuid ha estat eliminat amb el uuid com a clau  per poder accedir en temps O(1). Guardem la quantitat de  uuid no eliminats en una variable per no tenir que calcular  cada cop.  MusicData:  Guardem les metadades de cada cançó en un diccionari on  la clau es el uuid de la cançó (per poder accedir en temps  O(1)) i les metadades i file de la cançó son guardats en una  llista on podem accedir a cadascuna de les metadades i al  file en temps O(1) amb el mètodes get\_.  MusicPlayer:  Guardem la classe MusicData i treballem sobre aquesta amb  un mètode d’aquesta classe que permet comprovar si un file  esta guardat.  PlayList:  Com que les cançons s’han de reproduir en l’orde guardat  podem utilitzar una llista per guardar els uuids de les  cançons e iterar sobre aquesta cridant al mètode de  reproducció de la classe MusicPlayer guardat com a atribut,  al igual que podem obtenir l’uuid a través del mètode de la  classe MusicID.  SearchMetadata:  Consultem la metadada corresponent cridant al mètode get\_  corresponent de MusicData recorrent totes les cançons ja  guardades a MusicData a través d’un mètode iterador que  itera sobre les claus uuid de MusicData amb una list  comprenhension i així no tenim que guardar cap data  temporal dins de la classe SearcMeatada. Per fer les  operacions OR o AND utilitzem una la estructura de conjunts  que permet fer operació d’unió i intersecció sense guardar  valors repetits. |
| Cerques realitzades |  |
| Anotacions | Ens hem repartit les funcions i les hem implementat fins que hem tingut tot el projecte, Després ho hem pujat al github on hem mirat el codi i hem tractat d'arreglar els errors per tal d'obtenir la màxima qualificació. |

Diagrama

Descripción generada automáticamenteDiagrama

|  |  |
| --- | --- |
| Documentació Projecte E.D. - Fase II | |
| Autors | Carlos Oyardo Gutierrez (1569369) i Joel Tapia Salvador (1638962) |
| Qualificació “grade” |  |
| Canvis introduïts respecte a la primera part | Hem seguit el document per implementar la primera versió on clarament hem tingut errors a l’hora de executar el main proveït per exemple implementar o adaptar al nostre codi funcions com el camimescurt que inclouria disjktramodif i mindistance de exercicis fets a classe. Hem modificat com es guarden les dades de les cançons, per guardalres en un graf hash, un graf on guardem el nodes com a diccionari per poder accedir en O(1) a través de hash i els edges també els guardem con un diccionari de diccionaris per accedir en O(1). |
| Anàlisis del rendiment | A l’hora de calcular la similitud de les cançons es fa la comparació entre dos cançons sempre es calcula la similitud entre totes les cançons del corpus i per cadascuna es té que calcular la distancia (aplicar Disjktra que te complexitat O(N\*\*2)) per tant tenim una complexitat de O(N\*\*3), es podria reduir aquesta complexitat tenint en compte que les songs amb més similitud seran aquelles que es troben més a prop del la cançó de la qual busquem similitud, per tant es pot limitar el algoritme a començar a treballar amb els nodes més a prop del nostre (els seus veïns o qui es veí, o els veïns del seus veïns i així), fins a obtenir la quantitat de cançons demanades amb el paràmetre max\_list. |
| Proves i test | Cada vegada que hem fet una modificació, o en aquest cas, creat una classe nova fèiem unitest exclusius d’aquestes per comprovar el seu funcionament correcte per separat. Per fer les proves no hem utilitzat un Corpus de cançons addicionals al del campus virtual però en alguns testos o hem utilitzat informació inventada per provar mètodes o propietats específiques. |
| Anotacions |  |