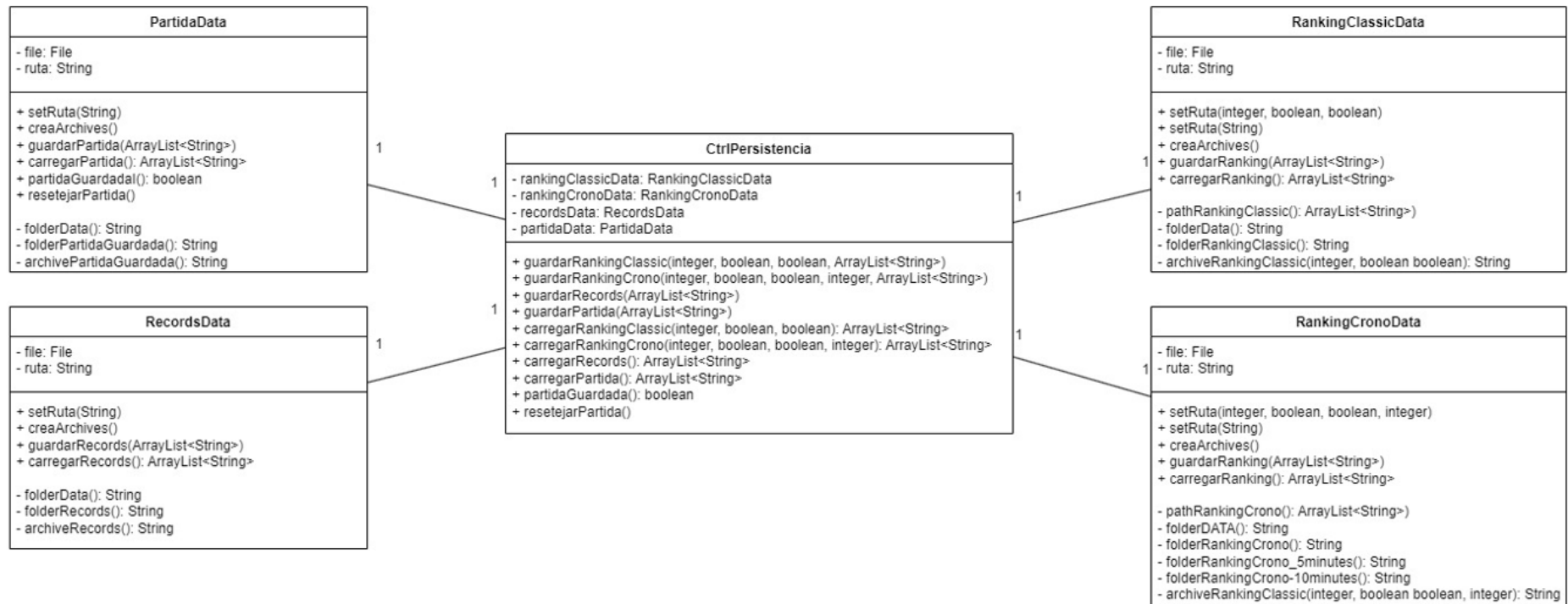


Diagrama classes Persistència



Com no es pot visualitzar bé en tamany A4, hem adjuntat aquest diagrama en el document “*Diagrama de classes Persistencia.pdf*”.

Descripció

CtrlPersistencia

Nom de la classe: CtrlPersistencia

Descripció:

Classe que serveix per a coordinar el funcionament de totes les classes de la persistència.

Cardinalitat:

- Cada CtrlPersistencia té un RankingClassicData.
- Cada CtrlPersistencia té un RankingCronoData.
- Cada CtrlPersistencia té un RecordsData.
- Cada CtrlPersistencia té una PartidaData.

Descripció atributs:

- **RankingClassicData rankingClassicData:** És l'atribut de la classe RankingClassicData.
- **RankingCronoData rankingCronoData:** És l'atribut de la classe RankingCronoData.
- **RecordsData recordsData:** És l'atribut de la classe RecordsData.
- **PartidaData partidaData:** És l'atribut de la classe PartidaData.

Descripció mètodes:

- **guardarRankingClassic(int numBoles, boolean modeAjuda, boolean colorsRepetits, ArrayList<String> ranking):** Aquest mètode s'encarrega de passar el ranking seleccionat a la classe RankingClassicData perquè el guardi en un arxiu.
- **carregarRankingClassic(int numBoles, boolean modeAjuda, boolean colorsRepetits):** Aquest mètode s'encarrega de passar el ranking seleccionat a la classe RankingClassicData perquè el carregui des de l'arxiu corresponent. Retorna un ArrayList de strings amb les posicions del ranking seleccionat.
- **guardarRankingCrono(int numBoles, boolean modeAjuda, boolean colorsRepetits, int cronometre, ArrayList<String> ranking):** Aquest mètode s'encarrega de passar el ranking seleccionat a la classe RankingCronoData perquè el carregui des de l'arxiu corresponent. Retorna un ArrayList de strings amb les posicions del ranking seleccionat.
- **carregarRankingCrono(int numBoles, boolean modeAjuda, boolean colorsRepetits, int cronometre):** Aquest mètode s'encarrega de passar el ranking seleccionat a la classe RankingCronoData perquè el carregui des de l'arxiu corresponent. Retorna un ArrayList de strings amb les posicions del ranking seleccionat.
- **guardarRecords(ArrayList<String> records):** Aquest mètode s'encarrega de passar els records a la classe RecordData perquè els guardi en un arxiu.
- **carregarRecords():** Aquest mètode s'encarrega de passar els records a la classe RecordData perquè el carregui des de l'arxiu corresponent. Retorna un ArrayList de strings amb els records.
- **guardarPartida(ArrayList<String> partida):** Aquest mètode s'encarrega de passar la partida a la classe PartidaData perquè la guardi en un arxiu.

- **carregarPartida():** Aquest mètode s'encarrega de passar la partida a la classe PartidaRecord perquè la carregui des de l'arxiu. Retorna un ArrayList de strings amb la partida.
- **partidaGuardada():** Aquest mètode s'encarrega de comprovar si hi ha una partida guardada. Retorna un booleà: true en cas d'haver-hi una partida guardada, false en cas contrari.
- **resetejarPartida():** Aquest mètode s'encarrega de ficar l'arxiu d'una partida amb el model estàndard per a "resetejarla".

PartidaData

Nom de la classe: PartidaData

Descripció:

Classe que serveix per a guardar i carregar la partida en un arxiu de text.

Cardinalitat:

- Cada PartidaData té un CtrlPersistencia .

Descripció atributs:

- **File file:** És un atribut de la classe File.
- **String ruta:** És el path d'on es guardarà l'arxiu.

Descripció mètodes:

- **setRuta(String path):** Aquest mètode s'encarrega de guardar una ruta en la classe.
- **folderDATA():** Aquest mètode s'encarrega de localitzar els paths de la carpeta DATA.
Retorna el path de la carpeta DATA.
- **folderPartidaGuardada():** Aquest mètode s'encarrega de localitzar el path de la carpeta PartidaGuardada. Retorna el path de la carpeta PartidaGuardada.
- **archivePartidaGuardada():** Aquest mètode s'encarrega de localitzar el path de l'arxiu seleccionat. Retorna el path del arxiu seleccionat.
- **partidaGuardada():** Aquest mètode s'encarrega de comprovar si hi ha una partida guardada. Retorna un booleà: true en cas que hi hagi una partida guardada, false en cas contrari.
- **resetejarPartida():** Aquest mètode s'encarrega de ficar l'arxiu d'una partida amb el model estàndard per a "resetejarla".
- **creaArchives():** Aquest mètode s'encarrega de comprovar que l'arxiu de Partida existeix. Si no existeix, el crea.
- **guardarPartida(ArrayList<String> datos):** Aquest mètode s'encarrega de guardar l'ArrayList que conté la informació de la partida a l'arxiu.
- **carregarPartida():** Aquest mètode s'encarrega de carregar el contingut de la partida guardada i retornarla. Retorna un ArrayList de strings amb el contingut de la partida.

RecordsData

Nom de la classe: RecordsData

Descripció:

Classe que serveix per a guardar i carregar la llista de records en un arxiu de text.

Cardinalitat:

- Cada RecordsData té un CtrlPersistencia .

Descripció atributs:

- **File file:** És un atribut de la classe File.
- **String ruta:** És el path d'on es guardarà l'arxiu.

Descripció mètodes:

- **setRuta(String path):** Aquest mètode s'encarrega de guardar una ruta en la classe.
- **folderDATA():** Aquest mètode s'encarrega de localitzar els paths de la carpeta DATA. Retorna el path de la carpeta DATA.
- **folderRecords():** Aquest mètode s'encarrega de localitzar el path de la carpeta Records. Retorna el path de la carpeta Records.
- **archiveRecords():** Aquest mètode s'encarrega de localitzar el path de l'arxiu seleccionat. Retorna el path del arxiu seleccionat.
- **creaArchives():** Aquest mètode s'encarrega de comprovar que l'arxiu de Records existeixi. Si no existeix, el crea.
- **guardarRecords(ArrayList<String> datos):** Aquest mètode s'encarrega de guardar l'ArrayList que conté la informació dels Records a l'arxiu.
- **carregarRecords():** Aquest mètode s'encarrega de carregar el contingut dels Records guardats i retornarlo. Retorna Un ArrayList de strings amb el contingut dels Records.

RankingClassicData

Nom de la classe: RankinClassicData

Descripció:

Classe que serveix per a guardar i carregar el ranking del mode clàssic en un arxiu de text.

Cardinalitat:

- Cada RankinClassicData té un CtrlPersistencia .

Descripció atributs:

- **File file:** És un atribut de la classe File.
- **String ruta:** És el path d'on es guardarà l'arxiu.

Descripció mètodes:

- **pathRankingsClassic():** Aquest mètode s'encarrega de crear un ArrayList de strings amb tots els paths dels rankings classic. Retorna un ArrayList de strings que conté el path de tots els rankings classic.
- **String folderDATA():** Aquest mètode s'encarrega de localitzar els paths de la carpeta DATA. Retorna el path de la carpeta DATA.

- **folderRankingClassic():** Aquest mètode s'encarrega de localitzar el path de la carpeta RankingClassic. Retorna el path de la carpeta RankingClassic.
- **archiveRankingClassic(int numBoles, boolean modeAjuda, boolean colorsRepetits):** Aquest mètode s'encarrega de localitzar el path de l'arxiu seleccionat. Retorna el path del arxiu seleccionat.
- **setRuta(int numBoles, boolean modeAjuda, boolean colorsRepetits):** Aquest mètode s'encarrega de guardar una ruta en la classe.
- **setRuta(String path):** Aquest mètode s'encarrega de guardar una ruta en la classe.
- **creaArchives():** Aquest mètode s'encarrega de comprovar que els arxius de ranking clàssic existeixin. Si no existeixen, els crea.
- **guardarRanking(ArrayList<String> datos):** Aquest mètode s'encarrega de guardar l'ArrayList que conté la informació del ranking amb el path guardat anteriorment a la classe.
- **carregarRanking():** Aquest mètode s'encarrega de carregar el contingut del ranking guardat i retornarlo. Retorna un ArrayList de strings amb el contingut del ranking seleccionat.

RankingCronoData

Nom de la classe: RankingCronoData

Descripció:

Classe que serveix per a guardar i carregar el ranking del mode crono en un arxiu de text.

Cardinalitat:

- Cada RankingCronoData té un CtrlPersistencia .

Descripció atributs:

- **File file:** És un atribut de la classe File.
- **String ruta:** És el path d'on es guardarà l'arxiu.

Descripció mètodes:

- **pathRankingsCrono():** Aquest mètode s'encarrega de crear un ArrayList de strings amb tots els paths dels rankings crono. Retorna un ArrayList de strings que conté el path de tots els rankings crono.
- **String folderDATA():** Aquest mètode s'encarrega de localitzar els paths de la carpeta DATA. Retorna el path de la carpeta DATA.
- **folderRankingCrono():** Aquest mètode s'encarrega de localitzar el path de la carpeta RankingCrono. Retorna el path de la carpeta RankingCrono.
- **folderRankingCrono_5minutes():** Aquest mètode s'encarrega de localitzar el path de la carpeta RankingCrono_5minutes. Retorna el path de la carpeta RankingCrono_5minutes.
- **folderRankingCrono_10minutes():** Aquest mètode s'encarrega de localitzar el path de la carpeta RankingCrono_10minutes. Retorna el path de la carpeta RankingCrono_10minutes.
- **archiveRankingCrono(int numBoles, boolean modeAjuda, boolean colorsRepetits, int cronometre):** Aquest mètode s'encarrega de localitzar el path de l'arxiu seleccionat. Retorna el path del arxiu seleccionat.

- **setRuta(int numBoles, boolean modeAjuda, boolean colorsRepetits, int cronometre):** Aquest mètode s'encarrega de guardar una ruta en la classe.
- **setRuta(String path):** Aquest mètode s'encarrega de guardar una ruta en la classe.
- **creaArchives():** Aquest mètode s'encarrega de comprovar que els arxius de ranking clàssic existeixin. Si no existeixen, els crea.
- **guardarRanking(ArrayList<String> datos):** Aquest mètode s'encarrega de guardar l'ArrayList que conté la informació del ranking amb el path guardat anteriorment a la classe.
- **carregarRanking():** Aquest mètode s'encarrega de carregar el contingut del ranking guardat i retornarlo. Retorna un ArrayList de strings amb el contingut del ranking seleccionat.