**Documentació: Entorns de desenvolupament**

### RA5 – Diagrames UML

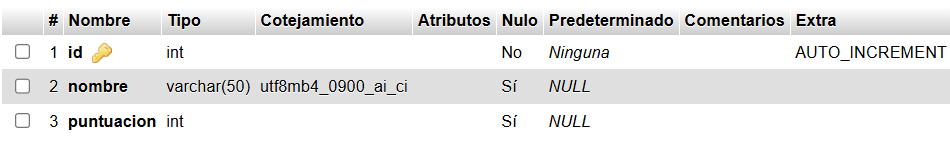
Un UML és un diagrama que representa visualment com està dissenyat un sistema, mostrant-ne les parts i com es relacionen. Al llarg d’aquest projecte hem realitzat un total de tres UML.

Per què tres UML?

* Abans: Per planificar i definir la idea inicial.
* A mig projecte: Per reflectir canvis i ajustar el disseny segons l’avanç.
* Al final: Per documentar el disseny real del projecte acabat.

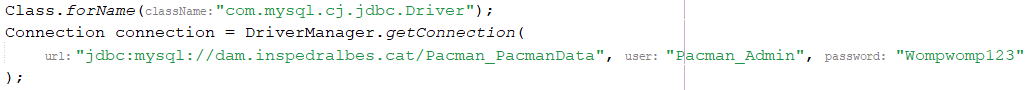
**Programació**

A l’hora d’utilitzar la Base de Dades en el nostre projecte de Pacman, vam arribar a la conclusió d'utilitzar-lo per emmagatzemar el nom de l’usuari i la seva puntuació amb el propòsit de tenir una taula de classificacions amb els 10 millors jugadors amb major puntuació.



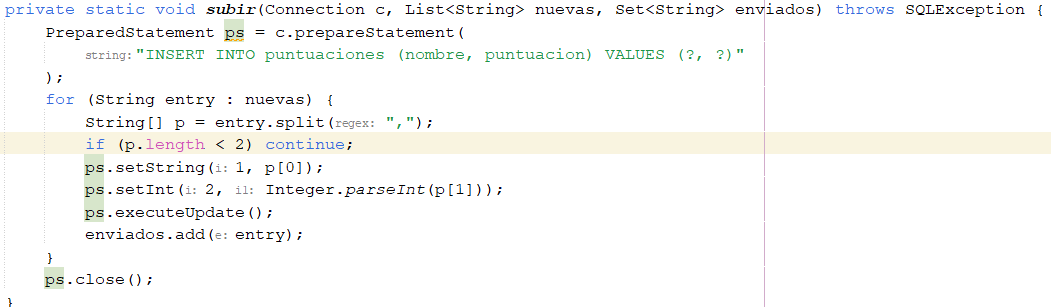
### RA8-RA9 – Connexió amb BDD

Per connectar-se a la Base de Dades, requereix una connexió i els permisos requerits per accedir a aquest.



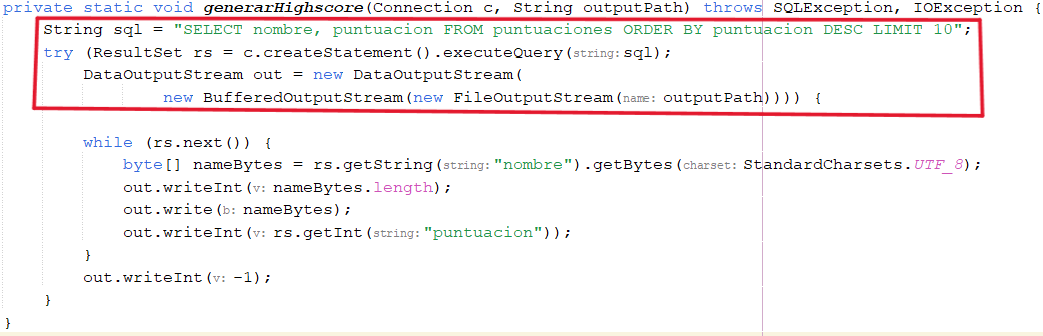
### RA8-RA9 – Inserció d'informació en la BDD

Per inserir informació a la Base de Dades, s’ha d’agafar la connexió que ja havíem creat abans per fer una declaració (“Statement”). En aquest cas, es crea una funció per inserir automàticament en nom amb la seva puntuació.



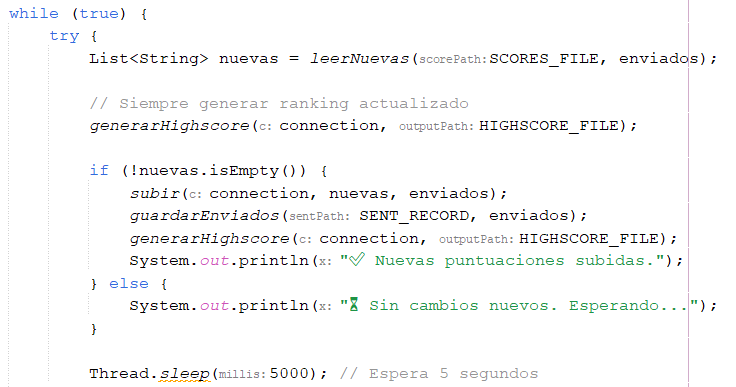
### RA8-RA9 – Recuperació d'informació d'una BDD

Per recuperar informació de la Base de Dades, es fa com abans, es fa una declaració a partir de la connexió amb la Base de Dades. En aquest cas, S’agafa la informació dels 10 jugadors amb la puntuació més alta.



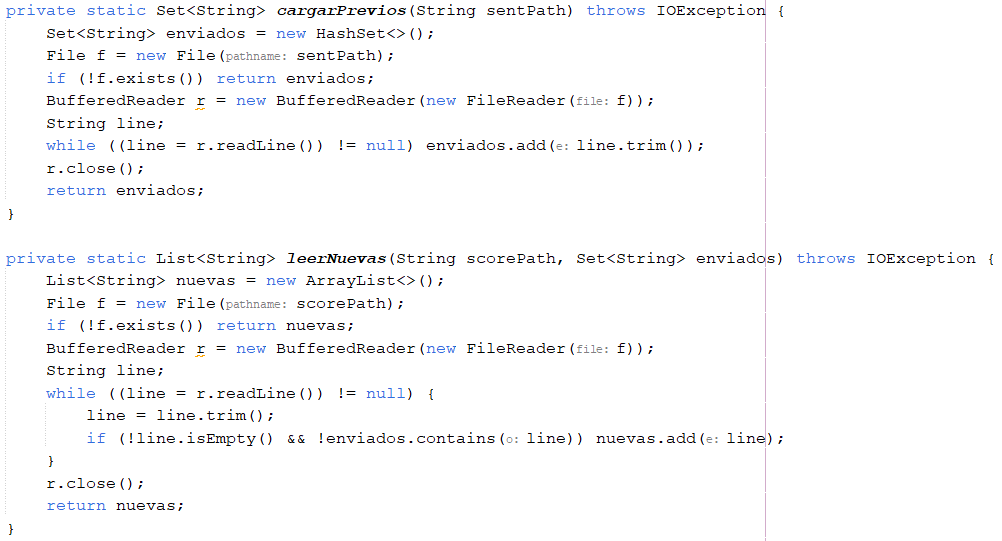
### RA8-RA9 – Actualització d'informació en una BDD

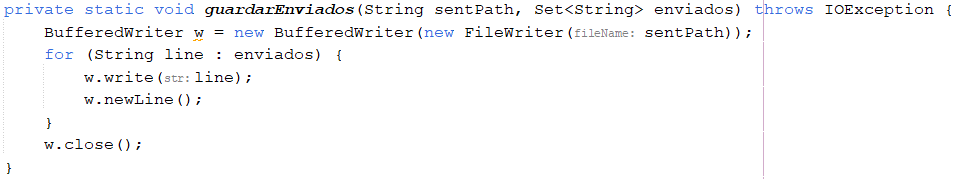
Per que la Base de Dades estigui sempre actualitzada, es comprova cada cert temps si hi ha nova informació. En cas que hi rebi de nova, l'actualitzarà i marcarà aquella informació com a actualitzada. Per tant, no s’enviarà la mateixa informació dues vegades a la base de dades.



### RA8-RA9 – Integració de la informació en un programa

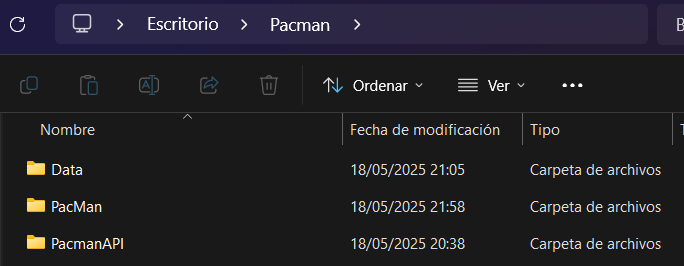
Tal com vaig comentar abans, una vegada s'hagin enviat, es marca i es guarda com a informació ja enviada. Per tant, no es pot enviar més d’una vegada.





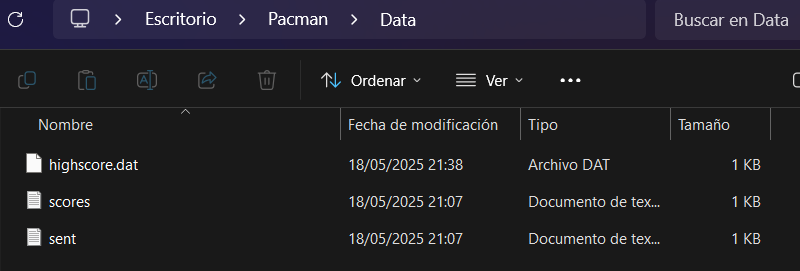
### RA5 – Entrada/sortida del llenguatge i les llibreries associades.

Per transportar la informació des de la BDD al Joc, es necessita una carpeta on s’enviarà la informació necessària perquè els dos programes facin el que necessitin. En aquest cas, es crearà la carpeta anomenada Data.



### RA5 – Fitxers per emmagatzemar i recuperar informació.

En cas de l’arxiu java, generarà un .dat amb les millors puntuacions per que el programa de Unity pugui llegir-les. I en cas del Unity, generarà un archiu .txt amb les puntuacions i els noms per guardar-les a la BDD.



### RA5 – Programes que utilitzen mètodes d'accés al contingut dels fitxers.

Per accedir a aquests, java ho detecta perquè és un String. En cas del Unity, ho ha de desxifrar per a després passar-ho a String i int.

