2. Descripció de les Estructures de Dades i algorismes utilitzats per a implementar les funcionalitats principals

La funcionalitat principal, la qual s'encarrega de generar un horari mitjançant unes dades d'entrada: aules, assignatures, matèries i restriccions (que son editables tan manualment com dinàmicament), més les dades referents a les suavitzacions d'una matèria (per no tenir en compte alguna restricció de grup, nivel o correquisit entre matèries) i les restriccions entre materies afegides (restriccions explicites de que una matèria no pot anar amb una altre).

El programa genera com a resultat una estructura de dades que guarda per a cada franja horària, les aules assignades en aquella hora amb l'assignatura associada:

Map<FranjaHoraria, Map<capaDatos.Aula, capaDatos.Asignatura>> cjt_asignaciones;

La gestió d'aquesta estructura de dades queda vinculada a la clase *CjtAsignaciones,* d'on l'algorisme es capaç d'omplir-la desde *controladorDominio* mitjançant les funcions:

public void addelement(FranjaHoraria fh, capaDatos.Aula au, capaDatos.Asignatura a)

public void delelement(FranjaHoraria fh, capaDatos.Aula au, capaDatos.Asignatura a)

Que afegeixen i esborren un element del mapa, respectivament.

Desde la funció *Generar(int i)* de controladorDominio, es calcula el backtracking desde i...n en el vector d'assignatures, assignant per cadascuna, una aula en tantes franges horàries com hores de classe tingui l'assignatura. S'utilitza la classe Restriccions per realitzar les comprovacions i la classe CjtAssignatures ja esmentada per guardar la solució.

Per cada assignatura, la funció *Generar(int i)* envia una comprovació a comprueba_restricciones de Restriccions per obtenir les primeres n franges horàries seguides que siguin vàlides, on n = número d'hores de classe de l'assignatura; junt amb l'aula assignada. Si rep una franja horària i una aula, l'algorisme es cridarà recursivament amb i+1; sinó, la funció retornarà false. Si un cop executat i+1, retorna true, l'algorisme retornarà true; mentre que si retorna false, buscarà la següent franja horària vàlida comprovant les restriccions desde la primera franja trobada +1. La funció pararà d'executar-se un cop la i sigui igual al tamany del vector d'assignatures a assignar.

La classe Restriccions guardarà una estructura de dades de suport durant l'execució, que ens servirà per tal de preveure quina serà la franja horària vàlida per cada assignatura. Per a cada assignatura, es guardarà el conjunt dels índex de franges horàries no permesos, junt amb els índex de les assignatures que provoquen la invalidació.

```
private ArrayList<Map<Integer, ArrayList<Integer>>> forwardChecking;
```

Comprueba_restricciones buscarà les primeres n franges horàries lliures, on n = al nombre d'hores de classe de l'assignatura, es a dir, no marcades en el forwardChecking, i que tinguin alguna aula del mateix tipus i amb capacitat apte, i retornarà una d'aquestes aules amb la primera franja horària trobada.

Abans de realitzar el retorn, l'algorisme marcarà les assignatures de i+1...n en el forwardChecking; només les franges horàries utilitzades en el cas que hi hagi alguna restricció que faci que no puguin anar juntes.

Si l'algorisme fa backtracking, es desmarcaran les franges marcades per l'assignació eliminada.