

## Especificació detallada del Model Conceptual de Dades (Capa Domini)

### Classes

#### **Classe:** PlaEstudis

PlaEstudis permet agrupar les Assignatures en conjunts de dades independents d'altres plans d'estudis.

#### **Atributs:**

- id (identificador)
- assignatures: és una estructura de dades que emmagatzema tots els objectes Assignatura pertanyents a un PlaEstudis.
- horari: quan es genera un horari pel PlaEstudis es guarda l'objecte Horari generat en aquest atribut per poder-lo consultar posteriorment.

#### **Mètodes:**

- PlaEstudis(String id): mètode creador de PlaEstudis.
- Void addAssignatura(Assignatura assig): afegeix una nova assignatura al PlaEstudis.
- Boolean eliminarAssignatura(String id): indica si s'ha pogut eliminar l'Assignatura amb identificador *id* del PlaEstudis.
- Boolean existsAssignatura(String id): indica si el PlaEstudis conté una Assignatura amb identificador *id*.
- String getID()
- Boolean hasHorari(): indica si PlaEstudis ha generat un Horari vàlid durant l'execució del programa.
- Void showAssignatures(): imprimeix pel canal *out* informació de les Assignatures contingudes pel PlaEstudis.
- String toStringAssignatures(): retorna un String amb informació concissa sobre quines assignatures conté PlaEstudis.
- String detallsAssignatura(String idAssig): retorna informació detallada de l'Assignatura *idAssig* si existeix en PlaEstudis.
- String[] toStringNomsAssignatures(): retorna un array de Strings només amb els identificador de les Assignatures de PlaEstudis.
- Void generarHorari(): genera un horari pel PlaEstudis i el guarda en l'atribut *horari*.
- Void afegirCorrequisits(String[] c): afegeix restriccions de correquisit en les Assignatures identificades per *c*.
- String getHorari(): retorna en un String l'horari del PlaEstudis prèviament generat.

#### **Classe:** Horari

La classe Horari conté l'algorisme de backtracking encarregat de trobar un horari vàlid donat el conjunt de dades de totes les Assignacions.

#### **Atributs:**

- conjuntAssignacions: conté una estructura de dades amb tots els objectes Assignacions sobre els qui ha de processar per assignar-los una Classe vàlida.

#### **Mètodes:**

- Horari(ArrayList<Assignacio> conjuntAssignacions): creadora de Horari.

- Void afegeixAssignacions(ArrayList<Assignacions> conjuntAssignacions): afegeix més Assignacions a *conjuntAssignacions* de Horari.
- Void findHorari(): inicia l'algorisme per generar un horari.
- Boolean selectClasse(int index): tracta l'Assignacio referenciada per *index* i li assigna un horari vàlid (és a dir, li assigna els objectes Classe necessaris).
- Stack<Classe> forward\_checking(Classe c): Donat la nova Classe reservada *c* elimina de totes les Assignacio les possibles Classe que han deixat de ser vàlides.
- Boolean chekNotEmpty(): indica si per totes les Assignacio que falten per tractar encara existeixen possibles Classe entre les quals escollir un horari.
- Void revertChanges(Stack<Classe> eliminades): reverteix les accions executades per l'algorisme de forward checking sobre *eliminades*.
- Void printHorari(): imprimeix pel canal *out* l'horari generat.
- Void printHorariAules().
- Void printHorariAssignatures().
- String toString(): retorna un String amb l'horari generat.

### **Classe:** Assignatura

Conté informació sobre l'assignatura així com restriccions que l'afecten directament.

### **Atributs:**

- id (identificador)
- nom
- nivell: nivell al que pertany l'assignatura.
- classes\_teoría: nombre de classes de teoria que han d'impartir els grups de l'assignatura.
- classes\_problemes: nombre de classes de problemes que han d'impartir els grups de l'assignatura.
- classes\_laboratori: nombre de classes de laboratori que han d'impartir els grups de l'assignatura.
- Duracio\_teoría: duració de les classes de teoria.
- Duracio\_problemes: duració de les classes de problemes.
- Duracio\_laboratori: duració de les classes de laboratori.
- Grups: estructura de dades que conté els grups de l'assignatura.
- Correquisits: restricció de correquisit de l'assignatura.

### **Mètodes:**

- Assignatura(String id, String nom, int nivell): creadora d'Assignatura.
- String GetId()
- int getNivell()
- void crearGrups(TreeMap<String, Grup> grups): es guarden els grups de Assignatura.
- Void setClasses(int nTeoria, int dTeoria, int nLaboratori, ind dLaboratori, int nProblemes, int dProbelemes)
- void showClasses(): imprimeix detalls de les classes de Assignatura.
- Void showGrups(): imprimeix informació dels grups de Assignatura.
- Void noSolapis\_Teoría\_i\_Problemes(): afegeix restriccions als grups de Assignatura

perquè les classes de teoria no es solapin amb els seus subgrups.

- `ArrayList<Assignacio> getAssignacions()`: retorna tot els conjunt d'Assignacions producte de relacionar Assignatura amb els seus grups.
- `Void addCorrequisit(String new_correquisit)`: afegeix una nova assignatura com a correquisit d'Assignatura.
- `Void addManyCorrequisits(ArrayList<String> new_correquisits)`: afegeix noves assignatures com a correquisits d'Assignatura.
- `String toString()`: retorna un String amb la informació bàsica de Assignatura.
- `String toStringCompleto()`: retorna un String amb informació detallada de Assignatura.

**Classe:** Grup

Implementa l'abstracció de "grup d'una assignatura".

**Atributs:**

- `idGrup`
- `capacitat`
- `hoariGrup`: indica si és de matins, tardes o ambdós.
- `Subgrup`: restricció de subgrup
- `tipus`: indica el tipus d'aula que necessita per poder fer classe.

**Mètodes:**

- `grup(String idGrup, int capacitat, String horariGrup, Tipus_Aula tipus)`: creadora de Grup.
- `String GetId()`
- `int getCapacitat()`
- `String getHorariAssig()`: indica en quina franja horària està assignat Grup
- `Tipus_Aula getTipus()`
- `RestriccioSubgrup getSubgrup()`: retorna la restricció subgrup de Grup.
- `Void afegirRestriccio(RestriccioSubgrup r)`
- `String toString()`

**Classe:** Aula

Implementa l'abstracció d'una Aula.

**Atributs:**

- `id` (identificador)
- `capacitat`
- `tipus`: indica de quin tipus és l'aula

**Mètodes:**

- `Aula(String id, int capacitat, Tipus_Aula tipus)`: constructura
- `String getId();`
- `int getCapacitat();`
- `Tipus_Aula getTipus();`
- `String toString();`

**Classe:** Classe

Aquesta Classe serveix per relacionar Aula amb una data concreta i el grup d'una assignatura, per tant és la unitat de dades que processa l'algorisme a l'hora d'assignar un hoari a una assignació.

**Atributs:**

- id\_assig
- id\_grup
- id\_aula
- hora\_inici
- hora\_fi
- dia

**Mètodes:**

- Classe(String id\_assig, String id\_grup, DiaSetmana dia, int inici, int fi, String id\_aula)
- DiaSetmana getDia()
- int getHoralnici()
- int getHoraFi()
- String getId\_assig()
- String getIdAula()
- int getDurada(): retorna la durada de la classe.
- Void showClasse(): imprimeix informació de Classe.
- String toString(): retorna en un String informació de Classe

**Classe:** Assignacio

Assignacio conté informació sobre assignatura i grup els quals són els seus identificadors. A més, ens serveix com a objecte on crear l'horari per al grup d'una assignatura, és a dir, Assignacio reuneix el conjunt d'objectes Classe que formen l'horari d'un grup.

**Atributs:**

- idGrup
- idAssig
- capacitat
- nivellAssig
- tAula: Tipu d'aula que necessita el grup d'Assignacio.
- HorariGrup: indica a quin interval de temps pertany el grup (matins o tardes).
- NumeroClasses: nombre de classes que necessita aquesta Assignacio.
- DuracioClasse: la durada de les classes d'aquest grup.
- NumeroClassesRestants: indica el nombre de classes que falta per assignar a l'horari del grup.
- inici\_possible: primera hora del dia que el grup pot fer classes.
- final\_possible: última hora del dia que el grup pot acabar les classes.
- possibles\_classes: conjunt d'objectes Classe que Assignació pot escollir per impartir classes i montar el seu horari.
- classes\_seleccionades: conjunt d'objectes Classe que Assignació ha escollit pel seu horari.
- Ocupacio: restricció que comprova si l'aula de Classe ja està reservada.
- Corequisit: restricció que comprova els corequisits d'Assignacio.
- Subgrup: restricció que comprova la restricció subgrup d'Assignacio.

**Mètodes:**

- Assignacio(String idGrup, Tipus\_Aula tAula, String id\_Assig, int nivellAssig, int numeroClasses, int duracioClasses, String horariGrup)
- String getIdAssig()
- String getIdGrup()
- int getNumeroClassesRestants()
- ArrayList<Classe> getAllPossiblesClasses()
- void afegirCorrequisit(Correquisit c)
- void afegirSubgrup(RestriccioSubgrup r)
- Map<String,Map<DiaSetmana,ArrayList<Classe>>>) generarPossiblesClasses(): genera tots els possibles objectes Classe entre els quals pot escollir Assignacio.
- Void printPossiblesClasses()
- void showAll()
- void afegeixPossibilitat(Classe c): afegeix la Classe c al conjunt de possibles classes d'Assignacio.
- ArrayList<Classe> forwardChecking(Classe c): dur a terme un forward checking tenint en compte que s'acaba de seleccionar la Classe c per l'horari d'Assignacio.
- ArrayList<Classe> nomesSeleccionades(): elimina totes les possibles classes d'Assignacio menys les seleccionades per l'horari.
- Void afegirSeleccionada(Classe c): afegeix una Classe al conjunt de seleccionades.
- Void eliminarSeleccionada(Classe c)
- boolean esMatins().
- ArrayList<Classe> getSeleccionades()
- boolean isEmpty(): indica si Assignacio encara poseeix objectes Classe entre les quals escollir classes pel seu horari.
- String toString()

**Classe:** Restriccio (Abstracte)

Abstraeix els tipus de restriccions que existeixen.

**Atributs:****Mètodes:**

- boolean solapenHores(int ai, int bi, int af, int bf): indica si un interval d'hores es solapa.
- ArrayList<Classe> deletePossibilities(Map<String,Map<DiaSetmana,ArrayList<Classe>>>, Classe c): et retorna tots els objectes Classe que Assignacio ha d'eliminar de les seves possibles classes perquè incompleixen restriccions.

**Classe:** RestriccioSubgrup

Implementa la restricció subgrup.

**Atributs:**

- pare: és l'objecte grup d'Assignació.

**Mètodes:**

- RestriccioSubgrup(Grup pare)
- String toString().

**Classe:** RestriccioCorequisit

**Atributs:**

- assignatures: el conjunt d'assignatures que té Assignacio com a correquisit.

**Mètodes:**

- boolean isEmpty(): indica si el correquisit té assignatures o està buit.
- Boolean addAssignatura(String id\_assignatura): afegeix una assignatura al correquisit.
- String toString()

**Classe:** RestriccioOcupacio

Implementa la restricció ocupació d'aules.

**Atributs:**

**Mètodes:**