

# CLASSES

## domain.classes

### Group.class

Representa un grup.

#### Atributs

<b>code [String]:</b>	Codi identificador únic. "LP40T", "LP41L"
<b>dayPeriod [Enum]:</b>	Període del dia en que ha d'anar el grup. "MORNING", "AFTERNOON" o "INDIFERENT".
<b>lectures [Llista de Strings]:</b>	Codi de les sessions del grup per cada setmana.
<b>numPeople [Integer]:</b>	Num. persones que pertanyen al grup.
<b>parentGroupCode [String]:</b>	Codi del grup pare al qual pertany. "LP40T", "F20T"
<b>subject [String]:</b>	Codi de l'assignatura a la qual pertany. "LP", "F"
<b>type [Enum]:</b>	Tipus de grup. "THEORY", "PROBLEMS"...

#### Mètodes

**Group(String code, Integer numPeople, String parentGroupCode, String subject, Type type, DayPeriod dayPeriod, ArrayList<String> lectures)**

code - Codi del grup.

numPeople - Núm. persones en el grup.

parentGroupCode - Codi del grup pare.

subject - Assignatura a la que pertany.

type - Tipus de grup.

dayPeriod - Període del dia.

lectures - Conjunt de sessions del grup.

Constructora estàndard.

#### **getCode()**

Retorna el codi del grup.

#### **getDayPeriod()**

Retorna el període del dia en que aquest grup pot fer classe.

#### **getLectures()**

Retorna la llista de lectures del grup.

#### **getNumOfPeople()**

Retorna el núm. de persones del grup.

***getParentGroupCode()***

Retorna el codi del grup pare.

***getSubject()***

Retorna el codi de l'assignatura del grup.

***getType()***

Retorna el tipus de classe del grup.

***toString()***

Retorna el String que identifica el grup.

**Lecture.class**

Representa una "lecture". Una classe setmanal d'un grup.

**Atributs**

***duration [Integer]:*** Duració de la sessió.

***group [String]:*** Grup al qual pertany la sessió. "LP40T", "LP41L"

***id [Boolean]:*** Codi identificador de la sessió dins del grup en el que està.

**Mètodes**

***Lecture(Integer id, String group, Integer duration)***

id - Identificació de la sessió.

group - Grup al qual pertany.

duration - Duració de la sessió.

Constructora estàndard.

***getDuration()***

Retorna el codi de l'aula.

***getGroup()***

Retorna el núm. màxim d'estudiants que hi caben.

***toString()***

Retorna el String que identifica la sessió.

**PosAssig.class**

Representa un conjunt d'assignacions candidates (aules i hores/dies) per quan es genera un horari.

**Atributs**

***assigMap [Map]:*** Mapa de les poss. assign. d'una sessió d'un grup en concret.

## **Mètodes**

**PosAssig(Map<Integer,Map <Integer, Set<String>>> assigMap)**

assigMap - Conjunt de possibles assignacions.

***dayIsEmpty(Integer dia)***

dia - Dia a mirar.

Indica si a posAssigMap, a un dia determinat no hi ha possibles assignacions.

***getMap()***

Retorna les possibles assignacions.

***hasDay(Integer dia)***

dia - Dia a mirar.

Donat un dia, indica si aquest està a posAssigMap o no.

***hasHourFromDay(Integer dia, Integer hora)***

dia - Dia a mirar.

hora - Hora a mirar.

Donat un dia i una hora, indica si aquesta està a posAssigMap o no.

***hasNoDays()***

Indica si a posAssigMap no hi ha possibles assignacions.

***hasRoomFromDayAndHour(Integer dia, Integer hora, String aula)***

dia - Dia a mirar.

hora - Hora a mirar.

aula - Aula a mirar.

Donat un dia, hora i aula, indica si posAssigMap conté l'aula donada en aquell dia i hora.

***removeDay(Integer dia)***

dia - Dia a mirar.

Elimina de posAssigMap totes les possibles assignacions per al dia donat

***removeHourFromDay(Integer dia, Integer hora)***

dia - Dia a mirar.

hora - Hora a mirar.

Elimina de posAssigMap totes les possibles assignacions per al dia i hora donats.

### ***removeRoomFromHourAndDay(Integer day, Integer hour, String room)***

day - Dia a mirar.

hour - Hora a mirar.

room - Aula a mirar.

Elimina de posAssigMap l'aula donada en el dia i hora donats

## **Restriction.class**

Representa una restricció.

### **Atributs**

***enabled [Boolean]:*** Indica si la restricció està activada.

***negotiable [Boolean]:*** Indica si la restricció és negociable.

### **Mètodes**

***disable()*** Desactiva la restricció.

***enable()*** Activa la restricció.

***isEnabled()*** Retorna si la restricció està activada.

***isNegotiable()*** Retorna si la restricció és negociable.

***toString()*** Retorna el String que identifica la restricció.

## **Room.class**

Representa una aula.

### **Atributs**

***code [String]:*** Codi identificador únic. "A6101", "A5E02"

***capacity [Integer]:*** Núm. màxim d'estudiants que hi caben. 20, 65

***hasComputers [Boolean]:*** Indica si l'aula disposa d'ordinadors per als alumnes.

### **Mètodes**

***public Room(String code, Integer capacity, Boolean hasComputers)***

code - Codi de l'aula.

capacity - Capacitat de l'aula.

hasComputers - Indica si l'aula té ordinadors o no.

Constructora estàndard.

### ***getCode()***

Retorna el codi de l'aula.

### ***getCapacity()***

Retorna el núm. màxim d'estudiants que hi caben.

***hasComputers()***

Retorna true si l'aula disposa d'ordinadors per als alumnes.

***toString()***

Retorna el String que identifica l'aula.

**Schedule.class**

Representa un horari.

**Atributs**

***schedule [Map de String[][]]***:Conté la informació de l'horari.

**Mètodes*****Schedule()***

Constructora estàndard.

***Schedule(String json)***

json - String que representa l'horari.

Constructora estàndard.

***getSchedule()***

Retorna l'horari.

***putLecture(String room, int day, int hour, String group)***

room - Room a mirar.

day - Dia a mirar.

hour - Hora a mirar.

group - Grup a posar.

Afegeix una sessió del grup a l'horari (en l'aula, hora i dia donats).

***removeLecture(String room, int day, int hour)***

room - Room a mirar.

day - Dia a mirar.

hour - Hora a mirar.

Elimina una assignació de l'horari (en l'aula, hora i dia donats).

***toJsonString()***

Retorna l'horari en format JSON.

***equals(Schedule sc)***

sc - Paràmetre a comparar.

True si sc és igual que this. Fals en cas contrari.

## **Subject.class**

Representa una assignatura.

### **Atributs**

- code [String]:** Codi identificador únic. "PROP", "M2"
- name [String]:** Nom complet. "Projectes de Programació", "Matemàtiques 2"
- level [String]:** Nivell de l'assignatura en el pla docent. "Q1", "Q5"
- coreqs [Llista de Strings]:** Codis de les assignatures que son corequisit d'aquesta.  
"['FM', 'M1']"
- groups [Llista de Strings]:** Codis dels grups d'aquesta assignatura.  
"['LP40T', 'LP41L', 'LP42L']"

### **Mètodes**

**Subject(String code, String name, String level, ArrayList<String> groups, ArrayList<String> coreqs)**

code - Codi de l'assignatura.

name - Nom complet de l'assignatura.

level - Nivell de l'assignatura.

groups - Grups de l'assignatura.

coreqs - Assignatures corequisites de l'assignatura que estem creant.

Creadora estàndard.

#### ***getCode()***

Retorna el codi de l'assignatura.

#### ***getCoreqs()***

Retorna la llista de correquisits de l'assignatura.

#### ***getGroups()***

Retorna la llista de grups de l'assignatura.

#### ***getLevel()***

Retorna el nivell de l'assignatura.

#### ***getName()***

Retorna el nom de l'assignatura.

#### ***toString()***

Retorna el String que identifica l'assignatura.

## **domain.classes.restrictions**

### **NaryRestriction.class**

Classe que engloba totes les restriccions n-àries.

#### **Atributs**

#### **Mètodes**

##### **NaryRestriction(boolean negotiable)**

negotiable - Indica si la restricció que estem mirant és negociable o no.

Constructora estàndard.

##### **validate(String lecture, String room, Integer day, Integer hour, Map<String, Subject> subjects, Map<String, Group> groups, Map<String, Lecture> lectures, Map<String, PosAssig> pAssigMap)**

lecture - Sessió que hem afegit a l'horari.

room - Aula on hem afegit la sessió.

day - Dia en el que hem afegit la sessió.

hour - Hora en la que hem afegit la sessió.

subjects - Conjunt d'assignatures de l'entorn.

groups - Conjunt de grups de l'entorn.

lectures - Conjunt de sessions de l'entorn.

pAssigMap - Conjunt de possibles assignacions per a cada sessió.

Elimina les assignacions que es consideraven possibles però que ja no compleixen la restricció un cop inserida la última sessió. Retorna:

- True si, un cop eliminat les aules en les que cada sessió no podia anar, totes les sessions restant poden anar com a mínim a una aula.
- False en cas contrari.

Les aules que s'eliminaran dependran de la restricció que estiguem validant.

### **CorrequisitRestriction.class**

Restricció sobre els correquisits de les assignatures. Dos grups amb el mateix parentCode de dues assignatures correquisites entre elles no poden anar a un mateix dia i hora. Ex: PROP 10 T no pot anar al mateix dia i hora que TC 15 P, però sí pot anar amb F 11 L i FM 10 T.

### **Atributs**

### **Mètodes**

#### **CorrequisitRestriction()**

Constructora estàndard.

#### ***toString()***

Retorna el String que identifica la restricció.

#### ***validate(String lecture, String room, Integer day, Integer hour, Map<String, Subject> subjects, Map<String, Group> groups, Map<String, Lecture> lectures, Map<String, PosAssig> pAssigMap)***

lecture - Sessió que hem afegit a l'horari.

room - Aula on hem afegit la sessió.

day - Dia en el que hem afegit la sessió.

hour - Hora en la que hem afegit la sessió.

subjects - Conjunt d'assignatures de l'entorn.

groups - Conjunt de grups de l'entorn.

lectures - Conjunt de sessions de l'entorn.

pAssigMap - Conjunt de possibles assignacions per a cada sessió.

Valida totes les possibles assignacions de les lectures restants sabent que hem afegit una lecture en un dia, hora i aula concrets.

### **LectureFromSameGroupOverlapRestriction.class**

Restricció sobre un mateix grup. Diverses sessions d'un mateix grup no poden anar en el mateix dia i hora. Ex: FM 10 T de 2 hores no pot anar al mateix dia i hora que FM 10 T de 1 hora.

### **Atributs**

### **Mètodes**

#### **LectureFromSameGroupOverlapRestriction()**

Constructora estàndard.

#### ***toString()***

Retorna el String que identifica la restricció.



***validate(String lecture, String room, Integer day, Integer hour, Map<String, Subject> subjects, Map<String, Group> groups, Map<String, Lecture> lectures, Map<String, PosAssig> pAssigMap)***

lecture - Sessió que hem afegit a l'horari.

room - Aula on hem afegit la sessió.

day - Dia en el que hem afegit la sessió.

hour - Hora en la que hem afegit la sessió.

subjects - Conjunt d'assignatures de l'entorn.

groups - Conjunt de grups de l'entorn.

lectures - Conjunt de sessions de l'entorn.

pAssigMap - Conjunt de possibles assignacions per a cada sessió.

True si, un cop eliminat les aules en les que cada sessió no podia anar, totes les sessions restant poden anar com a mínim a una aula. False en cas contrari. Les aules que s'eliminen per cada sessió restant són aquelles que farien anar dues sessions del mateix grup al mateix dia i hora.

## **ParentGroupOverlapRestriction.class**

Restricció sobre els grups. Un subgrup (31, 45...) no pot anar al mateix dia i hora que el seu grup pare (30, 40...). Ex: FM 11 L no pot anar al mateix dia i hora que FM 10 T, però sí pot anar amb FM 12 L.

### **Atributs**

### **Mètodes**

#### **ParentGroupOverlapRestriction()**

Constructora estàndard.

#### ***toString()***

Retorna el String que identifica la restricció.

***validate(String lecture, String room, Integer day, Integer hour, Map<String, Subject> subjects, Map<String, Group> groups, Map<String, Lecture> lectures, Map<String, PosAssig> pAssigMap)***

lecture - Sessió que hem afegit a l'horari.

room - Aula on hem afegit la sessió.

day - Dia en el que hem afegit la sessió.

hour - Hora en la que hem afegit la sessió.

subjects - Conjunt d'assignatures de l'entorn.

groups - Conjunt de grups de l'entorn.

lectures - Conjunt de sessions de l'entorn.

pAssigMap - Conjunt de possibles assignacions per a cada sessió.

True si, un cop eliminat les aules en les que cada sessió no podia anar, totes les sessions restant poden anar com a mínim a una aula. False en cas contrari. Les aules que s'eliminen

per cada sessió restant són aquelles que farien que un grup pare i un subgrup anessin al mateix dia i hora.

## **SubjectLevelRestriction.class**

Restricció sobre el nivell de les assignatures. Dos grups d'assignatures del mateix nivell no poden anar al mateix dia i hora si tenen el mateix codi i són del mateix tipus. Ex: FM 10 T no pot anar al mateix dia i hora que PRO1 10 T, però sí pot anar amb PRO1 11 L i PRO1 20 T.

### **Atributs**

### **Mètodes**

#### **SubjectLevelRestriction()**

Constructora estàndard.

#### **toString()**

Retorna el String que identifica la restricció.

***validate(String lecture, String room, Integer day, Integer hour, Map<String, Subject> subjects, Map<String, Group> groups, Map<String, Lecture> lectures, Map<String, PosAssig> pAssigMap)***

lecture - Sessió que hem afegit a l'horari.

room - Aula on hem afegit la sessió.

day - Dia en el que hem afegit la sessió.

hour - Hora en la que hem afegit la sessió.

subjects - Conjunt d'assignatures de l'entorn.

groups - Conjunt de grups de l'entorn.

lectures - Conjunt de sessions de l'entorn.

pAssigMap - Conjunt de possibles assignacions per a cada sessió.

True si, un cop eliminat les aules en les que cada sessió no podia anar, totes les sessions restant poden anar com a mínim a una aula. False en cas contrari. Les aules que s'eliminen per cada sessió restant són aquelles que farien anar dos grups, amb mateix codi i tipus, d'assignatures del mateix nivell al mateix dia i hora.

## **OccupiedRoomRestriction.class**

Restricció sobre la disponibilitat de les aules. Un grup no pot anar a una aula ocupada.

### **Atributs**

### **Mètodes**

#### **OccupiedRoomRestriction()**

Constructora estàndard.

### ***toString()***

Retorna el String que identifica la restricció.

***validate(String lecture, String room, Integer day, Integer hour, Map<String, Subject> subjects, Map<String, Group> groups, Map<String, Lecture> lectures, Map<String, PosAssig> pAssigMap)***

lecture - Sessió que hem afegit a l'horari.

room - Aula on hem afegit la sessió.

day - Dia en el que hem afegit la sessió.

hour - Hora en la que hem afegit la sessió.

subjects - Conjunt d'assignatures de l'entorn.

groups - Conjunt de grups de l'entorn.

lectures - Conjunt de sessions de l'entorn.

pAssigMap - Conjunt de possibles assignacions per a cada sessió.

True si, un cop eliminat les aules en les que cada sessió no podia anar, totes les sessions restant poden anar com a mínim a una aula. False en cas contrari. Les aules que s'eliminen per cada sessió restant són aquelles que farien anar dos grups al mateix dia, hora i aula.

## **UnaryRestriction.class**

Classe que engloba totes les restriccions un-àries.

### **Atributs**

### **Mètodes**

**UnaryRestriction(boolean negotiable)**

negotiable - Indica si la restricció que estem mirant és negociable o no.

Constructora estàndard.

**validate(Integer day, Integer hour, Integer duration)**

day - Dia en que, combinat amb l'hora, no pot anar el grup.

hour - Hora en que, combinada amb el dia, no pot anar el grup.

duration - Duració del grup.

True si el grup es pot posar en el day i hour. False en cas contrari.

## **DayPeriodRestriction.class**

Restricció sobre el nivell de les assignatures. Dos grups d'assignatures del mateix nivell no poden anar al mateix dia i hora si tenen el mateix codi i són del mateix tipus. Ex: FM 10 T no pot anar al mateix dia i hora que PRO1 10 T, però sí pot anar amb PRO1 11 L i PRO1 20 T.

### **Atributs**

**dayPeriod [Group.DayPeriod]:** Període del dia en el que ha d'anar el grup.  
**midDay [Integer]:** Hora en que passem a classes de tarda.

### **Mètodes**

**DayPeriodRestriction(Integer midDay, Group.DayPeriod dayPeriod)**

midDay - Hora en que passem a classes de tarda.

dayPeriod - Període del dia en el que ha d'anar el grup.

Constructora estàndard.

### ***toString()***

Retorna el String que identifica la restricció.

**validate(Integer day, Integer hour, Integer duration)**

day - Dia en que, combinat amb l'hora, no pot anar el grup.

hour - Hora en que, combinada amb el dia, no pot anar el grup.

duration - Duració del grup.

Retorna true si

- El grup ha d'anar de matins i  $hour + duration \leq midDay$ .
- El grup ha d'anar de tardes i  $midDay < (hour + duration) \leq$  Última hora de classes del dia.
- El grup pot anar tant de matins com tardes i  $hour + duration \leq$  Última hora de classes del dia.

False en cas contrari.

## **SpecificDayOrHourRestriction.class**

Restricció sobre el dia i l'hora dels grups. Un grup no podrà anar a un dia i hora que es digui. Ex: FM 10 T no pot anar Dilluns a les 09:00.

### **Atributs**

**day [Integer]:** Dia.

**hour [Integer]:** Hora.

### **Mètodes**

**SpecificDayOrHourRestriction(Integer day, Integer hour)**

day - Dia

hour - Hora

Constructora estàndard.

### ***toString()***

Retorna el String que identifica la restricció.

### **validate(Integer day, Integer hour, Integer duration)**

day - Dia en que, combinat amb l'hora, no pot anar el grup.

hour - Hora en que, combinada amb el dia, no pot anar el grup.

duration - Duració del grup.

True si day i hour són el dia i l'hora en el que el grup no pot anar. False en cas contrari.

## **domain.controllers**

### **CtrlDomain.class**

Controlador principal de domini de l'aplicació.

#### **Atributs**

**instance [CtrlDomain]:** Instància del controlador de domini.

**dataController [CtrlData]:** Instància del controlador de dades.

**subjects [Map<String, Subject>]:** Conjunt d'assignatures.

**rooms [Map<String, Room>]:** Conjunt d'aules.

**groups [Map<String, Group>]:** Conjunt de grups.

**lectures [Map<String, Lecture>]:** Conjunt de sessions.

**schedule [Schedule]:** Horari.

**unaryRestrictions [Map<String, Map<String, UnaryRestriction>>]:** Conjunt de restriccions un-àries..

**naryRestrictions [Map<String, NaryRestriction>]:** Conjunt de restriccions n-àries.

#### **Mètodes**

##### **CtrlDomain()**

Constructora estàndard.

##### **getInstance()**

Retorna la instància d'aquesta classe.

##### **getRoomNamesList()**

Retorna una llista de codis de les aules de l'entorn del domini.

##### **getGroupNamesList()**

Retorna una llista de codis dels grups de l'entorn del domini.

##### **getSubjectNamesList()**

Retorna una llista de codis d'assignatures de l'entorn del domini.

##### **getNaryRestrictions()**

Retorna una llista de codis de les restriccions n-àries de l'entorn del domini.

**erase()**

Borra les restriccions un-àries.

**getUnaryRestrictions()**

Retorna una llista de codis de les restriccions un-àries de l'entorn del domini.

**generateSchedule()**

Genera un horari amb les condicions de l'entorn.

**scheduleToJsonString()**

Retorna l'horari del domini convertit en Json.

**printSchedule()**

Imprimeix l'horari en una taula.

**importEnvironment(String filename)**

filename - Nom de l'entorn a importar.

Importa un entorn (aules, assignatures, aules) desde un arxiu.

**importSchedule(String filename)**

filename - Nom de l'arxiu d'horari a importar.

Importa un horari desde un arxiu.

**exportSchedule(String filename)**

param - filename Nom amb el que guardar l'horari.

Guarda l'horari de l'entorn a un arxiu.

**getEnvironmentFilesList()**

Retorna una llista de noms d'arxius d'entorn disponibles.

**getScheduleFilesList()**

Retorna una llista de noms d'arxius d'horari disponibles.

**CtrlSchedule.class**

Controlador principal de domini de l'aplicació.

**Atributs**

**instance [CtrlSchedule]:** Instància del controlador de schedule.

**Mètodes****CtrlSchedule()**

Constructora estàndard.

**generateSchedule(Map<String, Map<String, UnaryRestriction>> unaryRestrictions, Map<String, NaryRestriction> naryRestrictions, Map<String, Group> groups, Map<String, Room> rooms, Map<String, Subject> subjects, Map<String, Lecture> lectures, Schedule schedule)**

unaryRestrictions - Conjunt de restriccions unaries a tenir en compte.

naryRestrictions - Conjunt de restriccions n-aries a tenir en compte.

groups - Conjunt de grups que es volen incloure en l'horari.

rooms - Conjunt d'aules que es volen incloure en l'horari.

subjects - Conjunt de assignatures que tenen algun grup que es vol incloure a l'horari.

lectures - Conjunt de 'lectures' que es volen incloure en l'horari.

schedule - Objecte de la classe Schedule que contindrà l'horari generat al finalitzar l'algoritme.

Funció principal que genera un horari amb les dades donades.

Retorna true si s'ha trobat un horari valid, sino false.

**forwardCheck(String lecture, String room, Integer day, Integer hour, Map<String, Subject> subjects, Map<String, Group> groups, Map<String, Lecture> lectures, Map<String, PosAssig> pAssigMap, Map<String, NaryRestriction> naryRestrictions)**

lecture - Última 'lecture' que s'ha assignat a l'horari, i es vol comprovar la seva validesa.

room - Aula en la qual s'ha assignat la 'lecture'.

day - Dia en el que s'ha assignat la 'lecture'.

hour - Hora en la que s'ha assignat la 'lecture'.

subjects - Conjunt de assignatures que tenen algun grup que es vol incloure a l'horari.

groups - Conjunt de grups que es volen incloure en l'horari.

lectures - Conjunt de 'lectures' que es volen incloure en l'horari.

pAssigMap - Map que conté les possibles assignacions de cada 'lecture'.

naryRestrictions Conjunt de restriccions n-aries a tenir en compte.

Funció que comprova la validesa de l'última assignació i redueix les possibles assignacions si és possible utilitzant el mètode validate() de les restriccions..

Retorna true si l'assignació a comprovar és vàlida, sinó false.

**backjumping(Schedule schedule, PriorityQueue<Map.Entry<Integer, String>> heuristica, Map<String, Subject> subjects, Map<String, Group> groups, Map<String, Lecture> lectures, Map<String, PosAssig> pAssigMap, Map<String, NaryRestriction> naryRestrictions)**

schedule - Objecte de la classe Schedule que contindrà l'horari generat al finalitzar l'algoritme.

heuristica - Priority Queue de 'lectures', on les que tenen menor nombre de possibles assignacions estan a dalt.

subjects - Conjunt de assignatures que tenen algun grup que es vol incloure a l'horari.

groups - Conjunt de grups que es volen incloure en l'horari.

lectures - Conjunt de 'lectures' que es volen incloure en l'horari.

pAssigMap - Map que conté les possibles assignacions de cada 'lecture'.

naryRestrictions - Conjunt de restriccions n-aries a tenir en compte.

Funció que fa Backjumping, utilitzat en l'algoritme que genera un horari.  
Retorna true si s'ha generat correctament. False si no s'ha pogut generar cap horari valid.

## **persistence**

### **CtrlData.class**

Restricció sobre el dia i l'hora dels grups. Un grup no podrà anar a un dia i hora que es digui. Ex: FM 10 T no pot anar Dilluns a les 09:00.

#### **Atributs**

**day [CtrlData]:** Instància de la classe Data.

#### **Mètodes**

##### **getInstance()**

Retorna la instància d'aquesta classe.

##### **CtrlData()**

Constructora estàndard.

##### **readEnvironment(String filename)**

filename - Nom de l'arxiu d'entorn a importar.

Importa un entorn (aules, assignatures, aules) desde un arxiu.

##### **readSchedule(String filename)**

filename - Nom de l'arxiu d'horari a importar.

Importa un horari desde un arxiu.

##### **writeSchedule(String filename, String content)**

filename - Nom amb el que guardar l'horari.

content - Contingut de l'horari a guardar.

Guarda el contingut donat (que representa un horari) a un arxiu dins la carpeta d'horaris.

##### **readData(String filename)**

filename - Adreça de l'arxiu a llegir.

Llegeix els continguts d'un arxiu.

##### **writeData(String filename, String content)**

filename - Adreça de l'arxiu a on escriure (si no existeix es crearà).

Escriu un String a un arxiu local.



**getEnvironmentFilesList()**

Retorna una llista de noms d'arxius d'entorn disponibles.

**getScheduleFilesList()**

Retorna una llista de noms d'arxius d'horari disponibles.

**getFilesList(String path)**

path - Adreça de la carpeta.

Retorna una llista de noms d'arxius que es troben dins d'una carpeta.

Per a més informació consultar **javadoc**.