**GENERADOR D’HORARIS**

Andrea Querol de Porras – andrea.querol

Víctor Sánchez Gassull – victor.sanchez.gassull

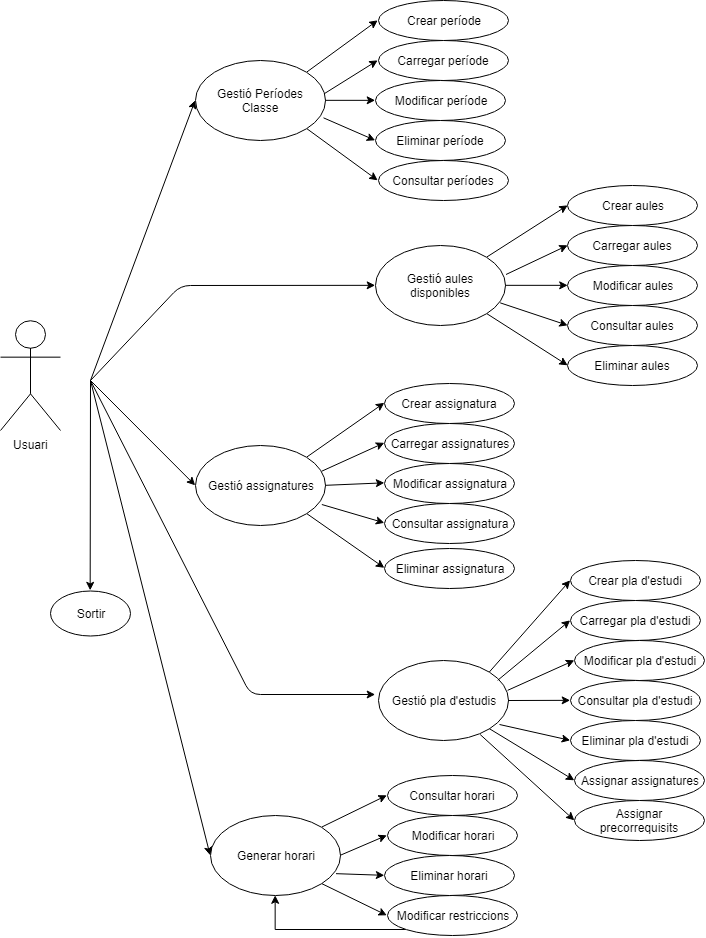
Bernat Torres Bellido – bernat.torres.bellido

Lliurament 1.0

**Índex**

# 1. Diagrama de casos d’ús

## 1.1 Diagrama



## Descripció

### 1.2.1 Gestió Quadrimestres Classe

Comportament: permet a l’usuari visualitzar els quadrimestres de classe creats fins al moment, seleccionar un i modificar-lo, consultar-lo o eliminar-lo; i també permet crear o carregar un nou quadrimestre. Si no hi ha cap generat, mostra directament l’opció de crear un quadrimestre o carregar-ne un.

### 1.2.2 Crear Quadrimestre

Crear quadrimestre: l’usuari haurà d’introduir els paràmetres necessaris per crear un quadrimestre: data d’inici i data final. El sistema valida valors i coherència de dades, i els registra.

* Error: Ja existeix un quadrimestre amb aquestes dates: canviar-les o abandonar.

### 1.2.3 Carregar quadrimestre

Carregar quadrimestres: l’usuari haurà de seleccionar l’arxiu que desitja utilitzar, on haurà de contenir la data d’inici i data final en el format requerit, pot tenir més d’un quadrimestre. El sistema llegeix l’arxiu, valida els valors i coherència de dades, i els registra, creant tants quadrimestres com contingui l’arxiu.

* Error: Ja existeix un quadrimestre amb aquestes dates: canviar-les o abandonar.

### 1.2.4 Modificar quadrimestre

L’usuari veu els valors que conté actualment el quadrimestre i permet introduir una nova data d’inici i/ data final. El sistema valida els nous valors i coherència de les dades, i els modifica.

* Error: la nova data d’inici és posterior a la data final existent: canviar la data d’inici, modificar la data final o abandonar.
* Error: la nova data final és anterior a la data d’inici existent: canviar la data final, modificar la data d’inici o abandonar.
* Error: les noves dates d’inici i data fi coincideixen amb un quadrimestre ja existent: canviar-les o abandonar.

### 1.2.5 Consultar quadrimestre

L’usuari visualitza les dates d’inici i final del quadrimestre.

### 1.2.6 Eliminar quadrimestre

L’usuari veu el botó d’eliminar, abans d’eliminar totalment el quadrimestre ha d’indicar que està segur de voler eliminar-lo, el sistema avisa que és un canvi irreversible. El sistema elimina el quadrimestre.

### 1.2.7 Gestió aules disponibles

Comportament: permet a l’usuari visualitzar les aules creades fins al moment, seleccionar una i modificar-la, consultar-la o eliminar-la; i també permet crear o carregar una nova aula. Si no hi ha cap generada, mostra directament l’opció de crear una nova aula o carregar-ne una.

### 1.2.8 Crear aula

L’usuari haurà d’introduir els paràmetres necessaris per crear una aula: nom, capacitat, tipus de l’aula i característiques. Seleccionarà les característiques: projector, ordinadors amb Ubuntu, ordinadors amb Linux i Windows, material de física, material d’embeded o material per xarxes, mitjançant checkboxes.

/\*Si és de tipus teoria ha d’indicar si l’aula té projector, a través d’un radio button. Si és de tipus laboratori ha d’introduir si té: projector, ordinadors amb Ubuntu, ordinadors amb Linux i Windows, material de física, material d’embeded o material per xarxes. Aquestes característiques de les aules es seleccionaran mitjançant checkboxes.\*/

* Error: el nom de l’aula no segueix el patró de noms del Campus Nord: introduir-lo seguint el patró o abandonar.
* Error: ja existeix una aula amb el nom especificat: canviar-lo o abandonar.

### 1.2.9 Carregar aula

L’usuari haurà de seleccionar l’arxiu que desitja utilitzar, on haurà de contenir el nom, capacitat i tipus d’aula (teoria o laboratori). També haurà d’estar indicat si té: projector, ordinadors amb Ubuntu, ordinadors amb Linux i Windows, material de física, material d’embeded o material per xarxes. Tota aquesta informació haurà d’estar en el format requerit; pot tenir més d’una aula. El sistema llegeix l’arxiu, valida els valors i coherència de dades, i els registra, creant tantes assignatures com contingui l’arxiu.

/\*Si és de tipus teoria haurà d’estar indicat si té projector o no; altrament, haurà de contenir la informació sobre el laboratori: projector, ordinadors amb Ubuntu, ordinadors amb Linux i Windows, material de física, material d’embeded o material per xarxes. Tota aquesta informació haurà d’estar en el format requerit; pot tenir més d’una aula. El sistema llegeix l’arxiu, valida els valors i coherència de dades, i els registra, creant tantes assignatures com contingui l’arxiu.\*/

* Error: Ja existeix alguna aula amb un dels noms especificats: canviar-lo, no crear aquella aula o abandonar.

### 1.2.10 Modificar aula

L’usuari veu els valors que conté actualment l’aula i permet canviar el nom, capacitat, tipus i les corresponents característiques (projector, Ubuntu, Linux i Windows, física, embeded, xarxes). El sistema valida els nous valors i coherència de les dades, i els modifica.

* Error: el nom de l’aula coincideix amb el d’una aula ja existent: canviar-lo o abandonar.
* Error: el nom de l’aula no segueix el patró de noms del Campus Nord: introduir-lo seguint el patró o abandonar.

### 1.2.11 Consultar aula

L’usuari veu el valor de les dades d’aula.

### 1.2.12 Eliminar aula

L’usuari veu el botó d’eliminar, abans d’eliminar totalment l’ayla ha d’indicar que està segur de voler eliminar-la, el sistema avisa que és un canvi irreversible. El sistema elimina l’aula.

### 1.2.13 Gestió assignatures

L’usuari pot visualitzar les assignatures creades fins al moment, seleccionar una i modificar-la, consultar-la o eliminar-la; i també permet crear o carregar una nova assignatura. Si no hi ha cap generada, mostra directament l’opció de crear una nova aula o carregar un arxiu.

### 1.2.14 Crear assignatura

L’usuari haurà d’introduir els paràmetres necessaris per crear una assignatura: nom, quadrimestres que estarà disponible (menú de selecció múltiple de Q1 i Q2), nivell dins del pla d’estudi (menú de selecció única: fase inicial, fase obligatòria i especialitat) i les característiques necessàries per les classes de teoria i laboratori (dos menús de selecció múltiple, un per teoria i un per laboratori: projector, ordinadors amb Ubuntu, ordinadors amb Linux i Windows, material de física, material d’embeded i material per xarxes. El sistema valida valors i coherència de dades, i els registra.

/\* Haurà d’indicar per les classes de teoria: hores setmanals i si necessita projector. També per les classes de laboratori haurà d’introduir: nombre d’hores setmanals, quantitat de laboratoris i què necessiten les aules (menú de selecció múltiple: projector, Ubuntu, Linux/Windows, física, embeded, xarxes). El sistema valida valors i coherència de dades, i els registra.\*/

* Error: Ja existeix una assignatura amb aquest nom: canviar-lo o abandonar.

### 1.2.15 Carregar assignatures

L’usuari haurà de seleccionar l’arxiu que desitja utilitzar, on haurà de contenir el nom, quadrimestres disponible, nivell dins del pla d’estudis i les característiques requerides per les aules de teoria i laboratori. Tota aquesta informació haurà d’estar en el format requerit; pot tenir més d’una assignatura. El sistema llegeix l’arxiu, valida els valors i coherència de dades, i els registra, creant tantes assignatures com contingui l’arxiu.

/\*requerida per les classes de teoria: hores setmanals i si necessita projector; i la informació de les classes de laboratori: hores setmanals, quantitat de laboratoris i necessitats de l’aula. Tota aquesta informació haurà d’estar en el format requerit; pot tenir més d’una assignatura. El sistema llegeix l’arxiu, valida els valors i coherència de dades, i els registra, creant tantes assignatures com contingui l’arxiu.\*/

* Error: Ja existeix alguna assignatura amb un dels noms especificats: canviar-lo, no crear aquella assignatura o abandonar.

### 1.2.16 Modificar assignatura

L’usuari veu els valors que conté actualment l’assignatura i permet canviar el nom, quadrimestres disponibles, nivell dins del pla d’estudis, les característiques d’aules de teoria i/o de laboratori. El sistema valida els nous valors i coherència de les dades, i els modifica.

* Error: el nom de l’assignatura coincideix amb el d’una assignatura ja existent: canviar-lo o abandonar.

### 1.2.17 Consultar assignatura

L’usuari visualitza els valors de les dades de l’assignatura.

### 1.2.18 Eliminar assignatures

L’usuari veu el botó d’eliminar, abans d’eliminar totalment l’assignatura ha d’indicar que està segur de voler eliminar-la, el sistema avisa que és un canvi irreversible. El sistema elimina l’assignatura.

### 1.2.19 Gestió pla d’estudis

L’usuari visualitzar els plans d’estudi creats fins al moment, seleccionar un i modificar-lo, consultar-lo, eliminar-lo, assignar-hi assignatures o assignar precorrequisits entre les assignatures del pla d’estudis. També permet crear o carregar un nou pla d’estudis. Si no hi ha cap generat, mostra directament l’opció de crear un nou pla d’estudis o carregar un arxiu.

### 1.2.20 Crear pla d’estudis

L’usuari haurà d’introduir el paràmetre necessaris per crear un pla d’estudis: el nom. El sistema valida el valor i coherència de dades, i el registra.

* Error: Ja existeix un pla d’estudis amb aquest nom: canviar-lo o abandonar.

### 1.2.21 Carregar plans d’estudis

L’usuari haurà de seleccionar l’arxiu que desitja utilitzar, on haurà de contenir el nom en el format requerit, pot tenir més d’un quadrimestre. A més, també pot tenir les assignatures que conté i la relació de precorrequisits entre elles. Pot contenir més d’un pla d’estudis. El sistema llegeix l’arxiu, valida els valors i coherència de dades, i els registra, creant tants plans d’estudis com contingui l’arxiu i fent les assignacions necessàries si en conté.

* Error: Ja existeix un pla d’estudis amb un dels noms especificats: canviar-lo, no crear aquell pla docent o abandonar.
* Error: ja s’ha assignat prèviament a un pla d’estudis una de les assignatures que figuren a l’arxiu: canviar-la a l’arxiu, eliminar l’assignació del pla d’estudis on està assignada o abandonar.

### 1.2.22 Modificar pla d’estudi

L’usuari veu els valors que conté actualment el quadrimestre i permet introduir un nou nom, eliminar assignacions d’assignatures (si en contenia). El sistema valida els nous valors i coherència de les dades, i els modifica.

* Error: la nova data d’inici és posterior a la data final existent: canviar la data d’inici, modificar la data final o abandonar.
* Error: la nova data final és anterior a la data d’inici existent: canviar la data final, modificar la data d’inici o abandonar.
* Error: les noves dates d’inici i data fi coincideixen amb un quadrimestre ja existent: canviar-les o abandonar.

### 1.2.21 Consultar pla d’estudis

L’usuari veu el nom del pla d’estudis i les assignatures que conté.

### 1.2.22 Eliminar pla d’estudis

L’usuari veu el botó d’eliminar, abans d’eliminar totalment el quadrimestre ha d’indicar que està segur de voler eliminar-lo, el sistema avisa que és un canvi irreversible. El sistema elimina el pla d’estudis.

### 1.2.23 Assignar assignatures

L’usuari veu les assignatures creades fins al moment i selecciona, mitjançant checkboxes, quines formen part del pla d’estudis.

### 1.2.24 Assignar precorrequisits

L’usuari visualitza les assignatures que hi ha al pla d’estudis. Podrà introduir la relació precorrequisit entre dues assignatures escrivint en la casella A el nom de l’assignatura que és precorrequisits de l’assignatura escrita a la casella B.

* Error: l’assignatura A no correspon amb cap assignatura del pla docent: canviar el nom introduït o abandonar.

### 1.2.25 Generar horari

L’usuari visualitza els noms dels plans d’estudi i mitjançant checkboxes podrà escollir quins plans d’estudi vol incloure en el generador.

* Error: no es pot generar cap horari amb les aules creades i els plans docents i restriccions seleccionats: suavitzar el conjunt de restriccions o abandonar.

### 1.2.26 Consultar horari

L’usuari visualitza els horaris generats fins al moment (veu el nom dels plans d’estudi que conté) i selecciona quin desitja consultar. Veurà l’horari en forma de taula: la primera columna contindrà les hores i la primera fila els dies de la setmana.

### 1.2.27 Modificar horari

L’usuari visualitza els horaris generats fins el moment i escull quin desitja modificar. Es carrega l’horari amb la mateixa visualització que en consultar horari, però podrà seleccionar qualsevol sessió i desplaçar-la pel horari i col·locar-la on desitgi. El sistema valida els nous valors i coherència amb els requisits, i els modifica.

* Error: La sessió que es vol moure no compleix els requisits indicats a la nova posició de l’horari: canviar-la a una altra franja horària o abandonar.

### 1.2.28 Eliminar horari

L’usuari veu el botó d’eliminar, abans d’eliminar totalment l’horari ha d’indicar que està segur de voler eliminar-lo, el sistema avisa que és un canvi irreversible. El sistema elimina el quadrimestre.

### 1.2.29 Modificar restriccions

L’usuari visualitza totes les restriccions amb checkboxes. Les que s’havien seleccionat per generar l’horari estaran marcades. L’usuari podrà marcar i desmarcar les restriccions. El sistema modifica les dades i genera un altre horari.

* Error: No es possible crear un horari amb les noves restriccions: suavitzar-les o abandonar.

# 2. Diagrama estàtic complet del model conceptual de dades

## 2.1 Diagrama UML

## 2.2 Descripció de les classes i funcions

S’han implementat constructores, getters i setters de totes les classes. Ometrem l’explicació d’aquestes funcions.

### 2.2.1 Assignatura

* isEqual(Assignatura a): compara dues assignatures basant-se en el nom, nivell i el quadrimestre en el que es donen.

### 2.2.2 Aula

Només té getters i setters.

### 2.2.CjtRestriccions

* borrarRS (int i) : borra l’element ièssim del vector de restriccióSolapar
* borrarRM (int i) : borra l’element ièssim del vector de restriccióMatins
* borrarRT (int i) : borra l’element ièssim del vector de restriccióTardes
* borrarRCA (int i) : borra l’element ièssim del vector de restriccióCaracterístiquesAula
* borrarRN (int i) : borra l’element ièssim del vector de restriccióNivell
* borrarRC (int i) : borra l’element ièssim del vector de restriccióCorrequisit
* addRS (RestriccióSolapar r): afegir la restricció passada per paràmetre al vector restriccióSolapar
* addRM (RestriccióMatins r): afegir la restricció passada per paràmetre al vector restriccióMatins
* addRT (RestriccióTardes r): afegir la restricció passada per paràmetre al vector restriccióTardes
* addRCA (RestriccióCaracterístiquesAula r): afegir la restricció passada per paràmetre al vector restriccióCaracterístiquesAula
* addRN (RestriccióNivell r): afegir la restricció passada per paràmetre al vector restriccióNivell
* addRC (RestriccióCorrequisit r): afegir la restricció passada per paràmetre al vector restriccióCorrequisit
* comprovarRestriccionsColocar(Sessio actual, ArrayList<Sessio> sessions, Integer dia, Integer hora, Horari horari): recorre tots els vectors de restriccions i comprova que en cap cas no es compleix la restricció, si es troba un que no la compleix retorna false.
* comprovarRestriccionsAula(Sessio actual, Aula aula): recorre el vector de característiques d’aula per comprovar que en tots els casos es compleixi la restricció Característica aula.

### 2.2.3 CjtUnitatDocent

* del (UnitatDocent ud): elimina ud del vector d’unitats docents de la classe
* get (integer i): retorna la ièsima unitat docent de l’arraylist d’unitats docents
* add (UnitatDocent ud): afegir ud a l’arraylist d’unitats
* size(): retorna la quantitat total d’unitats docents de l’arraylist d’unitats docents.

### 2.2.4 CtrlDomini

* afegirAulaUnitatDocent(String nom, Integer capacitat, ArrayList<CaracteristiquesAula> caracteristiques): afegeix una aula (nom, capacitat, característiques) a la unitat docent seleccionada.
* afegirAssignaturaPlaEstudis(String nom, Integer quadri, Integer nivell, ArrayList<CaracteristiquesAula> teo, ArrayList<CaracteristiquesAula> lab): afegeix una assignatura (nom, quadri nivell, teo, lab) al pla d’estudis seleccionat
* afegirAssignaturaPlaEstudis(String nom, Integer quadri, Integer nivell, String plaEstudis, ArrayList<CaracteristiquesAula> teo, ArrayList<CaracteristiquesAula> lab): afegeix una assignatura (nom, quadri, nivell, pla Estudis, teo, lab) al pla d’estudis seleccionat.
* afegirSessioQuadrimestre(Integer grup, String nomAssignatura): afegeix una sessió (grup, nomAssignatura) al quatrimestre seleccionat.
* existeixPlaEstudis (String nom): busca el pla d’estudis (nom) i si no el troba retorna false
* existeixAssignatura (String nom): busca si l’assignatura (nom) està al pla d’estudis seleccionat.
* getAssignatura (String nom): retorna l’assignatura (nom) del pla d’estudis seleccionat
* getAssignatura (Integer i): retorna la ièssima assignatura del pla d’estudis seleccionat.
* afegirPlaEstudis (String nom): afegeix un pla d’estudis (nom) a l’arraylist d’Unitat docent i plaEstudisSeleccionat passa a ser aquest.
* afegirUnitatDocent(String nom): afegeix una unitat docent (nom) a l’arraylist unitatsdocents i unitatDocentSeleccionat passa a ser aquesta
* afegirQuadrimestre(): afegeix un quadrimestre nou al pla d’estudis seleccionat i passa a ser el quatrimestre seleccionat.
* borrarUnitatDocent(Integer i): elimina la ièssima unitat docent de l’arraylist unitatsdocents i la unitatDocentSeleccionada passa a ser nul·la.
* llistaSessions(): retorna una concantenació de strings dels paràmetres de les sessions del quatrimestre seleccionat.
* crearRestriccioSolapar(Integer a, Integer b): crea una restriccióSolapar amb les sessions ièssima a i ièssima b del quatrimestre seleccionat. L’afegeix al conjunt de restriccions.
* crearRestriccioCorrequisit(Integer a, Integer b): crea una restriccióCorrequisit amb les sessions ièssima a i ièssima b del quatrimestre seleccionat. L’afegeix al conjunt de restriccions.
* crearRestriccioNivell(Integer a, Integer b): crea una restriccióNivell amb les sessions ièssima a i ièssima b del quatrimestre seleccionat. L’afegeix al conjunt de restriccions.
* crearRestriccioCaracteristicaAula(Integer a, ArrayList<CaracteristiquesAula> caracteristiques): crea una restriccióCaracterísticaAula amb la sessions ièssima a del quatrimestre seleccionat i característiques. L’afegeix al conjunt de restriccions.
* crearRestriccioTardes(Integer a): crea una restriccióTardes amb la sessió ièssima a del quatrimestre seleccionat i l’afegeix al conjunt de restriccions.
* crearRestriccioMatins(Integer a): crea una restriccióMatins amb la sessió ièssima a del quatrimestre seleccionat i l’afegeix al conjunt de restriccions.
* generarHorari(): genera i mostra un horari amb el pla d’estudis seleccionat, les sessions del quatrimestre seleccionat i les restriccions.
* getLlistaCaracteristiquesLab(String nom): retorna les característiques del laboratori de l’assignatura (nom) del pla d’estudis seleccionat
* getLlistaCaracteristiquesTeoria(String nom): retorna les característiques de la teoria de l’assignatura (nom) del pla d’estudis seleccionat
* midaUnitatsDocents(): retorna la quantiat d’unitats docents que hi ha creades.
* llistaAssignatures(): escriu per pantalla el nom de les assignatures del pla d’estudis seleccionat.
* llistaPlaEstudi(): escriu per pantalla el nom dels plans d’estudis de l’unitat docent seleccionat.
* llistaQuadrimestres(): escriu per pantalla el número dels quadrimestres del pla d’estudis seleccionat.
* llistaUnitatsDocents(): mostra per pantalla les unitats docents creades mostrant el seu nom.
* llistaAules(): escriu per pantalla el nom de les aules de la unitat docent seleccionada.
* borrarPlaEstudis(Integer b): elimina el ièssim pla d’estudis de l’unitat docent seleccionada.

### 2.2.5 Generador

* generarHorari(ArrayList<> aules) : Aquesta funcio s'encarrega d'organitzar els atributs, tot observant si hi ha suficients aules amb material disponible suficients per omplir les assignatures i treu les que no. on després cridarà a la funció que produïrà l'horari.
* potAnarAqui(Sesio sessio, Integer dia, Integer hora, Horari hor) : Aquesta funcio comprovarà si la sessio 'sessio' compleix els requisits per a poder ser inclosa a l'slot amb dia = 'dia' i hora = 'hora'.
* buscarAula (Sessio sessio, Integer dia, Integer hora, Horari hor) : Agafa les aules disponibles d'un slot horari (un dia i una hora) i retorna la primera aula que satisfaci les necessitats de la Sessio 'sessio'
* produirHorari (Horari hor, TaulesAules aules, ArrayList <Sessio> sessions, Integer dia, Integer hora) : comprovarà si es pot crear un horari amb les aules, Assignatures i Restriccions creades. En cas afirmatiu el mostrarà per pantalla.

### 2.2.6 Horari

* clonarHorari () : Retornarà un horari idèntic a l'horari p.i.
* mostrarHorari () : imprimirà per pantallà l'horari

### 2.2.7 Pair

* of(A first, B second): retorna un pair {A, B}
* equals(Object o): retorna si el pair p.i. es igual a l’objecte o.
* toString(): retorna el pair en una cadena de strings.

### 2.2.8 Pla Estudis

* afegirAssignatura () : Afegirà una assignaturà a l'ArrayList d'assignatures que pertanyen al pla d'estudis p.i.
* afegirQuadrimestre () : Afegirà un quadrimestre a l'ArrayList de quadrimestre del pla d'estudis p.i.
* borrarAssignatura () : borrarà una assignatura del pla d'estudis p.i.
* borrarQuadrimestre (): borrarà el quadrimestre de la unitat docent p.i.

### 2.2.9 Quadrimestre

* afegirSessio(Sessio sessio): afegeix una sessió (sessió) a l’arraylist sessions
* getSessio(int i): retorna la sessió ièssima de l’arraylist sessions
* borrarSessio(int i): esborra la sessió ièssima de l’arraylist sessions

### 2.2.10 Restricció

* pertany (Integer h\_i, Integer h\_f, integer h) : comprova si la hora 'h' coincidirà en el rang entre hora inici ('h\_i') i hora final ('h\_f')

### 2.2.11 RestriccióCaracterísticaAula

* CompleixRestriccio(Sessio comp, Aula aula): Comprova si una aula tindrà les mateixes característiques (o més) que les necessitats d'una assignatura.

### 2.2.12 RestriccióCorrequisit

* compleixRestriccio(Sessio actual, ArrayList<Sessio> sessions, Integer dia, Integer hora, Horari horari): Comprova si podrà afegir-se a l'slot horari, tot comparant el seu grup, Assignatura, amb la resta d'assignatures de l'slot horari. Es compararà amb les restriccions de Solapament que s'hagin creat.

### 2.2.13 RestriccióMatins

* compleixRestriccio (Sessio actual, ArrayList <Sessio> sessions, Integer dia, Integer hora, Horari horari): Comprova que la sesso 'actual' estigui a la hora de matí (en cas que li toqués estar de matí).

### 2.2.14 RestriccióNivell

* compleixRestriccio (Sessio actual, ArrayList <Sessio> sessions, Integer dia, Integer hora, Horari horari): la sessió 'actual' es compararà amb totes les sessions d'aquell slot horari i mirarà si hi ha sessions del mateix grup i nivell que ella.

### 2.2.15 RestriccióSolapar

* compleixRestriccio (Sessio actual, ArrayList <Sessio> sessions, Integer dia, Integer hora, Horari horari): donades dues sessions que no es podran solapar entre elles, comprova si la sessio 'actual' podrà solapar-se amb alguna sessio de l'slot horari on es vol afegir ('sessions')

### 2.2.16 RestriccióTardes

* compleixRestriccio (Sessio actual, ArrayList <Sessio> sessions, Integer dia, Integer hora, Horari horari) : Comprova si la sessio actual, en cas de ser d'estar organitzat com a tardes, si la hora en la que l'estem posant és de l'horari de tardes.

### 2.2.17 Sessió

* mostrarSessio() : retornarà un string format pel nom i el grup de la sessio p.i.
* sonIguals(Sessio comp) : compara dues sessions basantse amb el nom de la sessió i el grup a la qual pertanyen.

### 2.2.18 Taula

* getAtoms (Integer dia, Integer hora) : retornarà un arrayList amb els Types que formen la cel·la del dia 'dia' a la hora 'hora'
* borrar (A a, Integer dia, Integer hora) : borrarà un dels valors de la cel·la amb dia = 'dia' i hora = 'hora'

### 2.2.19 Taula Aules

* clonarTaulaAules(): retornarà una TaulaAula igual a la TaulaAula p.i.
* mostrarAtom(Aula aula): retornarà el nom de l’aula (aula).

### 2.2.20 Unitat Docent

* afegirAulaDisponible(Aula aula): afegeix una aula (aula) a l’arraylist d’aules
* afegirPlaEstudis(PlaEstudis pe): afegir un pla d’estudis (pe) a l’arraylist de plans d’estudis
* buscarPlaEstudis (String nomPlaEstudis): busca en el arraylist de plans d’estudis un pla d’estudis (nomPlaEstudis) i si el troba el retorna.
* getAula(String nom): busca en el arraylist d’aules una aula (nom) i si la troba la retorna.
* BorrarPlaEstudis(int i): elimina el ièssim pla d’estudis de l’arraylist de plans d’estudi
* borrarAula(Integer b): elimina la ièssima aula de l’arraylist d’aules.

# 3. Relació de les classes implementades

Les classes que ha implementat1 cadascú, amb els seus respectius drivers han sigut:

* Andrea Querol:
  + .
* Víctor Sánchez:
  + .
* Bernat Torres:
  + .

## 3.1 Andrea Querol

## 3.2 Víctor Sánchez

## 3.3 Bernat Torres

1. Per aquesta entrega la major càrrega d’implementació de classes l’han dut a terme el Víctor Sánchez i el Bernat Torres. L’Andrea Querol tindrà més càrrega a la capa de presentació de la propera entrega.

# 4. Descripció d’estructures i algorisme per la funcionalitat principal

Hem optat per fer servir un algorisme de tipus backtracking amb la variació de forward checking, per tal de generar més ràpid l'horari.

Com funciona l'algorisme?

L'algorisme agafa com a parametres un horari, una taula d'aules, un conjunt de sessions i un dia i hora.

Com hem vist, Horari i TaulaAules segueixen l'estructura de la classe Taula, Horari serà la instància sobre la que estarem generant la solució, i TaulaAules contidrà les Aules diferents per cada hora diferent.

El conjunt de sessions seràn les sessions que hem de posar a l'horari, quan aquest conjunt arribi a 0, haurem acabat i tindrem una solució correcte d'horari.

Aquesta versió a la que troba la primera solució para, més endavant es podran triar diferents solucions.

L'algorisme el que fa internament és, per un slot de l'horari (A la que anomenem ATOM) que l'agafem amb el dia i la hora que ens venen per parametres, iterem per totes les sessions que ens queden per possar i si poden anar en aquell slot, la possem i seguim mirant per les seguents avançant una hora. En cas de no poder anar en aquell slot l'algorisme seguirà provant amb les seguents.

En cas de no trobar una solució correcte, s'informarà a l'usuari que amb la seva configuració no s'ha pogut generar un horari.

Previament a generar l'horari, el programa buscarà casos on hi han sessions que requereixen aules molt estrictes, si es troba una sessió que no te cap aula que se li pot assignar (On produiria que no hi haguès solució a l'horari), es notifica a l'usuari quina sessió no té aules disponibles.