

Treball final de grau

Estudi: Grau en Enginyeria Informàtica

Títol: B2B MANAGMENT

Document: Memòria

Alumne: Adrià Alabau Blanquera

Tutors: Jordi Coll i Mateu Villaret

Departament: Informàtica, Matemàtica Aplicada i Estadística (IMAE)

Àrea: Llenguatges i Sistemes Informàtics

Convocatòria Juny 2020

Índex

1	Introducció	3
1.1	Motivació	3
1.2	Propòsit	4
1.3	Objectius	4
1.4	Punt de partida	4
2	Estudi de Viabilitat	6
2.1	Viabilitat econòmica	6
2.1.1	Personal	6
2.1.2	Equipament	6
2.1.3	Cost total	7
3	Metodologia	8
3.1	Implementació de metodologia	8
3.2	Distribució de tasques	9
4	Planificació	10
4.1	Anàlisi i presa de decisions	10
4.2	Disseny i distribució de tasques	10
5	Marc de treball i conceptes previs	12
5.1	SAT	12
5.2	Scala	13
5.2.1	ScalAT	13
5.3	Java FX i FXML	13
5.3.1	Funcionament	14
5.3.2	FXML	14
5.4	Bussines-to-Bussines	14
6	Requisits del sistema	17
6.1	Requisits funcionals	17
6.2	Requisits no funcionals	17
7	Estudi i decisions	18
7.1	Java FX i FXML	18
7.2	SAT	18
7.3	Scala i ScalAT	18
8	Anàlisi i disseny del sistema	19
8.1	Model de dades	19
8.2	Anàlisi de funcionalitats	20
8.2.1	Editar una sessió	22

8.2.2	Editar el nombre de sessions	23
8.2.3	Creació d'una entitat	24
8.2.4	Creació d'una reunió	24
8.2.5	Fixat d'una reunió	25
8.3	Interfície gràfica	27
8.3.1	Vista Principal	27
8.3.2	Vista editar sessions	30
8.3.3	Editar disponibilitat	31
8.3.4	Editar entitat	31
8.3.5	Editar reunió	33
8.4	Vista horari	34
8.4.1	<i>Drag-and-drop</i>	35
8.5	Model de restriccions	35
8.5.1	Paràmetres	35
8.5.2	Variables	35
8.5.3	Restriccions	36
8.6	Validació de dades	37
9	Implementació i proves	39
9.1	Problemes	39
9.1.1	Sistema de taules	39
9.1.2	Cost en temps de computació	39
9.2	Proves	39
9.2.1	Codificació SAT	39
9.2.2	Software	40
10	Implantació i resultats	43
10.1	Resultats experimentals	43
10.2	Normativa i legislació	45
11	Conclusions	46
12	Treball futur	47
12.1	Millores en la interfície gràfica	47
12.2	Codificació de les restriccions restants	47
12.3	Verificació de dades	47
12.4	Millores en el sistema de taules	47
12.5	Optimització i millora de restriccions	48
13	Bibliografia	49
14	Annex	50
14.1	Model de restriccions complet	50
14.1.1	Paràmetres	50
14.1.2	Variables	50
14.1.3	Restriccions	51
14.2	forum13.original	52
15	Manual d'usuari	57
15.1	Requisits de l'aplicació	57
15.2	Descripció de funcionament	57
15.2.1	Entrada de dades	57
15.2.2	Gestionar Horari	63

Capítol 1

Introducció

Gràcies al constant avanç de la tecnologia i la globalització del comerç, moltes empreses aposten per obrir negocis a altres països i formar aliances entre companyies de diferents parts del món. Això presenta una dificultat logística a l'hora de negociar els possibles contractes o negocis que es formin. En aquest punt van aparèixer les reunions *Bussines-to-Bussines* o B2B.

Les reunions *Bussines-to-Bussines* són una forma d'intercanvi entre dues o més empreses participants, de les quals sol intervenir un membre de cada una de les empreses amb l'objectiu d'exposar, negociar o tancar els detalls d'un contracte o una venda. Gràcies a la curta durada d'aquestes reunions i el fet que tenen lloc en esdeveniments importants, com congressos o conferències, permeten a una mateixa empresa dur a terme multitud de reunions amb un cost logístic reduït.

L'alta concentració d'empreses en un espai reduït, les seves diferències horàries i la seva disponibilitat de personal pot suposar un repte complicat gestionar. Aquest projecte consisteix en desenvolupar una eina gràfica que faciliti la tasca de confeccionar horaris a un organitzador d'esdeveniments tipus B2B.

Confeccionar horaris per un esdeveniment B2B és un problema combinatori dur (NP-hard). Per aquest motiu, utilitzarem una de les tècniques actualment més eficients per resoldre aquest tipus de problemes, que és la aproximació *model-and-solve*. Aquesta aproximació consisteix, en primer lloc, en codificar el problema com a un conjunt de variables i de restriccions sobre aquestes variables, de manera que, un cop es troba una assignació per les variables que satisfan les restriccions, aquesta representa una solució del problema.

Un cop definit el model de restriccions, s'utilitza un *solver* (solucionador) especialitzat de restriccions. En particular, en aquest projecte utilitzarem codificacions cap a el problema de la satisfactibilitat Booleana (SAT). Aquesta aproximació no només és eficient, sinó que ens permetrà delegar la tasca de trobar solucions a un solver especialitzat, i ens podrem centrar en desenvolupar una eina gràfica interactiva i intuïtiva.

1.1 Motivació

He escollit desenvolupar aquest projecte per diversos aspectes. Per una banda, volia posar en pràctica els coneixements apresos en el Grau en Enginyeria Informàtica (GEINF) relatius a l'anàlisi i desenvolupament de projectes, i els obtinguts mitjançant pràctiques a empreses. M'ha permès afrontar les dificultats que apareixen quan desenvolupes l'anàlisi realitzat d'un projecte i com evitar els errors comesos en un futur.

Per altra banda, volia ampliar els conceptes relacionats amb la resolució de problemes de satisfacció

de restriccions i aprendre a codificar-los amb el llenguatge introduït a l'assignatura *Programació Declarativa. Aplicacions*.

La idea principal del projecte és definida per la part dels clients, que necessitaven una aplicació gràfica que permetés resoldre B2B de forma general. També volien que el software permetés fer proves amb instàncies concretes i comparar els resultats obtinguts amb el software especialitzat del qual ja disposen.

1.2 Propòsit

El propòsit principal del projecte és desenvolupar una eina de software que permeti al personal d'un esdeveniment poder gestionar els diferents participants que assistiran i els seus interessos a reunir-se, essent capaç de generar un horari per a les reunions que es sol·licitin respectant els requeriments dels participants i les disponibilitats de taules i de temps.

Per a poder desenvolupar un software amb aquestes característiques, s'haurà de definir el problema plantejat de forma genèrica per tal de poder englobar una gran diversitat de tipus B2B, com per exemple, empreses amb més d'un participant, reunions entre més de dues empreses, la disposició de taules de característiques fixades i diferents, és a dir, tenir en compte que un esdeveniment disposarà de taules amb una capacitat limitada, i per tant, no totes les reunions es podrien dur a terme en una taula de característiques determinades.

Per tal que aquest software sigui capaç de dur les seves tasques a terme de forma eficient i atractiva a l'usuari, s'utilitzarà JavaFX per a la part gràfica, i Scala per codificar el problema de restriccions a SAT i així poder resoldre'l amb un SAT solver.

1.3 Objectius

Els objectius del projecte són els següents:

- La definició d'un problema B2B de forma genèrica, per tal que permeti ser adaptat a les necessitats diversos formats d'esdeveniments i clients.
- Dissenyar i desenvolupar una interfície visual que permeti gestionar els usuaris, les sessions i les reunions necessàries per a dur a terme un esdeveniment B2B.
- Desenvolupar una eina de resolució eficient per a la definició del problema B2B general. Al tractar-se d'un problema de satisfacció de restriccions, haurà de ser dissenyat mitjançant model i resolt via SAT.
- Crear una eina versàtil que permeti prefixar les reunions entrades, donant la possibilitat d'interactuar de forma natural, fent i desfent canvis assistint a l'usuari i verificant que les preferències indicades són possibles.
- Codificar restriccions SAT amb Scala i conèixer la sinergia que existeix entre Java i Scala.
- Aprendre el funcionament de JavaFX i desenvolupar una eina útil utilitzant tots els avantatges que ofereix.

1.4 Punt de partida

El projecte parteix amb la base d'una llibreria escrita en Scala anomenada ScalAT, utilitzada en l'assignatura de *Programació declarativa. Aplicacions*, de quart curs del Grau en Enginyeria Informàtica (GEINF). Aquesta llibreria s'utilitza per poder codificar un problema de satisfacció de restriccions com

ho són els B2B com a una fórmula de lògica proposicional, i fer servir un solver especialitzat per resoldre la fórmula. D'aquesta manera, la tasca de solucionar un problema B2B queda reduïda a codificar el problema amb la llibreria ScalAT, demanar una solució a la llibreria, i interpretar el resultat en termes del problema B2B original.

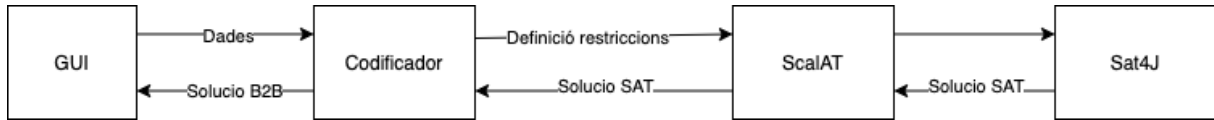


Figura 1.1: Diagrama d'interacció entre els components del software.

Pel que fa a la modalització del problema B2B a SAT, el projecte parteix del model presentat a [2]. De fet, els resultats mostrats en aquests dos treballs [2, 7] permeten constatar que l'aproximació SAT és estat de l'art per a resoldre aquesta mena de problemes. Tot i partir de l'esmentat model, algunes de les generalitzacions que proposem en el present projecte, requeriran petits canvis en algunes de les restriccions.

Capítol 2

Estudi de Viabilitat

Després d'una anàlisi del problema plantejat, i veient que la codificació del model SAT és possible, l'única restricció que es planteja en la viabilitat del projecte és l'econòmica.

2.1 Viabilitat econòmica

Per detallar la viabilitat econòmica del projecte, subdividirem entre costos humans i equipament.

2.1.1 Personal

Les tasques d'anàlisi i desenvolupament del software es poden dividir en tres: Anàlisi, disseny i desenvolupament. El salari per hores de cada un dels membres per hora serà:

- Analista: 20 €/h
- Dissenyador Interfície Gràfica: 12 €/h
- Programador: 12 €/h

Amb aquestes dades podem estimar el temps i preu total de les diferents tasques a dur a terme per cada especialista.

Tasca	Treballador	Durada	Cost
Anàlisi model de restriccions	Analista	20	400
Declaració d'instàncies de proves	Analista	15	300
Anàlisi estructura de l'aplicació	Analista	40	800
Disseny interfície gràfica	Dissenyador	20	240
Codificació model	Programador	40	480
Implementació del disseny visual	Programador	60	720
Implementació estructura de l'aplicació	Programador	90	1080
Depuració de codi i proves de sistema	Programador	30	360
Total		375	4380

2.1.2 Equipament

Els costos d'equipament tenen en conta el hardware i les llicències de software necessàries per el desenvolupament del projecte.

Item	Cost
Ordiandor programador	1450
<i>IntelliJ IDEA Community 2018.1</i>	0
<i>ScalAT</i>	0
<i>Sat4J</i>	0
<i>GitHub</i>	87
<i>Trello</i>	0
Total	1537

2.1.3 Cost total

La suma del cost total del projecte es 5917 €.

Capítol 3

Metodologia

La metodologia de desenvolupament utilitzada ha estat basada en el funcionament de la metodologia àgil *Scrum*. *Scrum* és un procés de bones pràctiques de treball que utilitzen equips de treball, principalment de desenvolupament de software, que consisteix a establir una sinergia horitzontal entre tots els treballadors i evitar els rols preestablerts, és a dir, en un equip en metodologia àgil, tots els membres haurien de poder dur a terme les tasques dels seus companys sense dificultat.

La metodologia *Scrum* utilitza un sistema de treball per *sprints*. Un *sprint* consisteix en un període de temps fixat, entre dues i quatre setmanes, on cada membre de l'equip ha de dur a terme les tasques que li han estat assignades. Habitualment, al final de cada *sprint* es realitza una revisió dels resultats de les tasques, en alguns casos també hi intervenen els clients per valorar el progrés i sol·licitar canvis.

Aquest sistema es considera incremental, ja que a cada *sprint* es poden aplicar canvis al disseny inicial i incloure o eliminar funcionalitats sense alterar de forma greu el ritme de desenvolupament del projecte.

El model basat en *sprints* incorpora un sistema de reunions propi. A l'inici de cada un d'ells, es realitza una reunió formal, de planificació, per a detallar les tasques que corresponent a cada membre i si aquestes seran assequibles en el temps establert. Al final de cada *sprint* es realitza una sessió retrospectiva, en la qual cada membre posa en comú les dificultats trobades així com les variacions que ha patit el seu pla de treball, com tasques afegides no planejades i temps dedicat a assistir als companys entre altres inconvenients. Per últim, es realitza una reunió informal diària, de no més de deu minuts, anomenada *daily*. L'objectiu d'aquesta és la sincronització de l'equip, és a dir, posar en comú problemes trobats, desencallar-los i resoldre tasques pendents entre membres de l'equip.

3.1 Implementació de metodologia

Amb l'ajuda dels dos tutors, Mateu Villaret i Jordi Coll, fent el paper de clients marcant els requisits funcionals de l'aplicació a cada termini, vam establir un sistema d'entregues setmanals de versions per establir un control de l'avenç de l'aplicació i poder modificar el producte de forma incremental.

La metodologia utilitzada no és idèntica a *Scrum*, ja que aquest estableix un mínim de dues setmanes abans de cada revisió, i la distribució de tasques entre els membres de l'equip, en aquest cas, només és un programador. Per altra banda, s'ha cregut convenient descartar les *daily*s, donada la composició de l'equip del projecte, ja que aquestes estan més enfocades a la comunicació entre membres de l'equip i no amb el client.

3.2 Distribució de tasques

Per simplificar la gestió de tasques, s'ha utilitzat l'eina en línia Trello per poder assignar les tasques a realitzar amb més o menys prioritat, les ja realitzades i les que estan actualment en procés.

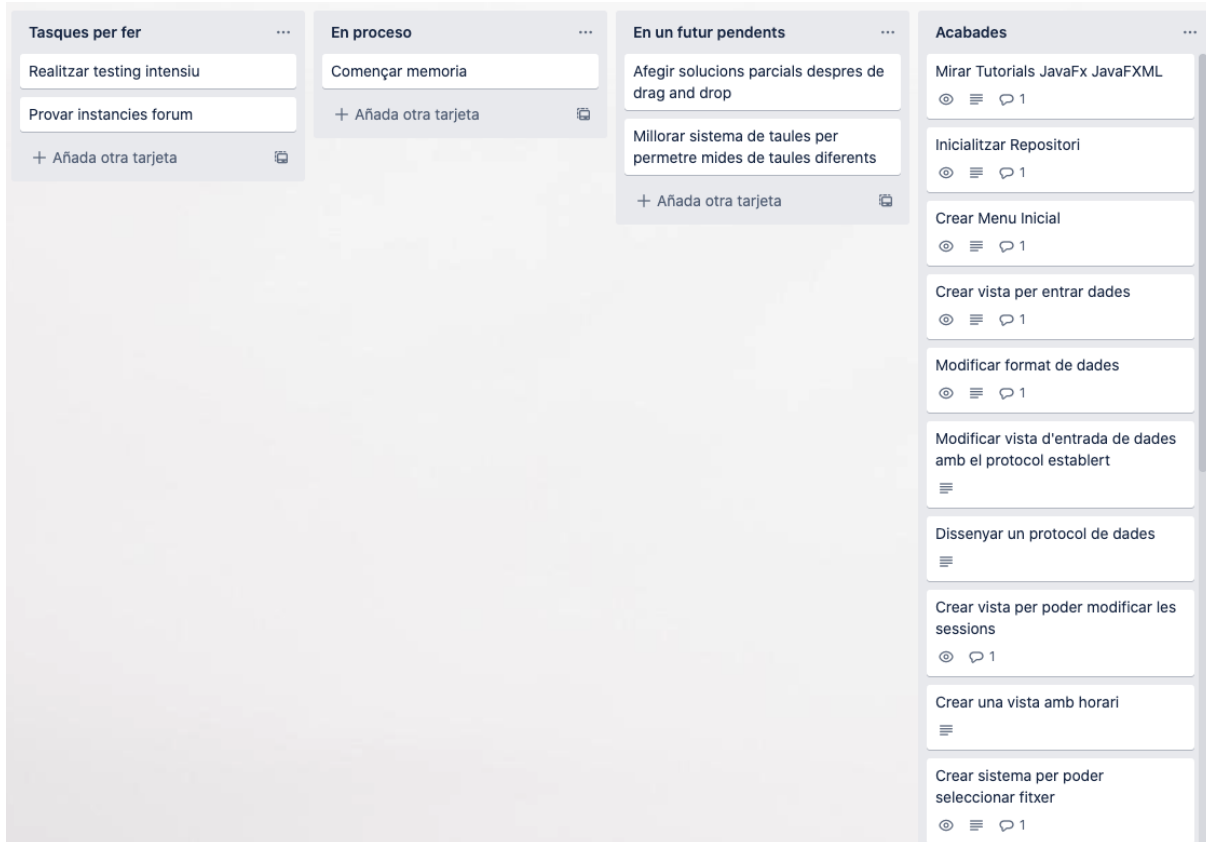


Figura 3.1: Captura de pantalla del gestor de tasques Trello.

Capítol 4

Planificació

En aquest capítol es descriuran els detalls de la planificació del projecte. Donat que aquest utilitza una metodologia àgil, és probable que les tasques establertes per cada bloc es vegin alterades.

4.1 Anàlisis i presa de decisions

La primera part i més important consisteix a analitzar quines de les restriccions es podrien implementar per a la primera versió del software, quin diagrama de classes seria necessari i quins serien els requisits funcionals.

4.2 Disseny i distribució de tasques

Després de diverses reunions amb els *clients*, els dos tutors, Jordi Coll i Mateu Villaret, es va plantejar una idea de disseny per a la interfície gràfica del sistema i es va distribuir en blocs les diferents tasques per tal de complir els requisits funcionals i poder treballar sobre una metodologia àgil.

Bloc 1 La primera fase del projecte consisteix a implementar el disseny d'una vista on es permeti a l'usuari entrar la informació essencial d'un esdeveniment, les empreses que hi assistiran i poder visualitzar el llistat d'aquestes de forma clara amb la possibilitat d'editar-lo. Implementar una classe genèrica amb el patró *Singleton* per a mantenir la informació essencial de forma única i accessible des de qualsevol punt de l'app.

Bloc 2 Afegir una vista per poder gestionar la informació detallada de les sessions que tindran lloc, així com les característiques de les taules que es tindran disponibles. S'haurà de permetre a l'usuari entrar diferents formats de taules en funció de la capacitat d'aquestes, així com la quantitat de les que es disposaran a cada una de les sessions.

Bloc 3 Implementar un sistema per poder indicar la disponibilitat de les empreses a cada moment de temps dintre de les sessions, així com el nombre de participants que assistiran per part seva. Es permetrà a l'usuari accedir a aquest apartat per editar la informació d'una empresa un cop afegida al llistat.

Bloc 4 Afegir un sistema per poder gestionar les diferents reunions, les empreses participants de cada una i, si ho requereixen, la sessió a la que estarien restringides.

Bloc 5 Implementar un sistema per poder guardar en un fitxer la informació relacionada amb l'esdeveniment. També es dissenyarà un sistema de restriccions en totes les dades que l'usuari pugui entrar, per evitar empreses repetides o reunions amb menys de dos participants, i s'aplicarà una verificació de les dades entrades.

Bloc 6 Codificar el model de restriccions en Scala. Utilitzar la llibreria ScalAT per implementar les diferents restriccions. Realitzar diversos jocs de proves i implementar un sistema de traduccions entre classes Java i Scala.

Bloc 7 Implementar una vista que permeti a l'usuari computar una solució òptima utilitzant la codificació de les restriccions juntament amb la informació de l'esdeveniment. Permetre també indicar preferències entre les reunions, fixar algunes d'elles en una hora i taula concreta. Incloure la capacitat de guardar el resultat o una solució parcial en un fitxer.

Bloc 8 Fase de realització de proves. Crear diversos esdeveniments i comprovar la integritat del sistema, documentar les errades i depurar-les. Incloure la capacitat de carregar instàncies de problemes, testejar els resultats i comprovar-los amb els obtinguts utilitzant altres eines.

Bloc 9 Documentar tot el projecte. Redactar el document actual, el sumari del projecte i la documentació relacionada amb el conjunt de restriccions del problema plantejat.

Bloc	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15
Bloc 1	■	■													
Bloc 2			■												
Bloc 3				■											
Bloc 4					■										
Bloc 5						■	■								
Bloc 6								■	■						
Bloc 7										■	■	■			
Bloc 8													■	■	
Bloc 9													■	■	■

Taula 4.1: Distribució de blocs per setmanes.

A la taula 4.1 es pot apreciar la planificació en setmanes dels diferents blocs de tasques. Pel fet que la metodologia utilitzada permet aplicar canvis al disseny, els blocs planificats es poden haver vist alterats.

Capítol 5

Marc de treball i conceptes previs

En aquest apartat es descriuen tots els conceptes necessaris per tal d'entendre al complet el projecte.

5.1 SAT

El problema de la Satisfactibilitat Booleana (o SAT) consisteix en determinar si existeix alguna assignació de valors de veritat (o interpretació) per a les variables d'una fórmula de lògica proposicional que permeti avaluar la fórmula a cert, és a dir, que satisfaci la fórmula. Per exemple, donada una fórmula SAT amb les variables booleans x_1, x_2, x_3 :

$$(x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_1 \vee x_3)$$

voldríem trobar una interpretació que la satisfaci. Per a aquesta fórmula, la interpretació $x_1 = \mathbf{cert}$, $x_2 = \mathbf{cert}$, $x_3 = \mathbf{fals}$ satisfà la fórmula. En canvi, per exemple la interpretació $x_1 = \mathbf{fals}$, $x_2 = \mathbf{fals}$, $x_3 = \mathbf{fals}$ no la satisfà. Quan existeixi alguna interpretació que satisfaci la fórmula, direm que la fórmula és *satisfactible*. Si no n'existeix cap diem que és *insatisfactible*.

El problema SAT es pot resoldre amb una màquina de Turing no-determinista en temps polinòmic (NP), i tot problema que pertanyi a NP es pot reduir en temps polinòmic a SAT (NP-hard). Per tant diem que és un problema NP-complet. Avui en dia encara no s'ha trobat cap algorisme polinòmic amb màquina determinista capaç de resoldre aquesta mena de problemes.

Típicament les fórmules SAT es presenten en format CNF. Una CNF, o Forma Normal Conjuntiva, és una conjunció (mitjançant l'operador \wedge) de clàusules. Una clàusula és una disjunció (amb operador \vee) de les literals. Un literal és una variable Booleana (literal positiu) o la negació (amb operador \neg) d'una variable Booleana (literal negatiu). Per tal que un problema SAT sigui satisfactible, totes aquestes clàusules s'han d'avaluar a cert per a una determinada interpretació. Una interpretació és una assignació de valors de veritat (**cert** i **fals**) per a totes les variables del problema. Una clàusula s'avalua a cert per una interpretació, si com a mínim la variable d'un literal positiu de la clàusula és assignada a **cert** per la interpretació o la variable d'un literal negatiu és assignada a **fals**.

Aquestes clàusules veure també com a restriccions, ja que restringeixen els valors que poden ser assignades les variables que la formen en funció de la seva composició.

Existeixen restriccions més complexes sobre conjunts de variables booleans que es fan servir sovint per a modelitzar problemes, com podrien ser les At-Most-One o Exactly-One. At-Most-One fa referència al conjunt de restriccions que limiten les variables a què, com a màxim, una sigui certa.

$$AMO(x_1, x_2, \dots, x_n)$$

De la mateixa manera, Exactly-One força al fet que una i només una variable sigui certa.

$$EO(x_1, x_2, \dots, x_n)$$

Al llarg del temps s'han desenvolupat diversos algorismes per solucionar els problemes SAT de manera eficient. Conseqüentment, s'han creat llibreries de codi que permeten implementar i solucionar problemes de SAT de forma senzilla. En aquest projecte s'utilitza la llibreria Sat4j per a solucionar un problema SAT i la llibreria ScalAT per a generar-lo.

5.2 Scala

Scala és un llenguatge de programació que uneix l'orientació a objectes amb la programació funcional. Permet utilitzar classes i objectes derivats de Java, gràcies a què està dissenyat per a ser compilat en Bytecode. També la programació funcional, ja que suporta funcions d'ordre superior, funcions que poden rebre altres funcions com a paràmetre, i també permet el reconeixement de patrons.

Gràcies a la seva part funcional, similar a llenguatges purament funcionals com Haskell, permet la declaració d'algorismes de forma clara i senzilla.

Aquest llenguatge ens va ser introduït a l'assignatura de quart *Programació declarativa. Aplicacions.*, i s'ha decidit incorporar-lo al projecte per investigar més sobre aquest.

Donat que aquest es compila com a Bytecode, permet que un objecte escrit en Scala sigui utilitzat dins un projecte Java com a una llibreria, ampliant així la seva capacitat de funcionament.

5.2.1 ScalAT

ScalAT és una llibreria escrita en Scala que permet codificar problemes de restriccions a SAT de forma senzilla.

Aquesta llibreria és presentada a l'assignatura *Programació declarativa. Aplicacions.* i utilitzada per a fer les primeres codificacions de problemes combinatoris de SAT.

Conte diversos mètodes per declarar variables Booleanes, afegir clàusules, i codificar les restriccions més comunes, com les At-Most-One, Exactly-One, At-Most-K o Exactly-K entre altres.

Aquesta llibreria encapsula la crida al SAT solver Sat4j [4], i ens permet determinar si la fórmula codificada és satisfactible. Si la fórmula és satisfactible, ScalAT permetrà obtenir el valor assignat a totes les variables. També ens permet obtenir informació rellevant sobre el procés de solució, com per exemple el temps que ha trigat o mètriques relacionades amb la mida de l'arbre de cerca. Amb la solució obtinguda es podrà mostrar el resultat a l'usuari.

5.3 Java FX i FXML

JavaFX és una llibreria que s'incorpora a Java i permet desenvolupar aplicacions amb interfície gràfica multiplataforma. Una aplicació JavaFx pot ser executada en qualsevol sistema operatiu que disposi d'una versió de Java Runtime Environment.

5.3.1 Funcionament

El funcionament principal de JavaFX consisteix en la gestió d'escenaris on es mostra la interfície gràfica del software. Cada escenari definit pel programador consisteix en una finestra habitual d'un sistema operatiu on s'inclouran els components visuals que compondran la vista.

Un *stage* és un escenari de vistes principal encapsulat per una finestra del sistema operatiu. Un *stage* de JavaFX sempre consta d'un component pare de tipus layout. Un layout és un format de component que té l'objectiu de contenir altres components inclosos en el seu interior delimitant la posició i mida que poden agafar. Existeixen diferents tipus de layouts, el principal i més utilitzat és el `BorderPane`, ja que en comptes de ser un contenidor d'objectes senzill sense delimitacions, com ho seria un `Pane`, permet la distribució d'aquests en cinc espais característics utilitzats per a construir el format habitual d'una aplicació. Aquests són el superior, per les barres amb títols i menús, els apartats laterals esquerre i dret per vistes de navegació i menús d'opcions, un apartat inferior per una secció amb els botons d'opcions i una secció central on apareixerà el contingut principal del software.

També s'ofereixen altres formats de layout, com els `VBox` i `HBox`, que com la seva inicial indica, vertical i horitzontal, agrupen els components inclosos per ordre en vertical i horitzontal limitant així la distribució. De forma similar, existeix el `GridPane` que permet la distribució en graella de mida determinada.

Per altra banda, existeix el layout `ScrollPane`, que permet tenir un component de tipus layout a dins seu, fixar una mida determinada i habilitar un desplaçament horitzontal i vertical per aquest.

Tots els components visuals que formen part de JavaFX deriven de la classe `Node`, per tant, tot objecte de tipus layout que pot tenir components inclosos, pot també contenir altres layouts.

5.3.2 FXML

FXML és un llenguatge, basat en el llenguatge de marques XML, que s'incorpora a JavaFX per a facilitar el disseny d'interfícies gràfiques i proporcionant l'utilització del patró de disseny Model-Vista-Controlador.

FXML permet simplificar la creació de les diferents vistes de l'interfície gràfica en fitxers XML on es declararà l'escena principal amb tots els components, i incorporar un fitxer controlador associat per a determinar les diferents accions que corresponen a les diferents accions dels components de la vista. Això agilitza el procés de desenvolupament d'una aplicació, ja que permet separar el disseny de la lògica.

De totes maneres, no incorpora cap component nou, tots els components presents a FXML, també ho són a JavaFX, i per tant, l'ús d'aquesta llibreria és purament per a millorar el disseny del codi.

FXML disposa d'una eina de disseny externa anomenada `Scene Builder` on es permet al programador crear i distribuir, de forma visual i intuïtiva, els diferents components que formen una escena, així com les seves característiques principals i les accions que han de complir.

5.4 Bussines-to-Bussines

Bussines-to-Bussines (B2B) és un model de reunions entre empreses (o altres tipus d'agents) que comparteixen els mateixos interessos.

Aquest model de reunions sol tenir lloc en esdeveniments de gran format, com conferències, congressos, on la majoria de les empreses participants tenen com a interès exposar els seus productes i crear nous negocis.

La majoria de les entitats participants tindran interès de reunir-se amb diverses empreses, mentre que

en molts casos el seu personal participant tindrà una disponibilitat reduïda. Això comporta un cost de gestió molt elevat.

El guany de popularitat del model B2B en els darrers anys ve donat al plantejament de reunions curtes i informals on les entitats exposen els seus punts de forma breu en una durada límit determinada. Això permet dur a terme multitud de reunions en un espai de temps de poques hores en un mateix lloc.

Hi ha moltes variants d'esdeveniments B2B amb diferents característiques. Per exemple es poden considerar agrupacions de reunions per sessions, permetre que els assistents indiquin franges horàries en què no estan disponibles, considerar reunions no només entre dues empreses sinó entre diverses empreses, etc. El primer pas abans de començar aquest projecte ha sigut definir un model teòric d'esdeveniment B2B que sigui prou genèric per ser utilitzat en diversos casos reals. A continuació es detallen els conceptes clau relacionats amb el problema B2B plantejat:

Esdeveniment Un esdeveniment pot ser un congrés, una fira d'empreses, una conferència, o similar. Una data assenyalada d'un o més dies on multitud d'entitats d'un o més àmbits es reuniran amb el propòsit d'exposar els seus productes o serveis i amb la intenció de crear nous negocis.

Gestor Anomenem gestor a la persona o grup de persones encarregades de programar l'horari de l'esdeveniment. El gestor serà l'usuari de l'aplicació desenvolupada.

Entitat Una entitat és una empresa o particular que assistirà a l'esdeveniment per tal de tenir reunions amb altres entitats.

Participant Les diferents entitats que assistiran disposaran de personal assistent, els diferents encarregats de participar en les reunions d'aquesta. Una entitat sempre tindrà un mínim d'un participant.

Reunió Una reunió és una trobada, de durada limitada pels gestors de l'esdeveniment, entre dues o més entitats, on participen un participant de cada entitat, per tal de tancar negocis, exposar productes o negociar els punts d'un contracte entre altres. La informació d'una reunió constarà del llistat d'entitats participants i opcionalment una sessió de preferència. La llista de reunions que s'han de fer formen part de l'entrada del problema, i en una solució totes aquestes reunions hauran d'estar col·locades.

Sessió Una sessió és un espai de temps limitat per diverses hores on tindran lloc diverses reunions. Un esdeveniment tindrà un mínim d'una sessió, podent-ne tenir varies. Un esdeveniment pot tenir diverses sessions en un mateix dia, com una sessió al matí i una de tarda, però mai dues sessions que es solapin. Aquestes sessions estan compostes per fragments de temps anomenats *Time Slots*, i disposaran d'una quantitat de taules de diferents capacitats.

Time Slot Espai de temps de durada fixada pels gestors, que correspon a la durada màxima d'una reunió, durant el qual podran tenir lloc diverses reunions, com a molt, tantes com taules disponibles hi hagi. Una sessió està formada per un determinat nombre de *Time Slots*, tots de la mateixa durada. Una entitat només pot tenir un nombre de reunions màxim en un mateix *Time Slot* corresponent al nombre de participants que aquesta disposa.

Forbidden Time Slot Les entitats podran tenir restriccions horàries per a fer reunions. Un *Forbidden Time Slot* és un *Time Slot* on una entitat no estarà disponible per a fer cap reunió. Si una entitat no estarà disponible durant tota una sessió, llavors es diu que tots els fragments de temps de la sessió, són *Forbidden Time Slots* per a l'entitat.

Taula Fem servir el mot *taula* per referir-nos a cadascun dels espais on tenen lloc les reunions, ja siguin taules, sales de reunions, etc. Un esdeveniment tindrà a la seva disposició diverses taules on es podran dur a terme determinades reunions. El nombre màxim de persones que càpiguen a la taula limitarà el format de reunió que es pot dur a terme en aquella taula. Una sessió disposarà d'un nombre determinat de cada tipus de taula segons el nombre de seients que té cadascuna.

Una solució del problema B2B consisteix en assignar un time slot i una taula a totes les reunions sol·licitades respectant les restriccions següents:

- Tota reunió s'ha de fer exactament a un time slot.
- Tota entitat pot tenir com a molt tantes reunions en un mateix time slot com participants tingui.
- Cap entitat pot tenir cap reunió en un dels seus forbidden time slot.
- Si s'ha fixat una reunió en un time slot i taula, aquesta no es podrà alterar en la solució.
- Un time slot pot tenir tantes reunions com taules disponibles en aquella sessió.
- Una reunió amb un cert nombre de participants no pot estar col·locada en una taula que no disposi de prou seients. Aquesta restricció s'ha exclòs en la versió actual de l'aplicació, i s'assumeix que totes les taules tenen una mida prou gran per celebrar-hi qualsevol reunió. Aquesta decisió s'ha pres perquè aquesta restricció no està directament suportada per ScalAT, i s'ha considerat que queda fora de l'abast del projecte. Tot i això, la interfície gràfica ja porta incorporat un sistema per a especificar taules de diferents mides, i està preparada per integrar fàcilment aquesta restricció si es codifica més endavant.

Cal destacar que el repartiment de reunions a taules concretes és computacionalment fàcil de fer a partir d'una solució que respecti que el nombre de reunions que es fan a cada sessió no és major que la quantitat de taules disponibles. Per tant, per fer l'aplicació més eficient, el repartiment de taules no es contempla en el model de restriccions codificat amb ScalAT, sinó que es farà posteriorment a obtenir una solució, tal com s'explica a la Secció 8.4.

Capítol 6

Requisits del sistema

Els requisits del sistema poden ser dividits en dos apartats.

6.1 Requisits funcionals

El Bussines-2-Bussines Managment ha de permetre:

- Gestionar les entitats participants en l'esdeveniment.
- Gestionar les diferents sessions en què es portarà a terme l'esdeveniment.
- Gestionar les diverses reunions que tindran les entitats.
- Establir les restriccions horàries de les entitats.
- Obtenir un horari per a les reunions sol·licitades.

6.2 Requisits no funcionals

Gràcies a la utilització de JavaFX, qualsevol ordinador amb una versió de Java Runtime Environment 13 i el solver sat4j inclòs en el paquet d'instal·lació, serà capaç d'executar el software.

Capítol 7

Estudi i decisions

En aquest capítol es descriuen els estudis realitzats i decisions preses durant el desenvolupament del projecte.

7.1 Java FX i FXML

Per tal de facilitar que l'aplicació sigui multiplataforma, i gràcies a la seva sinergia amb Scala, la part gràfica del software ha estat realitzada amb JavaFX. A part, s'hi han inclòs els aspectes de FXML per tal d'utilitzar un patró Model-Vista-Controlador en el desenvolupament de les diferents vistes per a simplificar la seva complexitat.

7.2 SAT

Un dels principals motius per utilitzar codificacions a SAT per resoldre el problema plantejat, és que SAT permet modelar amb facilitat les restriccions que apareixen als problemes B2B, i que és una de les tècniques més eficients per resoldre problemes tipus B2B [2].

Per altra banda, actualment existeixen SAT solvers molt eficients, i alguns d'ells estan desenvolupats amb Java, com el *Sat4j*. Això permet afegir la implementació del model de restriccions a una aplicació JavaFX.

Finalment, també s'ha decidit utilitzar SAT perquè és el focus de recerca del grup del qual formen part els tutors del projecte, el grup de recerca de Lògica i Programació.

7.3 Scala i ScalAT

El llenguatge Scala ha estat escollit per a realitzar la codificació per diversos motius. El primer és la disponibilitat de la llibreria ScalAT, que permet realitzar les codificacions de les restriccions clàssiques que apareixen en problemes tipus B2B, i utilitzar un *solver* eficient per obtenir una solució. Per altra banda, també aporta una aproximació funcional en la programació de les diferents restriccions, cosa que permet un codi més clar i concís.

Capítol 8

Anàlisi i disseny del sistema

Aquest apartat detalla tots els aspectes del software B2B Managment.

8.1 Model de dades

El software B2B Managment utilitza una classe principal, anomenada *MainData*, que utilitza un patró *Singleton* i conte tota la informació essencial de l'aplicació, el nom, la localització de l'esdeveniment, el nombre de sessions i la durada màxima d'una reunió.

Aquesta classe conte un llistat de Sessió, un llistat d'Entity i un llistat de Meeting.

La classe Sessio conte la informació sobre una sessió: la data fixada, l'hora d'inici i finalització d'aquesta i la computació de tots els *Time Slots* que conté.

Una Entity correspon a una entitat assistent a l'esdeveniment, i està formada pel seu identificador, nom i nombre de personal que assistirà. També conte el llistat de sessions a les quals assistirà i per conseqüència, els *Forbidden Time Slots*.

Una Meeting és la classe que representa una reunió de l'esdeveniment. Aquesta consta d'un variable per guardar la restricció de quina sessió ha d'assignar-se i el llistat d'entitats participants. També conté la informació necessària per a poder fixar en una posició determinada la reunió, és a dir, en una taula i *time slot* concrets.

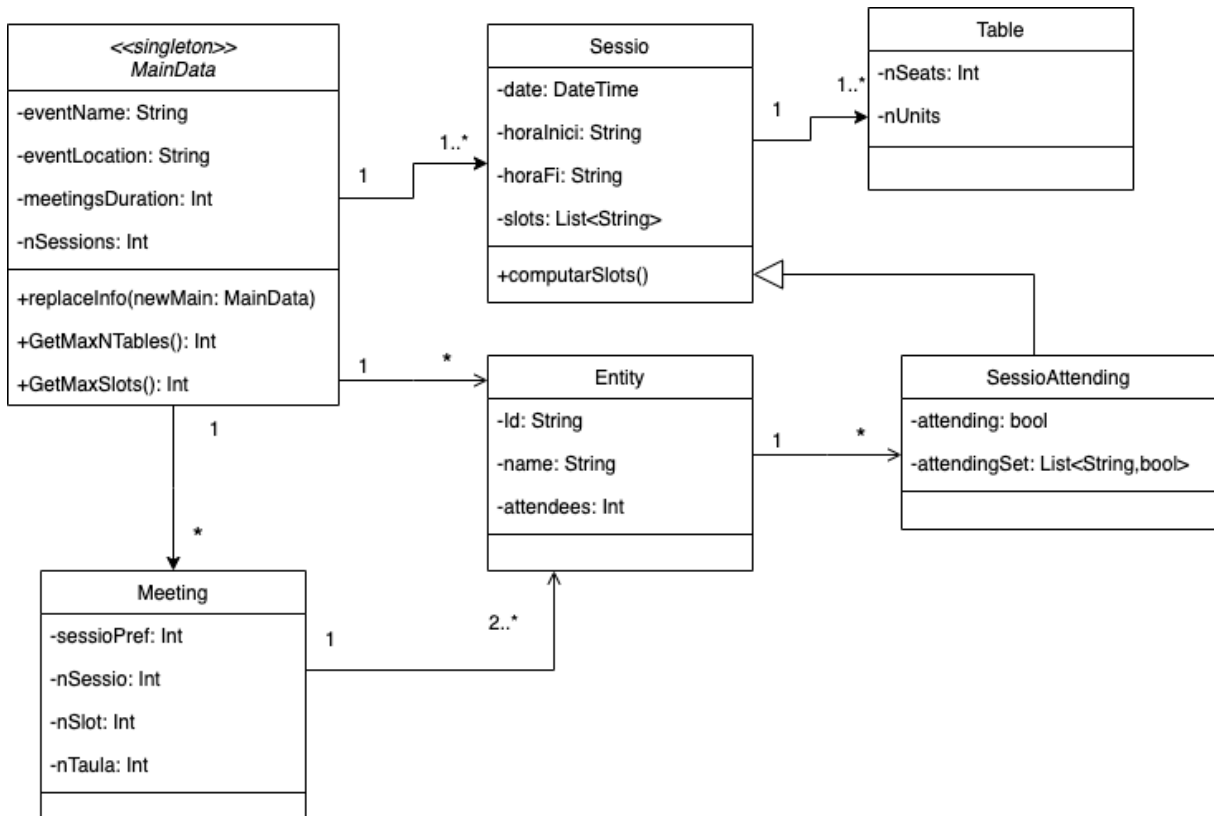


Figura 8.1: Diagrama de classes.

8.2 Anàlisi de funcionalitats

D'acord amb els requisits funcionals del software, s'ha de permetre a l'usuari gestionar i mantenir les diverses sessions, reunions i entitats, fixar reunions en *Time Slots* determinats i computar una solució final.



Figura 8.2: Diagrama de casos d'ús.

La majoria de casos d'ús requereixen unes accions molt bàsiques per part del gestor i tenen una resposta molt intuïtiva de l'interfície gràfica, per exemple, esborrar una entitat consta d'una sola acció per part del gestor i no té cap restricció de funcionament. Per aquests casos d'ús no s'ha desenvolupat cap diagrama. A continuació es detallen els casos d'ús una mica més complexos, que requereixen més accions per part del gestor, que provoquen més creació de continguts per part de l'aplicació o disposen de restriccions

complexes.

8.2.1 Editar una sessió

En el procés d'editar una sessió, l'usuari pot editar les hores d'inici i fi o la durada de les reunions. Per altra banda també pot editar el format de les taules.

En el cas d'editar la informació relacionada amb les hores de la sessió, el verificador editarà el llistat de *Time Slots* i aplicarà els canvis a l'assistència de totes les entitats.

Un objecte de la classe Entity correspon a una entitat assistent a l'esdeveniment, i està formada pel seu identificador, nom i nombre de personal que assistirà. També conté el llistat de sessions a les quals assistirà i per conseqüència, els *Forbidden Time Slots*.

Un objecte de la classe Meeting representa una reunió de l'esdeveniment. Aquesta consta d'un variable per guardar la preferència de quina sessió ha d'assignar-se i el llistat d'entitats participants. També conté la informació necessària per a poder fixar en una posició determinada la reunió.

En la figura 8.3 es pot apreciar el diagrama de flux d'editar una sessió.

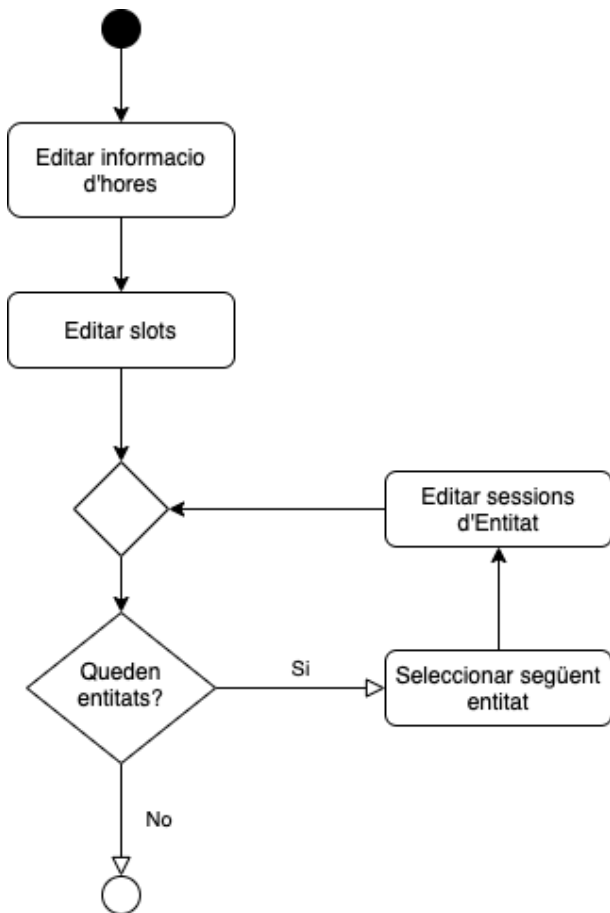


Figura 8.3: Diagrama de flux editar sessió.

8.2.2 Editar el nombre de sessions

La Figura 8.4 conté el diagrama de flux d'editar el nombre de sessions. Si l'usuari modifica el nombre de sessions disponibles, el verificador haurà de modificar les sessions mantingudes reduint o ampliant el llistat sempre per la part final, és a dir, si disposem de cinc sessions i l'usuari redueix a tres, només quedaran les tres primeres reunions.

Les entitats registrades al sistema disposen d'un llistat d'objectes de la classe *Sessio Attending*. Aquesta classe hereta de sessió i conté un Booleà que indica si s'assistirà a tota la sessió i un Booleà per a cada *time slot* de la sessió que indica si estarà disponible en aquell moment. El verificador eliminarà o afegirà els objectes del llistat de sessions que manté les entitats registrades. Totes les sessions creades noves tindran assistència completa.

Les reunions ja registrades que tinguin com a restricció una sessió que s'ha vist eliminada, passaran a no tenir restricció de sessió. Les que estiguin fixades en un time slot que forma part d'una sessió eliminada, passaran a no estar fixades.

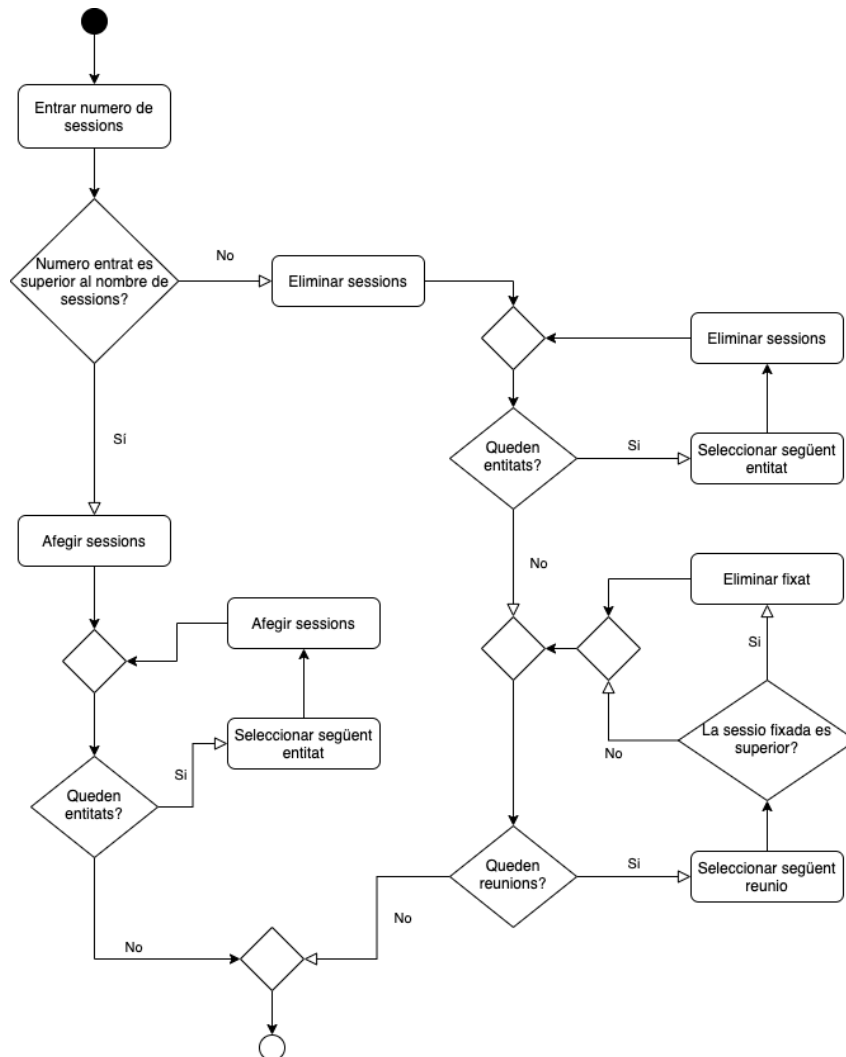


Figura 8.4: Diagrama de flux editar nombre de sessions.

8.2.3 Creació d'una entitat

Per a poder crear una nova entitat o editar-ne una ja existent, el verificador de dades comprovarà que no existeixi cap altra entitat amb l'identificador i el nom entrats.

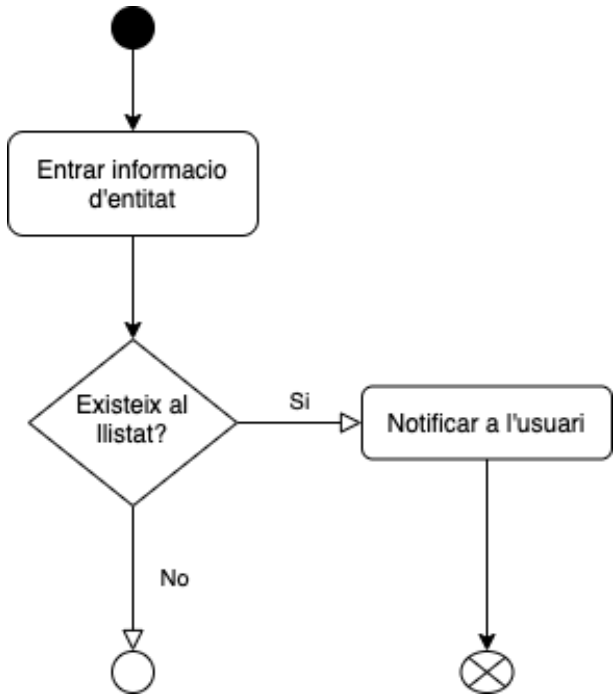


Figura 8.5: Diagrama de flux crear una entitat.

8.2.4 Creació d'una reunió

En la creació d'una reunió, es comprovarà que les entitats participants no estiguin duplicades i que n'hi hagi un mínim de dues. Si alguna de les entitats encara no ha estat registrada, es notificarà l'usuari però es permetrà la creació de la reunió. Quan l'usuari vulgui accedir a la vista d'horari, el verificador comprovarà de nou si totes les entitats existeixen, però aquest cop impedirà continuar si alguna d'aquestes no està registrada.

La Figura 8.6 conte el diagrama de flux amb el procés de crear una reunió.

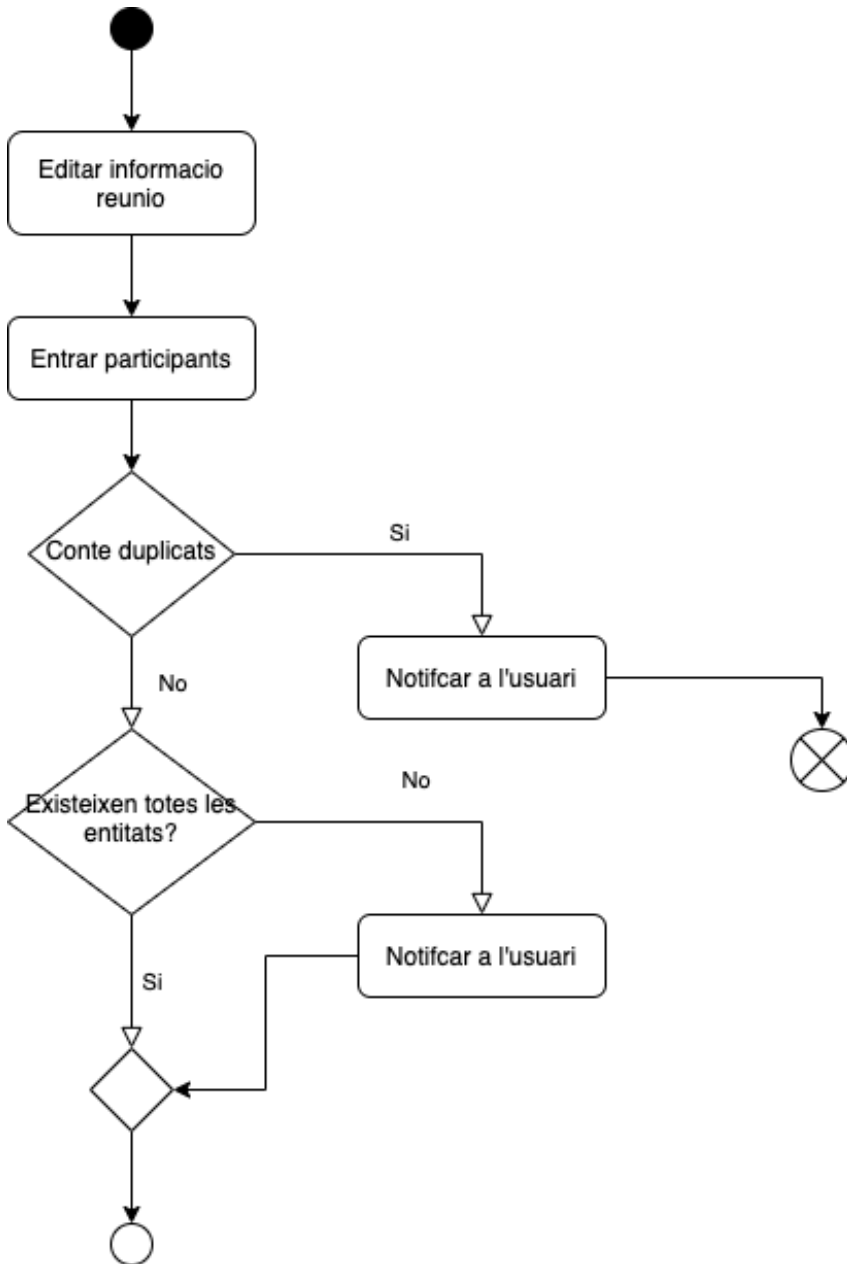


Figura 8.6: Diagrama de flux crear una reunió.

8.2.5 Fixat d'una reunió

Quan l'usuari vol col·locar una reunió en una sessió, *Time Slot* i taula determinats, el verificador comprovarà primer de tot que el *Time Slot* estigui permès per a totes les entitats de la reunió, seguidament comprovarà quantes vegades apareix cada participant en el mateix *Time Slot* i si disposa de suficients participants per atendre les reunions.

La Figura 8.7 conte el diagrama de flux que representa l'acció de fixar una reunió.

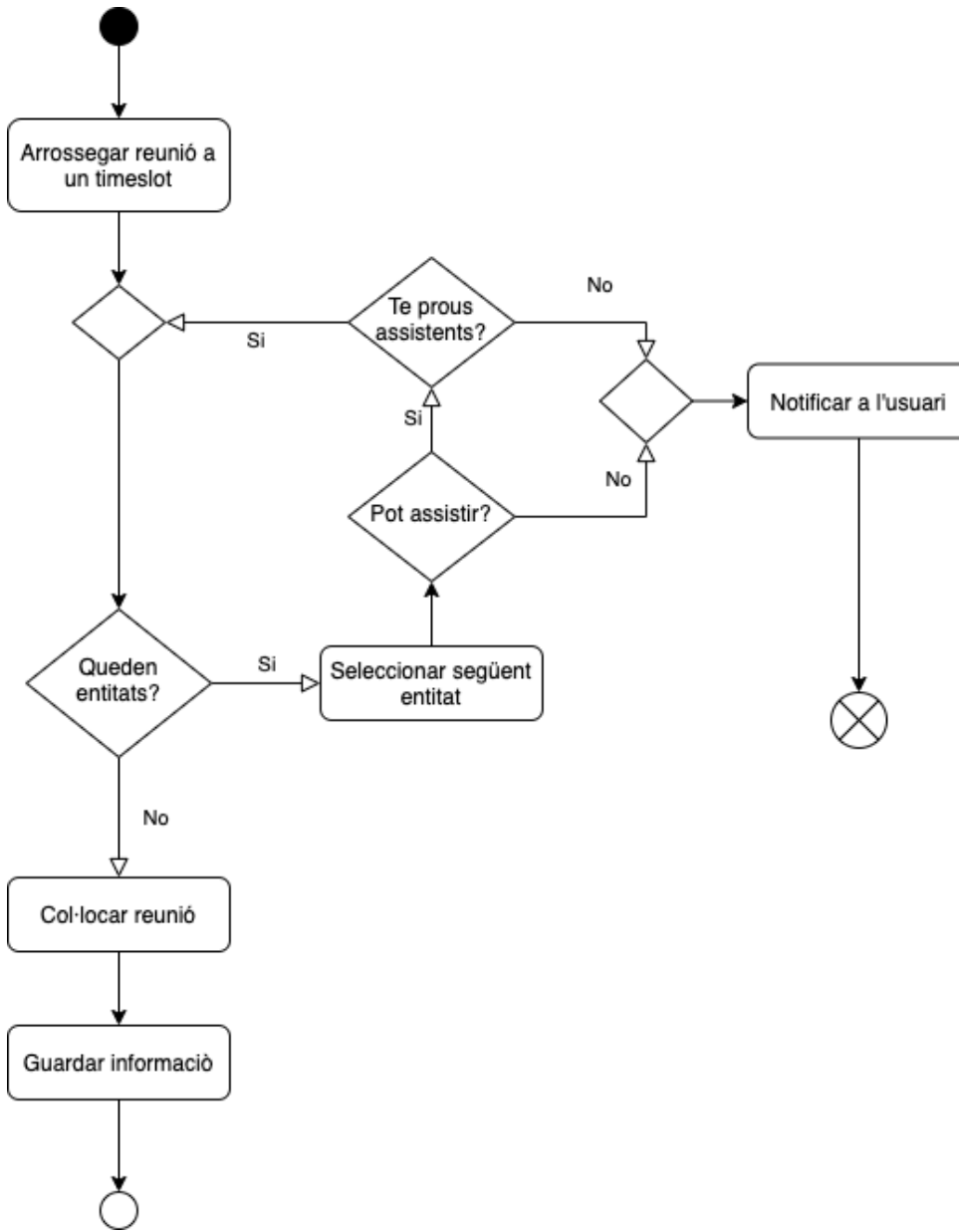


Figura 8.7: Diagrama de flux fixar una reunió.

8.3 Interfície gràfica

Aquest software està format per dues vistes principals i diverses vistes modals. Una vista modal és una interfície gràfica que apareix per separat, en alguns casos superposant una altra vista, la qual requereix la interacció de l'usuari i no li permet accedir a les vistes inferiors fins que aquesta és tancada o cancel·lada.

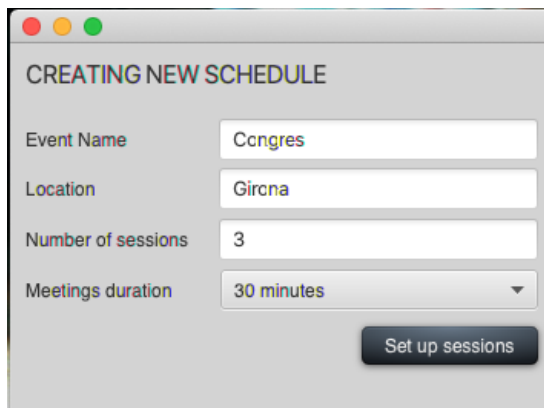
D'acord amb el patró de disseny Model Vista Controlador que implementa FXML, cada una de les vistes descrites posteriorment tenen associades una classe controlador, que s'encarrega de tota la lògica darrere dels diferents possibles accions.

En aquest apartat es descriuran les diferents vistes i les accions que un usuari pot dur a terme.

8.3.1 Vista Principal


Troblem una captura de la vista principal a la Figura 8.13. Per a facilitar la comprensió del detall d'aquesta vista, s'han enumerat els diferents apartats i es detallaran per separat.

En aquesta vista l'usuari gestor podrà introduir totes les dades principals referents a l'esdeveniment, de forma genèrica i deixant la informació més concreta amb valors per defecte. Està composta per un *BorderPane* principal i dividida en tres espais, el central, esquerra i inferior. La part esquerra correspon a l'apartat on un usuari pot entrar les dades de l'esdeveniment i la part de la dreta podrà visualitzar totes les dades entrades.



The image shows a window titled "CREATING NEW SCHEDULE". It contains four input fields: "Event Name" with the value "CcnGRES", "Location" with "Gircna", "Number of sessions" with "3", and "Meetings duration" with a dropdown menu set to "30 minutes". A "Set up sessions" button is located at the bottom right of the form.

Figura 8.8: Fragment de la Figura 8.13



The image shows a window titled "Set New Entity". It contains three input fields: "Entity Name", "Entity ID", and "Number of attendees". At the bottom, there are two buttons: "Set arribal info" and "Add entity".

Figura 8.9: Fragment de la Figura 8.13

- 1. Editor d'esdeveniments** Començant per la part superior (Figura 8.8), el gestor pot introduir el nom i localitat d'aquest, així com el nombre de sessions i la durada màxima de les reunions. Si ho desitja, podrà editar les sessions accedint a la vista prement el botó *Set up sessions*. En cas de no fer-ho, l'aplicació donarà valors per defecte a la configuració de cada sessió.

- 2. Editor d'entitats** Seguidament tenim l'apartat per introduir les dades d'una entitat (Figura 8.9): el seu nom, identificador i nombre de participants. Si el gestor ho desitja, pot accedir a la configuració d'assistència de l'entitat prement el botó *Set arribal info*. En cas de no fer-ho, s'assumirà que l'entitat estarà disponible en tots els *time slots*.

El botó *Add Entity* servirà per afegir les dades entrades a la taula d'entitats a la part central superior de l'aplicació.

5. Botons d'accio La part inferior de la vista està formada per un llistat de botons (Figura 8.12), els tres primers tenen a veure amb la càrrega i guardat de fitxers. El primer botó, *Load Instance*, permet a l'usuari seleccionar un fitxer de tipus instància, per a poder comprovar el comportament de l'aplicació i comparar els resultats amb proves realitzades amb aplicacions externes que resolen instàncies SAT. Aquest botó estarà ocult en fase de producció. Els dos següents botons, *Load* i *Save*, permeten a l'usuari guardar i carregar un fitxer en format JSON que contindrà tota la informació de l'esdeveniment. El botó anomenat *Next* permetrà a l'usuari avançar a la vista d'horari si el verificador considera que es compleixen totes les restriccions.



Figura 8.12: Fragment de la Figura 8.13

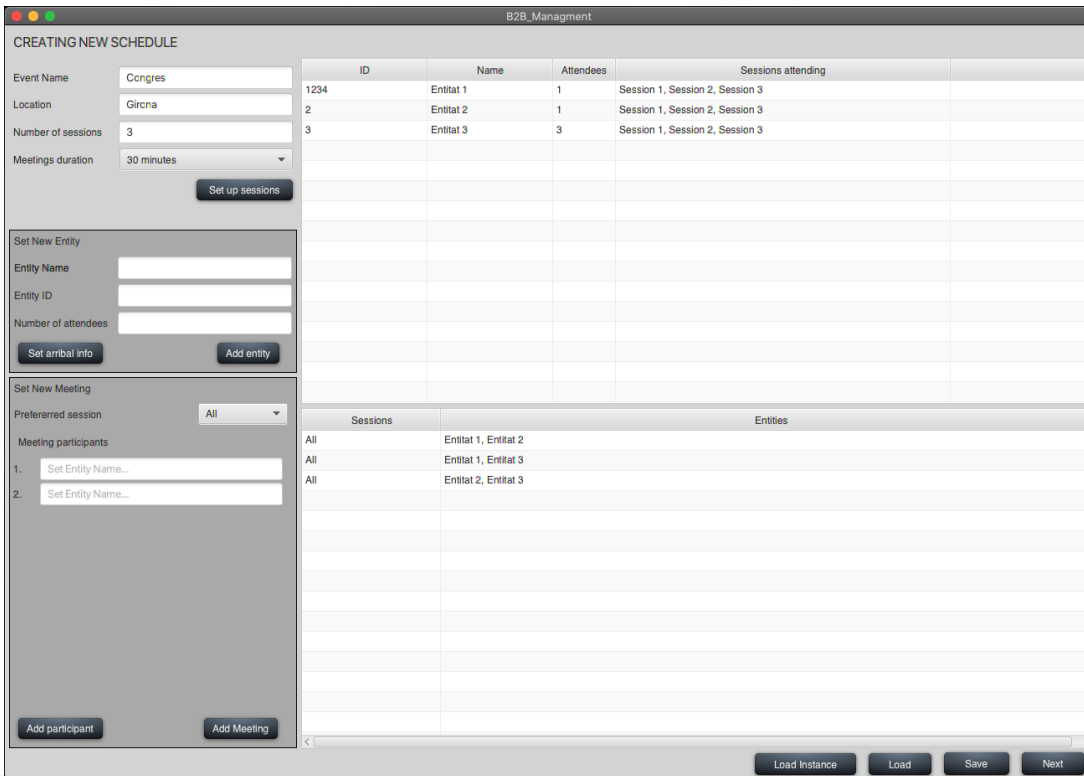


Figura 8.13: Captura de pantalla de la vista principal.

8.3.2 Vista editar sessions

En aquesta vista modal (Figura 8.14, l'usuari tindrà la possibilitat de personalitzar les sessions. Apareix un *GridPane* central amb les sessions com a columnes.

Es permet canviar l'hora d'inici i fi de cada sessió, així com entrar el nombre de taules disponibles per a cada una d'elles.

Pel fet que el conjunt de restriccions actual no permet la utilització de taules de diferents característiques, l'opció d'introduir i personalitzar la mida de taules ha estat ocultada. De totes maneres, en un futur es pot mostrar a l'usuari i el funcionament visual ja està implementat (Figura 8.15).

	Session 1	Session 2	Session 3
Date	6/6/2020	6/7/2020	6/8/2020
Start Hour	08:00	08:00	08:00
End Hour	13:30	13:30	13:30
Tables	11	11	11

Save

Figura 8.14: Captura de pantalla de la vista d'editar sessions.

	Session 1	Session 2	Session 3
Date	6/9/2020	6/9/2020	6/9/2020
Start Hour	08:00	08:00	08:00
End Hour	20:00	20:00	20:00
N Tables of 2	1	1	1
N Tables of 3	1	1	1

Add new table config Add Use * to add a custom table Save

Figura 8.15: Captura de pantalla de la vista d'editar sessions amb la configuració de taules.

8.3.3 Editar disponibilitat

La següent vista modal (Figura 8.16), l'usuari tindrà la possibilitat d'assignar els *Forbidden Time Slots* corresponents. Es podrà desactivar una sessió sencera amb el botó situat al costat del títol de cada sessió, o desactivar hores concretes.

Set when the entity will be available								
Session 1	<input checked="" type="checkbox"/>		Session 2	<input checked="" type="checkbox"/>		Session 3	<input checked="" type="checkbox"/>	
08:00	<input checked="" type="checkbox"/>		08:00	<input checked="" type="checkbox"/>		08:00	<input checked="" type="checkbox"/>	
08:30	<input checked="" type="checkbox"/>		08:30	<input checked="" type="checkbox"/>		08:30	<input checked="" type="checkbox"/>	
09:00	<input checked="" type="checkbox"/>		09:00	<input checked="" type="checkbox"/>		09:00	<input checked="" type="checkbox"/>	
09:30	<input checked="" type="checkbox"/>		09:30	<input checked="" type="checkbox"/>		09:30	<input checked="" type="checkbox"/>	
10:00	<input checked="" type="checkbox"/>		10:00	<input checked="" type="checkbox"/>		10:00	<input checked="" type="checkbox"/>	
10:30	<input checked="" type="checkbox"/>		10:30	<input checked="" type="checkbox"/>		10:30	<input checked="" type="checkbox"/>	
11:00	<input checked="" type="checkbox"/>		11:00	<input checked="" type="checkbox"/>		11:00	<input checked="" type="checkbox"/>	
11:30	<input checked="" type="checkbox"/>		11:30	<input checked="" type="checkbox"/>		11:30	<input checked="" type="checkbox"/>	
12:00	<input checked="" type="checkbox"/>		12:00	<input checked="" type="checkbox"/>		12:00	<input checked="" type="checkbox"/>	
12:30	<input checked="" type="checkbox"/>		12:30	<input checked="" type="checkbox"/>		12:30	<input checked="" type="checkbox"/>	

Save

Figura 8.16: Captura de pantalla de la vista d'editar la disponibilitat.

8.3.4 Editar entitat

Si es desitja editar la informació d'una entitat ja registrada, es mostrarà aquesta vista modal formada per les dues seccions que ja s'havien utilitzat per entrar una entitat (Figura 8.17). Es permetrà editar el nom i identificador sempre que els valors nous no es trobin ja registrats en una altra entitat.

Es podrà modificar l'assistència de l'entitat, desactivant sessions senceres o *Time Slots* determinats.

Per últim, s'ofereix un botó per tal d'eliminar l'entitat del llistat.

Entity Detail

Entity Name

Entity ID

Number of attendees

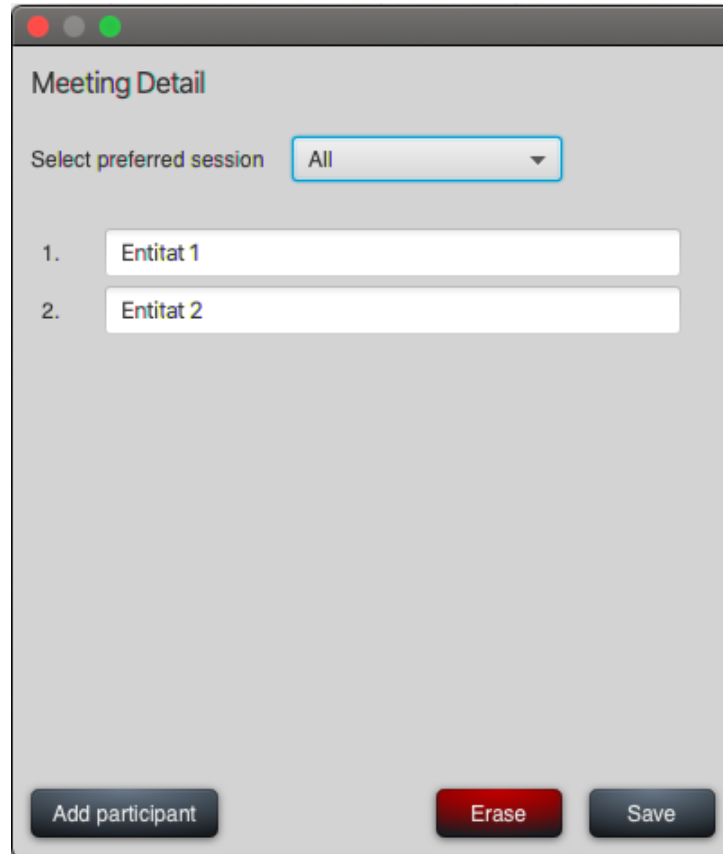
Session 1	<input checked="" type="checkbox"/>	Session 2	<input checked="" type="checkbox"/>	Session 3	<input checked="" type="checkbox"/>
08:00	<input checked="" type="checkbox"/>	08:00	<input checked="" type="checkbox"/>	08:00	<input checked="" type="checkbox"/>
08:30	<input checked="" type="checkbox"/>	08:30	<input checked="" type="checkbox"/>	08:30	<input checked="" type="checkbox"/>
09:00	<input checked="" type="checkbox"/>	09:00	<input checked="" type="checkbox"/>	09:00	<input checked="" type="checkbox"/>
09:30	<input checked="" type="checkbox"/>	09:30	<input checked="" type="checkbox"/>	09:30	<input checked="" type="checkbox"/>
10:00	<input checked="" type="checkbox"/>	10:00	<input checked="" type="checkbox"/>	10:00	<input checked="" type="checkbox"/>
10:30	<input checked="" type="checkbox"/>	10:30	<input checked="" type="checkbox"/>	10:30	<input checked="" type="checkbox"/>
11:00	<input checked="" type="checkbox"/>	11:00	<input checked="" type="checkbox"/>	11:00	<input checked="" type="checkbox"/>
11:30	<input checked="" type="checkbox"/>	11:30	<input checked="" type="checkbox"/>	11:30	<input checked="" type="checkbox"/>
12:00	<input checked="" type="checkbox"/>	12:00	<input checked="" type="checkbox"/>	12:00	<input checked="" type="checkbox"/>
12:30	<input checked="" type="checkbox"/>	12:30	<input checked="" type="checkbox"/>	12:30	<input checked="" type="checkbox"/>

Figura 8.17: Captura de pantalla de la vista d'editar entitats.

8.3.5 Editar reunió

Aquesta vista modal (Figura 8.18) reutilitza els components trobats prèviament en la vista principal per tal d'introduir les dades d'una reunió. Es permet editar i eliminar els participants d'una reunió, però només es permetrà guardar els canvis si la reunió final compleix els requisits necessaris.

També es permet eliminar la reunió del llistat.



The screenshot shows a modal window titled "Meeting Detail". At the top, there are three window control buttons (red, yellow, green). Below the title, there is a label "Select preferred session" followed by a dropdown menu currently set to "All". Underneath, there is a list of participants:

- 1. Entitat 1
- 2. Entitat 2

At the bottom of the window, there are three buttons: "Add participant", "Erase", and "Save".

Figura 8.18: Captura de pantalla de la vista d'editar entitats.

8.4 Vista horari

Aquesta és la segona vista principal de l'aplicació (Figura 8.19). En aquest apartat, l'usuari tindrà la possibilitat d'arrossegar, per mitja de *drag-and-drop* (Secció 8.4.1), les reunions i fixar-les en un determinat espai.

Està formada per dues seccions. La primera secció situada a l'esquerra mostra el llistat de reunions disponibles a col·locar en aquell instant i per la sessió seleccionada. En la part superior, ofereix un filtre per introduir el nom d'una entitat i remarcar totes les reunions en les quals participa.

La segona secció està formada per un *TabPane*, un *layout* que permet distribuir el contingut en pestanyes. En aquest cas, cada pestanya correspon a l'horari d'una sessió concreta.

Quan l'usuari selecciona l'horari d'una sessió, el llistat a l'esquerra es filtrarà per totes les reunions que poden tenir lloc i que encara no han estat fixades.

Si l'usuari decideix tornar a editar l'informació de l'esdeveniment, haurà de retornar a la vista prèvia premen el botó *Back* situat a la part inferior esquerra. Quan es realitza aquesta acció i s'havien fixat reunions prèviament, es demanarà a l'usuari si vol guardar de forma temporal la posició de les reunions col·locades. Editar l'informació registrada pot provocar que es perdin aquestes posicions, veure Secció 8.6.

Pel fet que es permet a l'usuari fixar reunions en determinats espais, també es permet a l'usuari guardar els canvis realitzats per poder-los tornar a carregar en un futur. El boto *Save* permet aquesta funcionalitat.

Per últim, el boto *Compute* executarà el codificador i *solver* per mostrar una solució, si existeix, de les dades entrades i es col·locaran totes les reunions a la posició obtinguda. D'acord amb la codificació a SAT que estem fent (veure Secció 8.5), i per fer tal de fer el càlcul de la solució més eficient, el *solver* només ens dirà a quina sessió es fa cada reunió. La distribució de les taules es fa automàticament a posteriori, omplenant primer les taules amb un índex més petit.

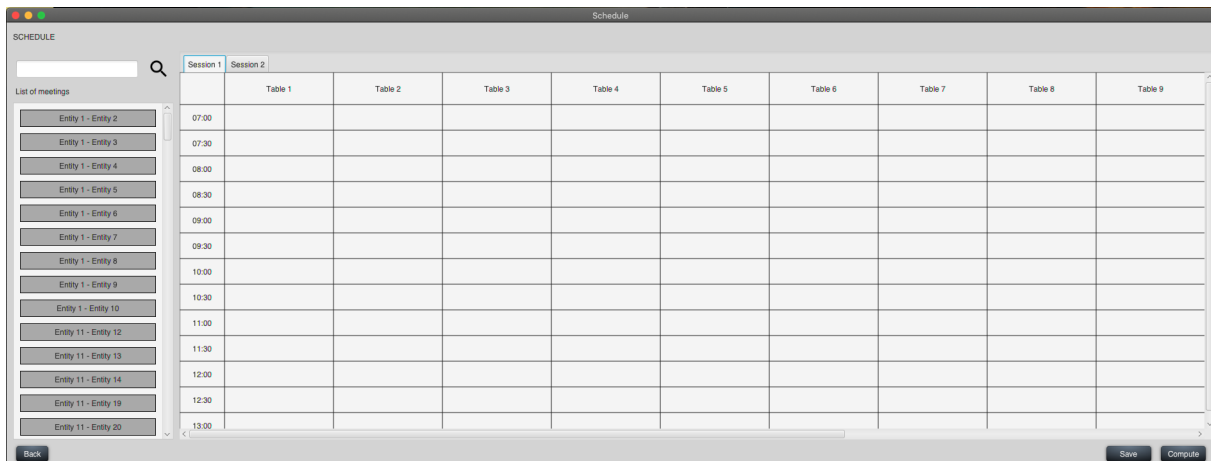


Figura 8.19: Captura de pantalla de la vista d'editar entitats.

8.4.1 *Drag-and-drop*

Aquesta vista permet la utilització del mecanisme *drag-and-drop* per a fixar les reunions situades al llistat esquerre a una posició de l'horari, per així atorgar-li un *time slot*, taula i sessió. També es pot arrossegar una reunió ja fixada a una altra posició, o bé, tornar-la al llistat de reunions pendents a col·locar.

Quan un requadre d'una reunió detecta l'acció *drag*, aquesta es transforma en una vista flotant que es pot desplaçar per la pantalla. Si aquest es deixa caure sobre un apartat incorrecte o bé fora de l'aplicació, es tornarà el requadre a la seva posició inicial.

Quan una casella de l'horari detecta que s'ha fet l'acció *drop*, aquesta cridarà el verificador de dades (vegeu Secció 8.6) per a comprovar si es pot col·locar la reunió. Si s'accepta la transició, aquesta reunió passarà a estar fixada i s'assignaran els valors de taula, *time slot* i sessió. En cas contrari, el requadre tornarà a la seva posició inicial.

El llistat de reunions per col·locar també acceptarà l'acció *drop*, desvinculant la reunió moguda de les dades que tenia prèviament.

8.5 Model de restriccions

En aquest apartat es detallaran les dades d'entrada que defineixen una instància (paràmetres), les variables Booleanes i les restriccions del model que es codifica amb ScalAT. Com ja s'ha esmentat a la introducció, el model que proposem és una adaptació del presentat a [2]. En concret, en el present projecte no s'optimitza cap funció objectiu ni es té en conta cap noció d'equitat entre la programació de reunions de les entitats, però en canvi es permeten reunions de més de dues entitats, així com que les empreses puguin tenir més d'un participant.

8.5.1 Paràmetres

nMeetings Nombre de reunions total.

nSlots Nombre de Slots total.

nSessions Nombre de sessions total.

attendesXParticipan Nombre de participants per cada entitat.

taulesXSessio Nombre de taules per cada sessió.

entityMeetings Llistat de reunions per a cada entitat.

forbidden Per a cada entitat, els slots als quals no assistirà, cada valor és un identificador d'un Slot.

sessioSlots Per a cada sessió, els identificadors dels Slots que en formen part.

sessioMeetings Per a cada sessió, els identificadors de les reunions que hi poden tenir lloc.

meetingSessions Per a cada reunió, els identificadors de les sessions a les quals pot portar-se a terme.

meetingEntities Per a cada reunió, els identificadors de les entitats que hi participen.

predef Llistat de les reunions prefixades mitjançant *drag-and-drop*, amb la sessió i Slot marcats.

8.5.2 Variables

Com que fem codificacions a SAT, totes les variables són Booleanes. El model conté una matriu de variables anomenada **schedule**, amb nombre de files *nMeetings*, i nombre de columnes *nSlots*. La variable de la fila *i*, columna *j* serà certa si i només si la reunió número *i* es fa en el time slot número *j*.

8.5.3 Restriccions

A continuació es detallen les restriccions del model. De cada restricció, se'n proporciona una definició formal i com s'ha implementat amb la llibreria ScalAT.

Restricció 1 La quantitat de reunions que pot tenir una entitat en un *Time Slot* determinat és com a molt el seu nombre d'assistent. S'utilitza una restricció At-Most-K.

$atMost(attendesXParticipan(p), \{schedule_{i,j} | i \in entityMeetings(p)\}) \forall p \in 0..nEntities, j \in 0..nSlots.$

```
for (p <- entityMeetings.indices; ts <- 0 until nSlots) {
  val list = (for (i <- entityMeetings(p)) yield (schedule(i)(ts))).toList
  e.addAMK(list, attendesXParticipan(p))
}
```

En el model original, aquesta restricció era un At-Most-One.

Restricció 2 Per a cada reunió predefinida, afegeix una clàusula que obligui a fer-la en el time slot predefinit.

$\{schedule_{i,j} | i = p.meeting, j = p.slot\} \forall p \in predef$

```
for (p <- predef) {
  e.addClause(schedule(p.meeting)(sessioSlots(p.sessio)(p.slot)) :: List())
}
```

Restricció 3 Una entitat no té reunions en Time Slots restringits.

$\neg\{schedule_{i,j} | i \in entityMeetings(p), j \in forbidden(p)\} \in \forall p \in 0..nEntities$

```
for (entity <- entityMeetings.indices; forb <- forbidden(entity); meet <-
  entityMeetings(entity)) {
  e.addClause((- (schedule(meet)(forb))) :: List())
}
```

Restricció 4a Per a cada reunió, afegim una restricció de tipus Exactly-One en el conjunt de Slots als quals pot tenir lloc.

$exactlyOne(\{schedule_{i,j} | x \in meetingSessions(i), j \in sessioSlots(x)\} \forall i \in 0..nMeetings$

Restricció 4b Si aquesta reunió està limitada a una sessió o varies, es marcaran com a no disponibles tots els Slots que formin part d'altres sessions.

$\{\neg schedule_{i,j} | j \in sessioSlots(x)\} \forall i \in 0..nMeetings, \forall x \in 0..nSessio \setminus meetingSessions(i)$

```
for (meetingI <- meetingSessions.indices) {
  e.addEQQuad((for (ses <- meetingSessions(meetingI); slot <- sessioSlots(ses))
    yield (schedule(meetingI)(slot))).toList)

  for (g2 <- 0 until nSessions if !meetingSessions(meetingI).contains(g2); slot
    <- sessioSlots(g2)) {
    e.addClause(-schedule(meetingI)(slot) :: List())
  }
}
```

Restricció 5 Per a cada Slot de temps, només es poden col·locar tantes reunions com taules disponibles.

$\{atMost(taulesXSessio(s), \{schedule_{i,j} \mid i \in sessioMeetings(s)\} \mid j \in sessioSlots(s)) \forall s \in 0..nSessions$

```
for (tSes <- 0 until nSessions; tSlot <- sessioSlots(tSes)) {
  e.addAMK((for (i <- sessioMeetings(tSes)) yield schedule(i)(tSlot)).toList,
    taulesXSessio(tSes))
}
```

Aquesta restricció és la que s'hauria de modificar si es volgués tenir en compte la capacitat de les taules disponibles. Ara mateix s'assumeix que qualsevol reunió es pot fer a qualsevol taula.

8.6 Validació de dades

Aquest software permet editar la informació entrada en tot moment, i això podria provocar inconsistències entre les dades registrades si no s'apliquen validacions de dades. Per exemple, si tenim un esdeveniment amb dues sessions, registrem una entitat i indiquem que tindrà assistència completa i posteriorment canviem que l'esdeveniment té tres sessions, l'entitat registrada no tindria cap informació sobre la tercera sessió. Per a controlar aquests casos no desitjats, s'ha dissenyat un validador de dades que controla i edita la informació cada cop que es realitzen canvis, es vol guardar informació o es vol navegar entre vistes principals. Aquest apartat detalla els diferents casos on el validador de dades modificarà de forma dràstica la informació registrada.

Editar número de sessions. Es modificarà l'assistència de totes les entitats entrades, és a dir, si s'han afegit noves sessions, totes les entitats passaran a tenir assistència total a les noves sessions. Per altra banda, totes les sessions eliminades s'esborraran de forma permetent començant pel final. Pel que fa a les reunions, eliminar sessions reiniciarà la sessió escollida d'aquelles reunions que tenien com a preferència una sessió que ara ja no existeix. Si aquesta reunió estava fixada en una taula i *time slot* d'una de les sessions esborrades, també es reiniciarà aquesta informació i quedarà com a no col·locada.

Modificar durades reunions. Modificar la durada de les reunions canvia per complet l'estructura de les sessions, per tant, es reiniciarà per complet tota la informació relacionada amb les sessions. Pel que fa a les entitats, es marcaran tots els *time slots* com a disponibles però no es modificarà l'assistència general a una sessió. Les reunions fixades deixaran d'estar-ho.

Canviar hora inici o fi d'una sessió. El procés de validació serà similar a l'apartat anterior però només afectarà la sessió modificada, és a dir, només es reiniciarà l'assistència d'entitats a la sessió corresponent i es descol·locaran les reunions col·locades a aquella sessió.

Canviar número de taules d'una sessió. En cas d'afegir més taules no es produirà cap canvi. Si es redueix el número, les reunions col·locades a una taula esborrada tornaran al llistat de reunions pendents de programar.

Guardat de dades. Per a poder guardar les dades a un fitxer, haurà d'estar registrat el nom de l'esdeveniment, ja que aquest serà el nom del fitxer final.

Crear una entitat. En crear una entitat nova, la única restricció que s'aplica és que no existeixi cap altra entitat amb el mateix identificador o nom.

Crear reunió. Per a crear una reunió, es controla que aquesta no tingui entitats duplicades i un mínim de dues entitats participants. Es permeten reunions duplicades, és a dir, poden existir dues reunions idèntiques, ja que dues entitats es poden voler reunir més d'una vegada.

Accés a la vista horari. Per a poder accedir a la vista horari, el validador de dades comprovarà que cap reunió conté una entitat participant que encara no està registrada al llistat d'entitats.

***Drag-and-drop* de reunions.** Per a poder fixar una reunió en una taula es comprovaran diversos factors. En primera instància es comprovarà que totes les entitats participants estaran disponibles en aquell *time slot*, és a dir, han marcat que estaran disponibles. Seguidament es comprovarà que totes les entitats tenen prou assistents, és a dir, si una entitat té més reunions ja fixades, haurà de tenir com a mínim un assistent més disponible.

Capítol 9

Implementació i proves

En aquest capítol es descriuran els problemes trobats i les diferents proves realitzades.

9.1 Problemes

Descripció dels problemes trobats durant el desenvolupament del projecte

9.1.1 Sistema de taules

La descripció inicial del problema permetia l'ús i configuració de taules per a cada sessió de diferents característiques. El disseny inicial permetia incloure per a les diferents sessions taules de mida concreta, això comportava que no totes les reunions es podrien col·locar en una taula determinada. Per exemple, si volem col·locar una reunió en la qual participen quatre entitats, aquesta només es podrà dur a terme en una taula amb quatre seients o més, si no es disposa de cap taula amb aquestes característiques, el problema és directament insatisfactible.

El conjunt de restriccions obtingut no contempla per al moment l'ús de taules de mida determinada. De totes maneres, la interfície gràfica i la lògica de gestió de taules diverses per a cada sessió ja ha estat implementada, però ocultada a l'usuari.

9.1.2 Cost en temps de computació

El temps actual de computació en el pitjor dels casos és exponencial. La utilització d'un validador de dades ha permès reduir evitar a l'usuari còmputos innecessaris, ja que es prohibeix definir solucions parcials quan es detecta que aquestes no tenen solució.

Tot i així, l'usuari de l'aplicació ha de ser conscient que el problema que es pretén resoldre és computacionalment dur, i que amb mides de problema grans pot ser que no obtingui una solució en un temps raonable.

9.2 Proves

En aquest apartat es descriuran les proves realitzades durant el desenvolupament del projecte.

9.2.1 Codificació SAT

El problema principal amb provar una codificació SAT és que no existeixen eines per a depurar el codi, si una solució que hauria de ser vàlida no ho és, només s'obté el resultat Insatisfactible, no quines variables

ho han causat. L'única forma de depurar el problema és revisar amb detall les dades entrades i la codificació de les restriccions en cerca d'errades.

Per a comprovar que l'eina desenvolupada pot resoldre instàncies de mides relativament grans, s'ha posat a prova amb instàncies extretes de l'article [2]. Aquestes instàncies ja havien estat resoltes amb eines eficients, i en tots els casos s'han obtingut resultats similars en temps.

9.2.2 Software

La realització de proves sobre el comportament de la interfície gràfica i la lògica del software es realitza mitjançant la creació d'escenaris de prova reduïts. Es dissenya un esdeveniment amb unes característiques concretes, amb poques reunions i entitats, de tal manera que és fàcil de saber si el resultat obtingut és el desitjat. Tots els comportaments no esperats es detallen duran l'execució i es depuren a posterior.

Example 9.2.2.1 *Volem provar la resolució a satisfactible per un determinat esdeveniment amb una sessió de quatre Time Slots, tres empreses i quatre reunions. Cada entitat disposa només d'un participant i disponibilitat total. Les reunions seran entre l'entitat 1 i l'entitat 2, entre l'entitat 1 i l'entitat 3, l'entitat 2 i l'entitat 3 i finalment, una reunió amb les tres entitats. Les solucions possibles al problema són, en qualsevol ordre, les quatre reunions col·locades una a cada reunió.*

The screenshot shows the 'CREATING NEW SCHEDULE' window in the B2B Management application. The window is divided into several sections:

- Event Details:** Fields for Event Name (Prova1), Location (Gircna), Number of sessions (1), and Meetings duration (30 minutes). A 'Set up sessions' button is below.
- Set New Entity:** Fields for Entity Name, Entity ID, and Number of attendees. Buttons for 'Set arrival info' and 'Add entity' are present.
- Set New Meeting:** A dropdown for 'Preferred session' (set to 'All') and two input fields for 'Meeting participants'. Buttons for 'Add participant' and 'Add Meeting' are at the bottom.
- Entity Table:** A table with columns ID, Name, Attendees, and Sessions attending. It contains three rows for Entitat 1, Entitat 2, and Entitat 3, each with 1 attendee and Session 1.
- Sessions Table:** A table with columns Sessions and Entities. It lists four sessions, each with 'All' attendees and a list of entities (Entitat 1, Entitat 2, Entitat 3).
- Navigation:** Buttons for 'Load Instance', 'Load', 'Save', and 'Next' are at the bottom right.

Figura 9.1: Captura de pantalla d'entrada de dades.

Un cop entrades les dades, fixem la reunió amb les tres entitats arrossegant-la al primer Time Slot. Seguidament computem una solució i obtenim el resultat.

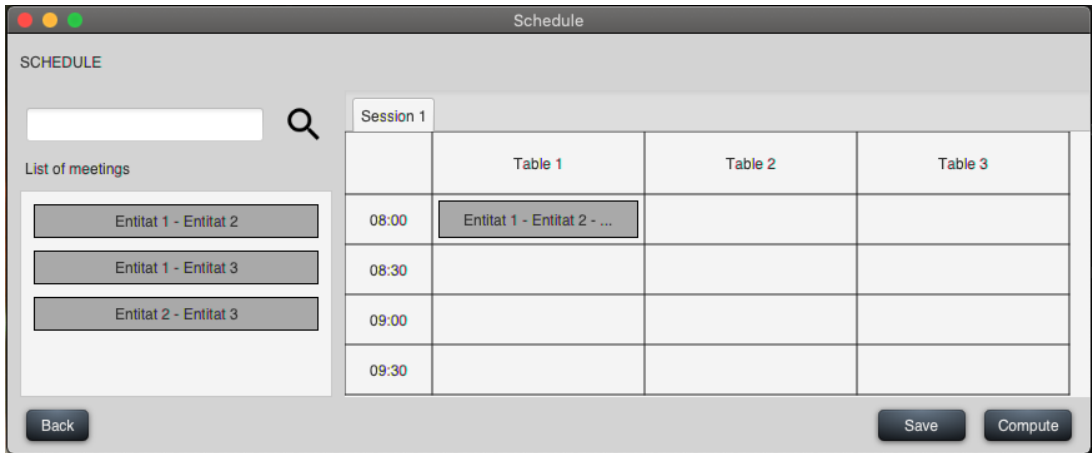


Figura 9.2: Captura de pantalla amb una reunió fixada.

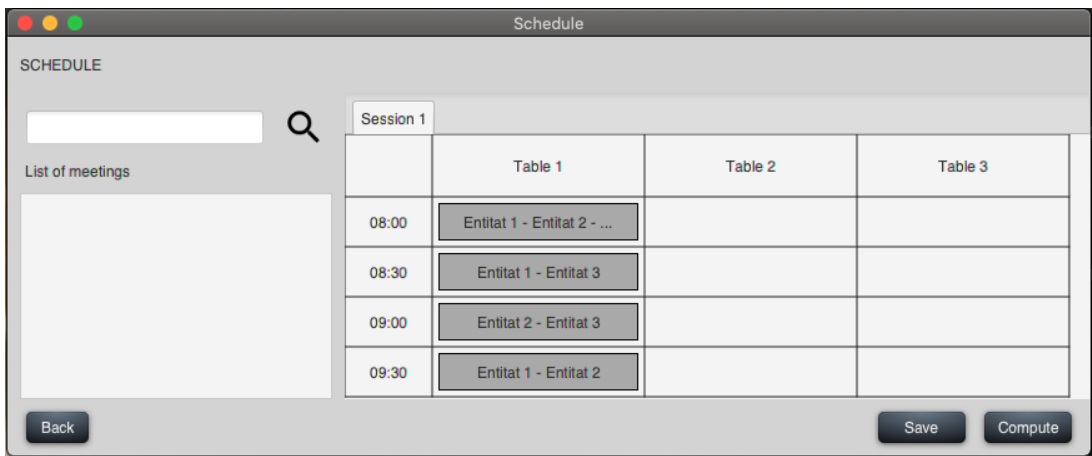


Figura 9.3: Captura de pantalla amb el resultat.

Example 9.2.2.2 *Volem provar la resolució a insatisfactible per un determinat esdeveniment. Utilitzarem la mateixa configuració que en el cas previ, però aquest cop l'esdeveniment tindrà dues sessions, i la reunió de tres entitats estarà limitada a la sessió 2. Per provocar que la instància sigui no satisfactible, marcarem que l'entitat 2 no pot assistir a la sessió 2.*

Entity Detail

Entity Name: Entitat 2

Entity ID: 2

Number of attendees: 1

Session 1		Session 2	
	<input checked="" type="checkbox"/>		<input type="checkbox"/>
08:00	<input checked="" type="checkbox"/>	08:00	<input checked="" type="checkbox"/>
08:30	<input checked="" type="checkbox"/>	08:30	<input checked="" type="checkbox"/>
09:00	<input checked="" type="checkbox"/>	09:00	<input checked="" type="checkbox"/>
09:30	<input checked="" type="checkbox"/>	09:30	<input checked="" type="checkbox"/>

Erase Save

Figura 9.4: Editar una entitat

Meeting Detail

Select preferred session: Session 2

- Entitat 1
- Entitat 2
- Entitat 3

Add participant Erase Save

Figura 9.5: Editar una reunió

Un cop modificades les dades, computem la solució i obtenim un missatge d'error indicant que no existeix possible solució.

Schedule

SCHEDULE

List of meetings

- Entitat 1 - Entitat 2
- Entitat 1 - Entitat 3
- Entitat 2 - Entitat 3

Warning

Warning

There is no solution to the corresponding input data

OK

Table 3

09:00			
09:30			

Back Save Compute

Figura 9.6: Resultat del comput.

Capítol 10

Implantació i resultats

10.1 Resultats experimentals

Per a realitzar una demostració del rendiment del software, s'utilitzarà una instància generada per a realitzar proves de gran calibre, anomenada *forum-13.original*, extreta de l'article [2]. La maquinària utilitzada per a computar els resultats té la següent configuració:

SO macOS Catalina 10.15.4

CPU Intel Core i7

RAM 16GB DDR3 a 1600 MHz

Aquesta instància conte la informació d'un esdeveniment amb dues sessions. A cada sessió es disposa de catorze taules. Hi han incloses setanta entitats amb les corresponents cent quaranta reunions.

La instància també disposa de la informació sobre els *Forbidden Time Slots* d'algunes de les empreses, limitant així els instants de temps als quals poden assistir.

Per altra banda, el format d'instància també pot contenir la posició fixada d'algunes de les reunions. En aquest cas particular, cap instància ha estat fixada.

A continuació es mostra el temps de còmput necessari de l'aplicació B2B Managment, en segons, per trobar una solució d'aquesta instància, i es compara amb el temps de còmput d'un model *MaxSAT* presentat a l'article [2]. Cal remarcar que el model MaxSAT no resol exactament el mateix problema que resol B2B Management, ja que presenta algunes diferències com ara una funció objectiu que cal optimitzar, o restriccions d'homogeneïtat sobre les solucions.

Els resultats obtingut utilitzant una eina externa amb el solver MaxSAT són 2.1 segons, mentre que el software B2B Management tarda 3.1 segons a computar el resultat i mostrar-lo per pantalla.

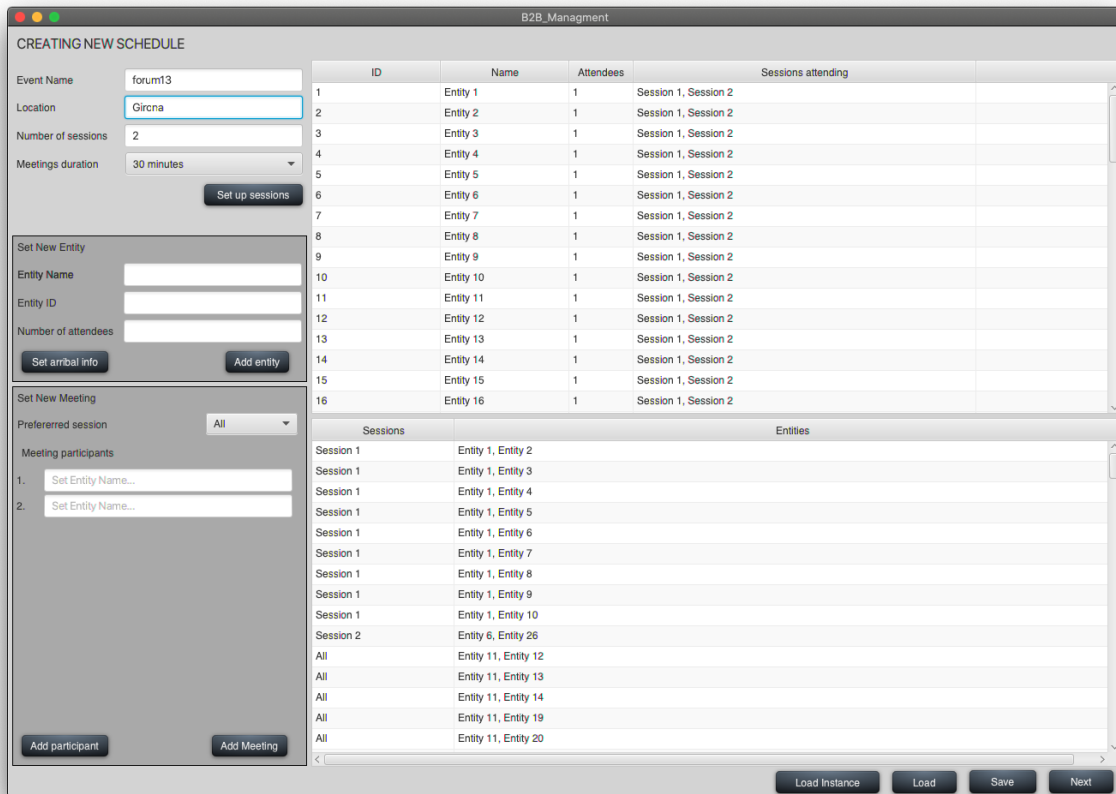


Figura 10.1: Dades d'entrada instància forum-13.original.

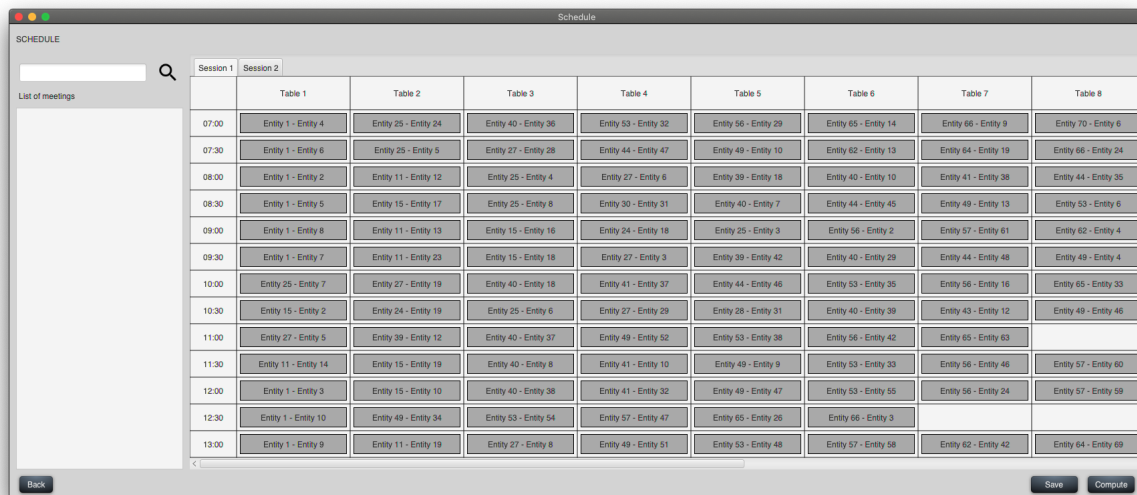


Figura 10.2: Resultat obtingut amb la instància forum-13.original.

10.2 Normativa i legislació

El projecte realitzat compleix al complet la legislació vigent. A causa del fet que no guarda cap mena d'informació de caràcter personal ni sensible.

Per altra banda, les llibreries utilitzades són de codi obert, per tant poden ser utilitzades en un projecte de software sempre que es mencioni el seu ús.

Capítol 11

Conclusions

Els objectius principals d'aquest projecte eren desenvolupar una eina que permetés gestionar un esdeveniment amb reunions B2B. Aquest objectiu ha estat completat aportant una eina funcional que realitza la tasca designada. A partir d'aquesta eina es poden aplicar diverses millores que atorguin més funcionalitat.

Per altra banda, estava com a objectiu posar en pràctica els coneixements obtinguts sobre la codificació SAT per a resoldre problemes de satisfacció de restriccions i l'optimització d'aquests. Això a permès posar en pràctica la programació funcional apresada i explorar amb més detall les possibilitats que permet el llenguatge Scala.

Un altre objectiu principal, en aquest cas més personal, era posar en pràctica els coneixements, apresos durant el grau universitari, en anàlisi i desenvolupament de software en tots els passos del projecte. Aquests coneixements han permès la realització d'un disseny amb l'incorporació de patrons de disseny i una estructura de dades correctament analitzada.

El desenvolupament d'aquest projecte ha comportat també l'estudi i aprenentatge del llenguatge FXML i la utilització de la llibreria JavaFX. He adquirit suficient experiència teòrica i pràctica per tal de poder analitzar i desenvolupar amb profunditat aplicacions en aquest llenguatge.

És important mencionar, que per qüestions de temps, s'ha decidit simplificar parts de les restriccions, com el format de taules de diferent nombre de seients, les quals podrien ser incorporades com a treball futur. S'hauria de dissenyar la part del model de restriccions i habilitar la part de configuració de taules actualment oculta.

La distribució dels blocs de tasques plantejats originalment (Secció 4.2) s'han vist mantinguts, l'única diferència a estat en la durada d'aquests blocs. Els blocs 2, 3 i 4 s'han vist allargats una setmana cadascun a causa de la corba d'aprenentatge de la llibreria JavaFX.

El projecte m'ha proporcionat coneixements en la planificació i desenvolupament d'un software professional, així com la pràctica per a ser incorporat en un equip de treball amb metodologia àgil.

Capítol 12

Treball futur

En aquest capítol es descriuran les opcions de millores per un treball futur que es podria realitzar al projecte.

12.1 Millores en la interfície gràfica

El disseny estètic de la interfície de l'usuari es podria millorar profundament, contractant a un professional de disseny. Això permetria tenir una estètica més treballada i còmode per a un usuari de l'aplicació.

Per altra banda, es podrien incorporar noves interfícies visuals per a permetre a un equip de recerca veure amb profunditat tota la interacció de la codificació amb el *solver* així com els resultats obtinguts en cada prova per a millorar la investigació i optimització de les restriccions.

12.2 Codificació de les restriccions restants

El conjunt de restriccions presentava un subconjunt no essencial per obtenir una solució, que implem-taven optimització en el resultat final.

Un subconjunt de restriccions permeten limitar que si una entitat té una reunió en un determinat *Time Slot*, es forci al fet que la següent reunió sigui en el *Time Slot* consecutiu i no es deixin espais de temps buits.

12.3 Verificació de dades

El software desenvolupat incorpora una verificació de dades a mesura que l'usuari va entrant les diverses entitats i reunions. De totes maneres, es podria incorporar un sistema més restrictiu per evitar errors per part de l'usuari mostrant pantalles d'ajuda i evitant que aquest entri formats de dades erronis. Per altra banda, es podria millorar la comprovació d'integritat de les dades abans de permetre accedir a la configuració de l'horari final.

12.4 Millores en el sistema de taules

Per raons de temps, el software utilitza un sistema de taules per a les reunions de mida indeterminada, és a dir, es permet utilitzar la mateixa taula per reunions amb dos participants com reunions amb vuit participants.

Una millora important seria el disseny i codificació de les restriccions pertinents per a limitar la mida de les taules a un format determinat i un nombre d'unitats determinat de cada tipus. Això proporcionaria una aproximació més real a un esdeveniment.

12.5 Optimització i millora de restriccions

Els conjunts de restriccions sempre presenten opcions a ser millorats i formes d'obtenir solucions amb menys temps i més òptims.

Capítol 13

Bibliografia

- [1] Armin Biere, Marijn Heule i Hans van Maaren. *Handbook of satisfiability*. Vol. 185. IOS press, 2009.
- [2] Bofill, Miquel and Garcia, Marc and Suy, Josep and Villaret, Mateu. “MaxSAT-based scheduling of B2B meetings”. A: *International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*. Springer. 2015, pàg. 65-73.
- [3] Miquel Bofill et al. “Scheduling B2B meetings”. A: *International Conference on Principles and Practice of Constraint Programming*. Springer. 2014, pàg. 781-796.
- [4] Daniel Le Berre and Anne Parrain. “The Sat4j library”. A: *Journal on Satisfiability, Boolean Modeling and Computation, Volume 7*. 2010, pàg. 59-64.
- [5] Martin Odersky i Bill Venners. *Programming in Scala*. Artima Inc, 2008.
- [6] Oracle. *JavaFX*. (accessed 2020-04-10). URL: <https://www.oracle.com/java/technologies/javase/javafx-overview.html>.
- [7] Gilles Pesant, Gregory Rix i Louis-Martin Rousseau. “A comparative study of mip and cp formulations for the b2b scheduling optimization problem”. A: *International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*. Springer. 2015, pàg. 306-321.
- [8] *The Scala Programming Language*. (accessed 2020-03-08). URL: <https://www.scala-lang.org/>.

Capítol 14

Annex

14.1 Model de restriccions complet

En aquest apartat es detallaran les dades d'entrada que defineixen una instància (paràmetres), les variables Booleanes i les restriccions del model que es codifica amb ScalAT.

14.1.1 Paràmetres

nEntities Nombre d'entitats total.

nMeetings Nombre de reunions total.

nSlots Nombre de Slots total.

nSessions Nombre de sessions total.

attendesXParticipan Nombre de participants per cada entitat.

taulesXSessio Nombre de taules per cada sessió.

entityMeetings Llistat de reunions per a cada entitat.

forbidden Per a cada entitat, els slots als quals no assistirà, cada valor és un identificador d'un Slot.

sessioSlots Per a cada sessió, els identificadors dels Slots que en formen part.

sessioMeetings Per a cada sessió, els identificadors de les reunions que hi poden tenir lloc.

meetingSessions Per a cada reunió, els identificadors de les sessions a les quals pot portar-se a terme.

meetingEntities Per a cada reunió, els identificadors de les entitats que hi participen.

predef Llistat de les reunions prefixades mitjançant *drag-and-drop*, amb la sessió i Slot marcats.

14.1.2 Variables

Com que fem codificacions a SAT, totes les variables són Booleanes. El model conté una matriu de variables anomenada **schedule**, amb nombre de files *nMeetings*, i nombre de columnes *nSlots*. La variable de la fila *i*, columna *j* serà certa si i només si la reunió número *i* es fa en el time slot número *j*.

14.1.3 Restriccions

En aquest apartat es detallen totes les restriccions per a resoldre un problema B2B.

Restricció 1 La quantitat de reunions que pot tenir una entitat en un *Time Slot* determinat és com a molt el seu nombre d'assistent. S'utilitza una restricció At-Most-K.

$$atMost(attendesXParticipan(p), \{schedule_{i,j} | i \in entityMeetings(p)\}) \forall p \in 0..nEntities, j \in 0..nSlots.$$

Restricció 2 Afegeix una clàusula certa per a cada reunió predefinida.

$$\{schedule_{i,j} | i = p.meeting, j = p.slot\} \forall p \in predef$$

Restricció 3 Una entitat no té reunions en Time Slots restringits.

$$\neg \{schedule_{i,j} | i \in entityMeetings(p), j \in forbidden(p)\} \in \forall p \in 0..nEntities$$

Restricció 4 Per a cada reunió, afegim una restricció de tipus Exactly-One en el conjunt de Slots als quals pot tenir lloc.

Restricció 5 Si aquesta reunió està limitada a una sessió o varies, es marcaran com a no disponibles tots els Slots que formin part d'altres sessions.

$$exactlyOne(\{schedule_{i,j} | x \in meetingSessions(i), j \in sessioSlots(x)\} \forall i \in 0..nMeetings$$

$$\{\neg schedule_{i,j} | j \in sessioSlots(x)\} \forall i \in 0..nMeetings, \forall x \in 0..nSessio \setminus meetingSessions(i)$$

Restricció 6 Per a cada Slot de temps, només es poden col·locar tantes reunions com taules disponibles.

$$\{atMost(taulesXSessio(s), \{schedule_{i,j} | i \in sessioMeetings(s)\} | j \in sessioSlots(s)\} \forall s \in 0..nSessions$$

14.2 forum13.original

```
nBusiness = 70;
nMeetings = 154;
nTables = 14;
nTotalSlots = 21;
nMorningSlots = 13;

requested = [|1, 2, 1,
|1, 3, 1,
|1, 4, 1,
|1, 5, 1,
|1, 6, 1,
|1, 7, 1,
|1, 8, 1,
|1, 9, 1,
|1, 10, 1,
|6, 26, 2,
|11, 12, 3,
|11, 13, 3,
|11, 14, 3,
|11, 19, 3,
|11, 20, 3,
|11, 21, 3,
|11, 22, 3,
|11, 23, 3,
|15, 2, 1,
|15, 10, 1,
|15, 16, 1,
|15, 17, 1,
|15, 18, 1,
|15, 19, 1,
|24, 18, 1,
|24, 19, 1,
|25, 3, 1,
|25, 4, 1,
|25, 5, 1,
|25, 6, 1,
|25, 7, 1,
|25, 8, 1,
|25, 24, 1,
|27, 3, 1,
|27, 5, 1,
|27, 6, 1,
|27, 8, 1,
|27, 19, 1,
|27, 28, 1,
|27, 29, 1,
|28, 31, 1,
|30, 31, 1,
|32, 31, 2,
|33, 26, 2,
|33, 31, 2,
|39, 12, 1,
|39, 18, 1,
|39, 42, 1,
```

|40, 7, 3,
|40, 8, 3,
|40, 10, 3,
|40, 18, 3,
|40, 29, 3,
|40, 36, 3,
|40, 37, 3,
|40, 38, 3,
|40, 39, 3,
|41, 10, 1,
|41, 32, 1,
|41, 37, 1,
|41, 38, 1,
|43, 12, 1,
|44, 35, 3,
|44, 45, 3,
|44, 46, 3,
|44, 47, 3,
|44, 48, 3,
|49, 4, 1,
|49, 9, 1,
|49, 10, 1,
|49, 13, 1,
|49, 34, 1,
|49, 46, 1,
|49, 47, 1,
|49, 50, 1,
|49, 51, 1,
|49, 52, 1,
|53, 6, 1,
|53, 7, 1,
|53, 30, 1,
|53, 32, 1,
|53, 33, 1,
|53, 35, 1,
|53, 36, 1,
|53, 38, 1,
|53, 48, 1,
|53, 54, 1,
|53, 55, 1,
|56, 2, 3,
|56, 3, 3,
|56, 10, 3,
|56, 14, 3,
|56, 16, 3,
|56, 24, 3,
|56, 28, 3,
|56, 29, 3,
|56, 42, 3,
|56, 46, 3,
|57, 26, 1,
|57, 47, 1,
|57, 58, 1,
|57, 59, 1,
|57, 60, 1,
|57, 61, 1,
|62, 4, 1,

```
|62, 12, 1,
|62, 13, 1,
|62, 23, 1,
|62, 24, 1,
|62, 39, 1,
|62, 42, 1,
|62, 46, 1,
|62, 54, 1,
|64, 19, 1,
|64, 29, 1,
|64, 46, 1,
|64, 50, 1,
|64, 52, 1,
|64, 63, 1,
|64, 68, 1,
|64, 69, 1,
|65, 14, 3,
|65, 24, 3,
|65, 26, 3,
|65, 33, 3,
|65, 38, 3,
|65, 45, 3,
|65, 46, 3,
|65, 52, 3,
|65, 54, 3,
|65, 60, 3,
|65, 63, 3,
|65, 64, 3,
|66, 2, 3,
|66, 3, 3,
|66, 4, 3,
|66, 9, 3,
|66, 18, 3,
|66, 24, 3,
|66, 26, 3,
|66, 29, 3,
|66, 34, 3,
|66, 39, 3,
|66, 45, 3,
|66, 50, 3,
|66, 59, 3,
|66, 63, 3,
|66, 67, 3,
|70, 6, 1,
|70, 7, 1,
|70, 8, 1,
|70, 32, 1,
|70, 34, 1,
|70, 35, 1,
|];
```

```
nMeetingsBusiness = [9, 4, 5, 5, 3, 6, 5, 5, 3, 6, 8, 4, 3, 3, 6, 2, 1, 5, 5, 1,
1, 1, 2, 7, 7, 5, 7, 3, 5, 2, 4, 4, 4, 3, 3, 2, 2, 4, 6, 9, 4, 3, 1, 5, 3, 6,
3, 2, 10, 3, 1, 3, 11, 3, 1, 10, 6, 1, 2, 2, 1, 9, 3, 9, 12, 15, 1, 1, 1, 6, ];
```

```
fixat = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
```



```
{0,},  
{0,},  
{0,},  
{0,},  
{0,4,},  
{0,},  
{0,13,},  
{0,},  
{0,},  
{0,},  
{0,},  
{0,},  
{0,},  
{0,5,},  
{0,},  
{0,},  
{0,},  
];
```

Capítol 15

Manual d'usuari

15.1 Requisits de l'aplicació

Per tal que l'aplicació B2B-Management sigui capaç de dur a terme la totalitat de la seva funcionalitat, l'executable Java del solver Sat4j ha de romandre en la mateixa carpeta que l'executable del software. Si no es compleix aquest punt, el software serà incapaç de computar una solució per l'instància creada.

15.2 Descripció de funcionament

Per a facilitar la comprensió i funcionament de l'aplicació, es detallarà la creació i edició de cada element que es pot afegir a un esdeveniment.

15.2.1 Entrada de dades

L'entrada de dades té lloc en la primera vista de l'aplicació (Figura 15.1) i les seves finestres específiques. En aquest apartat es detalla com entrar i editar cada part de les dades que componen un esdeveniment. S'adjunten captures de pantalla de cada fragment per a facilitar la comprensió i evitar dubtes.

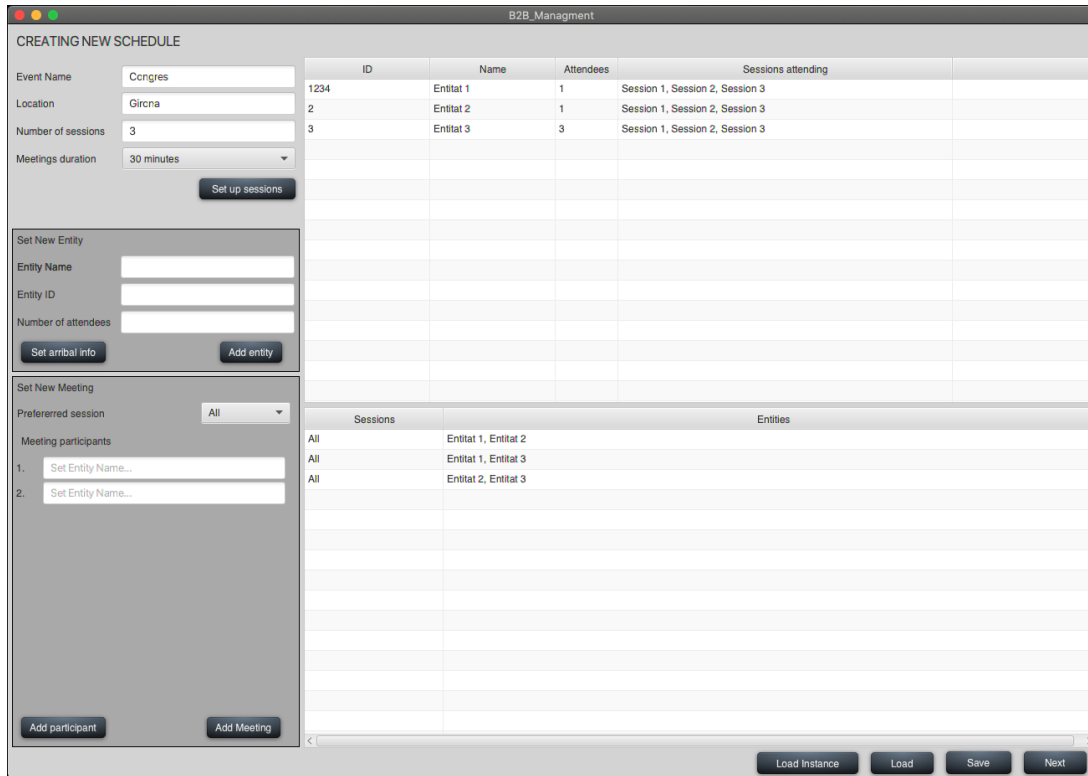


Figura 15.1: Vista general inicial.

Crear sessions

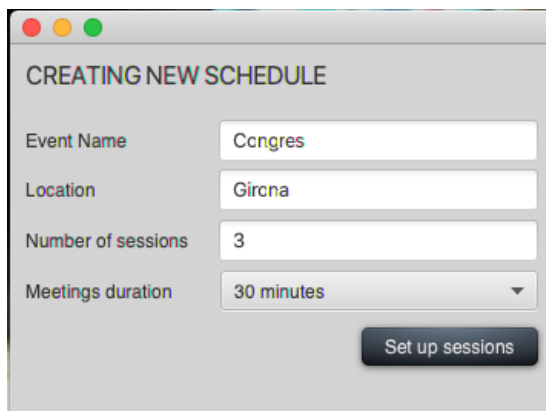


Figura 15.2: Fragment entrar dades de sessió.

En iniciar l'aplicació, el pas més important és introduir les dades principals referents a l'esdeveniment (Figura 15.2), el nom i ubicació, així com el nombre de sessions que tindrà i la durada prevista que tindran les sessions. Si l'usuari ho desitja, pot entrar la informació detallada de cada sessió pressionant el botó *Set up sessions*. Si no s'hi accedeix, l'aplicació donarà per defecte un horari de les 08:00 a les 20:00 hores i només una taula per sessió.

Editar sessions

	Session 1	Session 2	Session 3
Date	6/6/2020	6/7/2020	6/8/2020
Start Hour	08:00	08:00	08:00
End Hour	13:30	13:30	13:30
Tables	11	11	11

Save

Figura 15.3: Vista editar una sessió.

Si l'usuari decideix editar les sessions, accedirà a aquesta vista (Figura 15.3). En aquest apartat es mostren per columnes les diferents sessions disponibles, es pot editar la data, hora d'inici i hora fi de cada una. Per últim es poden entrar les taules disponibles a cada sessió. Cal remarcar que en editar aquesta informació quan ja hi ha entitats o reunions entrades, s'alterarà la seva informació relacionada amb les sessions. Sempre s'informarà l'usuari amb un missatge abans de realitzar cap canvi.

Crear entitat

Set New Entity

Entity Name: Entitat 4

Entity ID: 454321S

Number of attendees: 3

Set arribal info Add entity

Figura 15.4: Fragment crear entitat.

Seguidament s'entrarà la informació de totes les entitats que assistiran a l'esdeveniment. La informació essencial està formada pel nom, l'identificador i el nombre d'assistents. Si el gestor ho desitja, es pot configurar l'assistència pressionant el botó *Set arribal info*. Un cop tota la informació ha estat entrada, es pot registrar pressionat el botó *Add entity*. L'entitat s'afegirà al llistat i podrà ser editada posteriorment.

Editar assistència

Session 1	Session 2	Session 3
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
08:00 <input checked="" type="checkbox"/>	08:00 <input type="checkbox"/>	08:00 <input checked="" type="checkbox"/>
08:30 <input checked="" type="checkbox"/>	08:30 <input type="checkbox"/>	08:30 <input checked="" type="checkbox"/>
09:00 <input checked="" type="checkbox"/>	09:00 <input type="checkbox"/>	09:00 <input checked="" type="checkbox"/>
09:30 <input checked="" type="checkbox"/>	09:30 <input type="checkbox"/>	09:30 <input checked="" type="checkbox"/>
	10:00 <input type="checkbox"/>	10:00 <input checked="" type="checkbox"/>
	10:30 <input type="checkbox"/>	10:30 <input checked="" type="checkbox"/>
	11:00 <input type="checkbox"/>	11:00 <input checked="" type="checkbox"/>
	11:30 <input type="checkbox"/>	11:30 <input checked="" type="checkbox"/>
	12:00 <input checked="" type="checkbox"/>	12:00 <input checked="" type="checkbox"/>
	12:30 <input checked="" type="checkbox"/>	12:30 <input checked="" type="checkbox"/>
	13:00 <input checked="" type="checkbox"/>	13:00 <input checked="" type="checkbox"/>
	13:30 <input checked="" type="checkbox"/>	13:30 <input checked="" type="checkbox"/>
	14:00 <input checked="" type="checkbox"/>	14:00 <input checked="" type="checkbox"/>
	14:30 <input checked="" type="checkbox"/>	14:30 <input checked="" type="checkbox"/>
	15:00 <input checked="" type="checkbox"/>	15:00 <input checked="" type="checkbox"/>
	15:30 <input checked="" type="checkbox"/>	15:30 <input checked="" type="checkbox"/>

Si es desitja editar la disponibilitat d'una entitat, es pot indicar a quines sessions atindrà deixant seleccionats els *check*. En cas de només faltat durant un temps determinat, es pot indicar a quins instants de temps no estarà disponible. Un cop s'ha acabat d'entrar la informació, es pot guardar prement el botó *Save*. En cas de voler cancel·lar l'operació, es pot tancar la finestra de la forma tradicional.

Figura 15.5: Vista editar assistència.

Llistat d'entitats Un cop registrada una entitat, aquesta s'afegirà al llistat superior de l'aplicació principal (Figura 15.6). Si es desitja editar la seva informació, es podrà dur a terme fent doble clic sobre la fila corresponent.

ID	Name	Attendees	Sessions attending
1	Entitat 1	1	Session 1
2	Entitat 2	1	Session 1
3	Entitat 3	1	Session 1
454321S	Entitat 4	3	Session 1

Figura 15.6: Fragment llistat d'entitats.

Editar entitat

Entity Detail

Entity Name:

Entity ID:

Number of attendees:

Session 1	<input checked="" type="checkbox"/>	Session 2	<input checked="" type="checkbox"/>	Session 3	<input checked="" type="checkbox"/>
08:00	<input checked="" type="checkbox"/>	08:00	<input checked="" type="checkbox"/>	08:00	<input checked="" type="checkbox"/>
08:30	<input checked="" type="checkbox"/>	08:30	<input checked="" type="checkbox"/>	08:30	<input checked="" type="checkbox"/>
09:00	<input checked="" type="checkbox"/>	09:00	<input checked="" type="checkbox"/>	09:00	<input checked="" type="checkbox"/>
09:30	<input checked="" type="checkbox"/>	09:30	<input checked="" type="checkbox"/>	09:30	<input checked="" type="checkbox"/>
		10:00	<input checked="" type="checkbox"/>	10:00	<input checked="" type="checkbox"/>
		10:30	<input checked="" type="checkbox"/>	10:30	<input checked="" type="checkbox"/>
		11:00	<input checked="" type="checkbox"/>	11:00	<input checked="" type="checkbox"/>
		11:30	<input checked="" type="checkbox"/>	11:30	<input checked="" type="checkbox"/>
		12:00	<input checked="" type="checkbox"/>	12:00	<input checked="" type="checkbox"/>
		12:30	<input checked="" type="checkbox"/>	12:30	<input checked="" type="checkbox"/>
		13:00	<input checked="" type="checkbox"/>	13:00	<input checked="" type="checkbox"/>
		13:30	<input checked="" type="checkbox"/>	13:30	<input checked="" type="checkbox"/>
		14:00	<input checked="" type="checkbox"/>	14:00	<input checked="" type="checkbox"/>
		14:30	<input checked="" type="checkbox"/>	14:30	<input checked="" type="checkbox"/>
		15:00	<input checked="" type="checkbox"/>	15:00	<input checked="" type="checkbox"/>
		15:30	<input checked="" type="checkbox"/>	15:30	<input checked="" type="checkbox"/>
		16:00	<input checked="" type="checkbox"/>	16:00	<input checked="" type="checkbox"/>

Si s'ha comès algun error en l'entrada de dades d'una entitat o es volen realitzar canvis en la seva informació, es pot accedir a aquesta vista (Figura 15.7). El funcionament és el mateix que el detallat en els apartats previs de creació d'una entitat. Si es desitja eliminar l'entitat, es pot dur a terme prement el botó *Erase*.

Figura 15.7: Vista edició entitat.

Crear reunió

Set New Meeting

Preferred session: Session 2

Meeting participants

- Entity 4
- Entity 1
- Entity 5

Add participant Add Meeting

En aquest apartat (Figura 15.8) s'entrarà la informació de les reunions. Es pot seleccionar la sessió a la qual es vol restringir i seguidament es poden entrar les entitats participants pel seu nom. Inicialment només es mostren dos espais per a dues entitats, si es desitja afegir més entitats, es pot prémer el botó *Add participant* que afegirà un requadre nou cada cop. Quan la informació estigui completa, es prem el botó *Add Meeting* i aquesta es registrarà al llistat de reunions. Si algun dels participants entrats no existeix, es notificarà. Si s'han entrat repetits, s'aturarà el procés de registre.

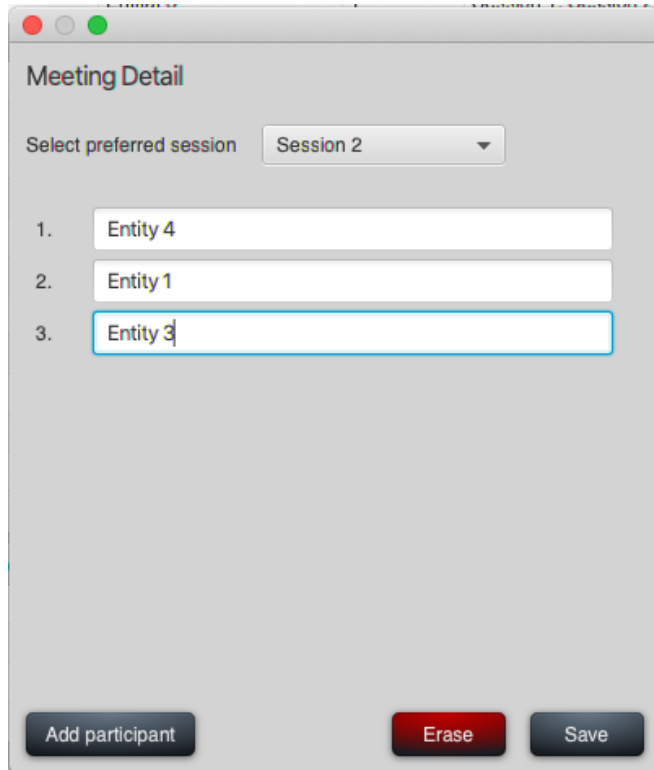
Figura 15.8: Fragment crear reunió.

Llistat de reunions Un cop registrada una reunió, aquesta s'afegirà al llistat inferior de l'aplicació principal (Figura 15.9). Si es desitja editar la informació d'una reunió, es podrà dur a terme fent doble clic sobre la fila corresponent.

Sessions	Entities
All	Entitat 1, Entitat 2, Entitat 3
All	Entitat 1, Entitat 2
All	Entitat 1, Entitat 3
All	Entitat 2, Entitat 3
Session 2	Entity 4, Entity 1, Entity 5

Figura 15.9: Fragment llistat de reunions.

Editar reunió



The screenshot shows a window titled "Meeting Detail". At the top, there is a label "Select preferred session" followed by a dropdown menu currently set to "Session 2". Below this, there is a list of three participants, each in a numbered box:

1. Entity 4
2. Entity 1
3. Entity 3

At the bottom of the window, there are three buttons: "Add participant" (dark grey), "Erase" (red), and "Save" (dark grey).

Figura 15.10: Vista edició reunió.

Si es vol editar la sessió de preferència o les entitats participants d'una reunió ja registrada, es podrà fer en aquest apartat (Figura 15.10). El funcionament és el mateix que en la vista principal.

Si es desitja eliminar la reunió, es pot dur a terme prement el botó *Erase*.

Guardar/Carregar En la vista principal (Figura 15.1) existeixen els botons *Save* i *Load*. Si es desitja guardar les dades registrades fins al moment, es podrà prémer el botó *Save* i s'haurà de seleccionar la carpeta a on es vol guardar el fitxer. En cas de voler carregar un fitxer guardat, s'haurà de prémer el botó *Load* i seleccionar un fitxer amb extensió JSON i amb el format de dades correcte.

Accedir a horari Un cop s'han entrat totes les dades d'un esdeveniment, s'haurà d'accedir a la vista d'horari prement el botó *Next* en la vista principal (Figura 15.1). Si totes les dades són correctes es podrà continuar.

En cas que alguna de les reunions contingui una entitat com a participant que no es trobi registrada al llistat, es notificarà a l'usuari i no s'avançarà.

15.2.2 Gestionar Horari

Un cop s'ha accedit a aquesta vista, es podran fixar les reunions desitjades i computar el resultat final. Aquesta vista (Figura 15.11) porta incorporat un buscador situat a la part superior esquerra. En introduir el nom d'una entitat i prémer el boto amb icona de lupa, es ressaltarà les reunions que tenen com a participant l'entitat.

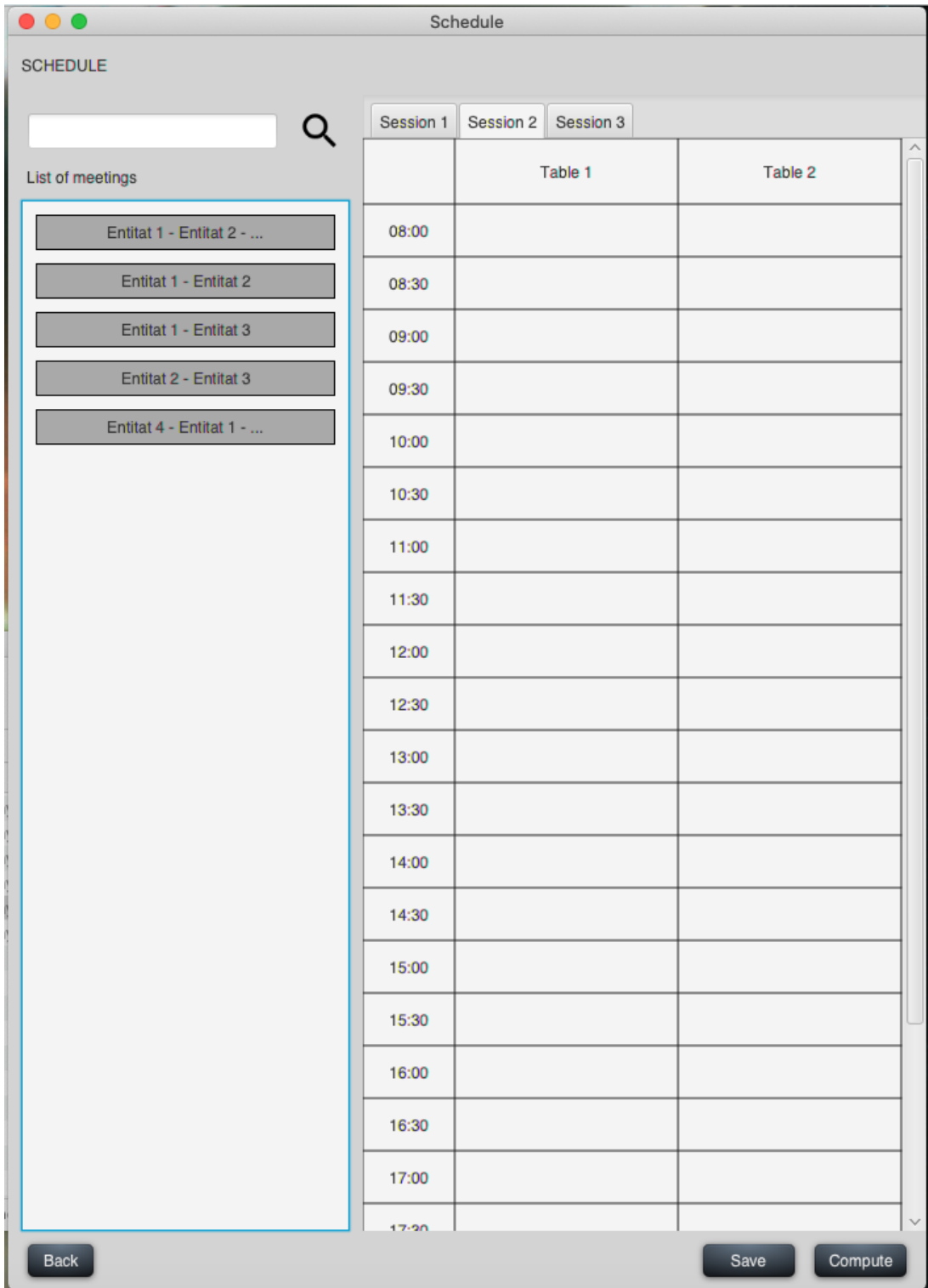


Figura 15.11: Vista horari.

Fixar reunió Per a poder fixar una reunió en una sessió, taula i *time slot*, es pot realitzar una acció *drag-and-drop* del requadre de la reunió desitjada sobre la posició final. Si alguna de les entitats de la reunió no està disponible, o bé no té prou assistents perquè en aquella franja horària ja té altres reunions, es notificarà a l'usuari i es tornarà el requadre al llistat.

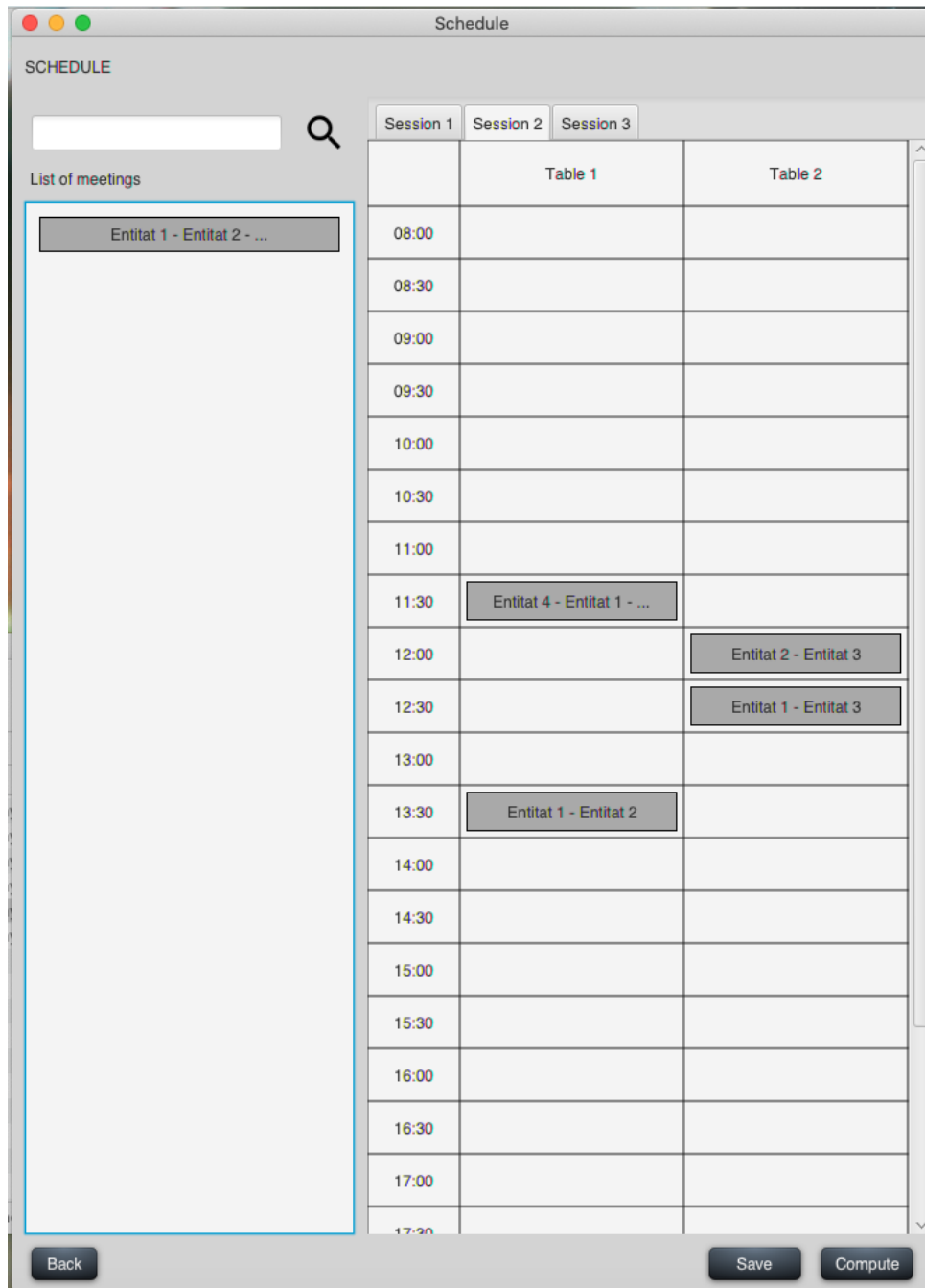


Figura 15.12: Fixat d'una reunió.

Computar solució Finalment es pot computar una solució prement el botó *Compute*. Si es troba solució, es col·locaran totes les reunions restants a la posició que se'ls ha atorgat.

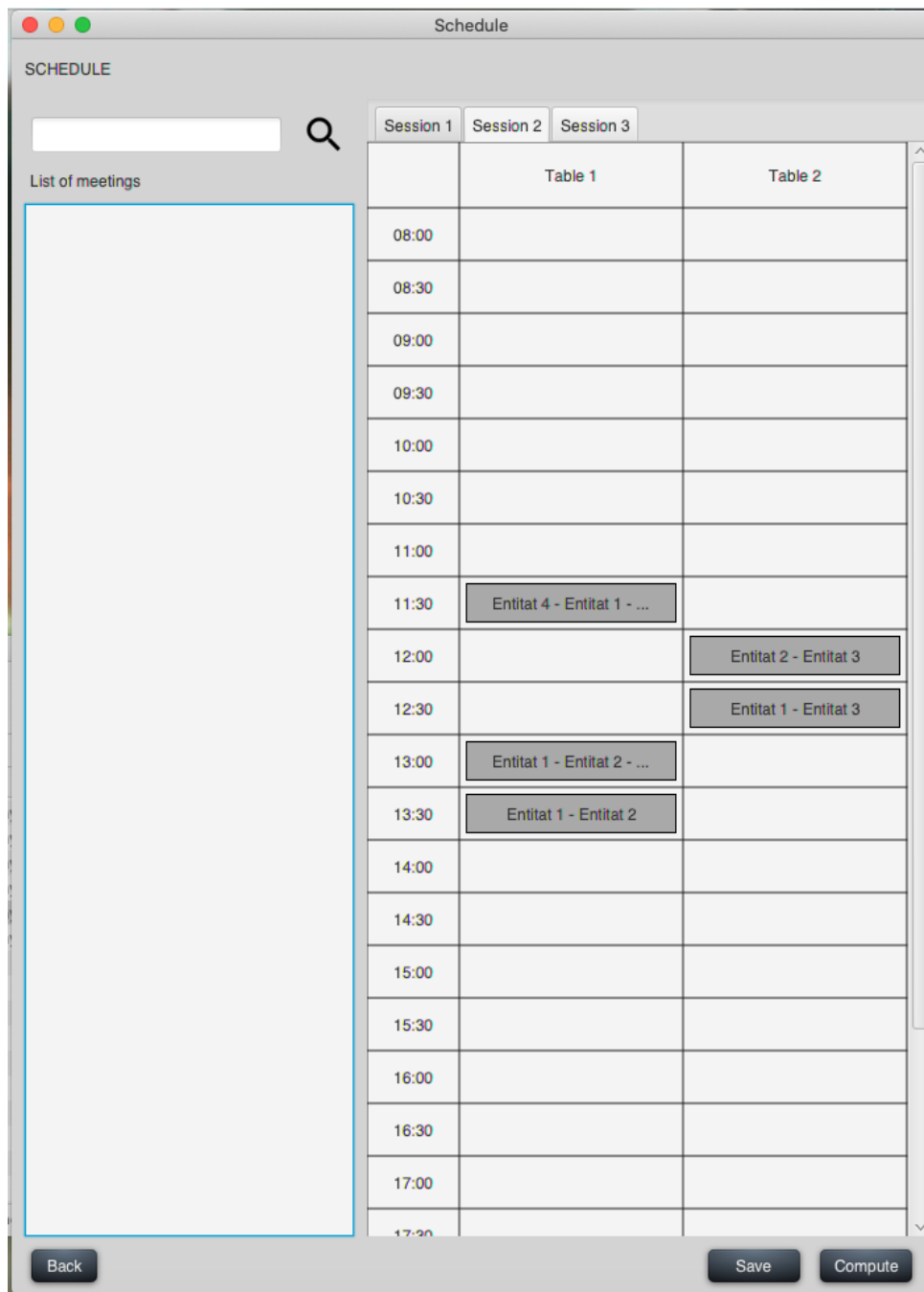


Figura 15.13: Fixat d'una reunió.

Guardar horari Si s'han prefixat reunions o s'ha computat una solució, es pot guardar el resultat obtingut en un fitxer prement el botó *Save*.
Si es desitja tornar a l'apartat d'entrada de dades, es pot prémer el botó *Back*. Es preguntarà a l'usuari si vol guardar les reunions col·locades de forma temporal.