

Assignador de docents per planificar un curs ampliant FET

Pere Ramon Erro Mas
Master Enginyeria en Informàtica

Samir Kanaan Izquierdo

23 de desembre de 2015



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Assignador de docents per planificar un curs ampliant FET</i>
Nom de l'autor:	<i>Pere Ramon Erro Mas</i>
Nom del consultor:	<i>Samir Kanaan Izquierdo</i>
Data de lliurament (mm/aaaa):	<i>12/2015</i>
Àrea del Treball Final:	<i>Intel·ligència Artificial</i>
Titulació:	<i>Master Enginyeria en Informàtica</i>
Resum del Treball (màxim 250 paraules):	
<p>Es tracta del desenvolupament de mòduls sobre base de codi existent aplicable a la plataforma de software d'empresa de codi obert Odoo per aconseguir una millor eina de gestió de centre docent amb l'objectiu d'automatitzar la presa de decisions amb uns paràmetres d'ús simples.</p> <p>En concret, el treball es centra en l'assignació de professorat a les classes que es preveuen impartir, considerant classe com a matèria que ha de rebre un grup.</p> <p>Els paràmetres que es tenen en compte són les hores mínimes i màximes que cada professor pot impartir, les hores mínimes i màximes que cada professor pot impartir de cada matèria i quin nivell té el professor per impartir cada matèria.</p> <p>Aquest és un problema complex i en el treball em serveixo dels coneixements adquirits d'intel·ligència artificial respecte a l'agrupació i l'optimització de resultats.</p> <p>Així mateix, el procés d'assignació de professors forma part tant del procés de configuració d'un curs docent com de la modificació d'un ja existent. Per aquest motiu, i com a fase de la gestió que usa la intel·ligència artificial, s'ha desenvolupat el lligam amb l'eina de generació d'horaris de codi obert FET, aconseguint una automatització aplicable a casos reals.</p>	

Abstract (in English, 250 words or less):

It is the development of modules on existing code base applicable to the software platform open source company Odoo for better management tool educational center with the aim of automating decision making with parameters Simple use.

In particular, the work focuses on the assignment of teachers to teach classes that are anticipated, considering class as a subject that has received a group.

The parameters taken into account are the minimum and maximum hours that each teacher can teach, minimum and maximum hours that each teacher can teach each subject and what level the teacher has to teach each subject.

This is a complex problem and work use acquired knowledge about artificial intelligence to optimize results.

Also, the process of assigning teachers thus part of the process of setting a course teaching how to modify an existing one. For this reason, and as stage management that uses artificial intelligence has developed links with the generation tool schedules open source FET, making automation applied to real cases.

Paraules clau (entre 4 i 8):

timetabling, teachers, assignation, odoo, fet

Índex

1	Introducció.....	1
1.1	Context i justificació del Treball.....	1
1.1.1	Aproximació al problema1	
1.1.2	Resum del problema1	
1.1.3	Particularitats del tipus de programari1	
1.1.4	Productes actuals2	
1.1.5	Producte base FET2	
1.1.6	Plataforma per la interfície d'usuari Odoo3	
1.1.7	Aportació en format codi obert4	
1.2	Objectius del Treball.....	4
1.2.1	Assignació de docents a les activitats4	
1.2.2	Incorporar la planificació en un programari d'empresa4	
1.2.3	Establir un esgraó més en el programari obert en aquest àmbit4	
1.3	Enfocament i mètode seguit.....	5
1.3.1	Utilització de mòduls Odoo propis5	
1.3.2	2a entrega5	
1.3.3	3a entrega5	
1.3.4	Entrega parcial5	
1.3.5	Darrera entrega6	
1.4	Planificació del Treball.....	6
1.4.1	Entrega 1 - Setembre6	
1.4.2	Entrega 2 - Octubre6	
1.4.3	Entrega 3 - Novembre6	
1.4.4	Entrega 4 - Desembre (abans del 23)6	
1.4.5	Llista de Tasques7	
1.5	Breu sumari de productes obtinguts.....	7
1.5.1	Mòduls School per al programari d'empresa7	
1.6	Breu descripció dels altres capítols de la memòria.....	7
1.6.1	L'algoritme en cru7	
1.6.2	Optimitzacions7	
1.6.3	Glossari, bibliografia, annexes7	
2	Producte resultant.....	8
2.1	Mòduls ERP base per arribar al punt origen del problema.....	8
2.1.1	Timed Groups8	
2.1.2	School Base8	
2.1.3	School Classe8	
2.1.4	School IWL (Impartition Week Line)8	
2.1.5	School Teacher Suitability9	
2.1.6	School Teacher Assignment9	
2.1.7	School FET9	
2.1.8	School Create Test Data9	
2.1.9	School Export Test Data9	
2.1.10	School Teacher Assignment and FET10	
2.2	School Teacher Assignment.....	10
2.3	School FET.....	11
2.3.1	Estructures persistents11	
2.3.2	Exportació de fitxer FET11	
2.3.3	Importació del fitxer FET12	
3	L'algoritme d'assignació de professors a classes.....	13
3.1	Generar estructures de dades:.....	13
3.2	Agrupar cursos i professors:.....	13
3.3	Assignar cursos : assigna_cursos.....	13
3.4	Algoritme parcial assigna_classes.....	14
4	L'optimització de l'algoritme.....	16

4.1	Introducció de % de cerca en l'espai de domini.....	16
4.2	Incorporació del multiprocessament.....	16
4.3	Compilació del codi Python.....	16
4.4	Reducció de l'espai de cerca.....	17
4.5	Estudi del paràmetre afinament.....	18
4.6	Millors temps quan es busquen menys.....	18
4.7	Incorporació de les solucions amb millors paràmetres.....	18
4.8	Sobra temps, aprofundim en l'arbre.....	19
5	Conclusions.....	20
5.1	Lliçons apreses.....	20
5.2	Anàlisi d'assoliment d'objectius.....	20
5.3	Observacions sobre la planificació.....	20
5.4	Línies de treball futures.....	21
5.4.1	Mantenir i millorar la documentació sobre els treballs en aquesta àrea	21
5.4.2	Millorar la base matemàtica del treball	21
5.4.3	Avançar en la integració dels algorismes del FET	21
5.4.4	Incorporar restriccions avançades en el procés automàtic d'assignació de professors i generació d'horaris.	21
6	Glossari.....	22
7	Bibliografia.....	24
8	Annexos.....	25
8.1	Comparativa de Software de Generador d'Horaris.....	25
8.2	Instruccions d'instal·lació Odoo sobre Linux Ubuntu 14.04.....	25
8.2.1	Introduction	25
8.2.2	The How To	26
8.2.3	Step 1. Build your server	26
8.2.4	Step 2. Create the Odoo user that will own and run the application	27
8.2.5	Step 3. Install and configure the database server, PostgreSQL	27
8.2.6	Step 4. Install the necessary Python libraries for the server	27
8.2.7	Step 5. Install the Odoo server	28
8.2.8	Step 6. Configuring the OpenERP application	28
8.2.9	Step 7. Installing the boot script	29
8.2.10	Step 8. Testing the server	29
8.2.11	Step 9. Automating Odoo startup and shutdown	30
8.3	Instal·lació del producte resultant, els mòduls school.....	31

Llista de figures

Il·lustracions

Il·lustració 1: Restriccions incloses en FET.....	3
Il·lustració 2: Gràfic Gantt de la planificació.....	7
Il·lustració 3: Mostra d'aplicació d'una solució trobada.....	10
Il·lustració 4: Pantalla de l'assistent d'assignació de professors.....	11
Il·lustració 5: FET utilitza arxius XML com a entrada i sortida.....	11
Il·lustració 6: Assistent d'importació del resultat de la generació del FET.....	12
Il·lustració 7: Incorporació multiprocessament.....	15
Il·lustració 8: Kernel compilat o no.....	16
Il·lustració 9: Optimització dels primers fills.....	16
Il·lustració 10: Concepte afinament.....	17
Il·lustració 11: Comprovació empírica menys solucions millors temps.....	17
Il·lustració 12: Pas de gegant en l'optimització.....	18
Il·lustració 13: Darrera optimització: aprofitant el temps sobrant.....	18

1 Introducció

1.1 Context i justificació del Treball

1.1.1 Aproximació al problema

Ens trobem amb un problema complex que es planteja normalment en col·legis i instituts, però que també seria equivalent en acadèmies o universitats.

La planificació és important. Concretant una mica més, estariem parlant de com compartir uns recursos en el temps per aconseguir realitzar el màxim d'activitats amb ells. D'entrada sembla una qüestió tant simple com seure en una taula i començar a assignar hores setmanals a les diferents activitats docents. En els casos de certa envergadura, passa però, que aviat ens trobem amb mancances d'espais i dubtes sobre si el que s'ha planificat era òptim. El problema s'embolica perquè a més de la pròpia limitació de recursos, tenim una sèrie de restriccions que poden arribar a ser simples com la no ubiqüitat dels recursos: un espai no pot estar ocupat per dues activitats diferents o una persona no pot estar a dos llocs diferents, igual que tampoc un equipament. Però també poden arribar a ser desconcertants com que no es pot fer activitats en una aula els mesos d'hivern perquè la calefacció falla i a primera hora encara és freda.

1.1.2 Resum del problema

Tenim: alumnes, activitats preparades per a ser gaudides per alumnes en grups, professors/monitors que poden dirigir una activitat, i espais o aules que encabeixen aquestes activitats.

Resumint la problemàtica, els alumnes poden tenir un horari individual i necessiten fer unes activitats específiques per cadascun d'ells, les activitats necessiten un espai determinat i poden tenir requeriments concrets, el professors també tenen el seu horari i unes matèries en les que són experts, i els espais tenen unes limitacions de capacitat.

1.1.3 Particularitats del tipus de programari

El perfil d'usuari d'aquest programari és un professional quina funció habitual no és la de submergir-se en la problemàtica de la planificació de recursos.

Així doncs, una gran part dels esforços dels desenvolupadors ha estat oferir una interacció amb l'usuari entenedora. Precisament aquesta vessant no és la que es cuida més en el codi lliure ja que sempre arriba un punt en que els experts en un software necessiten ser necessitats.

Però, en aquest cas és més important l'activitat empresarial de suport a l'usuari que requereix d'una mínima formació i consells en la programació del planificador.

Com una part molt important del suport, apareix la necessitat de importar/exportar dades per encaixar la programació i els resultats de la planificació en els processos de comunicació en el començament de les activitats i canvis sobrevinguts.

En definitiva, una clau important de l'èxit del desplegament d'aquest tipus de software consisteix, no tant en la qualitat de la solució, com en

l'embolcallat de l'usuari de forma que se li retiri qualsevol ombra a l'hora d'aconseguir el producte resultant, una planificació clara d'allò que inicialment semblava un guirigall.

1.1.4 Productes actuals

Codi propietari

- Peñalara - <https://www.penalara.com/queesghc.php>
Producte més complert. Veterania i resultats. Molt bon servei. De pagament. Actiu respecte col·laboracions amb altres empreses en el sector té connectors amb software de gestió corporatiu arreu.
- Kronowin - <http://www.adossis.com/>
Bon producte amb molts clients. La interfície s'ha quedat obsoleta usant llibreries antigues de Windows en contra de les interfícies web, i per això tindria menys facilitats d'ús.
- Untis - http://www2.grupet.at/home_es.php
D'origen universitari i molt utilitzat a varis països. De pagament.
- AscTimeTables - <http://www.asctimetables.com/#!/home/info>
Propietari. Té més centres anunciats. Consta de un bon tutorial però amb mancances de funcionalitats per centres complexos.
- Best Timetabling - <http://www.best-timetabling.com/es/generador-de-horarios/>
Propietari. Forma part de la suite de la marca Best.

Codi obert

- FET - <http://larescu.ro/liviu/fet/>
Codi obert amb llicència GNU. Interfície pobre respecte solucions de pagament. No complert del tot en mancar l'assignació de professors. Problema de connectivitat amb software de gestió existent.
- Tablix/Gtablix - <https://www.tablix.org/articles/about/>
Codi obert sense manteniment des de 2009 .
- UniTime - <http://www.unitime.org/>
Codi obert. Iniciat per separat per universitats nord americanes i europees. Té unes presentacions de resultat molt avançades.

En l'annexe 8.1 hi ha una taula amb una Comparativa de Software de Generador d'Horaris .

1.1.5 Producte base FET

Com hem introduït abans, FET és una aplicació d'escriptori que genera horaris sobre unes classes amb uns professors assignats adjudicant una hora de les determinades a la setmana i una aula. FET és un programari de codi obert i lliure amb llicència pública general Affero GNU Affero. Incorpora moltes restriccions per a programar la cerca de solucions . Pot treballar amb un algoritme de generació automàtica complerta però també amb una planificació semi automàtica o manual.

Com a entrada de dades podem utilitzar un arxiu XML modular però que és editable amb la mateixa interfície FET. D'altra banda també podem importar dades en format CSV.

És un projecte que Liviu Lalescu va començar a l'octubre de 2002 fent servir un algoritme genètic. En el juny del 2007 es va descobrir l'algoritme heurístic (intercanvi recursiu) i amb les aportacions de Volker Dirr, es va incorporar i encara corre ara, més ràpid i més potent per resoldre horaris difícils. De fet, simula un probable procediment manual per trobar una solució i trobem una descripció de l'algoritme aquí : [1]

- *Time constraints (misc)*
 - *Basic compulsory time constraints*
 - *Break (all teachers+all students not available)*
- *Time constraints (teachers)*
 - *A teacher is not available*
 - *Max days per week for a teacher*
 - *Max gaps per week for a teacher*
 - *Max gaps per week for all teachers*
 - *Max gaps per day for a teacher*
 - *Max gaps per day for all teachers*
 - *Max hours daily for a teacher*
 - *Max hours daily for all teachers*
 - *Min hours daily for a teacher*
 - *Min hours daily for all teachers*
 - *Max hours continuously for a teacher*
 - *Max hours continuously for all teachers*
 - *A teacher works in an hourly interval max days per week*
 - *All teachers works in an hourly interval max days per week*
- *Time constraints (students)*
 - *A students set is not available*
 - *Max gaps per week for a students set*
 - *Max gaps per week for all students*
 - *A students set begins early*
 - *All students begin early*
 - *Max hours daily for a students set*
 - *Max hours daily for all students*
 - *Min hours daily for a students set*
 - *Min hours daily for all students*
 - *Max hours continuously for a students set*
 - *Max hours continuously for all students*
- *Time constraints (activities)*
 - *An activity has a preferred starting time*
 - *An activity has a set of preferred time slots*
 - *A set of activities has a set of preferred time slots*
 - *A set of subactivities has a set of preferred time slots*
 - *An activity has a set of preferred starting times*
 - *A set of activities has a set of preferred starting times*
 - *A set of subactivities has a set of preferred starting times*
 - *Min n days between a set of activities*
 - *An activity ends students day*
 - *A set of activities ends students day*
 - *A set of activities has same starting time (day+hour)*
 - *A set of activities has same starting day (any hours)*
 - *A set of activities has same starting hour (any days)*
 - *2 activities are ordered*
 - *2 activities are consecutive*
 - *A set of activities are not overlapping*
 - *Min gaps (hours) between a set of activities*

FEATURES

- *Space constraints (misc)*
 - *Basic compulsory space constraints*
- *Space constraints (rooms)*
 - *A room is not available*
- *Space constraints (teachers)*
 - *A teacher has a home room*
 - *A teacher has a set of home rooms*
 - *Max building changes per day for a teacher*
 - *Max building changes per day for all teachers*
 - *Max building changes per week for a teacher*
 - *Max building changes per week for all teachers*
 - *Min gaps between building changes for a teacher*
 - *Min gaps between building changes for all teachers*
- *Space constraints (students)*
 - *A set of students has a home room*
 - *A set of students has a set of home rooms*
 - *Max building changes per day for a set of students*
 - *Max building changes per day for all students*
 - *Max building changes per week for a set of students*
 - *Max building changes per week for all students*
 - *Min gaps between building changes for a set of students*
 - *Min gaps between building changes for all students*
- *Space constraints (subjects)*
 - *A subject has a preferred room*
 - *A subject has a set of preferred rooms*
- *Space constraints (subjects and activity tags)*
 - *A subject + an activity tag have a preferred room*
 - *A subject + an activity tag have a set of preferred rooms*
- *Space constraints (activities)*
 - *An activity has a preferred room*
 - *An activity has a set of preferred rooms*

Il·lustració 1: Restriccions incloses en FET

En poques paraules, es defineix l'entorn amb la introducció d'aules, professors, grups i caselles de temps de l'horari. A partir d'aquí, es programen restriccions relacionades per cada un dels elements definits.

La sortida de resultats es pot obtenir en formats HTML, XML i CSV. A aquesta característica, afegida a l'entrada de dades per arxiu i la possibilitat d'execució per línia de comandes, possibilita l'automatització de passar les solucions d'assignació de professors pel programari de generació d'horaris.

1.1.6 Plataforma per la interfície d'usuari Odoo

Odoo és un programari de codi obert considerat com a RAD en l'entorn dels ERP . Està escrit en Python i té un recorregut d'aportacions de la comunitat des del 2005 amb l'incipient TinyERP creat per Fabien Pinckaers.

La utilitat és triple:

1. Empaquetar tot el codi en uns mòduls específics de forma que quedi ben endreçat.
2. Facilitat de documentació (Python ho facilita)
3. Una introducció de dades i la visualització de resultats directa en aquesta plataforma.

1.1.7 Aportació en format codi obert

El programari amb codi obert té molts punts forts respecte el privatiu. A destacar una gran escalabilitat per l'evolució del mateix, una garantia de resolució d'errors poc després de fer-se a la llum i amb conseqüències més controlades, i més resistent a les variacions del mercat. Si bé hi ha punts febles, com que les arrancades dels projectes són més lentes i, sobretot, la garantia del manteniment, una inversió en el programari de codi obert garanteix també que el programari privatiu millori i evolucioni, sense la sempre desitjable competència.

Aquest treball intenta ser una empenta per aquest software intel·ligent. En concret, el treball pretén completar el software amb codi obert més evolucionat, FET, per aconseguir una major aproximació a l'usuari tipus que demana menys complexitat de tot el procés.

1.2 Objectius del Treball

En general, es tracta d'aconseguir un «Planificador de Curs» per a un centre docent que amplii la funcionalitat de FET per construir horaris aplicant l'assignació de docents a una classe (grup-matèria) determinada per un pla docent, on els professors disposen de preferències d'ordre de tipus de matèria i horari.

1.2.1 Assignació de docents a les activitats

Actualment FET, i altres generadors d'horaris amb codi lliure parteixen del punt que les classes (activitats amb un grup d'alumnes i una matèria determinada) es troben definides amb els professors ja assignats. Amb aquest treball intentaré automatitzar també aquesta part de manera que les desiderates es podran complir millor en poder deslligar professors i classes concretes. Així, una classe que és òptim que es faci dilluns tarda, només caldria canviar el professor que en la seva desiderata demana tenir dilluns tarda lliure.

1.2.2 Incorporar la planificació en un programari d'empresa

FET i altres generadors d'horaris queden deslligats de la gestió del centre educatiu. Un segon objectiu es evitar aquest handicap. Per això utilitzaré el ERP de codi obert Odoo amb un gran recorregut, amplia biblioteca de llibreries, fàcilment modular i una gran i activa comunitat.

1.2.3 Establir un esgraó més en el programari obert en aquest àmbit

Aconseguir un binomi, programari i documentació, de suficient qualitat perquè el treball pugui ser reutilitzat de forma constructiva per altres iniciatives.

1.3 Enfocament i mètode seguit

1.3.1 Utilització de mòduls Odoo propis

Es tracta d'una decisió difícil davant dels dos braços de la balança: aprofitar codi i estructures existents, menys feina i resultats més estàndards, o bé, menys possibilitats d'ampliar funcionalitats que amb una reutilització quedarien limitades. En aquest cas, dels mòduls analitzats ([school de Serpent Consulting Services Pvt. Ltd](#), [Open Academy de Odoo S.A.](#) per a la versió 6.1) l'aprofitament dels mòduls existents no aconsegueix compensar, ni de bon tros, les complexitats i pèrdua de consistència a l'hora de desenvolupar els assistents d'assignació i planificació automàtiques.

1.3.2 2ª entrega

Durant el més d'octubre la meua tasca ha estat de convertir-me en un expert en la resolució d'aquest tipus de problemes analitzant les solucions que s'hagin publicat.

Preveig tres estratègies per aconseguir els objectius:

- Solució interna: Modificar el codi del FET per incloure la recerca de la millor assignació del professor.
- Solució mixta: Fer trampes incorporant l'assignació del professor com la tria d'un dels conjunts variables (grup-professor, aula professor)
- Solució externa: Desenvolupar una nova aplicació pont que usi el FET de forma totalment externa.

Per arribar a un coneixement per poder prendre una decisió m'han anat bé els nombrosos documents existents, com la Tesi doctoral de Tomás Müller [2], incorporada com a documentació extra en la WEB de UniTime .

1.3.3 3ª entrega

El temps dedicat fins a aquesta 3ª entrega l'he dedicat a preparar la plataforma i a estudiar quina de les 3 solucions convé posar en marxa. Un cop assolida l'estratègia de la solució externa amb codi propi d'assignació de professors, orientaré els treballs cap a la solució mixta, és a dir, aprofitar FET i els seus algoritmes per a aconseguir millors solucions: més ràpides i amb millor valoració.

1.3.4 Entrega parcial

De cara a fer una presentació més entenedora calia incorporar dades d'exemple en la mateixa plataforma. Per això he dedicat temps a generar dades de demostració. Per aconseguir-ho he incorporat retocs de tota mena:

Canviar el model general: nova relació entre matèria -course- i un curs -agrupació de matèries -offer-)

Millorar del codi: millor llegibilitat i qualitat d'algoritmes

Incorporació de nous assistents: generació de classes, línies horaris i sessions docents

Generació d'arxius XML per l'entorn de demostració que facilita la plataforma.

També he redactat una explicació condensada de l'algoritme d'assignació de professors a les classes, que complementa la pròpia del codi.

1.3.5 Darrera entrega

El treball de generar els resultats ha evidenciat mancances en l'algoritme per aconseguir suficients solucions. Un nombre important de solucions són necessaris per assegurar qualitat en el càlcul del generador d'horaris com a pas següent.

L'estudi i aplicació de les optimitzacions ha estat possible amb la parametrització de l'algoritme amb el concepte d'afinament.

El desenvolupament d'aquestes optimitzacions ha donat lloc a tot un capítol en aquesta memòria.

1.4 Planificació del Treball

L'eina utilitzada per la planificació del treball ha estat l'aplicació Planner. Costa una mica introduir-se però en ser simple de seguida s'aprofiten totes les funcionalitats.

Les fites temporals que es proposen tenen aquesta inicial traducció en el projecte concret:

1.4.1 Entrega 1 - Setembre

Objectius, estudi de la base sobre la que es treballarà (productes en el mercat i eines a utilitzar) i planificació

1.4.2 Entrega 2 - Octubre

Aprofundiment del coneixement del FET amb un resum de les tècniques que utilitza. Orientació de la solució i de la seva integració amb les dues eines existents FET i Odoó (modificació interna i/o desenvolupament per separat).

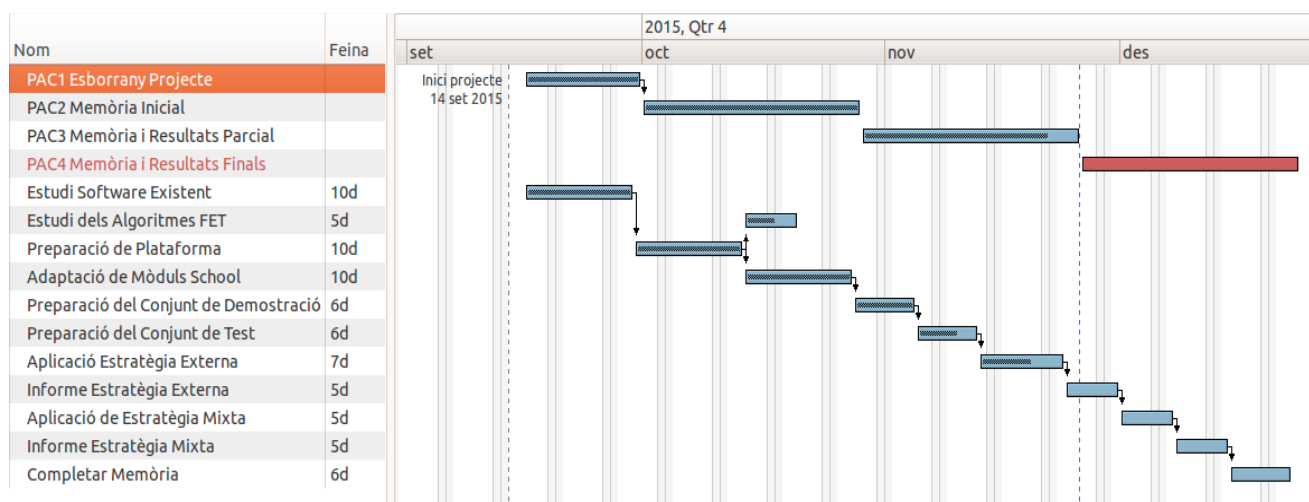
Preparació del conjunt de dades de test.

1.4.3 Entrega 3 - Novembre

Actualització de la memòria amb la clara finalitat de comprovar si es poden aconseguir tots els objectius plantejats, o convé replantejar la orientació del projecte. La idea és aprofitar els coneixements adquirits per reorientar si cal en tenir una millor visió d'aquest tipus de problemes complexos.

1.4.4 Entrega 4 - Desembre (abans del 23)

Desenvolupament màxim, preparació d'informes i comparatives, Preparació del treball,



Il·lustració 2: Gràfic Gantt de la planificació inicial

1.4.5 Llista de Tasques

De les tasques inicials, les dues dedicades a l'aplicació de l'estratègia mixta va haver de sacrificar-se per aprofundir en l'optimització de l'algoritme que inicialment no va donar els resultats objectius.

1. Estudi del software existent
2. Estudi dels Algoritmes FET
3. Preparació de la plataforma
4. Adaptació dels Mòduls School
5. Preparació del conjunt de demostració
6. Preparació del conjunt de test
7. Aplicació estratègia externa
8. Informe estratègia externa
9. Aplicació de l'estratègia mixta
10. Informe de l'estratègia mixta
11. Estudi i aplicació d'optimitzacions
12. Completar la memòria

1.5 Breu sumari de productes obtinguts

1.5.1 Mòduls School per al programari d'empresa

Els productes obtinguts són els desenvolupaments dels diferents mòduls Odoo que s'ha millorat respecte el codi existent o afegit i que s'expliquen amb més detall en l'apartat 2.1 d'aquesta memòria.

Cal dir que s'ha renovat tot el codi existent per adaptar-lo d'una versió anterior del programari OpenERP (v.6.1) a l'actual Odoo (v.8.0).

1.6 Breu descripció dels altres capítols de la memòria

1.6.1 L'algoritme en cru

En aquest apartat que omple el capítol 3, s'explica quin ha estat l'algoritme sobre el que s'ha treballat per la cerca de solucions del problema complex d'assignació de professors a les classes.

1.6.2 Optimitzacions

S'ha requerit afegir optimitzacions respecte l'algoritme de cerca exhaustiva aplicant tècniques basades agrupacions i en programació evolutiva.

El contingut d'aquest tema es troba en el capítol 4 que tracta sobre L'optimització de l'algoritme.

1.6.3 Glossari, bibliografia, annexes

Els capítols següents contenen aquests apartats que poden arribar a clarificar dubtes en la lectura o a ampliar coneixements que siguin d'interès.

2 Producte resultant

2.1 Mòduls ERP base per arribar al punt origen del problema

Són una adaptació de uns mòduls de versions anteriors del programari OpenERP que vaig iniciar jo mateix per encàrrec de l'empresa d'implantació de software obert Tecnoba de Sant Joan de Vilatorrada.

Aquest desenvolupament anava dirigit per a la gestió dels centres de secundària de Catalunya i, més concretament, per al control d'assistència.

El producte treballat en aquest TFM ha estat l'adaptació a la nova versió de la plataforma d'empresa, deixant, per motius d'orientació d'esforços, però també acadèmics de reduir matèria no objecte d'estudi, la resta de mòduls que no es dirigeixen a construir les estructures de dades necessàries per abordar el problema de donar solució a l'assignació de professors, i la següent de generar un horari. Tot i així, el codi del projecte té més de 7527 línies en Python i 3937 en codi de vistes XML.

El codi es troba allotjat en el lloc web de desenvolupament col·laboratiu de software GitHub <https://github.com/pereerro/schooly>.

El programari que modela l'entorn empresarial, es troba encapsulat en aquests mòduls que caracteritzaré breument:

2.1.1 Timed Groups

Aborda la gestió dels grups amb limitació temporal, dels subgrups, de les classificacions i de la inclusió automàtica de participants. Inclou gestió avançada com la de intersecció entre grups tot i respectant l'assignació temporal del temps coincident d'un participant en els dos grups.

2.1.2 School Base

Introdueix els conceptes base d'un centre educatiu com la matèria a impartir, el centre, el pla d'estudis (offer), la posada en pràctica d'un pla d'estudis, etc...

Com a conjunt de dades d'exemple, s'introdueixen 5 matèries i 6 pla d'estudis que incorporen cadascuna una versió d'aquestes matèries.

2.1.3 School Classe

Concreta més la gestió de la docència amb la classe i el curs docent, entre altres.

Completant el conjunt de dades d'exemple, es generen els cursos basats en els plans d'estudi corresponents i les classes necessàries, amb els seus subgrups corresponents.

2.1.4 School IWL (Impartition Week Line)

Es desenvolupa l'objecte principal necessari per la gestió dels recursos de forma periòdica, l'hora d'impartició. Possibilita la generació automàtica de les sessions docents tenint en compte la durada en setmanes dels cursos i la càrrega lectiva que s'hi ha previst. Això ho fa introduint l'objecte línia setmanal d'impartició de les classes.

La càrrega d'aquest mòdul en una base de dades marcada com de demostració executa, en aquest cas, aquesta generació automàtica de línies horàries que caldrà omplir amb professors.

Com a informació, en el codi per la versió anterior, en aquest mòdul s'hi han desenvolupat eines per fer desdoblaments i agrupacions desdobladades de diferents matèries. Aquesta pràctica permet reduir el nombre d'alumnes per sessió de certes matèries aprofitant recursos de professor extres i d'espai sobrants.

2.1.5 School Teacher Suitability

En aquest mòdul s'incorporen les estructures necessàries per portar la gestió de l'assignació de professors.

S'introdueix la dada d'hores mínimes i màximes de professor, les hores mínimes i màximes de professor per una matèria concreta, i la idoneïtat en percentatge de que un professors imparteixi una matèria.

2.1.6 School Teacher Assignment

S'introdueixen els assistents per obtenir solucions al problema d'assignació de professors a les classes.

En el punt 2.2 es presenta amb més detall.

2.1.7 School FET

Els objectes i assistents principals per una concreció de dia/hora i aula per les línies d'impartició.

Els dos darrer mòduls es documentaran amb més profunditat en els següents capítols.

2.1.8 School Create Test Data

Nou mòdul per generar professors i aules suficients per a les classes existents. També genera dades d'idoneïtat per a cadascun dels professors generats sobre les matèries de les classes sense professor.

La funció principal en aquest projecte ha estat la de completar les dades per poder efectuar els tests i les proves de perfeccionament de l'algoritme i de les optimitzacions.

Tant les hores màximes de professor com els marges que s'hi apliquen a aquestes (lack) o a les hores màximes de professor a un curs (lack2) es troben fixades en la preparació de l'assitent que apareix en el XML del mòdul:

```
<!--
Resource: school.create_test_data
-->
<record id="school_create_test_data_0" model="school.create_test_data">
  <field name="num_week_hours_per_teacher" eval="25" />
  <field name="lack" eval="5" />
  <field name="lack2" eval="5" />
</record>

<function model="school.create_test_data" name="create_test_data">
  <value eval="[ref('school_create_test_data_0'),]" />
</function>
```

2.1.9 School Export Test Data

Nou mòdul d'utilitat per fer persistent un estat determinat de la base de dades respecte les dades d'aquests mòdul.

2.1.10 School Teacher Assignment and FET

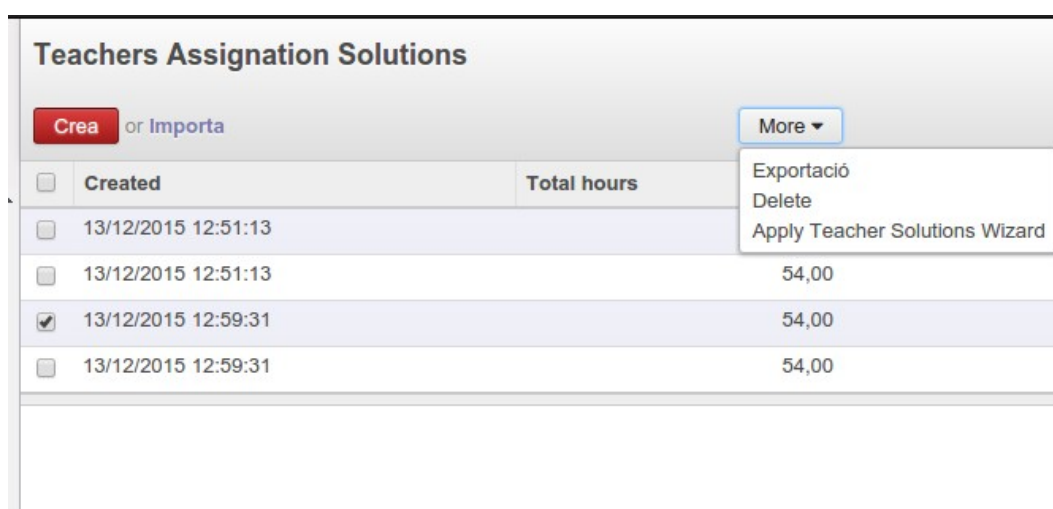
Darrer mòdul implementat que permet l'execució posterior de la generació d'horaris per a cada solució d'assignació de professors que s'ha aconseguit, i que s'ha marcat l'opció.

En aquest cas, l'aplicació de la solució de l'assignació també inclou l'aplicació de l'horari generat.

2.2 School Teacher Assignment

El desenvolupament dels models persistents situats a l'arrel de la carpeta té dos objectius:

1. Generar atributs calculats necessaris per l'algoritme de cerca de solucions d'assignació.
2. Introdueix els models que serviran com a base per mostrar les solucions.



<input type="checkbox"/>	Created	Total hours
<input type="checkbox"/>	13/12/2015 12:51:13	
<input type="checkbox"/>	13/12/2015 12:51:13	54,00
<input checked="" type="checkbox"/>	13/12/2015 12:59:31	54,00
<input type="checkbox"/>	13/12/2015 12:59:31	54,00

Il·lustració 3: Mostra d'aplicació d'una solució trobada

D'altra banda, l'assistent de cerca de solucions té dos mòduls situats en la carpeta wizard:

1. El que es basa en els objectes ORM propis de la plataforma.
2. El nucli de l'algoritme que es troba en l'arxiu `school_assign_teachers_kernel.py`. Aquest mòdul Python té una versió compilada amb Nuika que s'executa més ràpidament. També se li ha dotat d'execució directe des del terminal per aprofitar la codificació multiprocessador implementada com a resultat del treball d'optimització.

L'explicació de l'algoritme amb més detall es desenvolupa en el proper capítol 14.

Create Solutions

Max. solutions:

Time (min):

Max teachers hours: 60,00

Total classes hours: 54,00

Min teachers hours: 40,00

Course	Classes Hours	Max hours	Min hours
Matemàtiques 1r ESO	3,00	12,00	0,00
Llengua Catalana 1r ESO	3,00	12,00	0,00
Anglès 1r ESO	3,00	12,00	0,00
Matemàtiques 2n ESO	3,00	12,00	0,00
Llengua Catalana 2n ESO	3,00	12,00	0,00
Anglès 2n ESO	3,00	12,00	0,00
Matemàtiques 3r ESO	3,00	12,00	0,00
Llengua Catalana 3r ESO	3,00	12,00	0,00

Il·lustració 4: Pantalla de l'assistent d'assignació de professors

2.3 School FET

2.3.1 Estructures persistents

Aquest mòdul de programari estableix les estructures de dades que caldrà per fer la connectivitat amb el programa FET.

Amb el mòdul `school_tch_assig_and_fet` que s'explica en el punt 2.1.10 aquestes tasques queden automatitzades.

```
-<fet version="5.18.0">
  <Institution_Name>A Institution Name</Institution_Name>
  <Comments>Comentaris predeterminats</Comments>
  -<Hours_List>
    <Number>7</Number>
    <Name>08:00-09:00</Name>
    <Name>09:00-10:00</Name>
    <Name>10:00-11:00</Name>
    <Name>11:30-12:30</Name>
    <Name>12:30-13:30</Name>
    <Name>15:30-16:30</Name>
    <Name>16:30-17:30</Name>
  </Hours_List>
  -<Days_List>
    <Number>5</Number>
    <Name>monday</Name>
    <Name>tuesday</Name>
```

Il·lustració 5: FET utilitza arxius XML com a entrada i sortida

2.3.2 Exportació de fitxer FET

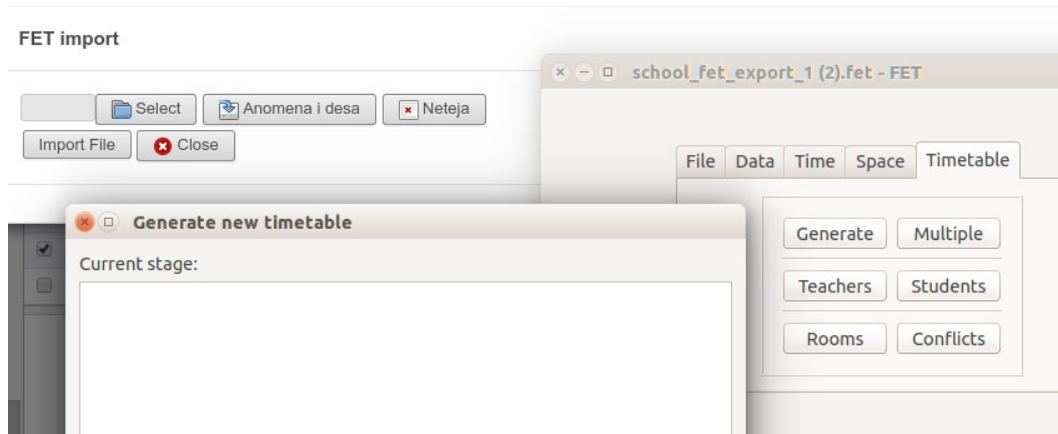
L'exportació del fitxer XML amb les activitats (classes), subactivitats (o línies horàries en Odoo), professors, aules, matèries, grups, i alumnes que després caldrà carregar en el programa extern d'escriptori.

S'hi incorporen automàticament com a restriccions, les hores que les aules o els professors es troben ocupats amb altres línies d'impartició no objectes de la cerca de hores i aules.

Aquest assistent es farà córrer de forma programada en les solucions d'assignació de professors que estiguin marcades per fer aquest següent pas.

2.3.3 Importació del fitxer FET

La importació del fitxer resultant de la planificació en FET, i que posa aules i horaris en les classes corresponents.



Il·lustració 6: Assistent d'importació del resultat de la generació del FET

3 L'algoritme d'assignació de professors a classes

Aquest algoritme té una base de cerca per restriccions. L'he ideat jo mateix. Ja amb el repte d'aquest treball, tenia l'esperança que se'm revelés un mètode d'optimització millor. Sense massa anàlisis matemàtic he intuït que es barrejaven diferents plans de dades en l'espai de solucions (assignació de classes i idoneïtat de professor per la matèria de la classe, a més de limitació temporal en funció de les classes), quines solucions en cada pla no podien fusionar-se sense augmentar la complexitat.

L'algoritme té varies fases:

- La primera tracta de preparar l'estructura de dades.
- La segona agrupa cursos i professors per poder fer el còmput d'assignacions a les classes corresponents sense que hi hagi interferències d'hores mínimes i màximes del professor si entra en dos còmputs diferents.
- Agafada l'agrupació individual, plantejem l'assignació de les classes de cadascun dels cursos per als professors que componen l'agrupació.
- Ho fem de manera recursiva a la vegada que actualitzem les hores assignades en l'estructura de dades que passem en la recursivitat.
- L'assignació de classes la establim permutant la llista ordenada per assegurar resultat i valor, i reduïda per guanyar velocitat, dels professors i aplicant-la a les classes.

A continuació explico l'algoritme amb mes detall:

3.1 Generar estructures de dades:

- **get_teachers_hours_blocked** : Hores bloquejades de professors, per cada professor i per professor i curs
- **get_classes_data_by_course** : Total hores de classe i dades de classes per curs. Dades son la llista de classes, la duració total de les classes i la duració i les hores setmanals per classe.
- **get_teachers_data_by_course** : Dades de professors per curs: total hores de professors màximes i mínimes, total hores per professor màxim i mínim, total hores per professor i curs màxim i mínim.

Es té en compte les hores bloquejades de professors, per cada professor i per professor i curs.

3.2 Agrupar cursos i professors:

- **agrupa_cursos_i_professors**

És habitual que els professors s'especialitzin en unes matèries determinades d'unes poques àrees o seminaris.

3.3 Assignar cursos : assigna_cursos

S'executa aquest procediment per a cada agrupació

Entrada:

- llista de cursos
- total d'hores de classes per assignar

- dades de classe per curs
- total hores de tots els professors màxim i mínim
- per a cada professor màxim i mínim a assignar
- per a cada professor i curs, màxim i mínim a assignar
- número màxim de solucions
- temps límit

Sortida:

Llista de solucions on cada solució consta de una parella amb el valor global de la solució calculada per les idoneïtats de professor i curs, i un diccionari on cada curs té una llista de professors per cada classe que s'ha passat com a tercer paràmetre.

Algoritme:

1. Fer un recorregut per tots els cursos de la llista que rep com a primer paràmetre. Per cadascun d'ells fa:
 1. Assigna les classes (**assigna_classes**) d'aquest curs als professors que tenen certa idoneïtat per aquest curs
 2. Per a cada solució d'assignació de classes per aquest curs obtinguda s'actualitza les dades d'hores de professor màximes i mínimes totals i per curs, treu el curs tractat de la llista, i torna a executar aquest algoritme recursivament amb la llista reduïda excepte quan la reducció no deixa cursos
 3. A les solucions d'assignació de cursos rebudes, incorpora la del curs actual i es queda amb les millors assignacions de cursos i classes relacionades.
2. Retorna les assignacions de cursos i classes relacionades

3.4 Algoritme parcial **assigna_classes**

Entrada:

- llista de classes
- per a cada professor màxim i mínim a assignar
- per a cada professor i curs, màxim i mínim a assignar
- hores totals que queden per assignar en aquest curs
- hores totals que queden per assignar en tots els cursos
- hores mínimes totals a omplir en aquest curs
- hores mínimes totals a omplir per a tots els cursos
- número màxim de solucions
- cursor que apunta a la classe a assignar
- coeficient per saber quants teachers s'agafarà de la llista ordenada per assignar sobre el número de classes
- temps límit

Sortida:

Llista de parelles amb el valor de l'assignació sumant les idoneïtats professor-curs aconseguides i la llista de professors assignats a les corresponents classes de la llista passada

Algoritme:

1. Ordena la llista de dades de professors per al curs de les classes que volem assignar tenint en compte per ordre d'importància: mínim de temps per al curs, mínim de temps total, percentatge d'idoneïtat, temps disponible per al curs, temps disponible total.

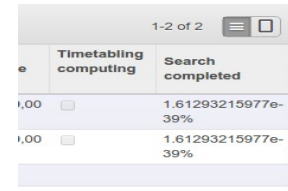
2. Agafem una quantitat de professors suficient (multiplicat pel paràmetre afinament) per assignar el total d'hores de les classes
3. Anem provant els professors de la llista.
 1. Comprovem que es compleixin les condicions d'hores màximes
 2. Amb la duració de la classe apuntada, s'actualitzen les dades d'hores a assignar i les mínimes disponibles de professor.
 3. S'augmenta el cursor de la següent classe i es fa recursió
 4. S'agafen les llistes de professors retornades i s'incorporen com a solucions si la valoració es millor
4. Retorna les assignacions de classes per al curs

4 L'optimització de l'algoritme

4.1 Introducció de % de cerca en l'espai de domini

Un cop preparats els jocs de proves, s'executen les primeres. Ja d'un inici es denota que s'esgota el temps límit amb molta diferència entre una execució amb una sola petició de solució i les que tenen vàries.

Per poder analitzar quin és el comportament de l'algoritme s'introdueix en una variable que va acumulant a tots els nivells percentualment, el recorregut que s'ha realitzat fins a complir el temps límit.



Timetabling computing	Search completed
1,00	1.61293215977e-39%
1,00	1.61293215977e-39%

4.2 Incorporació del multiprocessing

Tot i no formar part de l'optimització de l'algoritme, no es podia descartar una millora de les solucions aconseguint més recorregut de cerca en el domini de solucions amb el mateix temps objectiu.

Així, vaig modificar la codificació de l'algoritme per incorporar multiprocessing i, d'aquesta manera aprofitar els recursos de màquina que solen disposar el hardware actual.

Per això he usat, només en l'arxiu Python `school_assign_teachers_kernel.py` que conté el nucli del processament, el mòdul Python multiprocessing [3] .

Creating Solutions

Crea

 or [Importa](#)

<input type="checkbox"/>	State	Waiting since	Initiated	Seconds computing	Num Max Solutions	Time to compute	Num processors	Afina	Max value
<input type="checkbox"/>	Done	21/12/2015 19:09:44	21/12/2015 19:09:46	35	30	300	1	7,00	8949,00
<input type="checkbox"/>	Done	21/12/2015 19:12:16	21/12/2015 19:12:18	21	30	300	2	7,00	8949,00

Il·lustració 7: Incorporació multiprocessing

Com era d'esperar, els temps de processament es redueixen en el mateix ordre de divisió de la feina entre processadors: 2 processadors meitat de temps, etc...

4.3 Compilació del codi Python

Sense haver aconseguit els resultats esperats fent treballar tots els processadors possibles de la màquina, vaig abordar la compilació del codi Python sense haver d'interpretar-se el codi intermedi dels arxius pyc .

No esperava resultats molt millors, però era una oportunitat per comprovar la solució de compilació de codi Python.

Executed so... / Executed solutions									
Crea or Importa									
<input type="checkbox"/>	State	Waiting since	Initiated	Seconds computing	Num Max Solutions	Time to compute	Num processors	Afina	Max value
<input type="checkbox"/>	Done	21/12/2015 20:37:12	21/12/2015 20:37:15	15	30	300	1	7,00	8949,00
<input type="checkbox"/>	Done	21/12/2015 20:37:58	21/12/2015 20:38:29	8	30	300	2	7,00	8949,00
<input type="checkbox"/>	Done	21/12/2015 20:42:24	21/12/2015 20:42:28	17	30	300	1	7,00	8949,00
<input type="checkbox"/>	Done	21/12/2015 20:43:57	21/12/2015 20:43:59	8	30	300	2	7,00	8949,00

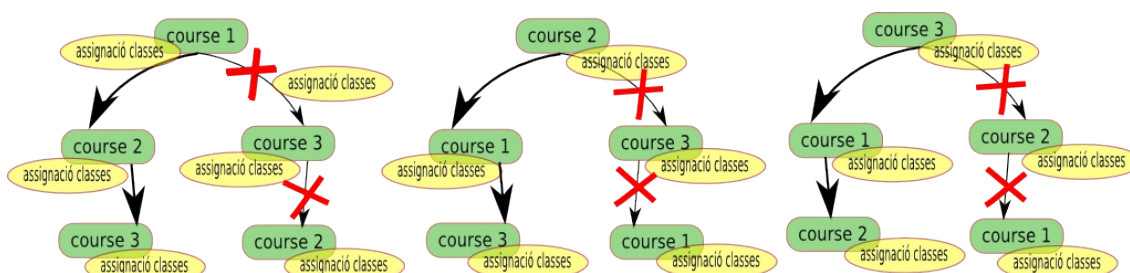
Il·lustració 8: Kernel compilat o no

Dels resultats de les execucions comprovem que només en el cas de l'execució amb un sol processador hi ha una mínima diferència del temps d'execució.

4.4 Reducció de l'espai de cerca

Si bé es cert que la implementació de l'algoritme inicial arribava ràpidament al cap de vall de l'arbre de cerca, perdia molt de temps en les branques inferiors sense tractar els diferents troncs. D'aquesta forma arribava la fi del temps sense tenir solucions completes i de valor raonablement bo.

Aquests troncs, en iniciar les assignacions per diferents cursos, serien un origen de solucions radicalment diferents. D'altra banda, amb aquesta evolució de la llista de professors reordenable a cada recursió, la qualitat del recorregut vertical està assegurada.



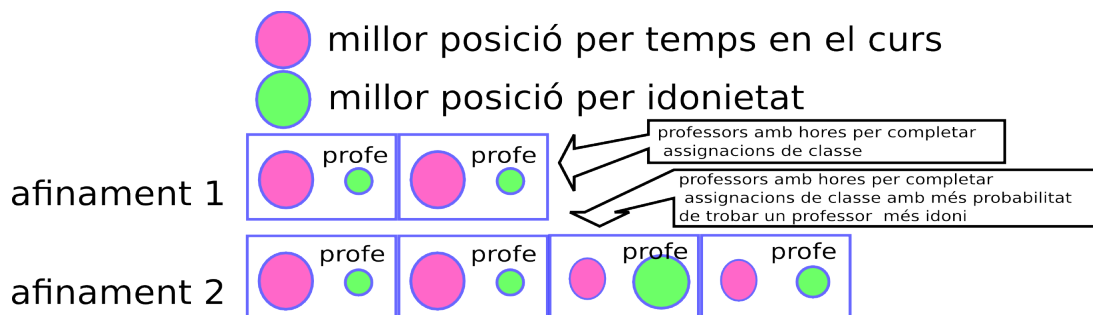
Il·lustració 9: Optimització dels primers fills

Aplicant aquesta millora s'ha passat a aconseguir molt bons resultats observant-los primer en els escenaris d'un sol processador i una sola solució amb un valor de percentatge de recorregut de l'ordre de 10 elevat a menys 40, reduint el temps a pocs segons quan sense aquesta optimització hagués calgut milers d'anys per trobar unes solucions similars.

L'altra inconvenient que s'ha evitat ha estat el de l'encallada en un inici d'assignacions a cursos de deixava la resta a assignar amb poques opcions per la llista de professors amb poques hores, tot i que la valoració de les assignacions aconseguides era alta.

4.5 Estudi del paràmetre afinament

Un cop fixada la reducció de l'espai de cerca, el darrer paràgraf de l'anterior punt ens introdueix en el paradigma següent: quant més professors amb alta idoneïtat es troben en les llistes, millors solucions (això és conseqüència evident), però de seguida passa que els darrers cursos es queden sense hores de professor.



Il·lustració 10: Concepte afinament

El cert és que l'origen d'aquesta optimització de l'algoritme deriva de l'observació empírica de les variacions del paràmetre d'afinament. Aquest paràmetre s'encarrega de poblar la llista de professors en les assignacions de les classes d'un curs concret.

4.6 Millors temps quan es busquen menys

També observat empíricament, tot i que demostrable amb un simple seguiment de l'algoritme, quantes menys solucions es busquen, menys es triga a obtenir-les.

Seconds computing	Num Max Solutions	Time to compute	Num processors	Afina	Max value	Timetabling computing	Search completed
5	1	300	1	7,00	8942,00	<input type="checkbox"/>	9.67759295863e-39%
1	1	300	1	2,00	7655,00	<input type="checkbox"/>	9.67759295863e-39%
301	5	300	1	4,00	8476,00	<input type="checkbox"/>	4.97329188379e-60%
300	10	300	1	7,00	0,00	<input type="checkbox"/>	1.10149825411e-70%
300	10	300	1	2,00	7406,00	<input type="checkbox"/>	3.29929753648e-70%

Il·lustració 11: Comprovació empírica menys solucions millors temps

4.7 Incorporació de les solucions amb millors paràmetres

De les darreres optimitzacions i amb l'objectiu de trobar les màximes solucions de màxima qualitat amb el menor temps, amplio l'algoritme perquè vagi acumulant solucions aplicant varies vegades l'algoritme parametritzant-lo inicialment amb una afinament màxim i una cerca de poques solucions, però reduint l'afinament a la vegada que augmentant el número de solucions buscades a mesura que es repeteix. Sempre amb la simplificació del primer fill en l'arbre de la permutació de cursos.

El resultat d'aquestes darreres modificacions són espectaculars, trobant moltes solucions i molt bones. De totes maneres trobo limitacions quan intento passar del centenar de solucions. Tot i així, considero un recorregut important en el perfeccionament de l'algoritme.

Seconds computing	Num Max Solutions	Time to compute	Num processors	Afina	Max value
11	30	600	2	8,00	8949,00
10	60	300	2	7,00	8949,00
301	200	300	2	9,00	8949,00

Il·lustració 12: Pas de gegant en l'optimització

4.8 Sobra temps, aprofundim en l'arbre...

Un cop fetes aquestes millores en l'algoritme, ens trobem que l'algoritme no esgota el temps que li donem per fer els càlculs.

Seconds computing	Num Max Solutions	Time to compute	Num processors	Afina	Deep Only One	Max value	Timetabling computing	Search completed
11	30	600	2	8,00	1	8949,00	<input type="checkbox"/>	1.61293215977e-39%
10	60	300	2	7,00	1	8949,00	<input type="checkbox"/>	8.06466079886e-40%
301	200	300	2	9,00	1	8949,00	<input type="checkbox"/>	3.87103721507e-40%
300	10	300	2	7,00	3	9070,00	<input type="checkbox"/>	9.27704813896e-38%
88	5	300	2	7,00	2	9070,00	<input type="checkbox"/>	6.77431507104e-38%

Il·lustració 13: Darrera optimització: aprofitant el temps sobrant

Per intentar trobar més solucions i alguna de millor, preparo poder fer un recorregut complet de l'arbre de permutacions de les assignacions de cursos per la part de dalt. Deixant les branques inferiors amb només el primer fill.

Tal com havíem previst, si bé és cert que es triga més a fer els càlculs, les solucions són lleugerament millors en fer més recorregut en el domini de cerca.

5 Conclusions

5.1 Lliçons apreses

Un punt molt important que m'ha quedat marcat amb foc ha estat que perquè es tingui un algoritme amb solucions segures, no vol dir que un visqui per veure-les (perdoneu l'humor negre fruit de certs moments de desesperació).

La percepció de tenir un algoritme robust quan es prova amb un joc de test petit va quedar ensorrada en veure el quasi nul recorregut que finalment feia la màquina sense cap sortida correcta.

Així, l'aplicació de variables de control i informes parcials ha estat una necessitat per descobrir els punts on calia actuar per assegurar solucions, encara que fos sacrificant possibles solucions una mica millors.

Crec que també seria destacable la intuïció que l'aproximació als problemes complexos cal fer-los amb un rigor matemàtic major. Possiblement, un canvi d'enfocament hagués permès aplicar un mètode d'optimització heurístic. Tot i que estudiada (aplicació d'algoritmes genètics en la generació d'horaris) [4], aquesta oportunitat no l'he sabuda veure.

5.2 Anàlisis d'assoliment d'objectius

Sota el meu parer, els objectius eren força ambiciosos.

En primer lloc, crec que s'ha aconseguit un bon programa per assignar professors a classes. A la vegada, s'ha facilitat la planificació dels horaris per encaixar les activitats a les hores setmanals que es puguin definir. No amago la realitat que cada centre té les seves pròpies característiques i que treballar amb aquesta aplicació no serà mai possible de forma directa. Tot i això, crec que facilito el pas perquè es pugui adaptar el codi resultant en un cas concret.

Dit això, entenc que les explicacions donades en els capítols anteriors em serveixen d'argument per considerar complerts els objectius inicialment establerts en una mesura, per a mi, satisfactòria.

5.3 Observacions sobre la planificació

La preparació dels mòduls per treballar en Odoo intentava aconseguir el doble objectiu de fer les proves amb la comoditat d'una interfície web i d'altra banda deixar desenvolupament fet per un possible aprofitament de les vistes en altres àmbits. Això ha fet endarrerir la posada en marxa de les proves i conseqüents millores de l'algoritme.

Sí es veritat, que un cop assentat l'algoritme en la plataforma, m'he dotat d'una eina còmode per implementar les modificacions que anaven sorgint de les proves.

Exposo, amb sinceritat, que la planificació no és el meu fort i deixo de banda aspectes molt importants en qualsevol treball. Són, sobretot, els relacionats amb la comunicació amb companys per buscar punts de vista diferents o, simplement, un cop de mà. Tot i així, m'he apartat una bona quantitat de temps i he gaudit de l'elaboració del treball sense excessives angixes.

En conclusió, el seguiment de la planificació ha estat suficient, potser no per evitar un sprint final, però sí per assegurar un producte final satisfactori.

5.4 Línies de treball futures

5.4.1 Mantenir i millorar la documentació sobre els treballs en aquesta àrea

Els treballs de recerca sobre aquests tipus de problemes complexos són molts, de fa molt de temps, i de gran qualitat. En la bibliografia inclosa en el treball només exposo els que he pogut abordar. No descarto que no només en el futur hi hagi moltes solucions més ben abordades.

Un exemple d'aquests treballs pot ser un de ja fa uns anys però molt interessant d'en R.J. Aust «An improvement algorithm for school timetabling» a la revista The Computer Journal, Vol 19 Number 4 pag 339-343 Abril 1975 [5]

5.4.2 Millorar la base matemàtica del treball

No haurà estat el primer problema que s'hagi abordat amb èxit sense una base matemàtica. Això no vol dir que no s'hi hagi hagut de treballar amb pic i pala quan potser no hagués calgut.

5.4.3 Avançar en la integració dels algoritmes del FET

Centrat en la millora de l'algoritme, he deixat la integració de FET en una crida com a programa extern. Amb la consecució de múltiples solucions, el fet que una d'elles sigui donada com a bona pel FET és probable. Aquest ha sigut un dels motius per centrar els esforços del treball en trobar moltes solucions en l'assignació de professors i de qualitat.

5.4.4 Incorporar restriccions avançades en el procés automàtic d'assignació de professors i generació d'horaris.

El codi entregat és a punt per rebre les preferències en forma de desiderates dels professors o altres restriccions d'aules o d'activitat.

Un petit pas a l'hora de la implantació amb aquestes desiderates seria fer una primera exportació per, un cop importades manualment en el programa d'escriptori FET, i incloses les personalitzacions, fer una exportació de les dades. Amb aquest XML, l'assignador de professors podria comprovar la bondat en FET de cada solució.

6 Glossari

Activitats.....	1, 4, 11, 20
Nomenclatura FET que es refereix a una classe amb professor.	
Algoritme.....	2p., 5pp., 9p., 13p., 16pp.
Conjunt de instruccions ordenades per a aconseguir resoldre una tasca.	
Biblioteca de llibreries.....	4
Conjunt de trossos de codi cadascun d'ells per resoldre una petita tasca amb l'ànim de reutilització.	
Cerca exhaustiva.....	7
Un algoritme que suposa buscar totes les possibilitats en un problema, per molt complex que sigui.	
Classe.....	ipp., 11pp., 18, 20
En aquest treball em refereixo com a classe la relació entre rebre una matèria per part d'un grup d'alumnes.	
Codi lliure.....	1, 4
Codi font d'un programa fruit d'un desenvolupament informàtic que es fa públic sense estar lligat a cap restricció. No confondre amb el programari lliure ja que el codi potser amb llicència (s'usa lliurement el programa, no el codi).	
Codi obert.....	ipp.
Ens referim al mateix que el codi lliure però amb restriccions determinades per una llicència d'ús que marca el creador o publicador del codi.	
Codificació multiprocessador.....	10
La codificació que té en compte les capacitats d'executar en paral·lel un programa o una part d'ell per part d'un hardware que n'és capaç.	
Desiderata.....	4
Nomenclatura d'àmbit docent que es refereix a les preferències personals del professorat respecte, per exemple, al seu horari, o altres decisions que afecten a la seva carrera professional.	
ERP.....	3p., 8
Són les sigles d'Enterprise Resource Planning que categoritzen un programari especialitzat en la gestió integral d'una empresa. Actualment s'utilitza més la empremta Software d'Empresa per remarcar la incorporació moderna de funcions socials.	
FET.....	ipp., 9pp., 19pp.
És un codi obert i lliure per planificar automàticament el horaris d'una escola, institut o universitat.	
Generació d'horaris.....	i, 3, 10
Es refereix a posar hora i aula a una activitat, classe amb professor.	
Github.....	8, 26
Repositori de codi amb amb facilitats de caire col·laboratiu entre desenvolupadors.	
Interacció amb l'usuari.....	1
És l'àmbit de la informàtica que es cuida que es seu ús sigui el màxim de desapercebut per l'usuari humà.	
Interfície.....	2p., 20
El conjunt de pantalles o similars que presenta una aplicació a l'usuari.	
Kernel.....	iv, 10, 16
Terme tècnic que es refereix al codi pur d'un algoritme determinat.	

Matèria.....	i, 4p., 8p., 13
Coneixements docents susceptibles de transmissió.	
Nuika.....	10
Compilador d'ús lliure de Python.	
Optimització.....	i, iip., 10, 13, 16pp., 20
Aconseguir millors resultats en un algoritme aplicant canvis. Hi ha mètodes estudiats que es poden aplicar a uns problemes tipus.	
Pla d'estudis.....	8
Contingut docent que requereix una partició en varis espais de temps en ser, to plegat, d'un volum considerable	
Planificador de Curs.....	4
Aquí em refereixo com la figura que planifica per complert el curs entés com un espai de temps amb uns plans d'estudis previstos.	
Plataforma de software.....	i
Plataforma que facilita el desenvolupament i execució de programari.	
Presa de decisions.....	i
L'acte en si de decidir quelcom que afectarà a l'esdevenir d'una empresa.	
Problema complex.....	i, 7
Normalment és aquell que necessita d'un estudi profund.	
Programació evolutiva.....	7
Mètode d'optimització que s'asimila a l'evolució biològica de les espècies.	
Programari obert.....	4
És un concepte similar al programari lliure però inclou un plus de facilitat en la col·laboració sobre el desenvolupament.	
Python.....	3p., 8, 10, 16
Llenguatge de programació interpretat (amb codi precompilat), orientat a funcions i amb esperit de lectura ràpida per facilitar la reutilització abaix nivell.	
RAD.....	3
Programari que facilita el desenvolupament d'aplicacions amb el paradigma que resa les sigles Rapid Application Development.	
Recursivitat.....	13
Tècnica de programació que consisteix en que un tros de codi, en un moment donat torna a executar-se un altre cop, continuant després en el punt amb les variacions de dades que s'hagin fet.	
Seminaris.....	13
En la nomenclatura docent, un conjunt de professors que tenen unes matèries característiques entre les seves especialitats.	
Subgrups.....	8
En els moduls school, els subgrups són de molts tipus com de particions de grup curs, grups específics d'alumnes amb convalidacions, grups determinats d'un grup classe que formen un desdoblament, etc...	
TFM.....	8
Treball Fi de Master, el que intenten ser totes aquestes pàgines.	

7 Bibliografia

1: , Descripció de l'algoritme FET, , <http://lalescu.ro/liviu/fet/doc/en/generation-algorithm-description.html>

2: Tomas Müller, , 2005, <http://www.unitime.org/papers/phd05.pdf>

3: Comunitat Python, Documentació Python multiprocessing, , <https://docs.python.org/2/library/multiprocessing.html#>

4: Dipesh Mittal, Hiral Doshi, Mohammed Sunara, Renuka Nagpure, Automatic Timetable Generation using Genetic Algorithm, 2015, <http://www.ijarcce.com/upload/2015/february-15/IJARCCE4I.pdf>

5: R.J. Aust, An improvement algorithm for school timetabling, 1975, <http://comjnl.oxfordjournals.org/content/19/4/339.full.pdf>

8 Annexos

8.1 Comparativa de Software de Generador d'Horaris

Software	Llicència	Modular	Versió de prova	Solucions			Organitzador de grups	Varietat de restriccions
				Solució completa	Solucions parcials	Control manual		
FET	GNU Affero	no	-	SI	SI	SI	no	Molt alta
Peñalara	Propietària	no	si	SI	SI	SI	no	Molt alta
Kronowin	Propietària	si		SI			Opcional	
Untis	Propietària	si	si	SI				
ascTimeTables	Propietària	no						
best Timetabling	Propietària	si	no	SI	?	SI	no	?
Tablix/Gtablix	GNU	no	-	SI	NO	NO	no	baixa
UniTime 4.0	Apache Soft							

Software	Prioritat a la integració	Assignació de professors	Entrades	Sortida	Anys	Llicències (anunciades)
FET	No	No	XML,CSV	XML,CSV,HTML		-
Peñalara	SI	Per àrees i per preferències			20	3000
Kronowin	NO	Per àrees i preferències			30	8000
Untis	NO	En mòdul apart			40	
ascTimeTables	NO				20	150000
best Timetabling	NO	?			10	150
Tablix/Gtablix	no	no	XML	HTML	2005-2009	-
UniTime 4.0						

8.2 Instruccions d'instal·lació Odoo sobre Linux Ubuntu 14.04

He reproduït íntegrament el contingut de la WEB amb les instruccions per instal·lar la plataforma Odoo per motius de utilitat bàsica a l'hora de preparar l'entorn d'execució del codi d'aquest treball.

L'autor és n'Alan Lord que el setembre de 2014 va publicar aquestes instruccions en el site <http://www.theopensource.com/>, tot i completades també arran de les aportacions dels comentaristes. A tots ells les gràcies per facilitar aquesta tasca tan mecànica.

8.2.1 Introduction

Welcome to the latest of our very popular OpenERP Odoo installation "How Tos".

The new release of Odoo 8.0 is a major upgrade introducing a great many new features and a new name.

Odoo 8.0 is not only better looking and easier to use, it also brings many improvements to the existing feature-set and adds a number of brand new features which extend the scope of the business needs covered by Odoo. Ecommerce, CMS, Integrated BI...

Rather than me blathering on about what's new, you can simply just go and [read the release notes here](#).

8.2.2 The How To

Following that introduction, I bet you can't wait to get your hands dirty...

Just one thing before we start: You can simply [download](#) a .deb (for Debian/Ubuntu type systems) or a .rpm (Redhat/CentOS) package of OpenERP and install that. Unfortunately that approach doesn't provide us ([Libertus Solutions](#)) with enough fine-grained control over where things get installed, and it restricts our flexibility to modify & customise, hence I prefer to do it a slightly more manual way (this install process below should only take about 10-15 minutes once the host machine has been built).

This time, rather than using a source tarball as the basis for installation we are going to take the code straight from the [Odoo 8.0 branch on Github](#). This should help when it comes to installing updates and bug fixes in the future by being able to issue a `git pull` command to update the code. *Bear in mind before doing a pull request you should always have backups and you may need to update your Odoo database(s) as well.*

So without further ado here we go:

8.2.3 Step 1. Build your server

I install just the bare minimum from the install routine (you may want to install the openssh-server during the install procedure or install subsequently depending on your needs).

After the server has restarted for the first time I install the openssh-server package (so we can connect to it remotely) and denyhosts to add a degree of brute-force attack protection. There are other protection applications available: I'm not saying this one is the best, but it's one that works and is easy to configure and manage. If you don't already, it's also worth looking at setting up key-based ssh access, rather than relying on passwords. This can also help to limit the potential of brute-force attacks. [NB: This isn't a How To on securing your server...]

```
sudo apt-get install openssh-server denyhosts
```

UPDATE: Note that it seems denyhosts is no longer being maintained and is not in the main Ubuntu repository any more. I'm aware of a possibly suitable alternative called fail2ban but have not used it yet; do your own research. Thanks to Rami for the cluebat!

UPDATE2: Thanks to Paul for the pointer. I have added python-unicodcsv to the list of dependencies. Apparently this is required to correctly restore backups.

Now make sure your server has all the latest versions & patches by doing an update:

```
sudo apt-get update
sudo apt-get dist-upgrade
```

Although not always essential it's probably a good idea to reboot your server now and make sure it all comes back up and you can login via ssh.

Now we're ready to start the Odoo install.

8.2.4 Step 2. Create the Odoo user that will own and run the application

```
sudo adduser --system --home=/opt/odoo --group odoo
```

This is a “system” user. It is there to own and run the application, it isn’t supposed to be a person type user with a login etc. In Ubuntu, a system user gets a UID below 1000, has no shell (it’s actually `/bin/false`) and has logins disabled. Note that I’ve specified a “home” of `/opt/odoo`, this is where the OpenERP server code will reside and is created automatically by the command above. The location of the server code is your choice of course, but be aware that some of the instructions and configuration files below may need to be altered if you decide to install to a different location.

[Note: If you want to run multiple versions of Odoo/OpenERP on the same server, the way I do it is to create multiple users with the correct version number as part of the name, e.g. `openerp70`, `openerp61` etc. If you also use this when creating the Postgres users too, you can have full separation of systems on the same server. I also use similarly named home directories, e.g. `/opt/odoo80`, `/opt/openerp70`, `/opt/openerp61` and config and start-up/shutdown files. You will also need to configure different ports for each instance or else only the first will start.]

A question I have been asked a few times is how to run the Odoo server as the odoo system user from the command line if it has no shell. This can be done quite easily:

```
sudo su - odoo -s /bin/bash
```

This will su your current terminal login to the odoo user (the “-” between su and odoo is correct) and use the shell `/bin/bash`. When this command is run you will be in odoo’s home directory: `/opt/odoo`.

When you have done what you need you can leave the odoo user’s shell by typing `exit`.

8.2.5 Step 3. Install and configure the database server, PostgreSQL

```
sudo apt-get install postgresql
```

Then configure the Odoo user on postgres:

First change to the postgres user so we have the necessary privileges to configure the database.

```
sudo su - postgres
```

Now create a new database user. This is so Odoo has access rights to connect to PostgreSQL and to create and drop databases. Remember what your choice of password is here; you will need it later on:

```
createuser --createdb --username postgres --no-createrole --no-  
superuser --pwprompt odoo  
Enter password for new role: *****  
Enter it again: *****
```

Finally exit from the postgres user account:

```
exit
```

8.2.6 Step 4. Install the necessary Python libraries for the server

```
sudo apt-get install python-cups python-dateutil python-decorator  
python-docutils python-feedparser \  
python-gdata python-geoip python-gevent python-imaging python-jinja2
```

```
python-ldap python-libxslt1
python-lxml python-mako python-mock python-openid python-passlib
python-psutil python-psycopg2
python-pybabel python-pychart python-pydot python-pyparsing python-
pypdf python-reportlab python-requests \
python-simplejson python-tz python-unicodcsv python-unittest2 python-
vatnumber python-vobject \
python-werkzeug python-xlwt python-yaml wkhtmltopdf
```

With that done, all the dependencies for installing Odoo 8.0 are now satisfied (note that there are some changes between this and the packages required for OpenERP 7.0).

UPDATE & NOTE: It has been pointed out to me that the Qweb templating engine in Odoo 8 is warning that the version of wkhtmltopdf is too old. It turns out that Ubuntu 14.04 packages version 0.9.9 of this library even though this is rather old. To update your Ubuntu server please follow the instructions on [this page](#). Many thanks to Ruben Kannan for pointing this out :-).

UPDATE & NOTE II: Zak suggests an alternative method to get and install the most recent version of wkhtmltopdf in [this comment below](#). Thanks Zak.

UPDATE & NOTE III: David suggested adding python-cups and python-geoip to the list of modules to install in [this comment below](#). Thanks David.

8.2.7 Step 5. Install the Odoo server

Install Git.

```
sudo apt-get install git
```

Switch to the Odoo user:

```
sudo su - odoo -s /bin/bash
```

Grab a copy of the most current Odoo 8 branch (*Note the “.” at the end of this command!*):

```
git clone https://www.github.com/odoo/odoo --depth 1 --branch 8.0
--single-branch .
```

(This might take a little while depending on the speed of your Internet connection.)

Note: Thanks to [Ian Beardslee](#) for the cluebat. Have now added --depth 1 to the command so it only retrieves the latest version without all the history. The download is now quite a bit quicker.

Once it's finished exit from the odoo user: `exit`.

8.2.8 Step 6. Configuring the OpenERP application

The default configuration file for the server (`/opt/odoo/debian/openerp-server.conf`) is actually very minimal and will, with only a small change work fine so we'll copy that file to where we need it and change it's ownership and permissions:

```
sudo cp /opt/odoo/debian/openerp-server.conf /etc/odoo-server.conf
sudo chown odoo: /etc/odoo-server.conf
sudo chmod 640 /etc/odoo-server.conf
```

The above commands make the file owned and writeable only by the odoo user and group and only readable by odoo and root.

To allow the odoo server to run initially, you should only need to change two lines in this file. Toward to the top of the file change the `linedb_password = False` to the same password you used back in step 3. Then modify the line `addons_path = /usr/lib/python2.7/dist-packages/openerp/addons so that it reads addons_path = /opt/odoo/addons` instead.

One other line we might as well add to the configuration file now, is to tell Odoo where to write its log file. To complement my suggested location below add the following line to the `odoo-server.conf` file:

```
logfile = /var/log/odoo/odoo-server.log
```

Use your favourite text editor here. I tend to use nano, e.g.

```
sudo nano /etc/odoo-server.conf
```

Once the configuration file is edited and saved, you can start the server just to check if it actually runs.

```
sudo su - odoo -s /bin/bash
/opt/odoo/openerp-server
```

If you end up with a few lines eventually saying OpenERP (Yes. The log still says OpenERP and not Odoo) is running and waiting for connections then you are all set.

If there are errors, you'll need to go back and find out where the problem is.

Otherwise simply enter `CTL+C` to stop the server and then `exit` to leave the `openerp` user account and go back to your own shell.

8.2.9 Step 7. Installing the boot script

For the final step we need to install a script which will be used to start-up and shut down the server automatically and also run the application as the correct user. There is a script you can use in `/opt/odoo/debian/init` but this will need a few small modifications to work with the system installed the way I have described above. [Here's a link](#) to the one I've already modified for Odoo version 8.

Similar to the configuration file, you need to either copy it or paste the contents of this script to a file in `/etc/init.d/` and call it `odoo-server`. Once it is in the right place you will need to make it executable and owned by root:

```
sudo chmod 755 /etc/init.d/odoo-server
sudo chown root: /etc/init.d/odoo-server
```

In the configuration file there's an entry for the server's log file. We need to create that directory first so that the server has somewhere to log to and also we must make it writeable by the `openerp` user:

```
sudo mkdir /var/log/odoo
sudo chown odoo:root /var/log/odoo
```

8.2.10 Step 8. Testing the server

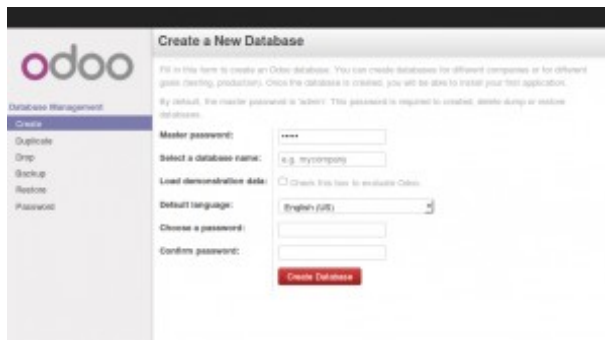
To start the Odoo server type:

```
sudo /etc/init.d/odoo-server start
```

You should now be able to view the logfile and see that the server has started.

```
less /var/log/odoo/odoo-server.log
```

If there are any problems starting the server you need to go back and check. There's really no point ploughing on if the server doesn't start...



Odoo 8 New Database

If the log file looks OK, now point your web browser at the domain or IP address of your Odoo server (or localhost if you are on the same machine) and use port 8069. The url will look something like this:

`http://IP_or_domain.com:8069`

What you should see is a screen like this one (it is the Database Management Screen because you have no Odoo databases yet):

What I do recommend you do at this point is to change the super admin password to something nice and strong (Click the “Password” menu). By default this password is just “admin” and knowing that, a user can create, backup, restore and **drop** databases! This password is stored in *plain text* in the `/etc/odoo-server.conf` file; *hence why we restricted access to just odoo and root*. When you change and save the new password the `/etc/odoo-server.conf` file will be re-written and will have a lot more options in it.

Now it's time to make sure the server stops properly too:

```
sudo /etc/init.d/odoo-server stop
```

Check the log file again to make sure it has stopped and/or look at your server's process list.

8.2.11 Step 9. Automating Odoo startup and shutdown

If everything above seems to be working OK, the final step is make the script start and stop automatically with the Ubuntu Server. To do this type:

```
sudo update-rc.d odoo-server defaults
```

You can now try rebooting you server if you like. Odoo should be running by the time you log back in.

If you type `ps aux | grep odoo` you should see a line similar to this:

```
odoo 1491 0.1 10.6 207132 53596 ? S1 22:23 0:02 python
/opt/odoo/openerp-server -c /etc/odoo-server.conf
```

Which shows that the server is running. And of course you can check the logfile or visit the server from your web browser too.



Odoo 8 Opening Screen

That's it! Next I would suggest you create a new database filling in the fields as desired. Once the database is initialised, you will be directed straight to the new main configuration screen which gives you a feel for the new User Interface in Odoo 8 and shows you how easy it is to set up a basic system.

8.3 Instal·lació del producte resultant, els mòduls school

El software resultant i les instruccions d'instal·lació del software es troben en el següent enllaç <https://github.com/pereerro/schooly> . Les modificacions se s'hi realitzin hi quedaran enregistrades com a «commits» nous i son sempre públiques.