

Escola Universitària Politécnica de Mataró

Centre adscrit a:



UNIVERSITAT POLITÈCNICA
DE CATALUNYA

Enginyeria Tècnica en Informàtica de Gestió

**AUTOMATITZACIÓ DE LA RESERVA D'ESPAIS
DOCENTS DEL TECNOCAMPUS DE MATARÓ**

Memòria

Sergio Álvarez Porto
PONENT: Eduard de Bru de Sala i Castells

TARDOR 2012



TecnoCampus
Mataró-Maresme

Dedicatòria

Dedico aquest projecte a la meva família que sempre m'ha donat suport i m'ha ajudat a seguir formant-me i ampliar els meus coneixements per tenir més oportunitats de cara al futur.

Agraïments

Vull agrair la seva col·laboració, en major o menor mesura a:

Manu Díaz, company d'enginyeria, per donar-me consells per resoldre problemes i millorar el disseny gràfic de l'aplicació. I a en Sergi Rojas, també company d'enginyeria, per donar-me suport, aconsellar-me i ajudar-me a trobar algunes eines de treball.

Resum

Es tracta de desenvolupar un software per automatitzar la reserva d'espais docents (aules i laboratoris) del Tecnocampus de Mataró. L'aplicatiu tindrà que obtenir la informació que generen els diferents centres universitaris des d'una base de dades MySQL, i generar un calendari de reserves per cada espai en el format Outlook corporatiu. S'ha assolit l'objectiu principal del projecte, però encara es pot continuar desenvolupant i millorar les funcionalitats de l'aplicació.

Resumen

Se trata de desarrollar un software para automatizar la reserva de espacios docentes (aulas y laboratorios) del Tecnocampus de Mataró. El aplicativo tendrá que obtener la información que generan los diferentes centros universitarios desde una base de datos MySQL, y generar un calendario de reservas para cada espacio en el formato Outlook corporativo. Se ha logrado el objetivo principal del proyecto, pero aún se puede seguir desarrollando y mejorar las funcionalidades de la aplicación.

Abstract

It is about developing software to automate the booking of teaching spaces (classrooms and laboratories) from Tecnocampus of Mataró. The application will have to get the information generated by various universities from a MySQL database and generate a schedule for each booking space in corporative Outlook format. Reached the goal of the project, but can still be developed and improve the functionality of the application.

Índex.

Índex de Figures.....	V
Índex de taules	IX
Glossari de termes	XI
1. Objectius.....	1
1.1. Propòsit.....	1
1.2. Finalitat.....	1
1.3. Objecte.....	1
1.4. Abast.....	1
2. Introducció	3
2.1. Què és Outlook?	3
2.2. Com funciona el calendari Outlook?	4
2.3. Restriccions a l'hora d'adjudicar cites.....	5
2.4. GP Untis 2011	5
3. Objectius i planificació del projecte	9
3.1. Objectius del projecte	9
3.2 Estudi de viabilitat.....	10
3.3 Planificació del projecte.....	12
3.4 Pressupost i estimació cost econòmic:.....	15
4. Elecció de l'entorn de desenvolupament.....	19

4.1. Introducció	19
4.2. Entorn de desenvolupament Java	19
4.3. PhpMyAdmin.....	20
4.4. Visual Basic	22
4.5. Gantt Project	24
5. Desenvolupament amb Netbeans	27
5.1. Introducció	27
5.2. Control d'errors.....	27
5.3. Control de sincronia	28
5.4. Patrons de disseny	28
5.5. JDBC vs JPA	29
5.6. Llibreries.....	31
6. Evolució del projecte.....	35
6.1 Introducció.....	35
7. Anàlisi i disseny de l'aplicació	39
7.1 Introducció.....	39
7.2. Requeriments	39
7.2.1. Requeriments funcionals.....	39
7.2.2. Requeriments no funcionals	40
7.3. Casos d'ús.....	40
7.3.1. Fitxa de Casos d'ús.....	40

7.3.2. Diagrama de Casos d'ús.....	44
7.3.3. Diagrama de classes.....	45
7.3.4. Esquema de BBDD.....	49
8. Com funciona l'aplicació	51
8.1. Introducció	51
8.2. L'aplicació en Java	51
8.2.1. Connectant a la Base de dades	51
8.2.2. Filtrant les dades.....	53
8.3. L'aplicació en Visual Basic	57
9. Conclusions.....	59
9.1. Valoració del projecte.....	59
9.2. Valoració del resultat.....	59
9.3. Expectatives de millora.....	60
10. Annex.....	61
11. Bibliografia	63

Índex de Figures.

Figura 2.1. Carpetes i eines d' Outlook.....	3
Figura 2.2. Eina Outlook, Estructura del calendari.....	4
Figura 2.3. Eina Outlook, calendari de cites.....	5
Figura 2.4. GP Untis, pantalla de càrrega de dades.....	7
Figura 2.5. GP Untis, horari per períodes i grups.....	7
Figura 3.1. Temps d'execució a les proves amb el mètode 1.....	12
Figura 3.2. Diagrama de Gantt de la planificació del projecte.....	14
Figura 3.3. Taula cost econòmic.....	17
Figura 4.1. Entorn de desenvolupament Java Netbeans.....	20
Figura 4.2. Eina per l'administració de BBDD MySQL, MySQL Workbench.....	21
Figura 4.3. Eina des de el navegador, phpMyAdmin.....	21
Figura 4.4. Entorn Visual Basic, pàgina d'inici.....	23
Figura 4.5. Entorn Visual Basic, interfície gràfica.....	23
Figura 4.6. Entorn Visual Basic, entorn de desenvolupament.....	24
Figura 4.7. Exemple de Diagrama de Gantt amb Gantt Project.....	25
Figura 4.8. Exemple de diagrama de PERT amb Gantt Project.....	25
Figura 5.1. Exemple de sincronia en Java.....	28

Figura 5.2. Unitat de persistència.xml.....	31
Figura 5.3. Llibreries del projecte final de carrera.....	32
Figura 5.4. Esquema resum amb informació de l'API Java-Outlook.....	33
Figura 5.5. Esquema resum amb informació de l'API Java-Excel.....	34
Figura 6.1. Esborrany del disseny inicial de pantalla única.....	35
Figura 6.2. Esborrany de l'esquema de processament.....	36
Figura 7.1. Diagrama casos d'ús.....	44
Figura 7.2. Diagrama de classe del paquet Domini.....	45
Figura 7.3. Diagrama de classe del paquet Aplicació.....	46
Figura 7.4. Diagrama de classe del paquet Presentació.....	46
Figura 7.5. Diagrama de classe del paquet Persistència.....	47
Figura 7.6. Diagrama de classe del paquet Persistència de Sortida.....	48
Figura 7.7. Esquema de la BBDD migrada.....	49
Figura 7.8. Esquema de taules de la BBDD útils pel projecte.....	49
Figura 8.1. Pantalla de connexió a base de dades.....	51
Figura 8.2. Exemple d'accés a base de dades.....	52
Figura 8.3. Missatge d'error d'accés a la BBDD.....	52
Figura 8.4. Pantalla de filtrat de dates i mètode de processament.....	53
Figura 8.5. Finestra del calendari de dates.....	54

Figura 8.6. Missatge d'error amb les dates.....	54
Figura 8.7. Última pantalla, processament de dades.....	55
Figura 8.8. Missatge informatiu del mètode 1.....	56
Figura 8.9. Interfície VB, informa de que tot ha estat correcte.....	56
Figura 8.10. Missatge d'error: període de prova API Outlook finalitzat.....	57
Figura 8.11. Missatge informatiu de com obtenir una llicència de l'API.....	57

Índex de taules

Taula 3.1. Planificació de tasques del projecte.....	13
Taula 7.1. Cas d'ús Seleccionar origen de dades.....	41
Taula 7.2. Cas d'ús Seleccionar interval de dates.....	41
Taula 7.3. Cas d'ús Preparar les dades fent servir el mètode 1.....	42
Taula 7.4. Cas d'ús Preparar les dades fent servir el mètode 2.....	43
Taula 7.5. Cas d'ús Enviar les dades fent servir el mètode 1.....	43
Taula 7.6. Cas d'ús Enviar les dades fent servir el mètode 2.....	44

Glossari de termes

Cita	Espai reservat al calendari Outlook amb informació de l'esdeveniment referent al dia, hora, motiu i ubicació. Es pot, a més, afegir informació com si fos un missatge de correu nou i enviar-la.
Classe	Hores lectives de les matèries que imparteixen els professors als alumnes. Poden ser de 30, 45 60 minuts o de qualsevol altra durada.
OLTP	OnLine Transaction Processing. tipus de processament que facilita i administra aplicacions per l'entrada de dades i recuperació i processament de transaccions.
OLAP	OnLine Analytical Processing. Alternativa útil al camp de la intel·ligència empresarial amb el propòsit d'agilitzar la consulta de grans quantitats de dades.
CRUD	Operacions CRUD, es refereix a les operacions sobre bases de dades de creació, lectura, actualització i esborrat (create, read, update, delete).
DNS	Domain Name Server. La seva principal funció és traduir noms intel·ligibles per les persones en identificadors associats als equips connectats a la xarxa amb l'objectiu de localitzar-los.
Mètrica	Metodologia de planificació, desenvolupament i manteniment de sistemes d'informació.

1. Objectius.

1.1. Propòsit.

Dissenyar i implementar una aplicació que permeti actualitzar de forma ràpida i eficient el calendari Outlook de reserves d'espais docents.

1.2. Finalitat.

Obtenir informació dels espais docents d'una base de dades comuna entre els diferents centres universitaris, processar la informació i actualitzar el calendari d' Outlook amb tota la informació classificada per dies, aules i hores.

1.3. Objecte.

Un programa Java amb un disseny gràfic senzill, un altre programa codificat en Visual Basic que farà la funció de macro per enviar la informació a Outlook, i diferents documents explicatius de les parts que componen aquest projecte.

1.4. Abast.

S'aconsegueix amb èxit la informació i s'envia sense problemes a Outlook.

Dirigit al responsable o responsables d'assignació d'horaris dels diferents centres universitaris.

2. Introducció

Abans de començar a entrar en matèria, és convenient explicar què és Outlook i com funciona el calendari que incorpora.

2.1. Què és Outlook?

És un programa d'organització ofimàtica que ofereix eines per l'administració del correu electrònic personal de forma senzilla, entre d'altres. Forma part del paquet office de Windows que ofereix un conjunt d'aplicacions d'escriptori, servidors i serveis pels SO de MS Windows i Mac OS X. També disposa d'unes llicències de pagament que ofereixen a l'usuari millores en el serveis o usos de l'aplicació per un preu entre 99 i 168 euros.

No s'entrarà en detall en explicar com funciona Outlook perquè no és tan important el programa en si, sinó una eina en concret que aquest ofereix.

L'eina en concret que ofereix Outlook i que fa servir el Tecnocampus de Mataró és el calendari.

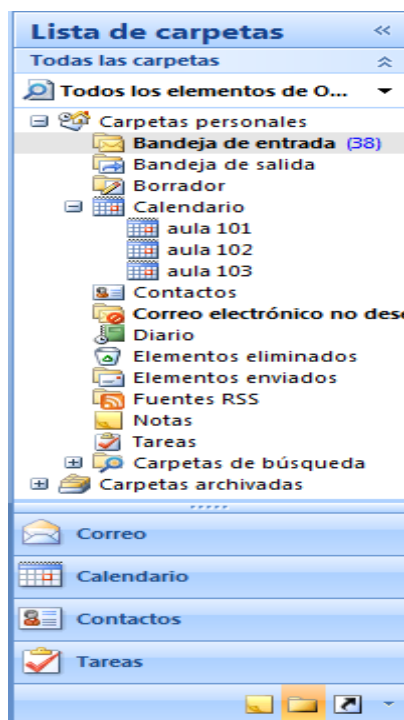


Figura 2.1. Carpetes i eines d' Outlook

2.2. Com funciona el calendari Outlook?

Outlook entre les moltes eines d'administració que ofereix, està la de calendari, que els centres universitaris del Tecnocampus fan servir per administrar i saber en tot moment l'ocupació d'aules i laboratoris dels centres.

El calendari té l'estructura normal que tindria qualsevol calendari, un llistat de dies entre els diferents mesos de l'any, d'un conjunt d'anys.

Permet afegir/eliminar cites als diferents dies amb la possibilitat de concretar hores, amb informació com el motiu de la cita, i que mostrarà en primera instància per diferenciar entre la resta de cites que pugui haver. També permet la creació de diferents calendaris, fet que permet una millor organització de les cites, tot de forma fàcil i eficient.

Tecnocampus fa ús d'aquesta eina per crear diferents calendaris. Cada una de les aules i laboratoris dels centres universitaris disposa d'un calendari propi, i cada calendari té en tot moment el detall (hora, dia, mes, any) de la ocupació de l'espai al que fa referència.

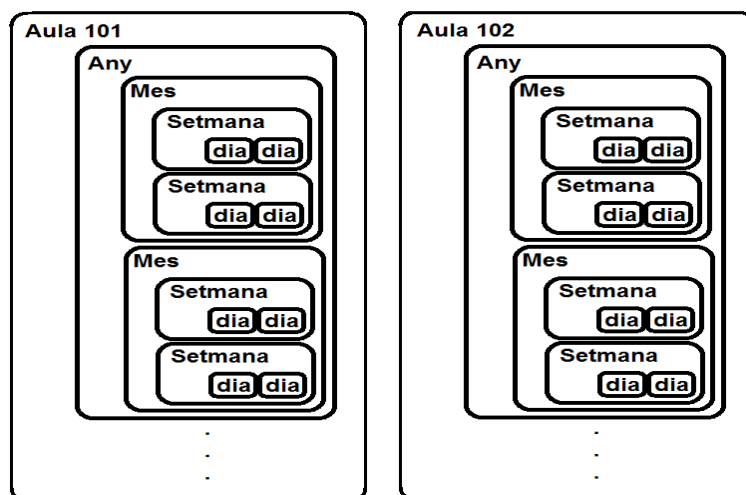


Figura 2.2. Eina Outlook, Estructura del calendari



Figura 2.3. Eina Outlook, calendari de cites.

2.3. Restriccions a l'hora d'adjudicar cites

Un dels objectius fonamentals de l'ús del calendari Outlook és per evitar el solapament de classes dins un mateix horari.

S'entén llavors com a solapament, que si en un hora concret d'un dia concret ja existeix informació en un dels calendaris (aules), aquella aula ja ha sigut reservada amb anterioritat per aquell moment del dia i no es podrà fer reserva de la mateixa aula al mateix moment, tot i que el calendari d' Outlook ho permet.

Existeixen casos en que a Outlook dins una mateixa aula en un mateix moment hi hagi dues cites i que no es consideri solapament. Aquest cas es podria donar, per exemple, si dos o més grups (s'entén com a grup la classificació que fa el centre de les diferents carreres) fan la mateixa assignatura.

2.4. GP Untis 2011

GP Untis és un software per a la gestió d'horaris, amb mòduls que permeten adaptar el programa a les necessitats del centre. Alguns d'aquests mòduls són:

- Horaris per seccions
- Horaris per períodes
- Horaris d'alumnes
- Franja d'assignatures
- Assignació de classes

A més permet generar horaris automàticament i ocupar espais per tot un període, fer diagnòstics de resultats, etc., tot amb una estructura clara i fàcilment comprensible.

Una de les possibilitats del GP Untis és la de compartir recursos concurrentment, en aquest cas necessita obtenir la informació d'una base de dades referenciada amb DNS del sistema de l'origen de dades (ODBC) de la màquina que el faci servir. És important que el nom amb el que es fa referència sigui "gpUntis", d'altra manera el software no ho detectarà i no serà capaç d'obtenir la informació.

Al iniciar el programa GP Untis demana un login i un password, a més d'altres dades com: 'codi del centre', 'curs escolar' i 'versió' de la informació obtinguda (veure Figura II).

Un cop carregada la informació de la base de dades, es poden realitzar diverses funcions, tals com: llistar professors, alumnes, aules, ..., llistar horaris, definir períodes, etc.

En el meu cas, aquest software ha estat usat majoritàriament per saber quin tipus d'informació trobaria, comprendre-la, i assegurar que la informació generada amb el software del projecte fos correcte.

Base de datos gp-U / Cargar datos externos

Código del Centro:
 840

Texto: EUPMT

Curso escolar:
 2011/2012

Texto:

Versión:
 1

Texto: T11

Usuario:
 debru

Clave de acceso:

Figura 2.4. GP Untis, pantalla de càrrega de dades

ETIG 3A Período: 12/9/2011 - 22/12/2011 Comparar horarios: activado Sólo horarios cambiados

16 Horas semanales. 0 Horas sin planificar. Grupo(s): EX11 EUPMT

	Lunes	Martes	Miércoles	Jueves	Viernes
8:00 - 8:50	Destino Proyectos Informatica FABREGAS JOAN	Matemáticas Aplicadas SEBA ENRIC	Matemáticas Aplicadas SEBA ENRIC	Destino Sistemas Informatic PALACIOS ALFONS aula 102	
9:00 - 9:50			Técnicas Destino SANCHEZ NURIA	Destino Proyectos Informatica FABREGAS JOAN aula 102	
10:10 - 11:00	Técnicas Destino SANCHEZ NURIA	Laboratori Informatica 3 BRU DE SALA EDUARD lab INF 1 ORL	Laboratori Informatica 3 BRU DE SALA EDUARD lab INF 1 ORL		
11:10 - 12:00					
12:10 - 13:00		Destino Sistemas Informatic PALACIOS ALFONS			
13:10 - 14:00					

Figura 2.5. GP Untis, horari per períodes i grups

3. Objectius i planificació del projecte

3.1. Objectius del projecte

L'objectiu d'aquest projecte és el desenvolupament d'una eina que permeti l'automatització de la reserva dels espais docents del Tecnocampus de Mataró. L'aplicatiu tindrà que obtenir la informació que generen els diferents centres universitaris des d'una base de dades MySQL, i generar un calendari de reserves per cada espai en el format Outlook corporatiu.

És recomanable tenir en compte la usabilitat en aquest projecte. L'aplicació ha de fer-se càrrec de proporcionar una interfície gràfica senzilla que faciliti al màxim la feina de l'usuari que s'encarregarà de fer anar el programa per tenir actualitzat el calendari Outlook.

En aquest projecte l'usuari haurà de tenir la impressió de que es tracta d'un programa senzill amb una interfície gràfica no gaire laboriosa. La dificultat del codi elaborat per a un bon funcionament del software no ha de veure's reflectida a la interfície, l'usuari no s'ha de trobar amb la necessitat de fer un gran esforç mental per entendre el funcionament dels diferents elements que apareixen; sense un estudi previ de l'eina, l'usuari ha de saber en tot moment què està fent i quin serà el resultat obtingut.

Aquesta aplicació ha de formar part de l'entorn de treball del responsable o responsables de l'assignació d'horaris. Així doncs, ha de tractar-se d'una aplicació que el responsable farà servir un cop al començament de cada semestre o trimestre, i que li permetrà tenir el calendari Outlook amb l'última informació de la BBDD comuna entre els diferents centres universitaris implicats.

L'usuari del software ha de tenir un coneixement de la informació que vol obtenir però no de com està guardada a la BBDD, per aquest motiu s'ofereix adjunt a la memòria un petit annex amb informació del programa, com funciona i coses que s'han de tenir en compte.

En quan al codi es tracta, d'estructurar i distribuir les diferents classes entre les diferents capes per permetre una mantenibilitat adequada, per si es donés el cas (i que si el projecte conclou amb èxit probablement sigui així...) que algun dia s'hagués de modificar part del

codi per algú altre que no fos el projectista, no li fos més costós en temps entendre i modificar el codi que fer el programa de nou.

El projecte presenta un altre objectiu amagat sota el desenvolupament de l'aplicació en Java: familiaritzar-se per part del projectista amb l'entorn del Visual Basic. En aquest projecte el Visual Basic s'utilitza per enviar de forma automàtica la informació a Outlook Calendar després de crear-la en Java.

3.2 Estudi de viabilitat

Primer s'ha de tenir clar en que consisteix l'estudi de viabilitat d'un sistema (EVS). Es tracta de fer l'anàlisi d'un conjunt de necessitats per a ser capaç de proposar una solució a curt termini, que tingui en compte les restriccions econòmiques, tècniques, legals i operatives.

L'estudi de viabilitat del sistema pel desenvolupament d'un software és una part molt important a tenir en compte dins el projecte, però no en tots els casos és imprescindible. No ho és si, com recomana Mètrica 3: *“Si la justificació econòmica és obvia, el risc tècnic baix, s'esperen pocs problemes legals i no existeix cap alternativa raonable, no és necessari profunditzar en l'estudi de viabilitat del sistema, analitzant possibles alternatives i realitzant una valoració i avaluació de les mateixes, sinó que aquest s'orientarà a l'especificació dels requisits, descripció del nou sistema i planificació”*.

Fent l'anàlisi les funcionalitats finals del software han sorgit preguntes respecte l'aplicació que s'estava desenvolupant: Per què es necessita aquest software?

Amb la informació donada, es pot arribar a dubtar de la necessitat del software produït en aquest projecte o del calendari mateix de l' Outlook. Llavors sorgeixen algunes preguntes com:

“Per què s'envia la informació a Outlook si ja la disposen en el GP Untis?”

GP Untis permet fer horaris i ocupar espais per tot un període que se li determini (període lectiu), però no permet reservar un espai només per un sol instant concret (hora, dia, mes i any) i això Outlook ho permet, és per aquest motiu que a més de GP Untis fan servir el calendari d' Outlook.

“Si el centre ja disposa d’un programa com GP Untis, que permet fer horaris, i del calendari d’ Outlook, per què volen aquest software?”

Cada centre té els seus responsables encarregats de l’assignació d’horaris, que li passen aquesta informació a la secretaria i aquesta introdueix la informació dels horaris un a un a Outlook. L’objectiu d’aquest projecte és evitar precisament això, es vol optimitzar aquest pas al màxim possible fent que amb uns pocs clics i un temps relativament curt s’envii tota la informació a Outlook.

Més endavant, a l’apartat 3.4 s’establirà una estimació del cost econòmic permetent valorar si és viable o no econòmicament aquest projecte.

Pel que fa a la part tècnica, no hauria d’existir risc per que en principi es satisfan els requeriments que es van considerar en un principi i no hauria d’haver cap problema en quant a la dificultat d’ús, es tracta d’un software senzill que no requereix d’una plantilla qualificada per dur-la a terme. De totes maneres l’encarregat de fer anar aquest programa serà el responsable de l’assignació d’horaris, que com a molt necessitarà saber com funcionen internament alguns elements del software especificats en un manual d’ús adjunt a la memòria. En quant a requisits del sistema, les proves s’han fet des d’un ordinador de sobretaula d’ús personal del projectista amb unes capacitats no molt elevades, no hauria de suposar cap problema pel maquinari del que disposen els centres Tecnocampus de Mataró. El software disposa de dos mètodes per processar i enviar les dades, un fent ús d’una API (de prova de tres mesos i posterior pagament), i l’altre d’una API gratuïta ajudat d’un programa extern codificat en Visual Basic. Els resultats de l’execució després de les pertinents millores en el rendiment del programa han sigut aquests:

Les proves s’han fet enviant prop de 8000 cites a Outlook, però s’han anat prenent notes del temps a les primeres 3000 cites, a les 3000 següents, i per últim el temps total d’execució

- Mètode 1 (API Java - Outlook):. El resultat ha sigut de quatre minuts per enviar les 3000 primeres, sis minuts per enviar les 3000 següents, i sis minuts més en enviar les 2000 restants. Un total aproximat de 16 minuts per enviar les 8000 cites.

```
creando appointment vez: 7934
creando appointment vez: 7935
creando appointment vez: 7936
Calendario actualizado correctamente
GENERACIÓN CORRECTA (total time: 16 minutes 4 seconds)
```

Figura 3.1. Temps d'execució a les proves amb el mètode 1.

- Mètode 2 (api Excel + Visual Basic): El resultat ha sigut de un minut per enviar les 3000 primeres cites, un minut i quinze segons més per enviar les 3000 següents, i prop d'un minut per les 2000 últimes cites. Un temps total aproximat de 3 minuts i 15 segons.

Aparentment quantes més cites han sigut enviades, més puja el temps que triga en enviar les restants, tot i que en un mètode el temps puja de forma exagerada. Cal comentar que no s'han fet sota les mateixes condicions, ja que quan es va acabar de codificar el mètode 2 i es van poder fer les proves, el període de proves amb el mètode 1 havia expirat i no es podia continuar fent proves de rendiment.

Per altra banda, cal comentar la inexistència de problemes de legalitat en quant a ús del software, sempre i quan es respectin les condicions d'ús de les API que es fan servir.

3.3 Planificació del projecte

Per tal d'assolir els objectius cal dur a terme una sèrie de tasques fins a finalitzar el projecte, i fer un seguiment de la planificació.

En un principi es va establir un temps aproximat de quatre mesos per realitzar el treball, que va des de l'inici de curs a mitjans de setembre fins a l'entrega del projecte i la memòria a mitjans de gener de l'any vinent. Ja que disposava de quatre mesos, la idea inicial era:

- Primer mes: Tenir definides les diferents funcionalitats de l'aplicació i haver començat amb l'estructura interna del codi del projecte.
- Segon mes: Haver acabat gran part del codi i començar a fer proves amb el processament de les dades.
- Tercer mes: Millora de la mantenibilitat i rendiment de l'aplicació.
- Quart mes: Elaboració de la memòria.

Cal comentar que el primer mes es va retardar i es va haver d'invertir part del segon mes per l'assoliment de l'objectiu del mes. Les primeres dues setmanes el treball realitzat va ser mínim per dificultats de comprensió de la base de dades i confusions en quant a l'objectiu de l'aplicació del projecte.

Nom Tasca	Data Inici	Data Fi
Estudi de l'entorn de desenvolupament	25/09/2012	26/09/2012
Estudi dels requisits i objectius	25/09/2012	06/10/2012
Estudi i comprensió de la BBDD	03/10/2012	01/11/2012
Disseny de l'aplicació	05/10/2012	09/10/2012
Contrast d'informació amb GP Untis	08/10/2012	09/10/2012
Codificació del mètode Excel de processament	10/10/2012	13/10/2012
Testing del mètode Excel de processament	15/10/2012	16/10/2012
Disseny i codificació d'interfícies	15/10/2012	10/11/2012
Estudi alternatives de processament (API J-O)	22/10/2012	24/10/2012
Substitució de JPA per JDBC	25/10/2012	26/10/2012

Reestructuració del codi (millora mantenibilitat)	26/10/2012	03/11/2012
Codificació i proves millora de rendiment	05/11/2012	17/11/2012
Formació en Visual Basic	19/11/2012	04/12/2012
Codificació Macro Visual Basic	23/11/2012	04/12/2012
Testing del programa en Visual Basic	04/12/2012	12/12/2012
Millores Visuals	12/12/2012	20/12/2012
Redacció de memòria	19/12/2012	14/01/2013
Entrega documentació	16/01/2013	18/01/2013
Preparació exposició oral	21/01/2012	31/01/2012

Taula 3.1. Planificació de tasques del projecte

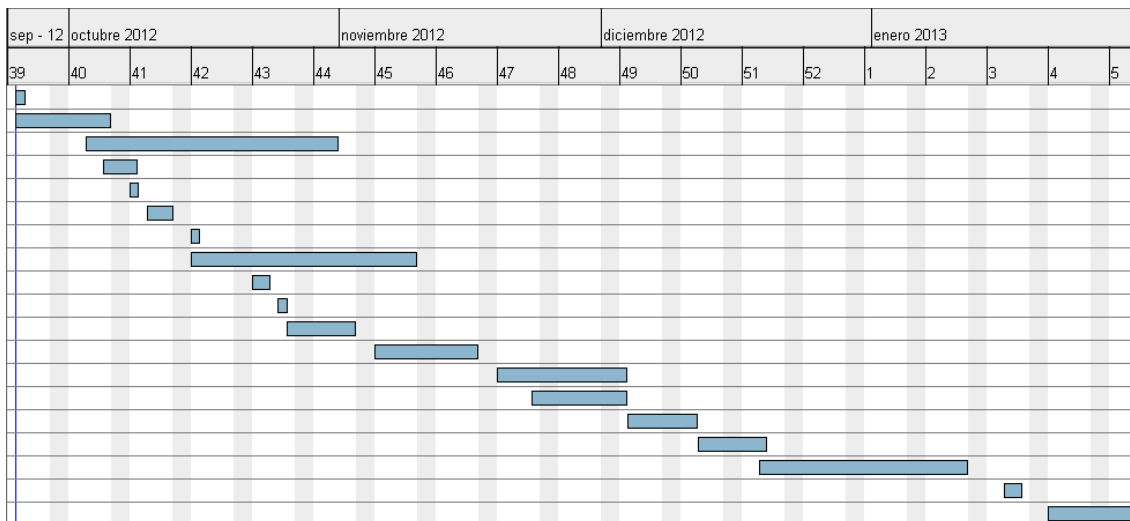


Figura 3.2. Diagrama de Gantt de la planificació del projecte

3.4 Pressupost i estimació cost econòmic:

Els costos per l'estimació del cost econòmic per el desenvolupament del software seran únicament, costos directes, imputables directament al pressupost pel desenvolupament: personal, software, hardware, formació, entre d'altres.

En tot el projecte la majoria de costos directes tals com software, hardware...suposen una petita part del que realment farà pujar el cost econòmic total. El cost directe que més fa pujar el cost econòmic total és el de personal, que en aquest cas es tracta d'un programador junior, amb un sou estimat d'uns 12€ l'hora, treballant unes cinc hores de mitjana al dia, cinc dies a la setmana més algun cap de setmana, durant quatre mesos.

Tenim llavors: 12€/hora per 5hores/dia per 5 dies per 4 setmanes al mes, resulta en uns 1.200€/mes, que estaria entre el que cobraria mensualment un programador junior en una multinacional (entre 13.000€ - 17.000€ l'any). Si s'agafa el sou mensual i es multiplica pels quatre mesos que dura el projecte, surt un cost econòmic total (tenint en compte només el sou de personal) de 4.800€.

Ara faltaria calcular el cost del software, hardware i formació.

El cost del software durant el desenvolupament de l'aplicació és de 0€, ja que el cost del software i sistema operatiu usats en aquest projecte estan inclosos dins el preu inicial de l'ordinador de treball, exceptuant el cas de l'API que connecta Java amb Outlook directament, que aquest resulta gratuït durant els primers 3 mesos de prova, però imputable si es vol continuar fent servir l'API més endavant.

El cost de l'API Java – Outlook és el següent:

Compra de llicències:

- Basic edition license: 197€ per developer
- Professional edition license: 479€
- Team edition license: 1.470€

A aquests costos s'ha d'afegir la renovació de llicència anual, que inclou actualitzacions i suport, surt a:

- Basic edition: 139€ per developer
- Professional edition: 337€
- Team edition: 999€

El cost del hardware usat és mínim. Es disposa d'un ordinador de sobretaula amb un preu inicial de 411€ sense IVA. Tenint en compte que l'ordinador té poc més de dos anys, i s'amortitza un 25% l'any, el valor actual d'aquest ordinador és de 205,5€.

Un altre cost directe a tenir en compte seria el de formació, amb una imputació del 15% per què no només és aplicable en aquest projecte, els coneixements adquirits poden ser útils per algun altre projecte. Resulta que la formació a estat via online, i amb documentació gratuïta, llavors el valor de formació està inclòs dins el cost d'electricitat i manteniment de l'adsl.

El desenvolupament del software s'ha realitzat en un domicili particular, fet que implica uns costos, no de mobiliari ni lloguer, però sí d'electricitat i manteniment de l'adsl.

El cost mensual de manteniment de l'adsl puja a 36€, del qual puc imputar un 15% per amortitzacions (5,4€), que multiplicat per els quatre mesos que dura el projecte fa un total de 21,6€.

Per saber la quantitat d'electricitat invertida en mantenir l'ordinador, es calcula a partir del numero d'hores de funcionament de la CPU (es suposarà que en cap moment està en standby) multiplicat pel numero de kW que es consumeixen cada hora, sumat al cost energètic de mantenir la pantalla de l'ordinador encesa.

Cost CPU: $6 \times 50,56 / 1000 = 0,30336$ kWh (una mitjana de 6 hores diàries de funcionament)

Cost Pantalla $6 \times 18,84 / 1000 = 0,11304$ kWh (per les 6 hores diàries de funcionament)

kWh total = $(0,30336 + 0,11304) * 80$ dies laborals = 33,312 kWh en tot el temps de projecte

Cost Total = $0,21€ * 33,312 = 6,99€$ (aprox. 7€)

Servei/Producte	Cost (€) els 4 mesos del projecte
Personal	4.800
Software	0, els tres primers mesos
Hardware	205,50
Electricitat	7,00
ADSL	21,6
TOTAL	5.034,10 €

Figura 3.3. Taula cost econòmic

4. Elecció de l'entorn de desenvolupament

4.1. Introducció

Quan es va començar amb el projecte, el centre va donar a escollir l'entorn de desenvolupament que trobés més adient, tenint en compte l'objectiu i sabent que el programa havia d'interactuar amb una base de dades MySQL.

El centre va proporcionar una còpia de l'script de creació de la base de dades que podia fer servir per treballar de forma local i un programa (GP Untis) que fan servir per administrar els horaris i que em servís d'orientació per entendre la BBDD.

Llavors per començar a treballar s'havia de triar l'entorn de desenvolupament per programar el software i una eina per administrar la base de dades, i familiaritzar-se amb la informació del GP Untis i saber quina era la informació que interessava extreure de la base de dades.

4.2. Entorn de desenvolupament Java

A l'hora d'escollir l'entorn de desenvolupament, de les opcions que hi havia, Java era la opció més clara, ja que estava familiaritzat amb l'entorn i sabia que ofereix una gran quantitat de facilitats que em permetrien assolir l'objectiu del treball.

Per treballar amb Java l'eina que es faria servir seria el Netbeans (Figura I).

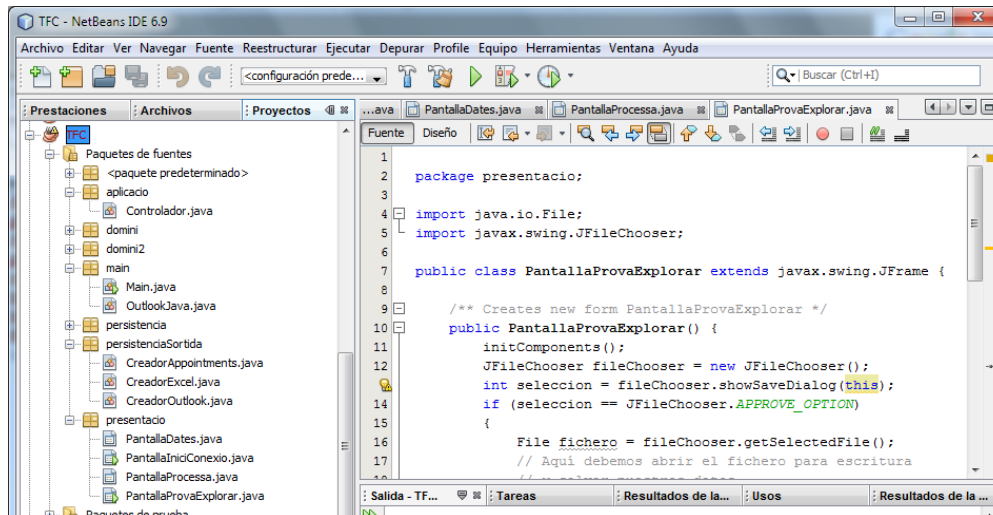


Figura 4.1. Entorn de desenvolupament Java Netbeans

Les aplicacions Java en Netbeans s'organitzen com a projectes. Cada projecte es pot dividir en:

- Paquets de fonts
- Paquets de prova
- Biblioteques/Llibreries
- Biblioteques de proves

El diferent codi i les pantalles que es dissenyin per l'aplicació es guardaran dins el "paquet de fonts". Dins aquest paquet es poden crear carpetes que ajudaran a organitzar el codi i els diferents arxius Java. Els arxius dins un paquet dins el "paquet de fonts" no podran interactuar amb altres arxius dins altres paquets si no s'especifica que es vol tenir accés a un paquet extern amb la paraula reservada Import i el nom del paquet (seguit d'un punt i el nom de la classe en concret si es desitja).

Dels diferents tipus d'arxius que es poden crear, el treball es focalitza en només dos: la classe Java, que conté únicament codi Java per realitzar una tasca, i els formularis JFrame, finestres d'aplicacions amb l'entorn gràfic.

4.3. PhpMyAdmin

A l'hora d'escollir una eina per administrar la base de dades havia d'escollir la que permetés de forma ràpida i senzilla accedir a la base de dades i treballar amb ella. Les dues

eines amb les que ja havia treballat anteriorment i que em permetien fer el que necessitava eren el *MySQL Workbench* (Figura II) i *phpMyAdmin* (Figura III).

Tot i que MySQL Workbench ofereix una interfície gràfica més usable que phpMyAdmin, els meus coneixements de l'eina era bastant limitats i aquesta elecció no afectaria a la forma de treballar de Netbeans amb la base de dades, únicament es faria servir per saber on havia de buscar la informació. A més, estava més familiaritzat amb l'eina navegador de phpMyAdmin per treballs anteriors.

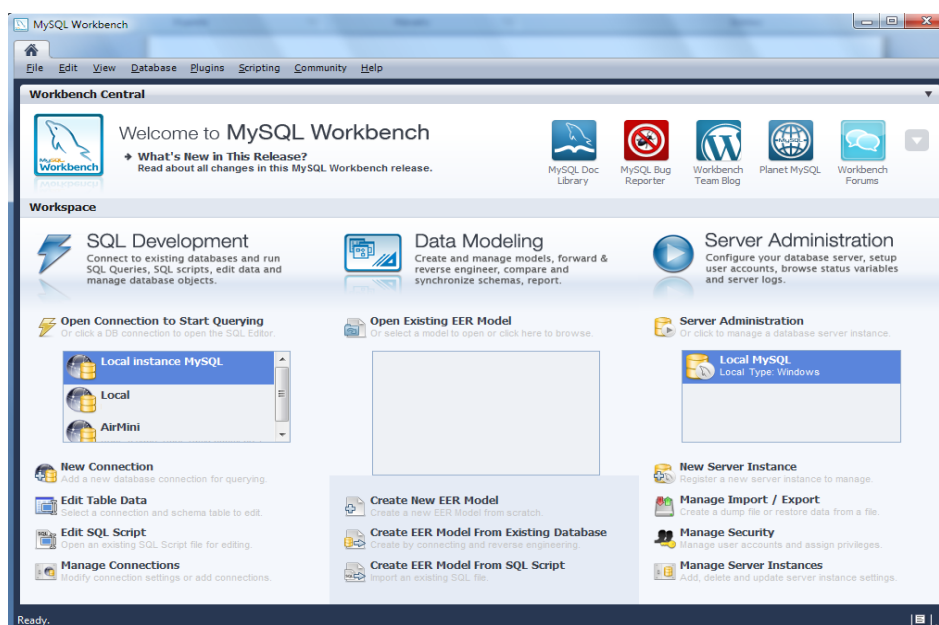


Figura 4.2. Eina per l'administració de BBDD MySQL, MySQL Workbench

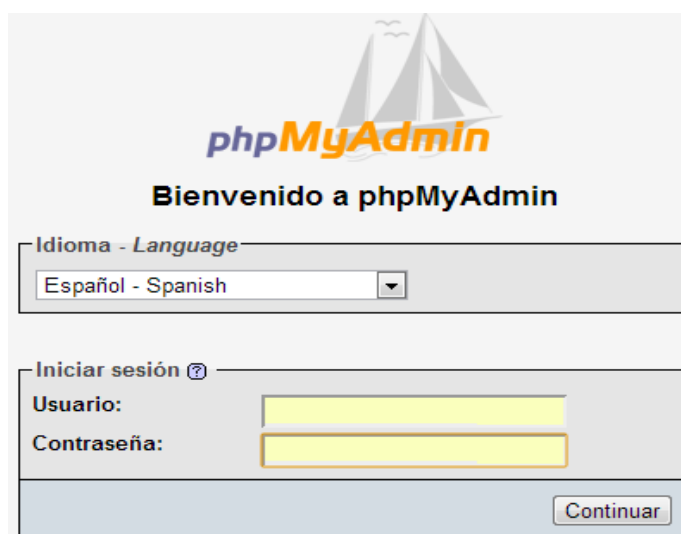


Figura 4.3. Eina des de el navegador, phpMyAdmin

PhpMyAdmin em va permetre importar fàcilment l' script de creació de BBDD i crear una còpia de la BBDD original. El resultat van ser quaranta quatre taules, la majoria d'elles buides, de les quals només necessitava cinc (Externelement, Externtime, School, Subjects, Teacher).

Cal comentar que aquesta base de dades era incomprendible, no existia cap relació entre les taules i estava mal dissenyada, bé per que es va dissenyar així de bon començament, o bé per que l'exportació des de la base de dades no va ser del tot correcte. El cas és que un treball senzill de no massa estona com seria el d'interpretar la base de dades, es va convertir en un complicat treball de comprensió d'uns quants dies i proves amb el GP Untis per saber si la informació era correcta.

4.4. Visual Basic

Un objectiu alternatiu al desenvolupament del software era la formació en Visual Basic. Això és degut a que per assolir els objectius del projecte, es necessitava programar una macro que enviés de forma automàtica la informació generada o obtinguda amb Java a Outlook, si Java mateix no ho permetia.

La forma en que es tenia planejat generar la informació obtinguda en Java, i el destí d'aquesta informació (Outlook), van fer que Visual Basic fos un candidat ferma com a complementari de Java Netbeans, ja que permet interactuar amb les diferents eines de Microsoft Office de forma senzilla i eficaç, a més de tenir un entorn gràfic i un llenguatge semblant al de Java Netbeans.

Per permetre que Visual Basic treballi amb les diferents eines *Office*, s'ha de fer "referència" a elles dins el projecte (a Netbeans seria com afegir llibreries). Per fer referència a una eina, per exemple Outlook, s'ha d'estar treballant amb un projecte VB i fer clic a "projecte", "afegir referència", i dins el llistat d'opcions buscar les referents a Outlook.

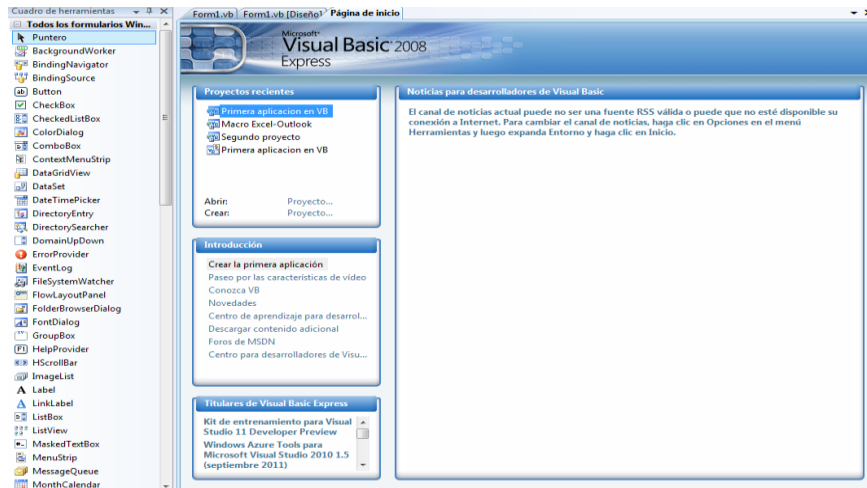


Figura 4.4. Entorn Visual Basic, pàgina d'inici

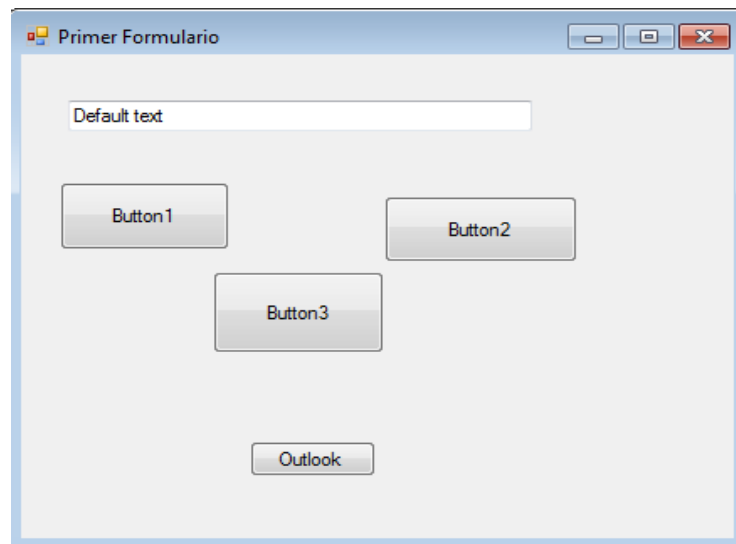


Figura 4.5. Entorn Visual Basic, interfície gràfica

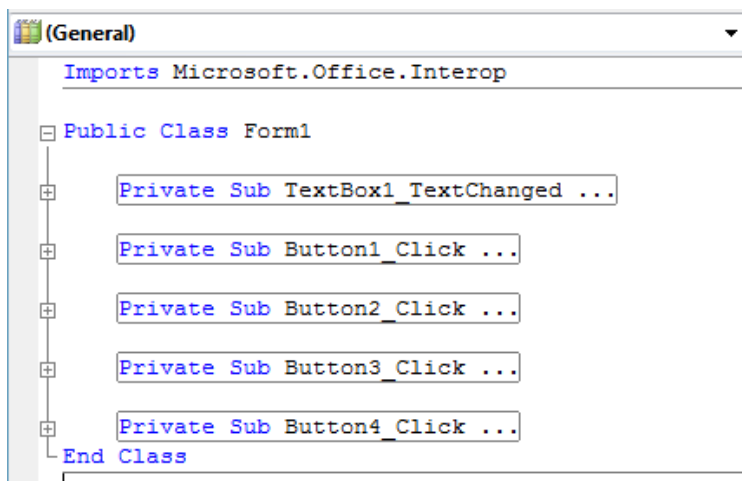


Figura 4.6. Entorn Visual Basic, entorn de desenvolupament

4.5. Gantt Project

És un programa de software lliure basat en Java, compatible amb Windows, Linux i Max OS X, que permet elaborar planificacions de projectes separant-lo en tasques o fer diagrames, entre d'altres.

Abans de conèixer aquest programa s'hauria fet servir Microsoft Project del paquet office per fer el diagrama de Gantt de la planificació del projecte, però no es disposava d'aquest al lloc de treball i instal·lar-ho donava problemes i buscant es va saber d'aquest programa molt útil per fer diagrames i que era perfecte pel projecte.

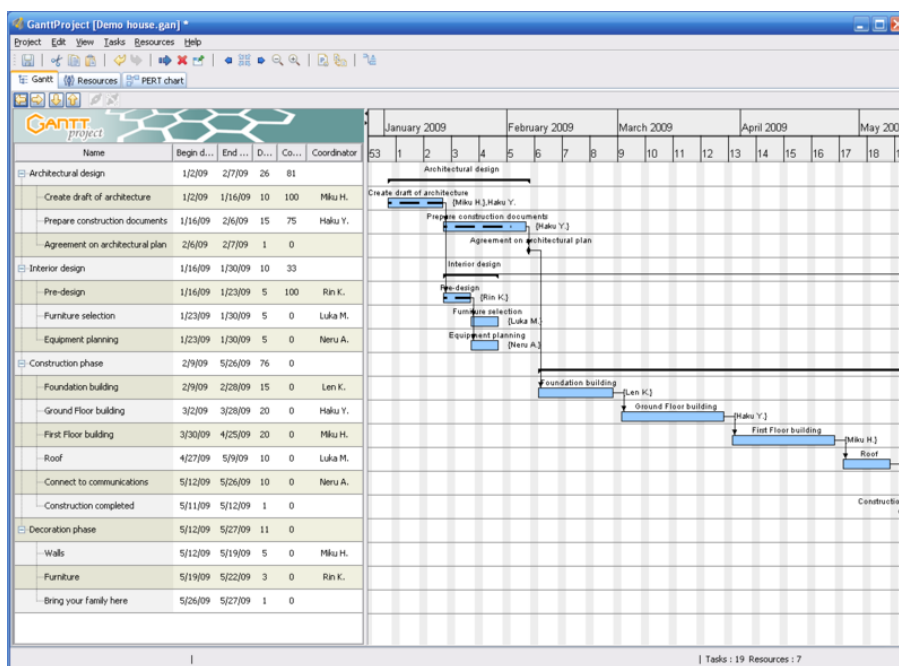


Figura 4.7. Exemple de Diagrama de Gantt amb Gantt Project

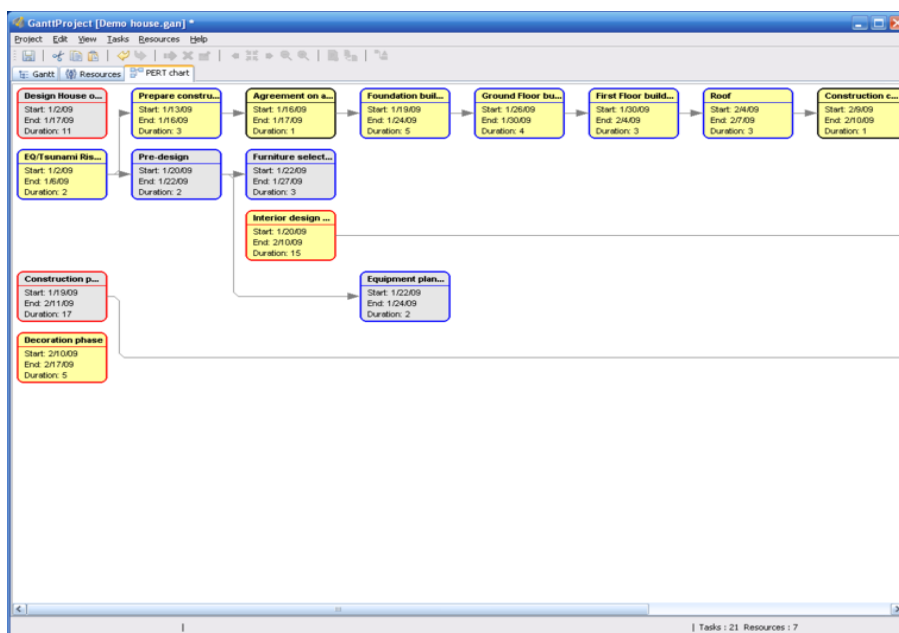


Figura 4.8. Exemple de diagrama de PERT amb Gantt Project

5. Desenvolupament amb Netbeans

5.1. Introducció

En aquest apartat s'acabarà de comentar algunes de les funcionalitats de Java Netbeans amb relació amb el projecte i es farà menció d'alguns mètodes o patrons usats per la part pràctica d'aquest projecte, que correspon a la codificació en Java.

5.2. Control d'errors

Una de les moltes funcionalitats que ofereix Java, i sense dubte una de les més útils és la de captura d'errors o "excepcions".

És molt habitual quan es programa en Java, sobretot si es tracta de projectes grans amb una gran quantitat de codi, que es produeixin dades inesperades o errors. Hi ha casos en que ja s'esperen errors (també coneguts com excepcions), i quan es produeixen s'espera que el programa respongui d'una determinada manera. Per aquests casos Java ofereix el que es coneix com Try-Catch, una eina que permet capturar aquestes excepcions i respondre d'una determinada manera.

```
Try{  
  
    (Aquí aniria la part del codi que pot produir l'error)  
  
}catch(Exception e){  
  
    (Aquí la resposta davant l'error produït)  
  
}
```

El paràmetre entre parèntesis després de la paraula reservada 'catch' indica el tipus d'excepció que es recollirà si es dona l'error. Existeixen molts tipus d'errors, que es poden capturar afegint un catch per cada tipus d'error (si es vol actuar de manera diferent davant cadascun d'ells) abans del de tipus Exception, que recull la resta d'errors que es produeixin.

Una resposta normal davant un error seria intentar reparar aquest error i informar a l'usuari de que s'ha produït i el motiu d'aquest.

5.3. Control de sincronia

Quan en un programa hi han diversos fils corrent simultàniament és possible que varis fils intentin accedir al mateix temps a un mateix lloc (un fitxer, una connexió, un array de dades...) i es possible que s'acabin entorpint entre ells. Per evitar que això passi, Java ofereix un mètode que permet sincronitzar els fils mitjançant la paraula clau *'synchronized'*.

Cal comentar que el mètode de sincronització fa referència a un objecte, és a dir, si hi ha més d'un mètode *synchronized* que fa referència a diferents objectes (per molt que ambdós facin servir el mateix nom de variable), l'execució d'ambdós no serà mútuament exclouent.

```
public synchronized static ConnexioBBDD getInstancia() throws Exception {
    if (instancia == null)
        instancia = new ConnexioBBDD();
    return instancia;
}
```

Figura 5.1. Exemple de sincronia en Java

5.4. Patrons de disseny

Un patró de disseny és una solució a un problema de disseny, la efectivitat del qual ha de haver estat comprovada i ha de ser reutilitzable, és a dir, ha de poder fer-se servir com a solució a diferents problemes de disseny en diferents circumstàncies.

Es va començar a implementar cada cop més l'ús de patrons de disseny a partir de la publicació del llibre *'Design Patterns'* pel grup *Gang of Four*, on es recollien els patrons de disseny més comuns, els quals es poden dividir en: patrons de creació, estructurals i de comportament.

Alguns dels patrons més comuns i que es fan servir en aquest projecte són:

- Patró Singleton: Assegura que una classe només tingui una instància d'ella, i proporciona un accés global a ella.
- Patró Adaptador: Permet cooperació entre classes que d'altra manera no podrien per tenir interfícies incompatibles.
- Patró Façana: Proporciona una classe/interfície única per un conjunt de classes fent que sigui més mantenible.

Un altre patró de disseny molt útil a l'hora d'organitzar els elements és el patró arquitectònic per capes, l'objectiu del qual és ordenar de forma lògica les diferents classes, minimitzant al màxim acoblaments innecessaris entre classes.

Una ordenació lògica (i que segueix aquest projecte) seria:

- Capa Presentació: on es troben totes aquelles classes o JFrames, que presenta el sistema a l'usuari i li transmet o captura informació.
- Capa Domini: és on es defineixen les dades dels diferents objectes del sistema.
- Capa Persistència: encarregada d'accedir a les dades i d'emmagatzemar-les. També pot ser de sortida, que s'encarrega de que la informació persisteixi fora del sistema.
- Capa Aplicació: s'encarrega de rebre informació i enviar una resposta a la capa presentació després de processar-la.

Habitualment la capa domini i capa persistència fan una sola capa anomenada capa de dades.

5.5. JDBC vs JPA

Habitualment quan es treballa en projectes que han d'accedir a una BBDD un es pot trobar davant la decisió de quina eina fer servir per fer les diferents consultes i manipulacions sobre la base de dades.

El millor que es pot fer és saber des d'un bon principi amb quin tipus d'aplicació es treballarà, i les diferents eines que es poden trobar. A continuació s'explica per sobre en que consisteix cadascuna per fer-se una idea quines opcions es poden descartar de bon principi.

- JPA: manera de fer la persistència Java sense acoblar-se amb Hibernate, TopLink, etc. Orientat a objectes, permet usar objectes regulars (POJOs).
- Hibernate: Una bona opció si tens un model objectual - relacional a mapejar.
- JDBC: Tota la persistència Java s'ha de codificar. Nivell més baix. Senzill però útil per consultes.
- DAO: Més patró que tecnologia. Subministra una interfície comuna entre la aplicació i una o més bases de dades. Interfície d'operacions CRUD.
- iBatis: A mig camí entre JDBC i Hibernate. Associa JavaBeans amb sentències SQL o procediments guardats en arxius XML, simplificant l'ús de la BBDD.
- JDO: Java Data Objects. Transparència dels serveis de persistència pel model de domini.

Dit això, les dues opcions principals a considerar per treballar el projecte van ser JPA i JDBC (també influeix el fet que eren eines amb les que ja s'havia treballat amb anterioritat i s'estalviaria un temps invertit en el possible aprenentatge d'un ús correcte de la mateixa).

Sabent que les dues opcions són aquestes, molta gent quan comença es pregunta, llavors què és millor: JPA o JDBC? Depèn.

Millor per a què? Les coses no són millors o pitjors fora d'un context, JPA o JDBC poden ser millor o pitjor que l'altre opció en relació a una circumstància:

Si la aplicació és del tipus OLTP, interactiva (per exemple destinada a gestionar transaccions), amb un ús que pot aprofitar coses com el cache multi nivell de Hibernate i has de poder instal·lar-la a diferents bases de dades, llavors millor JPA.

Si per altra banda la aplicació va més relacionada a empresa, o consultes de grans quantitats de dades, és a dir del tipus OLAP, mostrant informació només de lectura o per casos en que l'aplicació corre sobre una base de dades en particular, llavors millor JDBC.

Comentar també que els canvis que es facin en els objectes JPA poden afectar directament a la informació de la BBDD, aquest és un altre bon motiu pel que es pot triar JDBC en comptes de JPA. A més, fent proves en el projecte es va demostrar que per l'aplicació que s'estava treballant era més rendible si es feia servir JDBC en comptes de JPA.

Des del punt de vista del programador, JPA facilita molt la codificació, estalviant feina a l'hora de programar les diferents classes entitat o de persistència.

Per començar a treballar amb la base de dades, JPA necessita d'un arxiu de configuració XML (creat de forma automàtica si es crea una unitat de persistència que faci referència a la BBDD), que contindrà la informació referent a la base de dades a la que es fa referència, que a diferència de JDBC, no necessita d'aquest arxiu .XML però la configuració de connexió a la base de dades ha de ser codificada manualment.

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence-unit name="TFCPsthg" transaction-type="RESOURCE_LOCAL">
  <provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>
  <properties>
    <property name="javax.persistence.jdbc.url" value="jdbc:mysql://localhost:3306/tfc"/>
    <property name="javax.persistence.jdbc.password" value=":"/>
    <property name="javax.persistence.jdbc.driver" value="com.mysql.jdbc.Driver"/>
    <property name="javax.persistence.jdbc.user" value=":"/>
    <property name="eclipselink.ddl-generation" value="create-tables"/>
  </properties>
</persistence-unit>
```

Figura 5.2. Unitat de persistència.xml

Un altre motiu pel que JPA facilita el treball de codificació és el fet que les classes entitat (del paquet domini) i les classes persistència es poden crear automàticament. Les classes entitat es creen a partir de les taules de la base de dades, i tenen ja definides les principals consultes SQL, i les classes persistència es creen a partir de les classes entitat. Tot això estalvia temps de codificació, però igual que en JDBC, es pot afegir codi personalitzat.

El fet que moltes classes en JPA es creïn de forma automàtica fa que molt del codi no es faci servir, i obliga al programador a fer neteja de tot el codi innecessari, que si algun dia un altre programador ha de mantenir aquest aplicatiu, agrairà que tot el que hi hagi sigui per que es fa servir.

5.6. Lliberies

Aquest apartat si s'expliqués amb detall podria ser massa extens, ja que de lliberies ni ha moltes i existeix moltíssima informació al respecte, pel que només es farà una breu

explicació de com funcionen les llibreries i s'explicarà quines s'han fet servir en aquest projecte.

Les llibreries són una part fonamental en qualsevol projecte, no només en Java, si no en la majoria d'eines de programació, ja que ofereixen una quantitat de codi que simplifica la tasca de programació.

La plataforma de Java està pensada per ser independent del SO, per lo qual les aplicacions no poden recolzar-se en funcions depenent de cada sistema en concret. Les llibreries estàndard de Java contenen moltes de les funcions reutilitzables disponibles en els SO actuals.

Aquestes llibreries tenen tres propòsits dins l'entorn Java. Al igual que altres llibreries estàndard, ofereixen al programador un conjunt ben definit de funcions per realitzar diferents tasques habituals, com per exemple fer servir llistes d'elements o operar de forma sofisticada sobre cadenes de caràcters. Tasques tals com l'ús de funcions de xarxa o accés a fitxers externs, solen ser molt dependents de la funcionalitat de la plataforma destí.

Tot i això, no totes les plataformes suporten totes les funcions que una aplicació Java podria esperar. En aquests casos les llibreries poden 'emular' aquestes funcions fent servir el que estigui disponible o bé comprovar si la funcionalitat existeix.

Com ja s'ha explicat anteriorment a l'entorn de desenvolupament Java, Netbeans conté per cada projecte un paquet dedicat a les llibreries. A continuació es mostren les llibreries utilitzades en aquest projecte (Figura III).

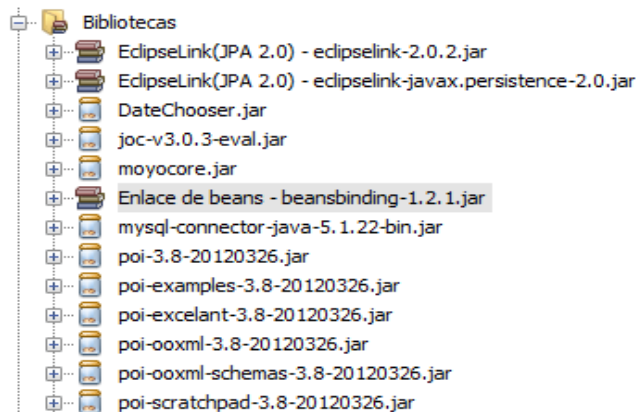


Figura 5.3. Llibreries del projecte final de carrera

Algunes llibreries es creen de forma automàtica quan es crea un tipus d'arxiu en concret, com és el cas de les llibreries EclipseLinks (JPA), que s'afegeixen automàticament quan es crea una unitat de persistència XML per implementar l'accés amb JPA. La raó per la qual està aquesta llibreria és que en un principi es va fer servir JPA com a eina per treballar amb la BBDD, però per raons de rendibilitat i recomanació del tutor docent es va acabar adaptant el projecte a JDBC.

Després hi ha altres llibreries que són necessàries per funcionalitats que s'han afegit al programa.

- DateChooser: llibreria que afecta al JFrame, afegeix eines a la paleta de components, tals com un calendari, que permet la selecció de dates a fi de treballar amb elles i permetre el filtrat de la informació per dates.
- Joc-v3.0.3-eval.jar + moyocore.jar: es tracta de llibreries de prova que ofereix el grup moyosoft, que permeten a l'aplicació Java interactuar amb Outlook. Algunes de les funcionalitats que ofereix són: gestionar contactes, gestionar calendari (cites, etc.), Acompanyant a la memòria del projecte hi ha uns arxius d'aquesta API on hi ha més informació al respecte, juntament a la web oficial del grup.
- Mysql-connector-Java-5.1.22-bin.jar: Llibreria que permet accedir a la base de dades.
- Llibreries POI: permeten al software Java interactuar amb Excel, permetent així crear un arxiu Excel amb la informació obtinguda de la BBDD.

API Java-Outlook	
És gratuït?	Està disponible una versió gratuïta de l'API vàlida fins a 3 mesos, passat aquest període s'ha de pagar si es vol continuar fent ús d'aquesta API.
Es proporciona exemples d'ús?	Sí. Junt amb la memòria es proporciona una sèrie de documents i arxius, entre ells exemples d'ús de l'API.
Quines versions Outlook i Java suporta?	El connector és compatible amb les versions d'Outlook 2000, 2002, 2003, XP, 2007 i 2010. Pel que fa a Java, es necessita la versió 1.4 o posterior.
Proporciona control d'errors?	Sí. Les llibreries inclouen llançadors d'excepcions per determinar el bon funcionament de les funcions que aquestes ofereixen.

Figura 5.4. Esquema resum amb informació de l'API Java-Outlook.

POI Java-Excel	
És gratuït?	Completament gratuït. No existeix versió de prova.
Es proporciona exemples d'ús?	Sí. Junt amb la memòria es proporciona una sèrie de documents i arxius, entre ells exemples d'ús de l'API.
Quines versions Outlook i Java suporta?	El connector ofereix compatibilitat amb les versions d'Excel des de la 97 fins a la 2007+.
Proporciona control d'errors?	Sí. Les llibreries inclouen llançadors d'excepcions per determinar el bon funcionament de les funcions que aquestes ofereixen.

Figura 5.5. Esquema resum amb informació de l'API Java-Excel.

6. Evolució del projecte

6.1 Introducció

És important dedicar un petit apartat a l'evolució del projecte, per tal de deixar constància de que s'ha produït aquesta evolució, no només pel que fa al codi sinó també al disseny gràfic, que de bon principi es va pensar com una cosa molt senzilla, i que ha anat evolucionant per les necessitats d'implementació de noves funcionalitats i requeriments.

Al principi es va pensar el projecte com una aplicació senzilla, amb poc més que una pantalla, per escollir la informació que es volia i un botó per acceptar. A mesura que s'anava avançant s'anaven afegint funcionalitats i es veia com l'espai de la pantalla que s'havia pensat de bon començament anava quedant insuficient. Es va acabar dividint l'aplicació en tres pantalles (o JFrames) per aconseguir ficar tot el que es desitjava.

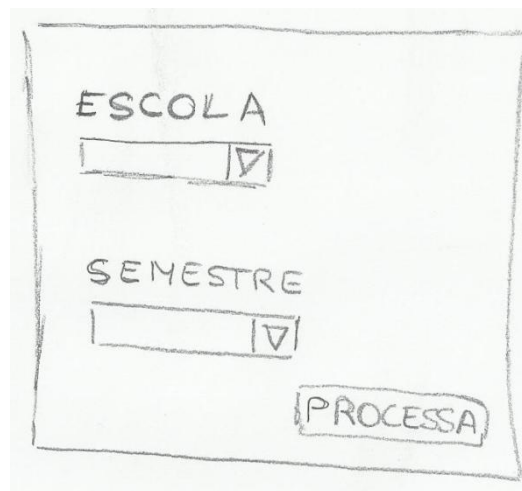


Figura 6.1. Esborrany del disseny inicial de pantalla única

Pel que fa al codi, abans d'implementar JPA per tractar la base de dades, van sorgir alguns inconvenients que van fer impossible treballar amb la base de dades, i es va implementar un patró adaptador per simular que accedim a la base de dades però en realitat es treballaria amb informació dins del propi programa. Això es va fer per no deixar el projecte parat i començar a codificar la part controladora i les pantalles mentre es trobava una solució al problema (que finalment no va ser res més que una dificultat a l'hora d'interpretar correctament la base de dades).

Un cop solucionat l'error d'interpretació, es va començar a treballar la base de dades mitjançant JPA, on les classes entitat i la persistència s'havien de codificar poc, pràcticament es feien de forma automàtica, les dades d'accés a la BBDD es trobaven en la unitat de persistència XML i només s'havia de codificar la part controladora, corresponent a la capa Aplicació.

Donat que, pel que fa a la BBDD, el software no requereix més que llegir la informació, es va pensar que la millor opció era treballar amb JDBC, això milloraria la rendibilitat considerablement, a més d'evitar alguns errors que es donaven a l'hora d'agafar la informació amb els mètodes creats de forma dinàmica amb JPA. També es va decidir canviar a JDBC per motius de seguretat, no fos que per accident o descuit es tractés un objecte de la base de dades obtingut amb JPA, es canviés la informació i afectés a la base de dades real.

Aconseguir l'objectiu d'obtenir correctament la informació de base de dades, quedava pensar la forma d'enviar-la a Outlook. Es van pensar dues possibilitats: trobar la forma d'enviar directament des de Java Netbeans la informació a Outlook, o crear un arxiu extern amb la informació extreta i després amb una macro enviar aquesta informació a Outlook.

El fet que s'haguessin proposat dues formes d'enviar la informació a Outlook (i totes dues necessàries), va fer que les pantalles evolucionessin per oferir la possibilitat a l'usuari d'escollir el mètode de processament (enviar dades).

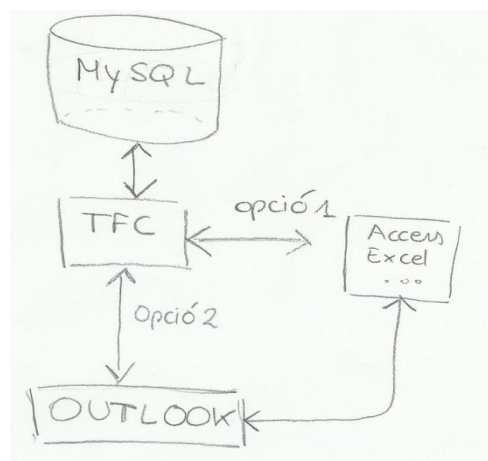


Figura 6.2. Esborrany de l'esquema de processament

Finalment es va trobar la manera d'enviar la informació directament des de Java a Outlook amb l'API de moyosoft (mètode 2), i de crear l'arxiu extern (.xls) amb la informació de BBDD per ser enviada posteriorment a Outlook fent una crida a l'execució d'un software codificat en Visual Basic encarregat de llegir la informació i enviar-la (mètode 1).

7. Anàlisi i disseny de l'aplicació

7.1 Introducció

L'anàlisi i disseny de l'aplicació és una part fonamental en tot projecte, però en el cas concret d'aquest projecte té un pes més important l'anàlisi que no el disseny. Al tractar-se d'una aplicació que probablement es farà servir un cop o dos al començament de cada semestre i prou, no és tant important l'aspecte visual d'aquesta. Per altra banda, haver analitzat prèviament els requeriments inicials i tenir clar l'objectiu d'aquest software ajudarà a fer les coses bé de primeres, minimitzant al màxim el temps que es podria perdre tirant enrere o modificant l'estructura del codi.

7.2. Requeriments

A continuació s'exposen una sèrie de requeriments a complir dins el projecte per a la satisfacció del client. Entre tots els requeriments, n'hi ha de vitals per al bon funcionament del software, i n'hi ha alguns que més que necessaris són "millores" o modificacions a gust del client. Al llarg del procés d'elaboració de l'aplicació han anat sorgint nous requeriments que en un principi no s'havien trobat o no es creien necessaris.

7.2.1. Requeriments funcionals

- S'ha de permetre en tot moment tirar enrere dins l'aplicatiu.
- S'ha de poder seleccionar l'origen de dades, és a dir, on anirà a buscar la informació el programa.
- S'ha de poder escollir el mètode de processament (amb API Outlook o per creació d'arxiu Excel).
- S'ha de permetre a l'usuari indicar el destí de l'arxiu Excel en cas que esculli aquest mètode.
- S'ha de permetre a l'usuari escollir les versions a processar.
- El sistema enviarà un missatge de "tot correcte" quan s'hagi acabat el processament.
- El sistema informarà de qualsevol error que es produeixi.

7.2.2. Requeriments no funcionals

- El sistema crearà un arxiu .txt amb les incidències produïdes, si se'n produís cap.
- L'aplicació podrà executar-se sense problemes en qualsevol ordinador amb accés a una base de dades amb el mateix nom de taules i atributs necessaris pel software.
- El sistema no ha de mostrar els caràcters de contrasenya al camp password d'accés a base de dades remot.
- A la selecció de dates entre les que filtrarà el calendari, la data 1 ha de ser anterior a la data 2.
- A la selecció de dates entre les que filtrarà el calendari, la diferència entre la data 1 i la data 2 no pot ser superior a dos anys.
- A la selecció de dates entre les que filtrarà el calendari, les dues dates han de tenir algun valor.
- El sistema ha de tenir permisos sobre Outlook, en cas contrari preguntarà en tot moment si es vol enviar les cites.

7.3. Casos d'ús

Els casos d'ús descriuen els passos o activitats a realitzar per dur a terme un procés sense descriure el funcionament intern o disseny.

Seguidament es descriuen els casos d'ús més destacats de l'aplicació amb informació detallada de cada un.

7.3.1. Fitxa de Casos d'ús

Fitxa dels casos d'ús i informació detallada (Objectiu, descripció, Precondició i post condició, flux normal i alternatiu).

Casos d'ús: Seleccionar origen de dades, seleccionar interval de dates, preparar dades mètode 1, preparar dades mètode 2, enviar dades mètode 1, enviar dades mètode 2.

SELBBDD	Seleccionar origen de dades
Objectiu	Connectar amb una BBDD d'on obtenir la informació
Descripció	Per defecte el sistema dona la opció de treballar amb una BBDD local definida al propi programa. L'usuari si vol pot escollir on vol anar a buscar la informació.
Precondició	
Flux Normal	<ol style="list-style-type: none"> 1- L'usuari inicia el programa 2- El sistema mostra la pantalla de selecció d'origen de dades. Per defecte es treballa en local 3- L'usuari escull l'origen de dades. L'usuari pot escollir connectar-se a una BBDD remota 4- Si l'usuari ha escollit connectar-se de forma remota, omplirà els camps (IP, DNS, user, pass). En cas contrari passa directament al punt 5 5- L'usuari prem el boto de continuar 6- El sistema connectarà amb la BBDD i mostrarà la següent pantalla
Postcondició	S'ha connectat a una BBDD, ara es pot obtenir la informació desitjada.
Flux Alternatiu	<ul style="list-style-type: none"> - En tot moment l'usuari pot tancar el programa. - L'usuari pot canviar l'origen de dades en qualsevol moment, sempre i quan no hagi premut el botó de continuar. 6- No es troba l'origen de dades, el sistema mostra un missatge d'error amb el motiu de l'error i torna al punt 3.

Taula 7.1. Cas d'ús Seleccionar origen de dades

A continuació el cas d'ús SELDATES. S'ha extret aquest subcas d'ús amb motiu de que es repeteix en dos casos d'ús més a continuació.

SELDATES	Seleccionar interval de dates
Objectiu	Definir entre quines dates es processarà el calendari
Descripció	Es tracta d'una api per seleccionar dates, que farà la funció de filtrat, per permetre a l'usuari escollir entre quines dates vol omplir el calendari.
Precondició	Que s'hagi executat el cas d'ús <i>SELBBDD</i> , s'ha de tenir accés a una BBDD.
Flux Normal	<p>Els punts 1, 2 i 3 es repetiran per definir la Data 1 i la Data 2.</p> <ol style="list-style-type: none"> 1- L'usuari prem la icona al costat dret de la data definida a Data

	2- El sistema mostra un calendari on l'usuari pot escollir la data que vol 3- L'usuari navega en el calendari i fa doble clic al dia que vol fixar com a data
Postcondició	S'ha escollit un interval de dates pel filtratge del calendari.
Flux Alternatiu	- En tot moment l'usuari pot tancar el programa.

Taula 7.2. Cas d'ús Seleccionar interval de dates

PREPDADESMET1	Preparar les dades fent servir el mètode 1
Objectiu	Deixar llestes les dades per ser enviades
Descripció	Es farà servir el mètode que fa ús d'un arxiu Excel per preparar les dades per ser enviades.
Precondició	Que s'hagi executat el cas d'ús <i>SELBDD</i> , s'ha de tenir accés a una BBDD.
Flux Normal	1- L'usuari executarà el subcas d'ús <i>SELDATES</i> 2- L'usuari escull la opció <i>POI Java - Excel</i> com a mètode per processar les dades 3- L'usuari prem el botó per continuar 4- El sistema mostra la següent pantalla, amb opció d'escollir on vol crear l'arxiu Excel i selecció de versions a processar. 5- L'usuari escull on vol crear l'arxiu Excel. 6- L'usuari selecciona, per cada centre universitari, les versions de les quals vol extreure informació per enviar al calendari Outlook. 7- L'usuari prem el botó per processar les dades. 8- El sistema comença el cas d'ús <i>ENVDADESMET1</i> i informa a l'usuari.
Postcondició	S'ha escollit com es vol processar les dades i seran enviades a Outlook.
Flux Alternatiu	- En tot moment l'usuari pot tancar el programa. 4.1- La Data 1 és superior a la Data 2, el sistema no ho permet, el sistema informa de l'error i es torna al punt 1. 4.2- Alguna o cap de les dues dates ha estat definida, el sistema informa de l'error i es torna al punt 1. 4.3- La diferència de dates entre Data 1 i Data 2 es superior a 2 anys. El sistema no ho permet, informa de l'error i es torna al punt 1. 4.4- La BBDD no conté la informació esperada. El sistema informa de l'error i tanca el programa. 5.1- L'usuari no escull cap destí per l'arxiu Excel o cancel·la. El sistema per defecte crearà l'arxiu a l'arrel del projecte.

Taula 7.3. Cas d'ús Preparar les dades fent servir el mètode 1

PREPDADESMET2	Preparar les dades fent servir el mètode 2
Objectiu	Deixar llestes les dades per ser enviades
Descripció	Es farà servir el mètode que fa ús de l'API Java - Outlook per preparar les dades, per ser enviades.
Precondició	Que s'hagi executat el cas d'ús <i>SELBBDD</i> , s'ha de tenir accés a una BBDD.
Flux Normal	<ol style="list-style-type: none"> 1- L'usuari executarà el subcas d'ús <i>SELDATES</i> 2- L'usuari escull la opció <i>API Java - Outlook</i> com a mètode per processar les dades 3- L'usuari prem el botó per continuar 4- El sistema mostra la següent pantalla, amb opció de seleccionar les versions a processar. 5- L'usuari selecciona, per cada centre universitari, les versions de les quals vol extreure informació per enviar al calendari Outlook. 6- L'usuari prem el botó per processar les dades. 7- El sistema comença el cas d'ús <i>ENVDADESMET2</i>.
Postcondició	S'ha escollit com es vol processar les dades i seran enviades a Outlook.
Flux Alternatiu	<p>- En tot moment l'usuari pot tancar el programa.</p> <ol style="list-style-type: none"> 4.1- La Data 1 és superior a la Data 2, el sistema no ho permet, el sistema informa de l'error i es torna al punt 1. 4.2- Alguna o cap de les dues dates ha estat definida, el sistema informa de l'error i es torna al punt 1. 4.3- La diferència de dates entre Data 1 i Data 2 es superior a 2 anys. El sistema no ho permet, informa de l'error i es torna al punt 1. 4.4- La BBDD no conté la informació esperada. El sistema informa de l'error i tanca el programa.

Taula 7.4. Cas d'ús Preparar les dades fent servir el mètode 2

ENVDADESMET1	Enviar les dades fent servir el mètode 1
Objectiu	Actualitzar el calendari Outlook correctament
Descripció	Es farà servir el mètode que fa ús de l'arxiu Excel i un programa extern codificat en Visual Basic per enviar les dades al calendari Outlook.
Precondició	Que s'hagi executat el cas d'ús <i>PREPDADESMET1</i> .
Flux Normal	<ol style="list-style-type: none"> 1- El sistema crea un arxiu Excel amb la informació a enviar i guarda el path de l'arxiu en un document de text. 2- El sistema executa la macro codificada en Visual Basic. 3- El sistema del programa macro llegeix l'arxiu de text per saber el path de l'arxiu Excel 4- El sistema del programa macro llegeix les dades de l'arxiu Excel, obté la informació que interessa i l'envia a Outlook en forma de cita. 5- El sistema del programa macro informa que el calendari s'ha actualitzat

	correctament.
Postcondició	S'ha processat la informació i ha sigut enviada al calendari Outlook.
Flux Alternatiu	

Taula 7.5. Cas d'ús Enviar les dades fent servir el mètode 1

ENVDADESMET2	Enviar les dades fent servir el mètode 2
Objectiu	Actualitzar el calendari Outlook correctament
Descripció	Es farà servir el mètode que fa ús de l'API Java - Outlook per enviar les dades al calendari Outlook.
Precondició	Que s'hagi executat el cas d'ús PREPDADESMET2.
Flux Normal	<p>1- El sistema agafa les dades recollides i filtra la informació que interessa per crear les cites.</p> <p>2- El sistema crea la cita i l'envia a Outlook (aquest pas es repeteix tantes vegades com cites es creïn).</p> <p>3- El sistema tanca la connexió amb Outlook i informa a l'usuari que el calendari s'ha actualitzat correctament.</p>
Postcondició	S'ha processat la informació i ha sigut enviada al calendari Outlook.
Flux Alternatiu	<p>2.1- No es té dret d'administrador per accedir a Outlook. El sistema pregunta a l'usuari si vol enviar la cita corresponent.</p> <p>2.2- No es troba el calendari corresponent a l'aula. El sistema afegeix les dades a un arxiu Excel que es crearà a l'arrel del projecte.</p>

Taula 7.6. Cas d'ús Enviar les dades fent servir el mètode 2

7.3.2. Diagrama de Casos d'ús

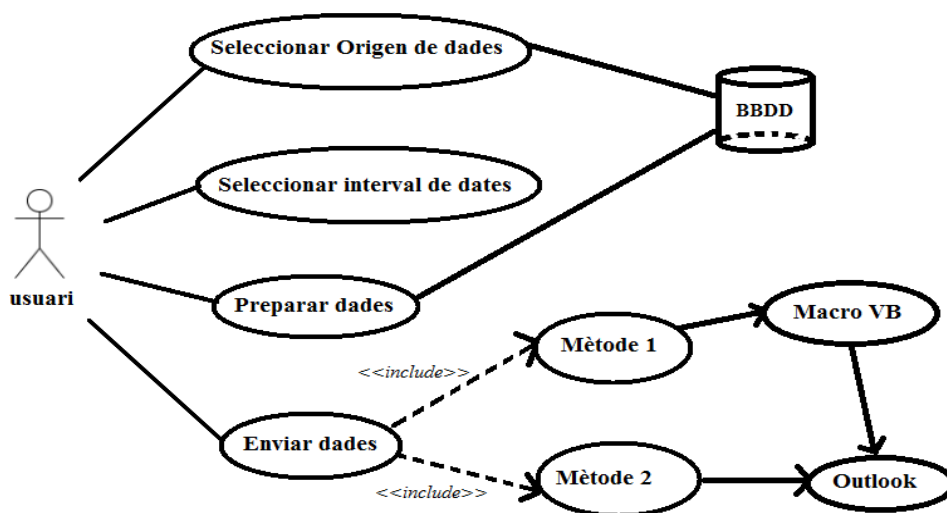


Figura 7.1. Diagrama casos d'ús

7.3.3. Diagrama de classes

Paquet Domini



Figura 7.2. Diagrama de classe del paquet Domini

Paquet Aplicació

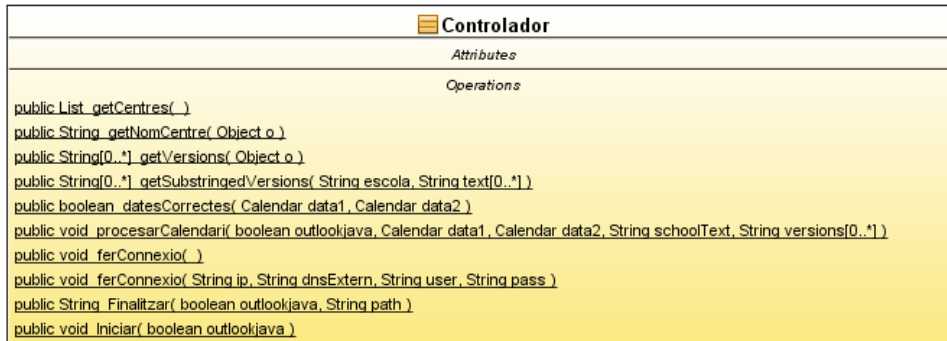


Figura 7.3. Diagrama de classe del paquet Aplicació

Paquet Presentació

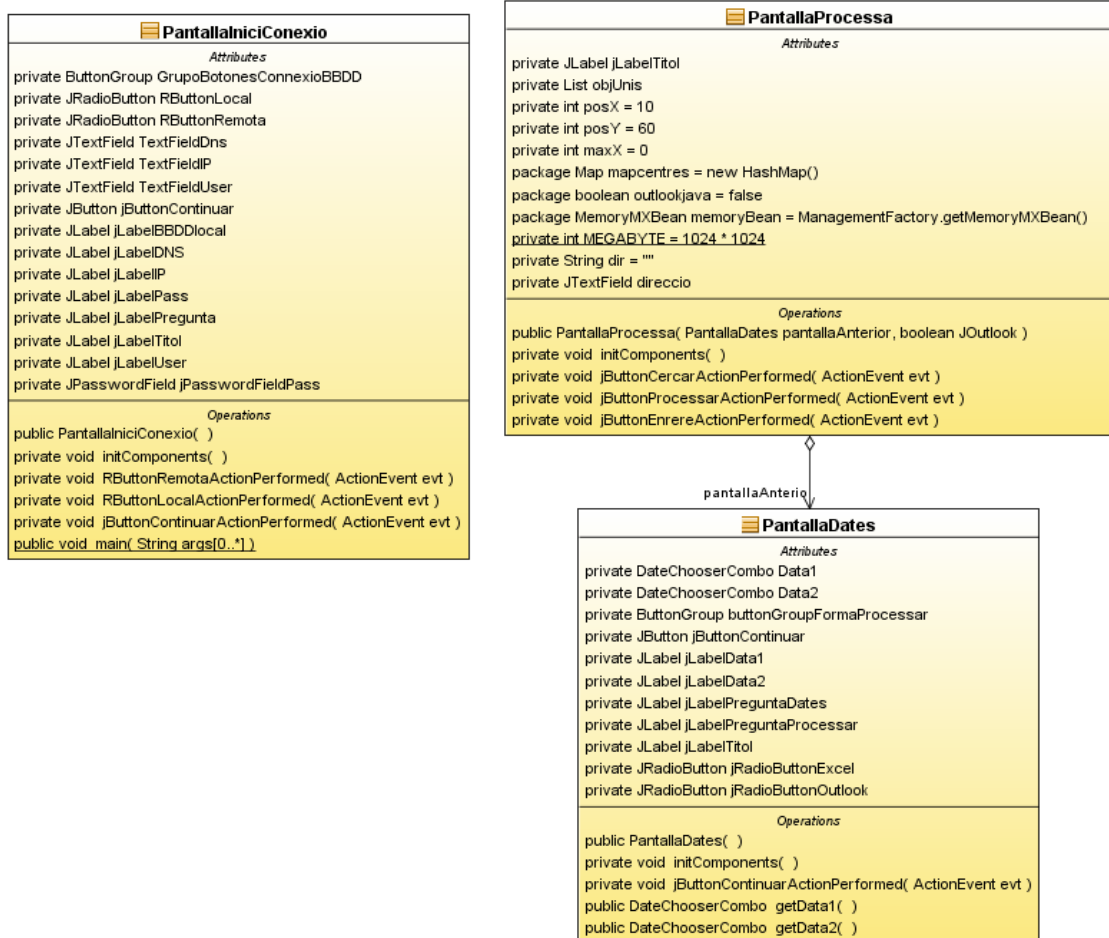


Figura 7.4. Diagrama de classe del paquet Presentació

Paquet Persistència

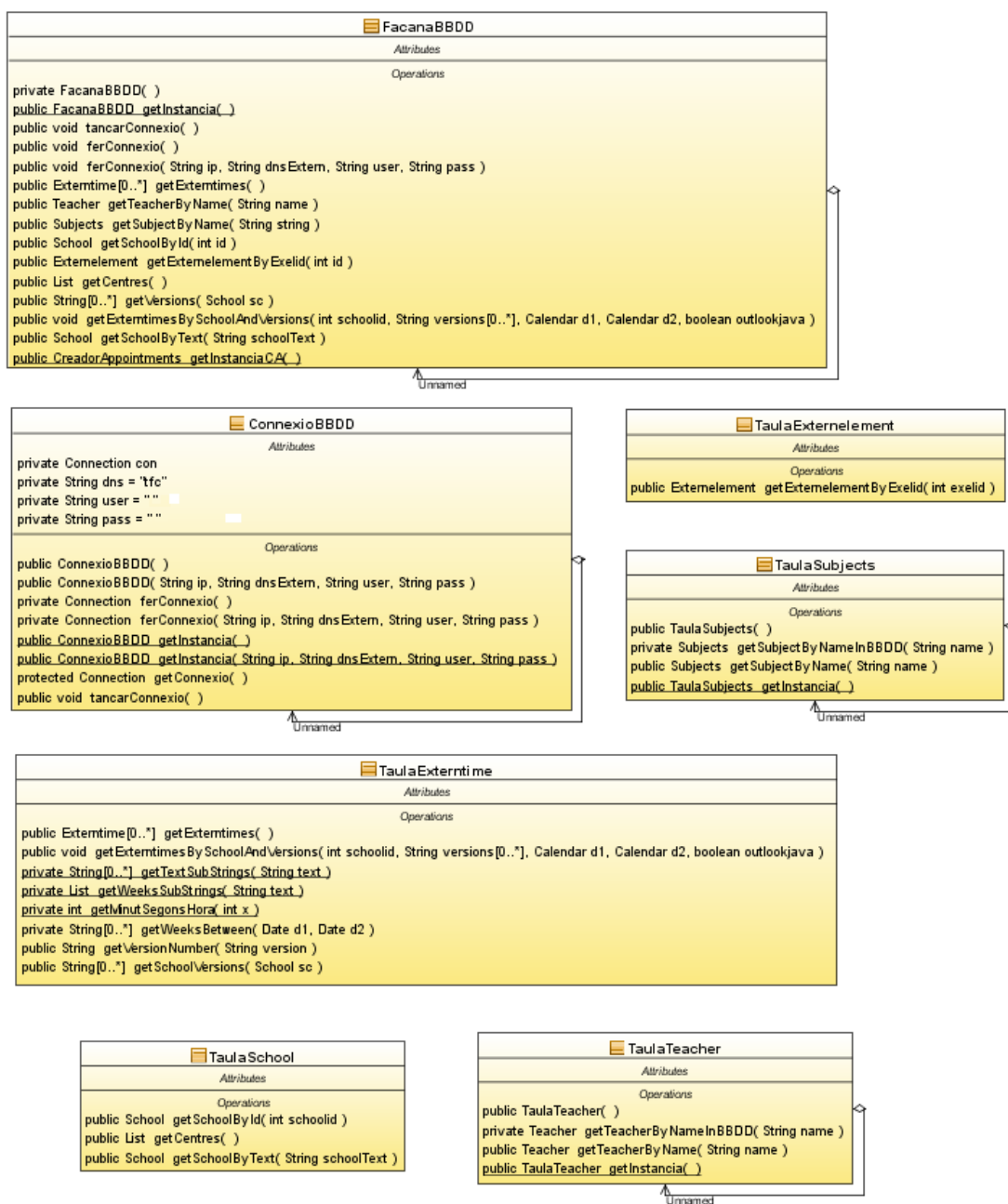


Figura 7.5. Diagrama de classe del paquet Persistència

Paquet Persistència de Sortida

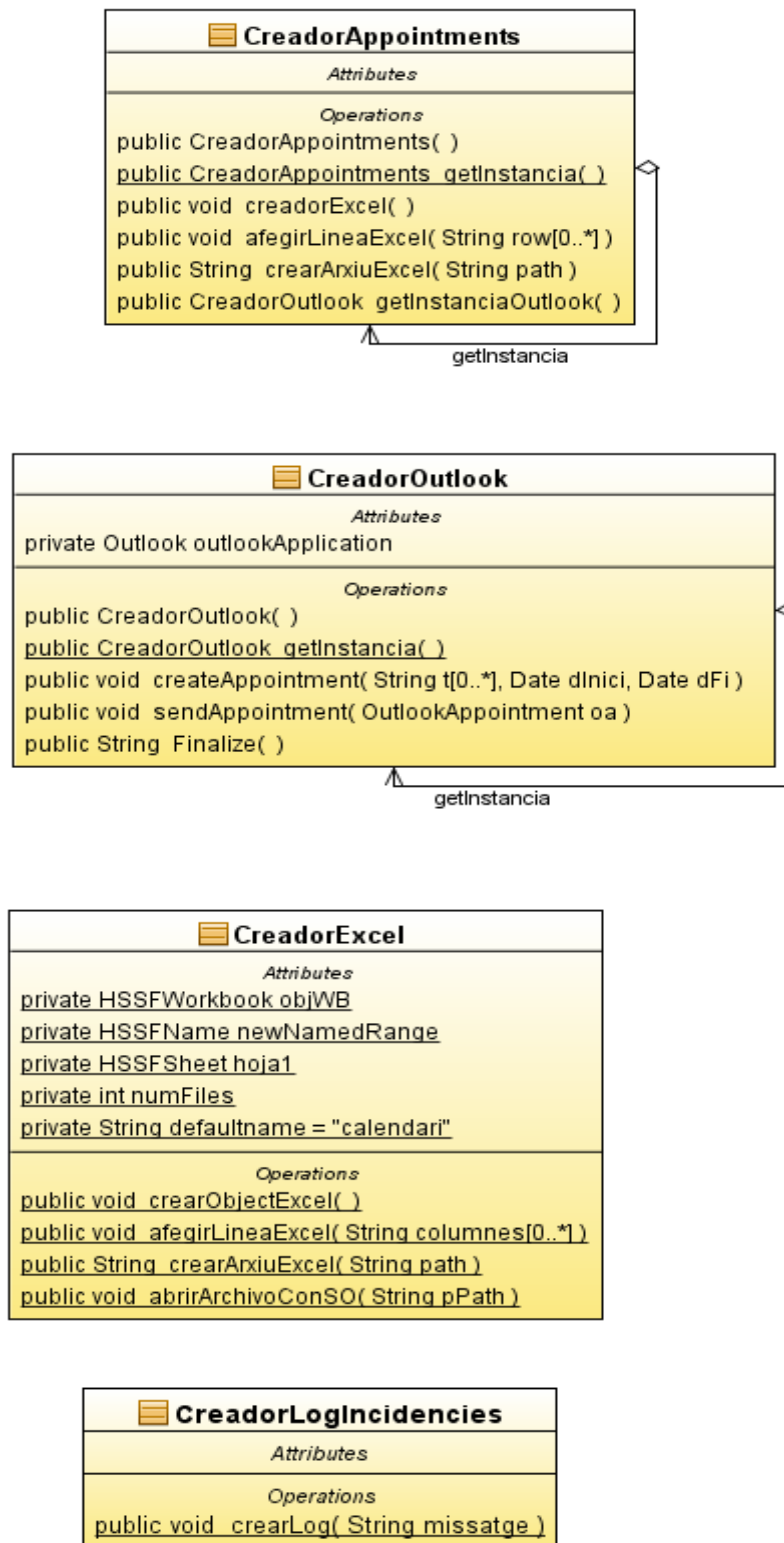


Figura 7.6. Diagrama de classe del paquet Persistència de Sortida

7.3.4. Esquema de BBDD



Figura 7.7. Esquema de la BBDD migrada

Com es pot veure a la imatge (Figura I), a la base de dades facilitada per treballar no existeix cap relació entre taules, fet que va dificultar molt la comprensió de la mateixa en el seu moment. Quaranta quatre taules, totes o pràcticament totes elles amb (la que menys) deu columnes d'atributs, pràcticament totes les claus primàries compostes i claus foranies inexistents.

A continuació un esquema de la BBDD, relacionada per fer-la més entenedora, només amb les taules que s'han fet servir per obtenir la informació al software.

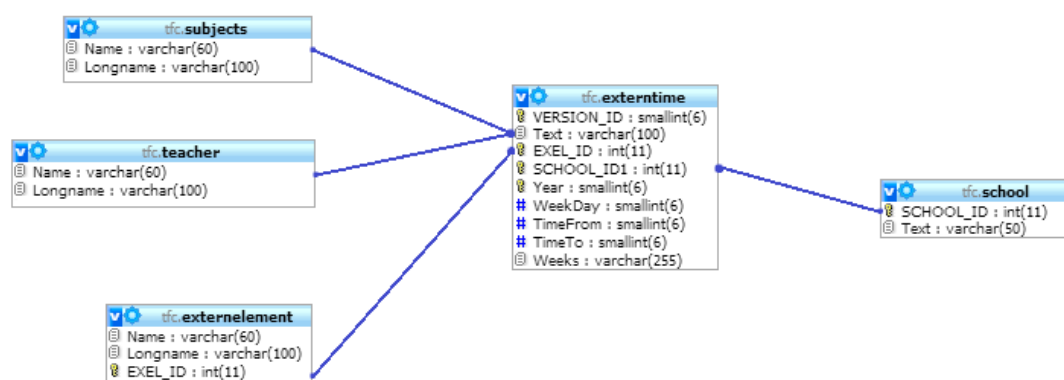


Figura 7.8. Esquema de taules de la BBDD útils pel projecte.

Tal i com mostra la imatge de la Figura II, la informació gira entorn la taula 'externtime', que conté tota la informació o la referència a ella, que necessitem enviar a Outlook.

De l'atribut *text* de la taula 'externtime' surten dues fletxes cap a 'subjects' i 'teachers', això es deu a que l'atribut *text* conté un String format per: *sigles dels estudis i curs + sigles de l'assignatura + sigles del professor*.

Exemple: *GEI 1A,MATI,BLANC* -> fa referència a 'Grau Enginyeria Informàtica 1A', Matemàtiques 1, BLANCH RAMON.

8. Com funciona l'aplicació

8.1. Introducció

L'objectiu d'aquest apartat es mostrar i comentar com funciona el software dissenyat en aquest projecte, les diferents pantalles que conté, etc.

8.2. L'aplicació en Java

8.2.1. Connectant a la Base de dades

Al iniciar l'aplicació la primera pantalla que es mostra es la d'accés a una base de dades. Com es veu a la Figura I, l'usuari pot triar si connectar-se de forma local o si vol accedir a una base de dades remota. Que sigui remota implica que no ha estat definida dins el software, pot ser també local però els paràmetres d'accés no seran els definits dins l'aplicatiu.

Per defecte ve marcada la opció de local, que com es mostra a l'etiqueta connectarà a localhost pel port 3306. No es mostra a l'etiqueta però els paràmetres *usuari* i *contrasenya* són root i espai buit, respectivament.

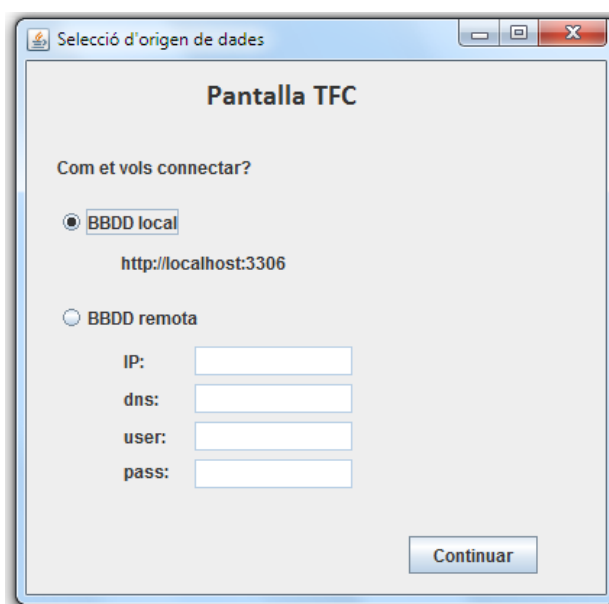


Figura 8.1. Pantalla de connexió a base de dades

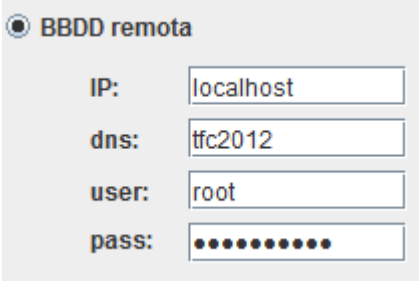
En cas de no voler connectar amb la opció BBDD local, l'usuari haurà de marcar la opció de BBDD remota (automàticament l'altre opció quedarà desmarcada), i haurà d'emplenar les caselles d'IP, DNS, user i pass.

IP: Es tracta de la direcció on es troba la base de dades.

DNS: Fa referència al nom de la base de dades on anirà a buscar la informació

User i Pass: fan referència al nom d'usuari i contrasenya, respectivament, necessaris per accedir a la base de dades.

Exemple:



El formulari mostra l'opció "BBDD remota" seleccionada amb un botó radio. A continuació, hi ha quatre camps de text etiquetats: "IP:" amb el valor "localhost", "dns:" amb el valor "tfc2012", "user:" amb el valor "root", i "pass:" amb una sèrie de punts ocults.

Figura 8.2. Exemple d'accés a base de dades

En aquest cas, també s'accedirà de forma local (localhost), però amb els paràmetres desitjats. D'aquesta forma si l'usuari i la contrasenya són correctes, s'accedirà a la base de dades 'tfc2012'. D'altre manera, si no s'aconsegueix accedir a la base de dades, bé per que no troba la base de dades a la IP especificada, o bé per que l'usuari i/o contrasenya són incorrectes, llençarà un missatge d'error informant que alguna cosa ha anat malament.

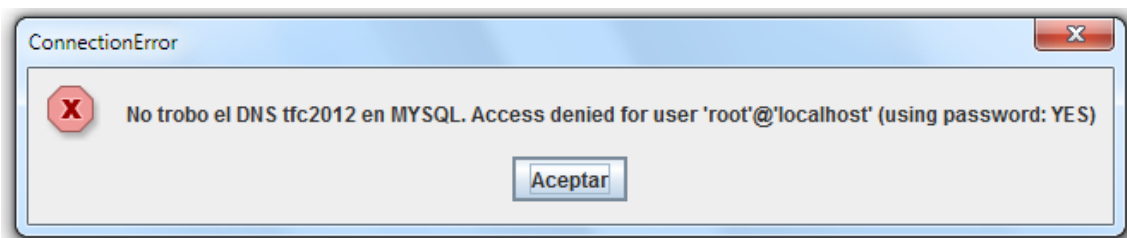


Figura 8.3. Missatge d'error d'accés a la BBDD

També es pot donar el cas d'accedir a una base de dades existent, però que no conté la informació que s'espera. En aquest cas permetrà obrir la següent pantalla però no permetrà processar cap dada.

8.2.2. Filtrant les dades

Quan ja es té accés a la base de dades, s'obre pas al següent JFrame, encarregat de filtrar la informació i d'escollir el mètode de processament (Figura IV).

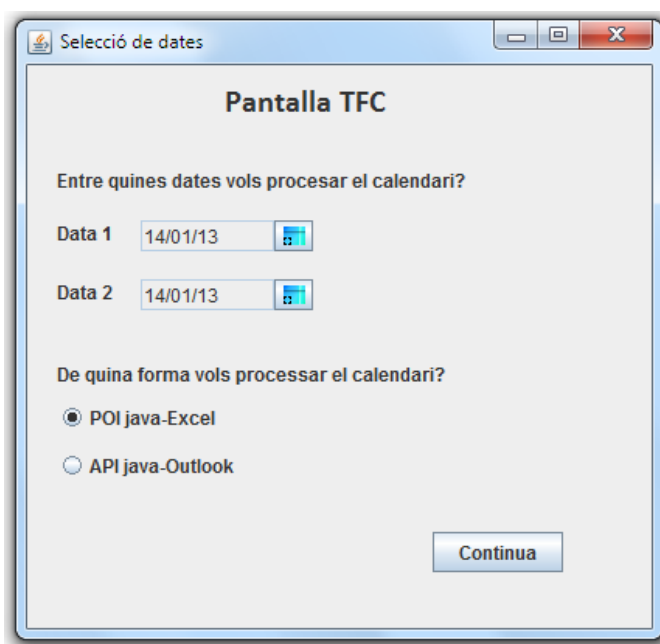


Figura 8.4. Pantalla de filtrat de dates i mètode de processament

Aquest JFrame disposa d'una api que permet seleccionar dates, i que per defecte mostra les dates del dia actual. Per escollir les dates es fa clic a sobre de la icona blava, llavors s'obre una finestra en format calendari que permet escollir les dates desitjades fent doble clic sobre el dia en concret.



Figura 8.5. Finestra del calendari de dates

Existeixen una sèrie de requeriments si es vol continuar a l'hora d'escollir el filtrat de dates, d'altra manera l'aplicació llençarà un missatge informant de l'error.

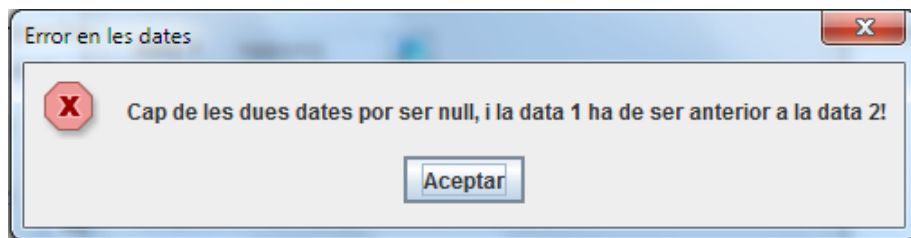


Figura 8.6. Missatge d'error amb les dates

Seguidament l'usuari podrà escollir el mètode de processament de les dades. Les dues opcions són mútuament excloents, també el procés que segueix cadascun dels dos mètodes és completament diferent i el resultat no és exactament el mateix.

8.2.3 Processament de la informació

Si es prem el botó de continuar a la pantalla de filtrat de dades, apareix una última pantalla que on l'usuari haurà de seleccionar les versions de les quals vol extreure informació.

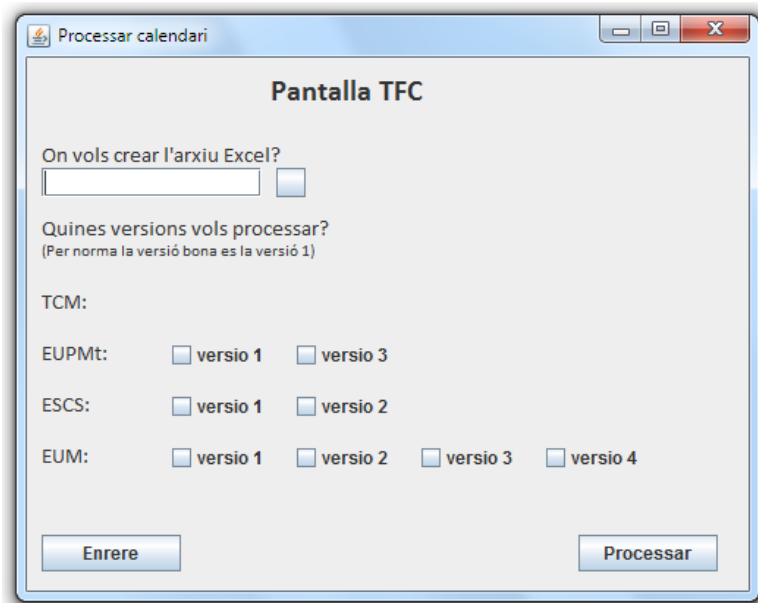


Figura 8.7. Última pantalla, processament de dades

Nota: La informació dels diferents horaris a la BBDD es pot classificar per versions (versió 1, versió 2, ...), i tot i que la informació necessària ve donada a la versió 1, l'usuari pot escollir de quines versions vol processar el calendari, si es produís el cas que n'hi hagués més d'una, és per aquest motiu que és recomanable que l'usuari sàpiga quines versions l'interessa. Aquest petit filtre de versions ve donat per motiu que a la BBDD amb la que es va treballar en un principi a l'hora de fer el programa existien més d'una versió per cada centre universitari, tot i que a les següents BBDD dels darrers anys només existia una única versió, és per això que es va creure convenient donar la possibilitat d'escollir quines versions es volien processar.

Com s'ha comentat en l'apartat anterior, el resultat variarà lleugerament si s'escull un mètode o un altre per processar el calendari, també ho farà la forma de treballar les dades.

Si s'ha escollit el mètode POI Java – Excel (mètode 1), apareixerà un espai on l'usuari podrà escollir un destí i un nom d'arxiu. Això és degut a que el mètode 1, quan obté la informació de la base de dades crea un arxiu Excel (directament importable a Outlook) amb aquesta informació i el nom escollit per l'usuari al destí que aquest hagi triat. Llavors el sistema informa que l'arxiu ha sigut creat correctament, crea un arxiu amb el path de l'arxiu generat i comunica que s'està enviant la informació a Outlook (Figura VIII).

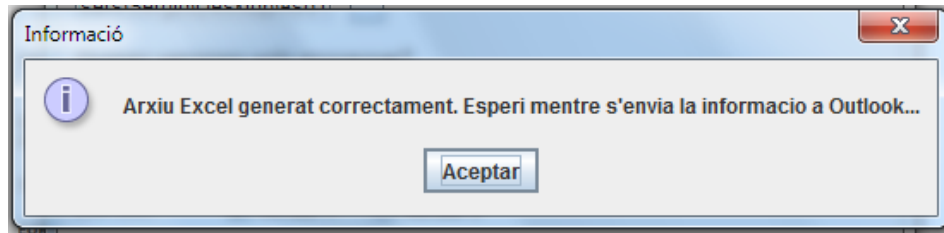


Figura 8.8. Missatge informatiu del mètode 1

Per enviar aquesta informació Java executa un programa extern codificat en Visual Basic, situat en una carpeta a l'arrel del programa. El funcionament d'aquesta macro en Visual Basic es comenta a l'apartat 8.3. Només comentar que un cop la macro ha fet la seva funció, s'obre una pantalla amb un missatge informant de que el calendari ha estat actualitzat amb èxit (Figura IX).

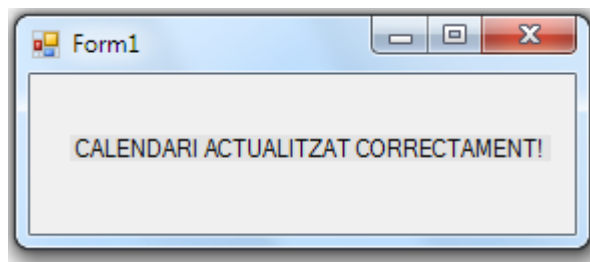


Figura 8.9. Interfície VB, informa de que tot ha estat correcte

Si s'ha escollit el mètode amb l'API Java – Outlook (mètode 2), tot el treball d'enviar la informació es fa des de la mateixa aplicació Java.

La idea de donar a escollir dues formes d'enviar la informació ve donada per que una envia la informació directament des de l'aplicació Java i permet interactuar amb Outlook des de Java i l'altre requereix d'un programa extern per fer-ho. A més, la diferència en quant a resultat respecte els dos mètodes, és que el mètode 2 distribueix la informació entre els diferents calendaris corresponent a les aules del centre universitari, i el mètode 1 fica tota la informació al calendari per defecte.

Cal comentar que l'API que fa servir el mètode disposa d'un període de tres mesos de prova, passats els quals s'haurà de comprar una llicència d'ús per fer-la servir. Si passa aquest temps i no es disposa de la llicència el mateix Java llençarà un error informant de la necessitat d'obtenir la llicència i informació de com fer-ho.

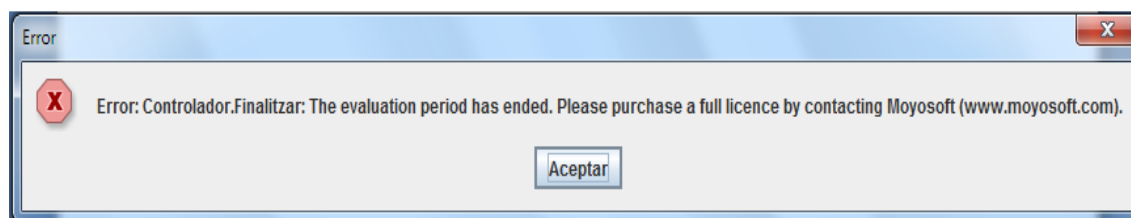


Figura 8.10. Missatge d'error: període de prova API Outlook finalitzat

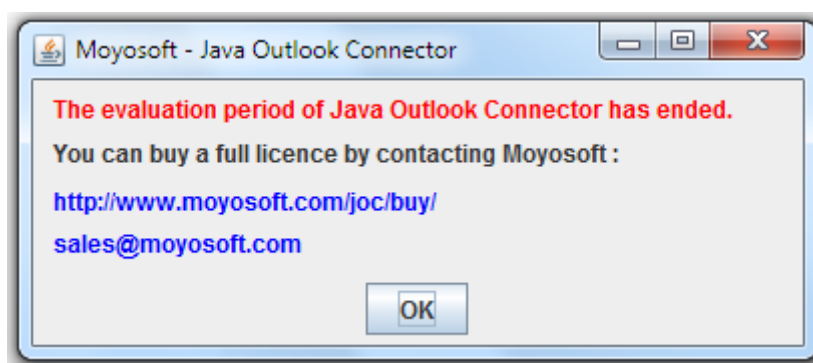


Figura 8.11. Missatge informatiu de com obtenir una llicència de l'API

8.3. L'aplicació en Visual Basic

Per la necessitat de trobar una alternativa a l'enviament de les dades a Outlook, es va haver de codificar un programa en Visual Basic que fes la funció de macro i enviés informació d'un arxiu Excel a Outlook.

Primer de tot, per permetre que Visual Basic treballi amb Outlook i Excel, s'ha d'afegir una referència a aquestes dues eines office. Per fer-ho es fa clic a projecte, afegir referència, i a la casella de COM es busquen les referències. Un cop afegides ja es pot treballar amb objectes Outlook i Excel.

És important que la macro VB estigui dins la carpeta *Macro Excel-Outlook* dins l'arrel del programa Java. Com ja s'ha comentat a l'apartat anterior, el programa Java quan fa servir

el mètode 1 crea un arxiu amb el path de l'arxiu Excel generat, i doncs és creat dins aquesta carpeta. Llavors el programa VB va a buscar dins la carpeta arrel un arxiu de text amb el nom de path i llegeix la informació d'aquest per saber on anar a buscar per llegir les dades.

Un cop VB té accés a l'arxiu Excel, llegeix línia per línia i envia cites a Outlook fins que troba que no hi ha més cites per enviar.

9. Conclusions

9.1. Valoració del projecte

L'elaboració d'aquest software ha estat un punt important en l'aprenentatge de l'eina Java, ja que ha servit per ampliar els coneixements en termes d'accés a bases de dades i processament de les dades obtingudes. També ha servit per millorar la forma de treballar, organitzant la feina i el diferent codi, millorant la mantenibilitat de l'aplicació.

A més ha servit per fer una petita introducció al món de la programació en Visual Basic, una eina desconeguda completament abans de l'elaboració d'aquest projecte.

En conclusió, aquest projecte ha servit per obrir els ulls i saber que Java ofereix molt més del que es pensava inicialment, i que té un ventall de possibilitats enorme.

9.2. Valoració del resultat

A pocs dies d'entrega del projecte, es considera que el software compleix els objectius marcats.

S'han fet proves del seu funcionament i es controlen molts dels possibles errors que es puguin produir, i es crea un arxiu d'incidències informant de la data i el motiu de l'error.

Quan en un principi l'eina ja estava pràcticament finalitzada, es va intentar millorar el rendiment d'aquesta i els resultats van ser molt satisfactoris. Inicialment l'eina trigava més de mitja hora en enviar les dades a Outlook i passat aquest temps donava error de memòria per l'excés de dades enviades simultàniament (al voltant de 8000 cites). Actualment s'ha aconseguit un temps mínim de 4 minuts per enviar la mateixa quantitat de cites.

També es creu que la mantenibilitat és força bona, molta part del codi que pugui portar a confusió esta documentada i ben organitzada en paquets.

9.3. Expectatives de millora

Tot i assolit l'objectiu principal del projecte, en cap cas es considera aquesta eina com tancada. Encara es pot continuar desenvolupant i millorant les seves funcionalitat.

Algunes de les millores que es podrien afegir serien:

- Si s'ha d'escriure en un aula i aquesta està plena, que informi a l'usuari.
- Si el solapament no és idèntic, preguntar a l'usuari què fer (mostrar que hi ha i que es vol ficar). Afegir possibilitat de repetir acció per cada solapament trobat.

10. Annex

La memòria del projecte ve acompanyada d'un CD amb:

- Projecte Java del software desenvolupat
- Llibreria usada en aquest projecte
- Petit apèndix de manual d'usuari del programa
- Carpeta de la macro codificada en Visual Basic (codi + executable)
- Documentació i arxius referents a l'API Java – Outlook
- Documentació i arxius referents al POI Java – Excel
- Documentació i arxius referents al driver connector MySQL
- Scripts de creació de base de dades usats
- Exemple calendari Excel generat pel software
- Memòria en format PDF

11. Bibliografia

- [1] <http://www.officerecovery.com/es/outlookundelete/licensing.htm> Llicències Outlook
- [2] <http://www.tufuncion.com/trabajo-programador> Sou programador Junior
- [3] <http://apiconz.wordpress.com/2007/10/22/como-generar-hojas-de-excel-en-una-aplicacion-java-usando-apache-poi/> Exemple d'ús POI Java – Excel
- [4] <http://www.javaprogrammingforums.com/java-swing-tutorials/278-how-add-actionlistener-jbutton-swing.html> Afegir *Listeners* a la codificació d'interfícies dinàmicament
- [5] http://chuwiki.chuidiang.org/index.php?title=Uso_de_Layouts#El_Layout_null Informació per a la codificació d'interfícies de forma dinàmica
- [6] <http://www.coderanch.com/t/279182/java/Starting-line-FileWriter-class> Crear arxiu de text
- [7] <http://www.calendario-365.es/calendario-2010.html> Calendaris per setmanes
- [8] <http://www.sc.ehu.es/sbweb/fisica/cursoJava/fundamentos/clases1/string.htm> Manipular Strings
- [9] <http://www.rdebug.com/2011/01/lanzar-aplicaciones-del-so-desde-java.html> Executar arxius externs des de Java

