



Escola Politècnica Superior
d'Enginyeria de Vilanova i la Geltrú

UNIVERSITAT POLITÈCNICA DE CATALUNYA

PROJECTE FI DE CARRERA

TÍTOL: Aplicació de gestió (ERP) per una escola bressol

AUTOR: *Victor Lossius Millet*

TITULACIÓ: *Enginyeria Tècnica en Informàtica de Gestió*

DIRECTOR: *Jordi Esteve Cusiné*

DEPARTAMENT: *Llenguatges i Sistemes Informàtics*

DATA: *27 de Juny de 2011*

TÍTOL: *Aplicació de gestió (ERP) per una escola bressol*

COGNOMS: *Lossius Millet*

NOM: *Victor*

TITULACIÓ: *Enginyeria Tècnica Informàtica*

ESPECIALITAT: *Gestió*

PLA: *1992*

DIRECTOR: *Jordi Esteve Cusiné*

DEPARTAMENT: *Llenguatges i Sistemes Informàtics*

QUALIFICACIÓ DEL PFC

TRIBUNAL

PRESIDENT

SECRETARI

VOCAL

DATA DE LECTURA:

Aquest Projecte té en compte aspectes mediambientals: Sí No

PROJECTE FI DE CARRERA

RESUM (màxim 50 línies)

Amb aquest projecte final de carrera es vol crear una aplicació per gestionar una escola bressol.

Aquesta aplicació s'encarregarà de controlar tota la gestió dels alumnes matriculats a l'escola, guardant de cadascun d'ells tota la informació possible, així com també de gestionar els horaris, hores extra i menjador per cadascun dels dies que assisteixen a l'escola. A més, es gestionarà la facturació mensual per cadascun dels alumnes matriculats, tenint en compte el tipus d'ús que s'ha fet de l'escola. També es vol tenir un control de tot el material escolar de l'escola i sobre quin és l'estat de cadascun d'ells i l'aula a la que està assignada. Altrament, l'aplicació també ha de gestionar tota la resta d'informació que conté una escola bressol, com ara les educadores, aules i les vacances.

La tecnologia escollida per realitzar aquest projecte és OpenERP, un ERP (Planificació de Recursos Empresarials) lliure i multiplataforma, amb client tant d'escriptori com web. OpenERP ens ofereix diferents mòduls operatius, on aquests són modificables per poder satisfer les nostres necessitats empresarials.

Paraules clau (màxim 10):

<i>OpenERP</i>	<i>Python</i>	<i>PostgreSQL</i>	<i>XML</i>
<i>ERP</i>	<i>Alumne</i>	<i>Assistent</i>	<i>Educadora</i>
<i>Gestió</i>	<i>Escola</i>		

Agraïments

Vull agrair aquest projecte a les següents persones:

Al meu tutor Jordi Esteve, per la seva ajuda i temps dedicat durant la realització del projecte.

Al professor Bernardino Casas per l'ajuda inicial quan estava realment perdut i per ajudar-me a trobar el tutor adequat.

A la meva xicota Mireia per donar-me la idea d'aquest PFC , per tot el suport donat, per animar-me i per estimar-me.

Als meus companys Rafa, David i Juanjo, per les incomptables tardes d'estudi realitzades durant el transcurs de la carrera.

Gràcies a tots.

Índex

1. Introducció i objectius	13
1.1 Objectius	13
1.2 Justificació	13
2. Descripció del projecte	15
2.1 Gestió dels nens	15
2.1.1 Altes i baixes	16
2.1.2 Consultes i modificacions	17
2.2 Gestió de les educadores	17
2.3 Gestió de les aules i material de l'escola	17
2.4 Facturació	18
2.5 Manuals del sistema	18
3. Avaluació tecnològica	19
3.1 ERP	19
3.2 Escollir ERP	20
3.2.1 Principals ERPs lliures	20
3.2.2 Opció escollida: OpenERP	21
3.3 OpenERP	21
3.3.1 Tipus d'arxius	24
4. Planificació inicial	27
5. Anàlisi i especificació	29
5.1 Especificació de requeriments	29
5.1.1 Requeriments referents als alumnes	29
5.1.2 Requeriments referents als responsables dels alumnes	29
5.1.3 Requeriments referents a les participacions dels alumnes	29
5.1.4 Requeriments referents a les educadores	30
5.1.5 Requeriments referents als càrrecs de les educadores	30
5.1.6 Requeriments referents al material escolar	30
5.1.7 Requeriments referents a les aules	30
5.1.8 Requeriments referents als menús	30
5.1.9 Requeriments referents a la facturació	31
5.1.10 Requeriments referents a les vacances	31
5.1.11 Requeriments no funcionals	31
5.2 Model Conceptual	31
5.2.1 Glossari de conceptes	31
5.3 Casos d'ús del sistema	34
5.3.1 Diagrama de casos d'ús	34
5.3.2 Casos d'ús	35
5.3.2.1 <i>Crear nou alumne</i>	35
5.3.2.2 <i>Modificar Educadora</i>	40
5.3.2.3 <i>Esborrar Material</i>	42
5.3.2.4 <i>Assistent Consultar Alumnes Menjador</i>	43
5.3.2.5 <i>Assistent Crear Factura</i>	45
5.3.2.6 <i>Assistent Crear Participacions</i>	47
6. Disseny	51
6.1 Aprofitament dels mòduls oficials i extres	51
6.2 Diagrames de col·laboració	54
6.2.1 Cas d'ús: Crear nou alumne	54
6.2.1.1 <i>nouAlumne (alumne)</i>	54
6.2.1.2 <i>afegirDades (dades, aula)</i>	55
6.2.1.3 <i>assignarResponsable (responsable)</i>	55
6.2.1.4 <i>crearParticipació (dia)</i>	56
6.2.1.5 <i>afegirDadesParticipació (dades)</i>	57
6.2.1.6 <i>fiAlumne ()</i>	58
6.2.2 Cas d'ús: Modificar educadora	58
6.2.2.1 <i>esborrarMaterial (exemplar)</i>	60
6.2.2.1 <i>seleccionarEducadora (educadora)</i>	58

6.2.2.2 <i>modificarDades (dades, aula, càrrec)</i>	59
6.2.2.3 <i>fiEducadora ()</i>	59
6.2.3 Cas d'ús: Esborrar Material	60
6.3 Model de components	61
6.3.1 Visibilitats d'atribut	61
6.3.2 Visibilitats de paràmetre	62
6.3.3 Visibilitats locals	63
7. Implementació	65
7.1 Versió d'OpenERP	65
7.2 Mòduls desenvolupats	65
7.2.1 Mòdul kindergarten	65
7.3 Model	67
7.3.1 Objecte Material Escolar	67
7.3.2 Objecte Alumne	69
7.4 Controlador	69
7.4.1 Objecte Participació	70
7.5 Vista	72
7.5.1 Vista arbre	72
7.5.2 Vista formulari	74
7.5.3 Vista cerca	76
7.5.4 Accions	77
7.5.5 Menús	77
7.5.6 Dreceres	77
7.6 Implementació d'altres recursos	78
7.6.1 Assistents (wizards)	78
7.6.2 Introducció de dades automàticament	79
7.6.3 Seguretat i permisos	79
7.6.4 Traduccions	80
7.7 Test i jocs de proves	80
8. Planificació final i anàlisi econòmica	83
8.1 Planificació final	83
8.1.1 Comparativa entre planificació inicial i resultat final	86
8.2 Anàlisi econòmica	87
9. Conclusions	91
9.1 Conclusions personals	91
9.2 Conclusions tècniques	91
9.3 Possibles ampliacions	92
9.4 Aspectes mediambientals	92
9.5 Anàlisi d'accessibilitat	93
Annex I: Manual d'administrador	95
1.1 Instal·lació del servidor OpenERP	95
1.1.1 Instal·lació del servidor en Windows	95
1.1.2 Instal·lació del servidor en Linux	95
1.2 Execució de servidor OpenERP	96
1.2.1 Windows	96
1.2.2 Linux	96
1.3 Creació d'una base de dades	97
1.4 Instal·lació dels mòduls	99
1.5 Creació d'usuaris	100
Annex II: Manual d'usuari	105
2.1 Instal·lació del client OpenERP	105
2.1.1 Instal·lació del client en Windows	105
2.1.2 Instal·lació del client en Linux	105
2.2 Execució del client	105
2.3 Interfície d'usuari	107
2.4 Ús general	108
2.4.1 Gestió d'alumnes	109
2.4.1.1 Alumnes	109
2.4.1.2 Responsables	117
2.4.1.3 Historial de Factures	117

2.4.2 Gestió d'educadores.....	118
2.4.2.1 Educadores.....	118
2.4.2.2 Càrrecs.....	119
2.4.3 Aules.....	120
2.4.3.1 Aules.....	120
2.4.3.2 Material Escolar.....	121
2.4.3.3 Consultar material escolar per renovar.....	122
2.4.4 Configuració.....	122
2.4.4.1 Vacances.....	123
2.4.4.2 Menús diaris.....	123
2.4.4.3 Totes les participacions.....	123
2.4.4.4 Tipus de quotes.....	124
2.4.5 Assistents consulta.....	124
Annex III: Programes usats.....	127
Bibliografia.....	129

1. Introducció i objectius

El projecte final de carrera és l'últim esglaió que ha de superar l'estudiant per tal de finalitzar la carrera universitària, i és on l'alumne, com a futur enginyer, ha de posar en pràctica tots els coneixements obtinguts durant els seus estudis.

Durant els estudis es realitzen problemes concrets amb certes limitacions, però és en el Projecte Final de Carrera on l'estudiant ha de trobar solució a un problema sense les limitacions del món acadèmic i més proper al món real.

1.1 Objectius

Amb aquest PFC es volen assolir una sèrie d'objectius:

- Realitzar un anàlisi complet al problema proposat pel client. D'aquest anàlisi s'extraurà la informació necessària per fer el disseny.
- Implementar la solució dissenyada, així com redactar la memòria i els manuals per l'ús de l'aplicació.
- Introduir-nos en el món dels ERPs, així com assolir els coneixements necessaris per la utilització d'aquesta tecnologia cada dia més present en les empreses.
- Per últim, es vol minimitzar els costos de desenvolupament d'aquest projecte, usant sempre que sigui possible, el màxim d'eines lliures.

1.2 Justificació

Quan arriba el moment de començar a pensar en el PFC, molts alumnes com jo, ens trobem a la situació que no sabem exactament que realitzar. En el meu cas, tenia la idea que volia treballar amb un ERP, una tecnologia molt demandada a les empreses, i en la que volia aprofundir els meus coneixements obtinguts anteriorment treballant en una empresa privada. Més concretament, vaig estar col·laborant en el disseny i testeig d'un sistema SAP pel departament de Recursos Humans de l'empresa T-systems Iberia.

Encara que tenia clar que volia dirigir el meu PFC cap al món dels ERP, no tenia gens clar com fer-ho. Fins que fruit d'una conversa amb un familiar que treballa com educadora, va sorgir la idea d'aquest PFC. Amb la qual cosa, encara que aquest projecte no fos aplicable en una empresa, si que disposava d'una persona que faria el paper de client i m'explicaria els requisits que hauria de tenir aquesta aplicació.

Un cop fet l'anàlisi i descripció del projecte, em vaig posar amb contacte amb el que ha estat el meu tutor del PFC, Jordi Esteve, per tal de explicar-li la idea del projecte i veure amb quina tecnologia podia treballar-hi. Ell em va comentar que l'anterior quadrimestre un estudiant havia estat treballant en la realització d'un mòdul per un menjador escolar amb la tecnologia OpenERP, i que estaria molt bé poder utilitzar aquest mòdul per tal d'incorporar la funció de menjador al meu projecte, que en un principi no havia tingut en compte.

2. Descripció del projecte

La idea d'aquest PFC neix degut a la falta de programari de gestió d'una escola bressol que permeti gestionar la totalitat d'aquesta. Es vol disposar d'una eina on es pugui gestionar tota la informació de l'escola de manera més eficient i productiva. A més, un dels requisits és aconseguir que estigui tot dins la mateixa aplicació i sigui el més intuïtiu possible.

L'objectiu principal serà doncs, crear i dissenyar una aplicació de gestió per a una escola bressol, de manera que emmagatzemi i gestioni tota la informació necessària pel funcionament de l'escola. D'aquesta manera, voldrem:

- Mantenir i gestionar les dades dels nens juntament amb les del seu/s responsable/s.
- Gestionar l'assistència a l'escola dels nens amb les dades que això comporta: horari, servei acollida i menjador.
- Mantenir i gestionar les dades de les educadores.
- Mantenir i gestionar les dades de les aules així com el material de l'escola.
- Gestionar la facturació.

2.1 Gestió dels nens

Es vol mantenir totes les dades possibles dels nens: NIF, nom, cognoms, data de naixement, aula assignada, així com les dades del/s seu/s responsables: NIF, nom, cognom, telèfons (fíx, mòbil i telèfon de la feina), adreça, població, ocupació i compte bancari. Cal tenir en compte que cada nen/a ha de tenir com a mínim un responsable per tal de que es pugui matricular com alumne de l'escola, això és important ja que per qualsevol problema amb el nen/a, sempre es truca al responsable d'aquest. Encara que la situació normal és mantenir com a responsables el pare i la mare (en alguns casos, a part dels pares, es manté també algun dels avis com a responsable), es pot donar el cas per qualsevol raó on només es mantingui un d'ells, on en casos extraordinaris sigui un tutor el responsable del nen.

Encara que l'escola tingui un horari d'obertura i de tancament, és habitual que cada nen faci un horari

diferent depenent de les necessitats dels pares. L'horari de l'escola és el següent:

- Acollida: de 7:45h a 8:45h. Hora extra pels pares que volen deixar el seu fill abans d'hora.
- Matí: de 8:45h a 12:00h.
- Matí i migdia: de 8:45h a 15:00h. Menjador inclòs.
- Tot el dia: de 8:45h a 17:00h. Menjador inclòs.

Per tant l'aplicació ens ha de permetre poder seleccionar l'horari que farà cada nen, si n'en fa ús del servei d'acollida, així com si aquest es queda a dinar a l'escola. Es pot donar el cas que un nen/a només vingui 3 hores al dia pel matí així com pot haver-hi un altre cas on un nen/a faci l'hora d'acollida i l'horari de tot el dia. Cal afegir que, per normativa de l'escola, el responsable del nen pot modificar el seu horari per un dia (o dies) concret si aquest ho desitja, encara que lo habitual és que un nen faci sempre el mateix horari. El sistema també ha d'oferir una opció per tal de consultar l'assistència a l'escola pel dia escollit, així com consultar els nens que assistiran a l'hora d'acollida. D'aquesta manera l'educadora que treballi a primera hora del matí podrà saber quins nens vindran a l'acollida.

Pel que fa el funcionament del servei de menjador, l'aplicació ens ha de permetre gestionar el menjador diari dels nens, així com la possibilitat de que en qualsevol moment es pugui seleccionar que un nen es quedi a dinar un dia en concret. L'aplicació també ens ha de permetre gestionar el tipus de menú de l'escola, així com poder consultar quants nens es queden a dinar pel dia escollit, per tal de facilitar la feina al cuiner.

A més a més, es vol que l'aplicació ofereixi la possibilitat per afegir-hi informació addicional, on es pugui comentar per exemple a quin responsable trucar primer en cas de tenir algun problema, mantenir la informació necessària sobre el/s medicament/s que s'ha de prendre un nen si es dona el cas, poder comentar si el nen té alguna al·lèrgia a algun tipus de menjar o medicament, o estar informats si el nen té algun tipus de discapacitat.

2.1.1 Altes i baixes

Per tal de mantenir tota la informació esmentada, necessitarem que l'aplicació ens ofereixi la possibilitat d'afegir un nen quan els pares vulguin matricular-lo. Per fer això caldrà que els pares (o responsables) ens omplin un formulari amb tota la informació necessària, per tal que l'administrador pugui introduir aquestes dades al sistema.

Necessitarem també que l'aplicació ens ofereixi la possibilitat de donar de baixa, ja sigui per baixa temporal o baixa permanent. Un nen estarà actiu mentre se li pugui facturar un mes, pot estar de baixa temporal per alguna causa justificada, període el qual se li facturarà. Quan un nen abandoni l'escola tindrem una baixa permanent, i, per tant, aquest passarà a estar en estat de baixa (o inactiu).

És important que l'aplicació ens deixi controlar els períodes de temps que un nen ha estat actiu (on volem incloure les baixes temporals). És per això que mai eliminarem un nen del sistema, només es donarà l'excepció quan un nen es matriculi i no arribi a incorporar-se a l'escola, o quan aquest comenci l'etapa escolar.

2.1.2 Consultes i modificacions

Dels nens també ens interessa consultar totes les seves dades quan es necessiti, ja sigui per buscar el telèfon del seu responsable o per mirar l'horari que hi realitza. Les dades també poden ser modificables des del mateix lloc de consulta, un cop modificades, substituiran les actuals.

2.2 Gestió de les educadores

S'ha de contemplar el manteniment de les dades de les educadores que treballen a l'escola. Per cada educadora necessitarem saber: NIF, nom, cognoms, adreça, població, telèfon, jornada laboral, aula que té assignada i càrrec que realitza dins l'escola (ja sigui directora, educadora, educadora de suport o personal de cuina). Amb les educadores, l'aplicació també necessitarà les opcions per donar d'alta i de baixa. Una educadora estarà activa mentre estigui treballant, en el moment que deixi d'exercir la feina, passarà a estar en estat de baixa.

Les educadores es podran donar d'alta, consultar i modificar en qualsevol moment, però a l'igual que el cas dels nens mai s'eliminaran, excepte quan una educadora contractada no arribi a incorporar-se mai a l'escola. Això és perquè es desitja que el sistema pugui controlar els períodes de temps que l'educadora ha estat treballant a l'escola i el càrrec que exercia. A més, sempre pot haver-hi la possibilitat que una educadora que hagi treballat anteriorment a l'escola torni al cap d'uns anys, amb la qual cosa, es mantindria tota la informació sobre el seu càrrec actual i l'exercit anteriorment.

2.3 Gestió de les aules i material de l'escola

Respecte les aules, s'ha de contemplar les diferents característiques d'aquestes: el nom de l'aula, metres quadrats, la ratio (capacitat) de nens que té i el material que té assignat. Cada nen estarà assignat a una aula on hi passarà normalment la majoria de la jornada, això vindrà donat segons l'edat del nen, des de l'aula on hi estaran els nens menors d'un any fins l'aula on hi estaran els nens més grans de 2-3 anys. Per exemple, una escola bressol podria tenir les següents aules:

- Una aula de P0 (de 0 a 1 any). Capacitat per 8 nens.
- Dues aules per P1 (de 1 a 2 anys). Capacitat per 13 nens.
- Dues aules per P2 (de 2 a 3 anys). Capacitat per 20 nens.

Cada aula tindrà assignada com a mínim una educadora, que serà responsable de la gestió diària de les necessitats d'aquesta.

Pel que fa al material de l'escola, s'ha de portar un control d'aquest. Entenem per material educatiu, els llibres infantils, música, joguines i material plàstic. De cadascun d'ells es vol saber el títol, format, l'estat en que es troba (perfecte o deteriorat), l'aula que té assignada, així com un codi identificatiu per cada objecte, ja poden haver-hi repetits. El material de l'escola, pot donar-se d'alta, modificar-se o eliminar-se en qualsevol moment. A més, es vol disposar d'una eina per consultar el material deteriorat, per tal d'estalviar temps a l'hora de renovar-lo.

2.4 Facturació

El primer dia del mes, es generaran tots els rebuts dels nens que han estat actius durant el mes anterior. Cadascun d'aquests s'enviarà a l'adreça del nen pertinent, on constarà el nom de l'escola, la data i el número del rebut, i, a més a més, el nom del nen al que fa referència, el nom del responsable/s i l'import a pagar. L'import a pagar serà la suma de la quota fixa del més (independentment del tipus d'horari que hagi fet el nen), la quota del servei d'acollida i la quota del menjador.

Pel que fa la quota del servei d'acollida, aquesta pot ser esporàdica o fixa. La quota d'acollida fixa té un preu estipulat, i en el cas de la quota d'acollida esporàdica, es calcularà a partir del preu d'aquest servei multiplicat pels dies que se n'ha fet ús. Un cop calculat la quota esporàdica, es cobrarà la més adient pel client, és a dir, la de menor cost. La quota del menjador, es calcularà de la mateixa manera. Quan es faci efectiu un rebut, s'ha de marcar com pagat.

2.5 Manuals del sistema

Per tal de completar l'aplicació, caldrà disposar dels pertinents manuals d'instal·lació i de utilització per l'usuari final amb la descripció del seu funcionament. Aquests s'ajuntaran al final d'aquesta memòria.

3. Avaluació tecnològica

En el moment d'escollir la tecnologia que volem utilitzar per resoldre el projecte plantejat, s'ha de tenir en compte el llenguatge de programació que volem utilitzar, el sistema gestor de base de dades que volem, quin tipus de servidor s'utilitzarà, entre d'altres.

Partim de la idea que el client vol una solució d'escriptori en comptes d'una aplicació web per temes d'agilitat, i vol que les dades estiguin centralitzades en un servidor on diferents usuaris puguin accedir-hi a la vegada. També es vol que l'aplicació sigui multiplataforma, i que les tecnologies a utilitzar siguin lliures, per no tenir que pagar llicències.

3.1 ERP

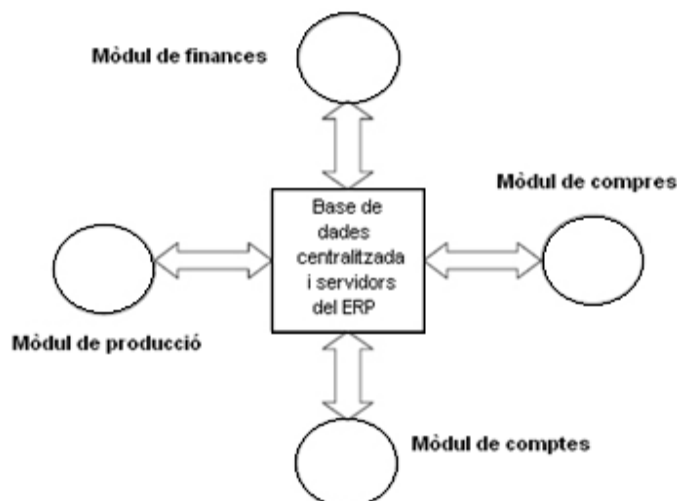
Un ERP (Planificació de Recursos Empresarials) és un sistema d'informació que integra i gestiona moltes de les habituals pràctiques que es realitzen en una empresa. Un ERP es caracteritza respecte a una altra eina de gestió, en la integració màxima de les seves diferents parts en una mateixa aplicació.

Els objectius principals d'un sistema ERP són:

- Optimització dels processos empresarials.
- Accés a tota la informació de forma fiable, precisa i oportuna (integritat de les dades).
- Possibilitat de compartir informació entre tots els components de l'organització.
- Eliminació de dades i operacions innecessàries.
- Base de dades centralitzada.

Les principals característiques que distingeixen un ERP de qualsevol altre software empresarial, és que tenen que ser sistemes integrals, modulars i adaptables:

- Integrals, és a dir, que permetin que els diferents departaments d'una empresa es relacionin entre sí, on el resultat d'un procés dependrà d'aquesta integració.
- Modulars, es divideixen en diferents mòduls les diferents necessitats d'una empresa, on aquesta farà ús de la part que necessiti, ja sigui: recursos humans, finances, vendes, compres, entre d'altres.
- Adaptables, on una empresa pugui incorporar fàcilment a un mòdul les necessitats que aquesta necessiti.



Il·lustració 1 - Esquema de l'arquitectura modular d'un ERP

3.2 Escollir ERP

3.2.1 Principals ERPs lliures

En el mercat es poden trobar multitud de ERPs privatis com ara SAP o WorkPLAN, però per tal de minimitzar els costos, ens limitarem a estudiar les opcions de codi obert més importants que tenim:

- **Compiere**, aquesta aplicació ERP-CRM (Customer Relationship Management) està pensada per a petites i mitjanes empreses. Està desenvolupada totalment amb Java amb base de dades Oracle, i com altres ERPs, es pot ampliar afegint-hi mòduls. Encara que Compiere està governada per una llicència de software lliure (Compiere Public License), existeixen diverses llibreries internes de les quals no es proporcionen el codi amb el producte, i fins i tot, algunes de pagament. A més, s'ha de pagar en cas que vulguem accedir a la documentació de desenvolupament i el suport tècnic.
- **Openbravo**, ERP de codi obert destinada a petites i mitjanes empreses. L'estructura de dades està basada amb una versió antiga de Compiere, projecte el qual ja no manté cap compatibilitat. És una aplicació amb arquitectura client/servidor web, el que li permet ser utilitzada des de qualsevol navegador web. Escrita en Java s'executa sobre Apache i Tomcat, i té suport per a bases de dades PostgreSQL i Oracle.
- **FacturaLux**, és un ERP orientat a treballar amb grans bases de dades i processos administratius com ara l'administració, gestió comercial i finances. Aquest ERP ofereix mòduls des de la gestió

financera i comercial fins a l'adaptació a processos complexes de producció. Treballa amb bases de dades Postgres i MySQL, i la seva interfície gràfica es basa en la llibreria Qt.

- **OpenERP**, és un software ERP amb llicència GPL (General Public License) on les seves principals característiques són aportar solucions en diferents àmbits com la gestió de treballadors, clients, facturació, compres, vendes, etc. Es basa en l'arquitectura client/servidor, el que permet treballar a tots els usuaris sobre el mateix repositori de dades. A més, és multiplataforma i disposa de client web, amb la qual cosa, li permet treballar des de qualsevol navegador web. OpenERP utilitza PostgreSQL com sistema manegador de bases de dades i ha estat programat en Python.
- **Fisterra**, és un projecte que pretén crear un ERP genèric fet amb software lliure. En l'actualitat suporta: administració de clients, facturació, gestió de stock i de pagaments, entre d'altres. La implementació usa Gnome SDK i PostgreSQL.

3.2.2 Opció escollida: OpenERP

A l'hora de decantar-me per OpenERP, va tenir un gran pes, que aquest sistema sigui més flexible, amigable i potent que altres ERPs que existeixen en el mercat, a més ens dona molta facilitat i senzillesa a l'hora de crear nous mòduls. Aquesta facilitat es gràcies a que es disposa d'un framework de desenvolupament anomenat OpenObject que permet desenvolupar nous mòduls amb un potent ús d'herències per estendre els mòduls existents. OpenERP disposa de més de 500 mòduls entre els oficials (120) i els extres, amb la qual cosa, tenim un gran ventall de mòduls per poder adaptar-los a les nostres necessitats. A més, disposa d'una gran comunitat al darrera amb una alta participació dels seus membre, tant en aportació de mòduls com en ajuda.

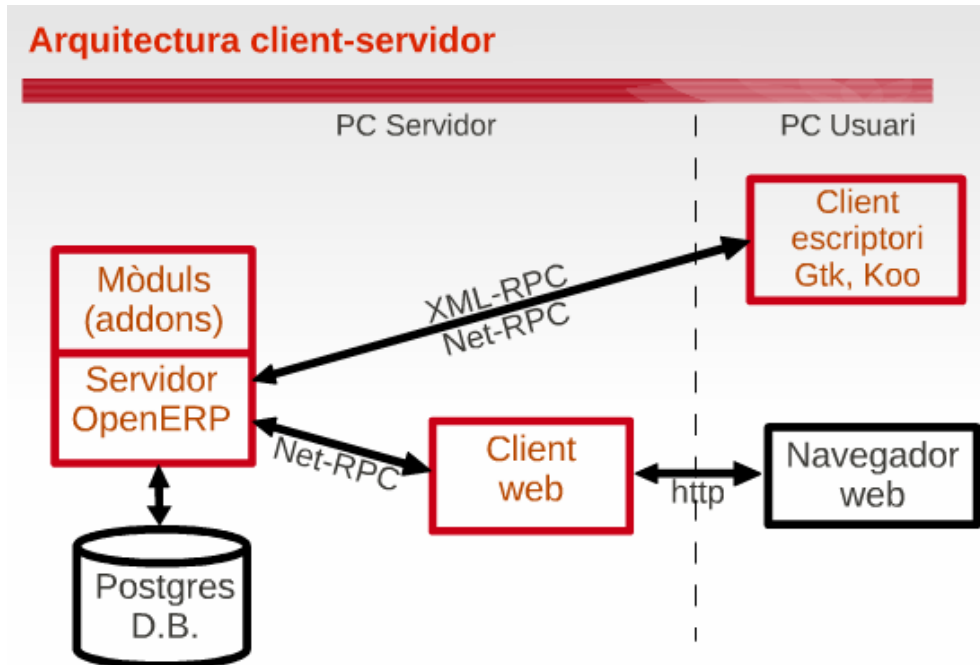
Finalment, el que em va acabar de decidir per escollir aquesta opció, és que el meu tutor té experiència amb OpenERP, això significaria que tindria un suport molt gran de cara els dubtes i problemes que m'haguessin d'aparèixer en el transcurs del projecte.

3.3 OpenERP

Com hem comentat abans, OpenERP és un complet sistema de gestió empresarial que cobreix les necessitats de gestió, vendes, compres, facturació, entre molts d'altres. Disposa de 120 mòduls oficials, i molts més d'extres, arribant a la xifra de més de 500 mòduls disponibles, tots ells amb possibilitat de adaptar-los a les

nostres necessitats sempre que ens convingui.

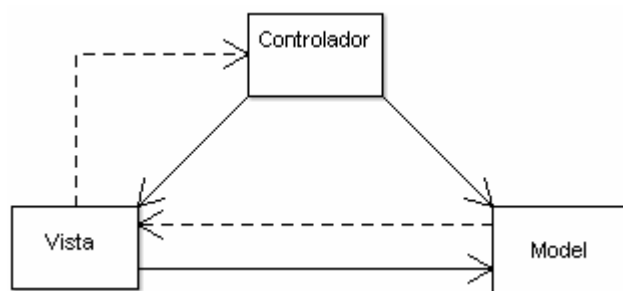
OpenERP usa una arquitectura client/servidor. El client envia les peticions al servidor, i aquest ataca la base de dades, retornant els valors demanats pel client.



Il·lustració 2 - Arquitectura Client/Servidor (Què és OpenERP? – Jordi Esteve)

Com podem observar en l'anterior il·lustració, hi ha dos tipus de clients: client d'escriptori i client web. El client d'escriptori usa la llibreria Gtk i es connecta directament al servidor. Aquest client és multiplataforma i es pot instal·lar en Mac OS, Linux i Windows. En el cas del client web, podem accedir-hi des de qualsevol navegador (Firefox, Opera, etc.) introduint la direcció del servidor i automàticament se'ns connectarà al servidor. Cal dir, que el servidor es pot tenir instal·lat en Windows o Linux.

OpenERP usa l'esquema Model Vista Controlador (MVC). Aquest model separa les dades d'una aplicació, la interfície i la lògica de control en tres components diferents.

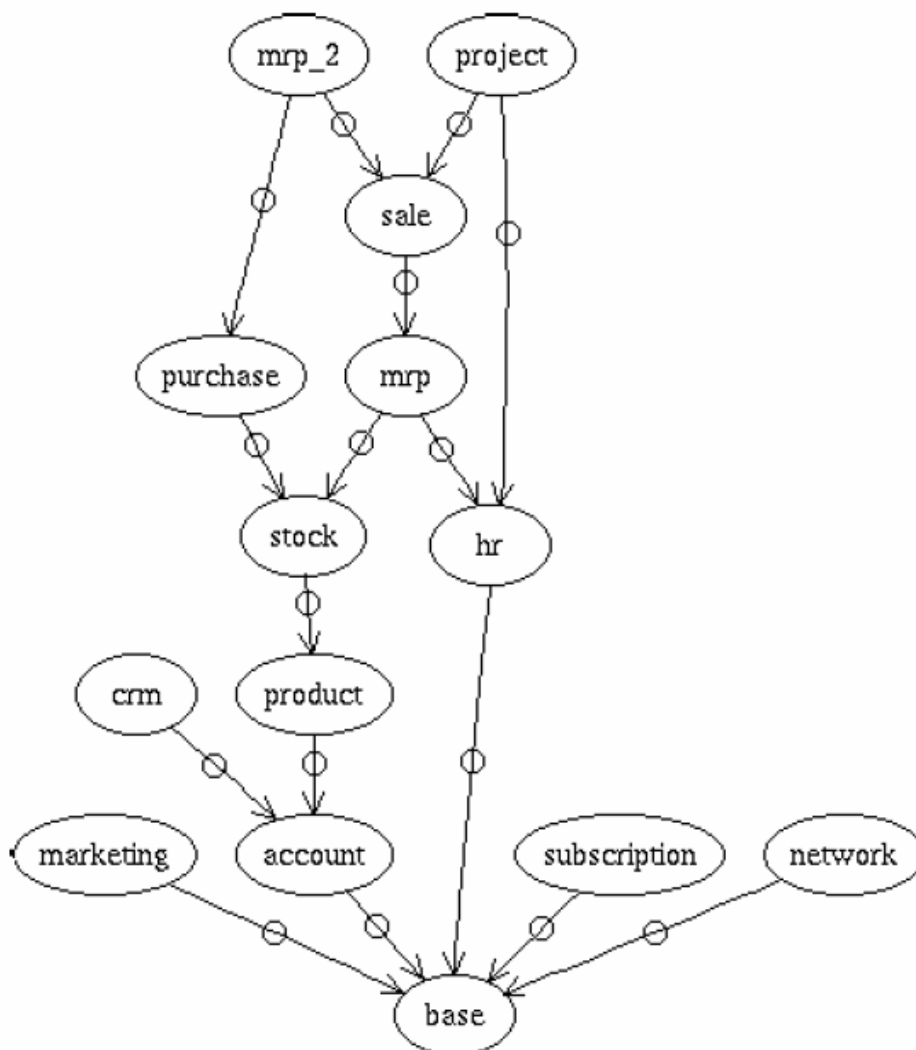


Il·lustració 3 - Esquema Model Vista Controlador (http://es.wikipedia.org/wiki/Modelo_Vista_Controlador)

El model s'encarrega de gestionar les dades, la vista presenta el model en un format adequat per interactuar amb l'usuari, i el controlador és l'encarregat de respondre a les accions i interaccions entre l'usuari i la vista. D'aquestes tres parts de l'esquema, en el cas d'OpenERP es distribueix:

- Pel model s'utilitza un diccionari Python que es mapeja automàticament sobre una base de dades PostgreSQL.
- Per la vista usa arxius en XML.
- Pel controlador usa arxius en Python.

OpenERP divideix les diferents tasques de gestió d'una empresa en diferents mòduls, on cada mòdul aporta una solució a unes necessitats concretes. Podem entendre un mòdul com una extensió del entorn OpenERP que hi afegeix una funció de negoci. Quan instal·lem OpenERP, s'instal·la per defecte el mòdul base, el qual ja conté les característiques bàsiques com ara gestió de contactes, empreses, usuaris, etc. Es poden instal·lar diferents mòduls, tenint en compte que hi ha mòduls que depenen d'altres.



Il·lustració 4 - Esquema que mostra els principals mòduls d'OpenERP amb les seves dependències (http://openerp.com/wiki/index.php/Documentation:UserManual/The_Modules)

A l'hora de trobar una solució a un problema amb OpenERP, el que hem de fer és estudiar els diferents mòduls que ens poden servir pel nostre objectiu. Un cop fet això, podem:

- Deixar el mòdul sense modificar-lo ja que ens soluciona totalment el nostre problema.
- Adaptar aquest mòdul a les nostres necessitats mitjançant l'herència.
- Crear un nou mòdul en cas que no puguem aprofitar-lo.

3.3.1 Tipus d'arxius

Internament un mòdul és una carpeta que conté els arxius de controlador i de vista per aquell mòdul. Aquests arxius són:

- `__init__.py`: arxiu Python que indica els arxius amb la lògica del mòdul.
- `__openerp__.py`: arxiu Python que conté informació del mòdul com l'autor, versió, descripció, arxius dels que depèn, els noms de les vistes, etc.
- `nom_modul.py`: arxiu que conté la lògica del mòdul en Python.
- `nom_modul_view.xml`: arxiu que conté les vistes en xml.
- `/security`: carpeta que conté un arxiu CSV on es defineixen les regles d'accés i permisos d'usuari. També conté un arxiu xml on es defineixen els grups d'usuari que tindrà el mòdul.
- `/wizard`: carpeta que conté els assistents del mòdul. Encara que actualment, podem prescindir d'aquesta carpeta i afegir els arxius dels assistents directament a la carpeta del mòdul.
- `/report`: carpeta amb els informes del mòdul
- `/i18n`: dins d'aquesta carpeta es troba un arxiu `.po` on es defineixen les traduccions del mòdul.

Dins de l'arxiu que conté la lògica del mòdul, es poden definir diferents objectes, on un objecte OpenERP s'entén com un recurs necessari en la gestió de l'empresa. Dins d'aquests objectes es defineixen diferents tipus de camps, on aquests poden ser altres objectes o diferents tipus de variables:

- Camps elementals: Camps bàsics que poden ser de diferents tipus: char, boolean, integer, float, date, datetime, time, selection, binary, reference.
- Camps relacionals: Són camps que relacionen objectes, hi ha tres tipus: many2one, one2many, many2many.
- Camps funcionals: camps que es calculen a partir d'altres.

Quan s'instal·la el mòdul, aquest camps automàticament es creen a la base de dades.

Sobre els objectes podem aplicar-hi accions (mètodes), on hi ha accions predefinides com pot ser la creació, escriptura o eliminació de dades. Aquestes accions, es poden redefinir en cas que vulguem adaptar-les.

Perquè l'usuari pugui veure els camps descrits anteriorment als models, s'han de definir les vistes en arxius amb format xml. La vista d'un mòdul pot tenir les següents formes:

- Vista formulari (form), permet crear o modificar els camps.
- Vista arbre o llista (tree), mostra les dades en format arbre o llista.
- Vista calendari (calendar), mostra els recursos en un calendari amb les dates segons s'hagin introduït.
- Vista gràfica (graph), mostra les dades en un gràfic de barres o pastís.
- Vista cerca (search), ens serveix per modificar la cerca de registres a la vista arbre.

A més d'aquests arxius, un mòdul pot contenir altres tipus d'arxius:

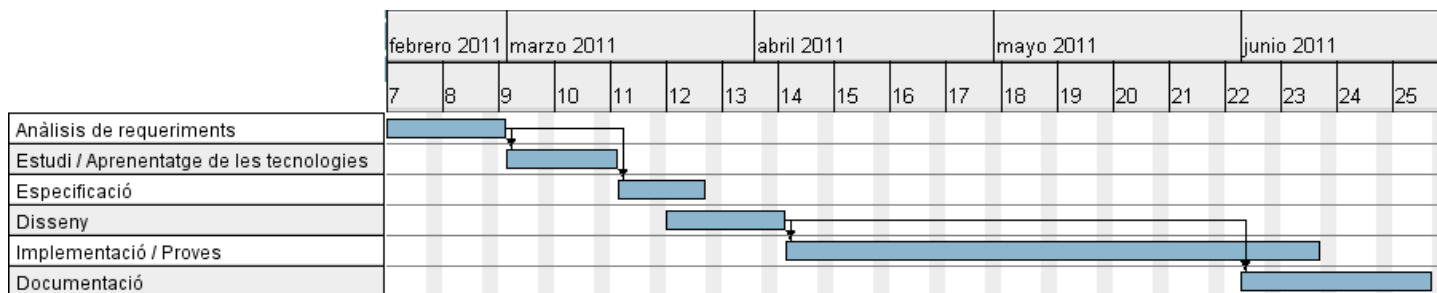
- Informes (reports), serveixen com plantilles per realitzar informes personalitzats amb les dades que vulguem. Es generen en format PDF, Openoffice, HTML o XML.
- Fluxos de treball (workflows), permeten definir estats per les dades i les seves transicions.
- Assistents (wizards), accions que serveixen per guiar a l'usuari a realitzar determinades tasques.

4. Planificació inicial

Aquest projecte s'inicia a mitjans de febrer de 2011, i en un principi hi havia una planificació inicial on estava previst finalitzar aquest projecte al novembre de 2011. Aquesta planificació es va modificar en els primers compassos del projecte quan la meva disponibilitat horària per dedicar-li temps al projecte va ser major, i vaig veure possible acabar-lo a temps el mateix quadrimestre. Per tant, a continuació es pot veure l'explicació de la planificació inicial tenint previst acabar al juny de 2011.

A l'hora de planificar el projecte, s'ha dividit en diferents parts, on cadascuna d'aquestes comporta una tasca del projecte:

1. Anàlisi del projecte. En aquesta fase inicial, s'analitza el problema, i s'estableixen els passos a seguir per realitzar el projecte de manera correcte. Aquesta fase té una durada de dues setmanes.
2. Estudi de la tecnologia. En aquesta fase s'estudia la tecnologia a utilitzar, i un cop escollida ens familiaritzarem amb ella: OpenERP, Python, XML i PostgreSQL. Té una durada de dues setmanes.
3. Especificació. A partir de la descripció del projecte feta, extraurem els requeriments funcionals, no funcionals i d'interfície. A més, també es faran el model conceptual i els casos d'ús del sistema. Temps estimat d'una/dues setmanes.
4. Disseny. Es realitzarà el disseny de la interacció del sistema així com el de la interfície d'usuari. Temps estimat d'una/dues setmanes.
5. Implementació. En aquest període es crea la solució software dissenyada per resoldre la problemàtica inicial. Té una durada inicial de vuit setmanes.
6. Test i proves. Es provarà el software amb dades i casos reals, d'aquestes proves es corregiran els errors que vagin sorgint. Té una durada de dues setmanes, i es té pensat realitzar a la vegada que es va implementant el sistema.
7. Documentació. Redacció de la memòria del PFC juntament amb els manuals d'ús. Temps estimat de 3 setmanes.



Il·lustració 5 - Diagrama de GANTT que mostra les etapes del projecte

S'ha considerat que:

- Les setmanes tenen 7 dies de treball
- Els dies tenen 5-6 hores laborals aproximadament.
- Les vacances de setmana santa es consideren dies de treball així com els altres dies festius que hi puguin haver.

5. Anàlisi i especificació

5.1 Especificació de requeriments

De la definició del problema i conceptes podem dividir aquests en diferents grups: alumnes, responsables, participacions, educadores, càrrecs, material escolar, aules, menús, facturació i vacances. Els requeriments que ha de contenir cada grup són els següents:

5.1.1 Requeriments referents als alumnes

R.1.1 Donar d'alta nou alumne.

R.1.2 Modificar un alumne.

R.1.3 Consultar un alumne.

R.1.4 Donar de baixa un alumne.

R.1.5 Assignar un alumne a una aula.

R.1.6 Assignar un responsable a l'alumne

R.1.7 Esborrar alumne (només quan es dóna d'alta i no arriba a incorporar-se, o quan comenci l'etapa escolar).

5.1.2 Requeriments referents als responsables dels alumnes

R.2.1 Crear nou responsable.

R.2.2 Modificar responsable.

R.2.3 Consultar responsable.

R.2.4 Esborrar responsable.

5.1.3 Requeriments referents a les participacions dels alumnes

R.3.1 Crear nova participació per un alumne.

R.3.2 Crear participacions durant un interval de temps per l'alumne seleccionat.

R.3.3 Modificar participació.

R.3.4 Consultar participació.

R.3.5 Consultar totes les participacions d'un alumne.

R.3.6 Esborrar participació.

R.3.7 Marcar una participació com no assistida.

R.3.8 Consultar assistència pel dia seleccionat.

R.3.9 Consultar menjador pel dia seleccionat.

R.3.10 Consultar acollida pel dia seleccionat.

5.1.4 Requeriments referents a les educadores

- R.4.1 Donar d'alta nova educadora.
- R.4.2 Modificar una educadora.
- R.4.3 Consultar una educadora.
- R.4.4 Assignar una educadora a una aula.
- R.4.5 Donar de baixa una educadora.
- R.4.6 Esborrar educadora (només quan es dóna d'alta i no arriba a incorporar-se).

5.1.5 Requeriments referents als càrrecs de les educadores

- R.5.1 Crear un nou càrrec.
- R.5.2 Modificar un càrrec.
- R.5.3 Consultar un càrrec.
- R.5.4 Esborrar un càrrec.

5.1.6 Requeriments referents al material escolar

- R.6.1 Crear un exemplar de material escolar.
- R.6.2 Modificar un exemplar de material escolar.
- R.6.3 Consultar un exemplar de material escolar.
- R.6.4 Assignar un exemplar de material escolar a una aula.
- R.6.5 Esborrar un exemplar de material escolar.
- R.6.6 Consultar material escolar per renovar

5.1.7 Requeriments referents a les aules

- R.7.1 Crear una aula.
- R.7.2 Modificar característiques d'una aula.
- R.7.3 Consultar característiques d'una aula.
- R.7.4 Esborrar una aula.
- R.7.5 Consultar material assignat a una aula.
- R.7.6 Consultar alumnes assignats a una aula.
- R.7.7 Consultar educadores assignades a una aula.

5.1.8 Requeriments referents als menús

- R.8.1 Crear un nou menú.
- R.8.2 Modificar un menú.
- R.8.3 Consultar els menús.
- R.8.4 Esborrar un menú.

5.1.9 Requeriments referents a la facturació

R.9.1 Crear una factura.

R.9.2 Modificar una factura.

R.9.3 Consultar una factura.

R.9.4 Esborrar una factura.

5.1.10 Requeriments referents a les vacances

R.10.1 Crear un nou període de vacances.

R.10.2 Modificar un període de vacances.

R.10.3 Consultar un període de vacances.

R.10.4 Esborrar un període de vacances.

5.1.11 Requeriments no funcionals

- S'ha de poder accedir a la informació des de diferents llocs i diferents usuaris alhora.
- Tota la informació es guarda en una mateixa base de dades.
- El sistema ha de ser el més intuïtiu possible i senzill d'utilitzar, just el que l'usuari necessita.

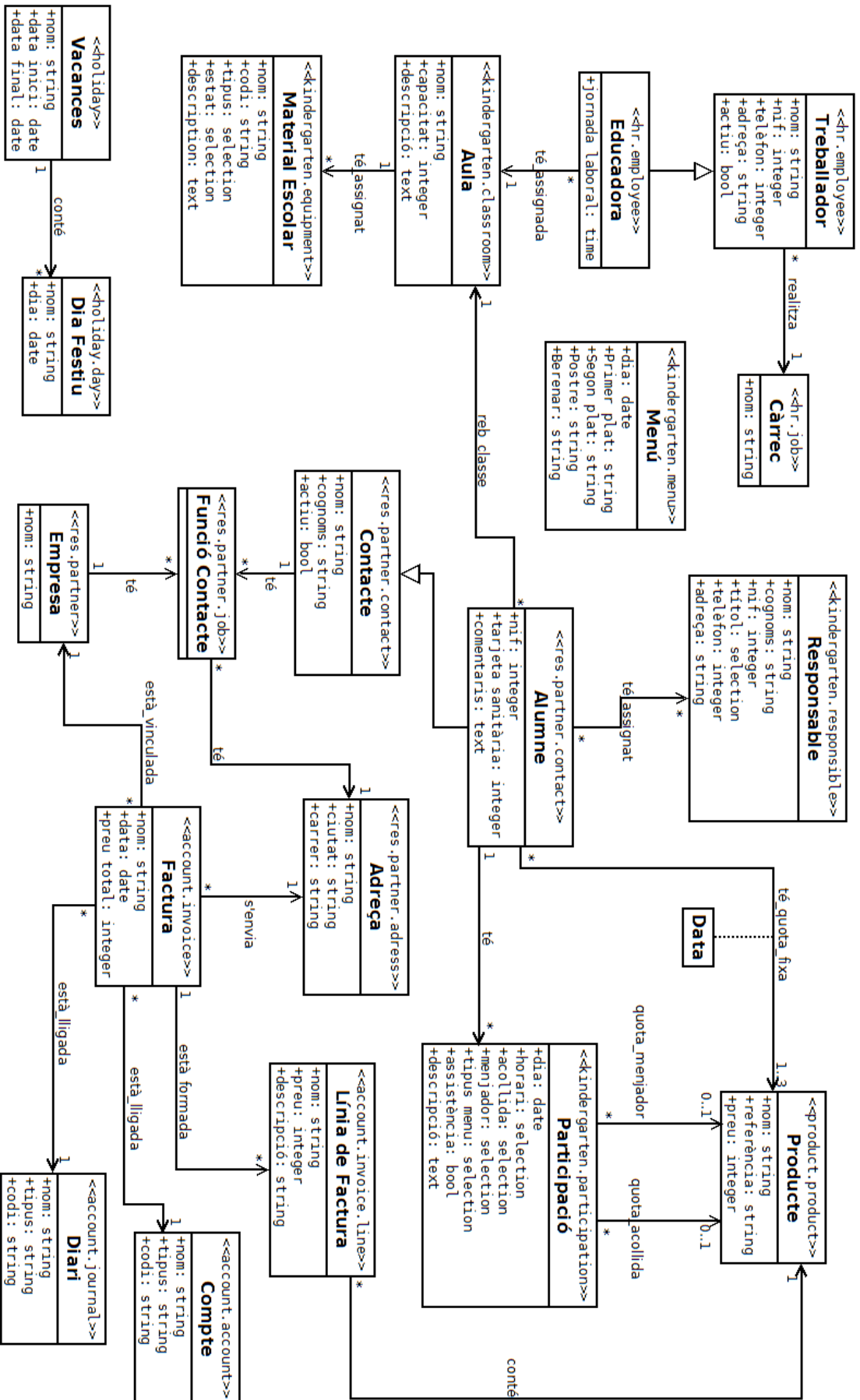
5.2 Model Conceptual

A continuació es pot veure el model conceptual del sistema, que mostra els conceptes del projecte, les relacions que existeixen entre elles i quin tipus de relació és. Primer veiem el glossari de conceptes que apareixen en aquest:

5.2.1 Glossari de conceptes

- **Alumne:** És el nen o nena matriculat a l'escola bressol.
- **Participació:** Representa l'ús que fa l'alumne de l'escola bressol per un dia concret. Ens informa de l'horari realitzat, si n'ha fet ús del servei d'acollida, si n'ha fet ús del menjador i si ha assistit a l'escola.
- **Adreça:** Direcció on s'envia la factura.
- **Responsable:** Entenem per responsable la persona o persones encarregades de l'alumne, ja siguin els seus pares, familiars o tutors.
- **Educadora:** És una treballadora del centre.
- **Càrrec:** Ens serveix per indicar el nivell de responsabilitat dins de l'escola: directora, educadora, educadora de suport, educadora en pràctiques o personal de cuina.

- **Aula:** És el lloc assignat d'un alumne on farà classe.
- **Material Escolar:** És el material de l'escola, com poden ser: joguines, música, llibres infantils i material plàstic.
- **Menú:** És el menú de l'escola per un dia.
- **Empresa:** És el destinatari de les factures, entenem per empresa la família o responsable del alumne el qual se li enviaran les factures.
- **Factura:** És un document associat a una Empresa i a un alumne. És on es guarda els usos que ha fet l'alumne de l'escola en un mes concret, i que es cobrarà a l'empresa.
- **Línia de factura:** Una línia de factura indica l'ús que ha fet l'alumne de l'escola; aquest ús vindrà donat pel producte que tingui associat.
- **Producte:** Entenem per productes, les diferents quotes que té l'escola. Hi hauran tres tipus de quotes fixes (quota mensual fixa, quota menjador fixa i quota acollida fixa) i dos tipus de quota esporàdica (quota menjador esporàdica i quota acollida esporàdica). Aquestes quotes són productes de tipus servei.
- **Vacances:** Període de temps on els alumnes no hi poden assistir a l'escola i no es poden crear participacions.
- **Dia de festa:** És cadascun dels dies de dins d'un interval de temps de vacances.

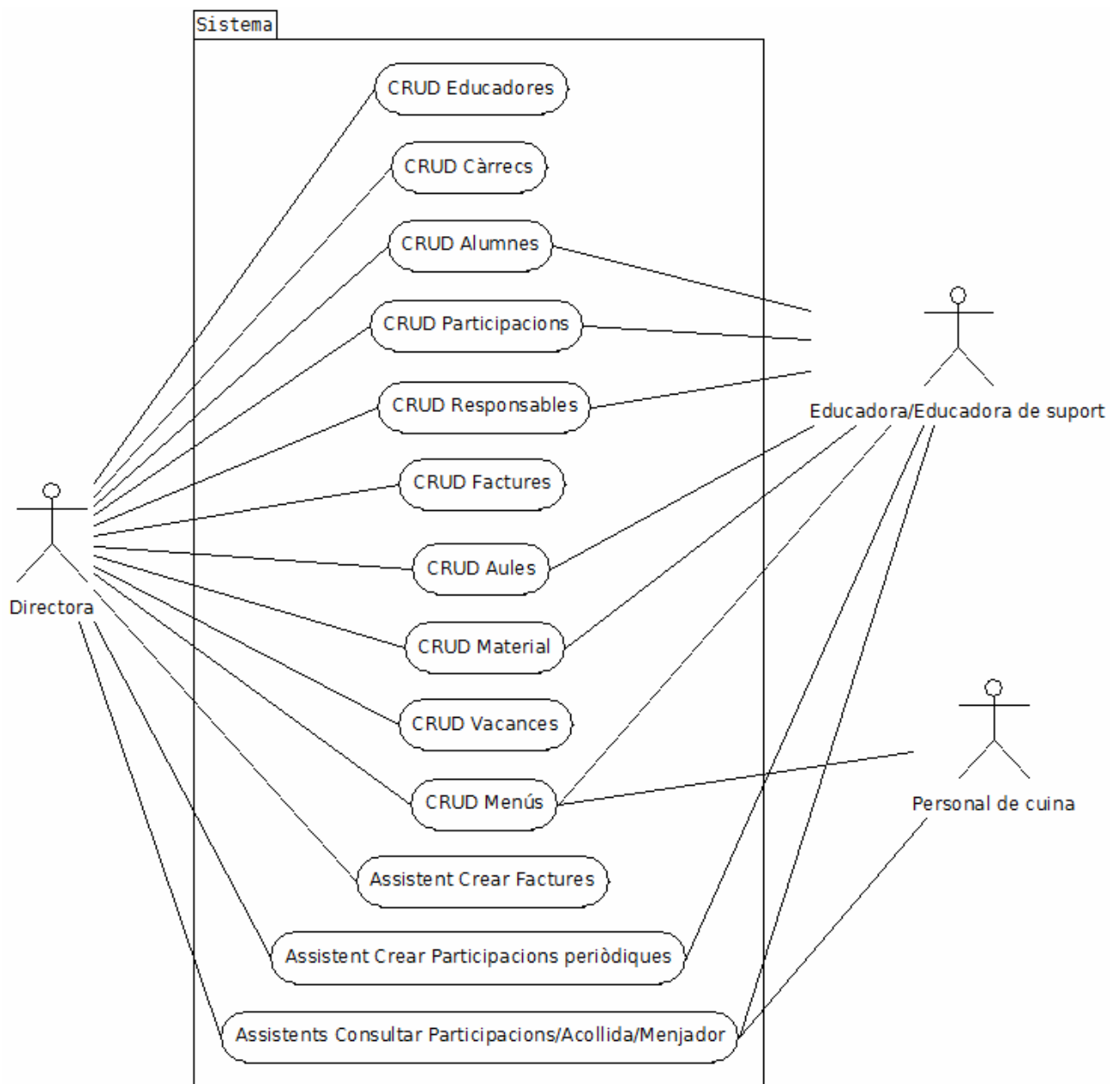


II-lustració 6 - Model Conceptual

5.3 Casos d'ús del sistema

5.3.1 Diagrama de casos d'ús

Per simplificar el diagrama, utilitzarem l'acrònim CRUD (Create, Retrieve, Update, Delete) per englobar els quatre casos d'ús bàsics de cada objecte, que són: crear, obtenir, actualitzar i esborrar.



Il·lustració 7 - Diagrama del model de casos d'ús

En la il·lustració 7 es mostra el diagrama de casos d'ús pels conceptes vistos al model conceptual, i on podem observar totes les interaccions possibles dels usuaris amb el sistema

En el diagrama s'han definit els següents actors:

- Directora: És la directora de l'escola bressol. Tindrà accés a tots els elements del mòdul ja que la seva feina es controlar durant tot el curs que tot vagi bé, gestionant educadores, alumnes, material, etc.
- Educadora: És qualsevol educadora assignada a una aula i responsable d'aquesta. Té accés al material escolar, aules, menús i consultar/modificar alumnes, així com gestionar les participacions d'aquests.
- Educadora de suport: És qualsevol educadora que té assignada una aula però que pot impartir classe a una altre aula segons les necessitats d'aquestes. Té els mateixos permisos que les educadores.
- Personal de cuina: Encarregat del funcionament de la cuina de l'escola. Té accés als menús de l'escola per tal de crear-los, així com a l'assistent consultar menjador, per tal de fer una previsió dels alumnes que es queden a dinar.

A més a més dels actors definits anteriorment, existeix el perfil d'administrador, que tindrà ple accés a tots els elements del sistema i serà l'encarregat del correcte funcionament d'aquest.

5.3.2 Casos d'ús

Com es pot veure a la il·lustració 7, els CRUD són la base dels casos d'ús, els més utilitzats, i a més, són força semblants entre ells. Per tant, a continuació només es mostren alguns d'aquests casos d'ús del sistema, però no s'arriben a mostrar tots degut a la seva semblança.

5.3.2.1 Crear nou alumne

Cas d'ús:	Crear nou alumne
Actors:	Directora
Descripció:	La finalitat d'aquest cas d'ús és donar d'alta un nou alumne en el sistema.
Requeriments:	R.1.1 Donar d'alta nou alumne
Curs normal d'esdeveniments:	
Accions dels actors	Sistema
1. L'actor prem el botó "Alumnes".	

	2. El sistema mostra la vista arbre de tots els alumnes registrats.
3. L'actor prem el botó "Nou registre".	
	4. El sistema mostra la vista formulari per a un nou alumne.
5. L'actor omple el formulari amb les dades obligatòries referents a l'alumne.	
6. L'actor introdueix si ho veu necessari les altres dades referents a l'alumne: comentaris, fotografia, nacionalitat, etc.	
7. L'actor seleccionar el camp "Aula" de l'alumne.	
	8. El sistema mostra les aules de l'escola.
9. L'actor seleccionar l'aula i prem el botó "Acceptar".	
10. L'actor selecciona la pestanya "responsables".	
	11. El sistema mostra la vista arbre per als responsables del alumne.
12. L'actor prem el botó "Afegir".	
	13. El sistema mostrarà tots els responsables registrats al sistema.
14. L'actor prem el botó "Nou registre" (en cas que el responsable ja existeixi, fa doble click sobre ell).	
	15. El sistema mostra la vista formulari pel responsable.
16. L'actor introdueix les dades del responsable (o les verifica en cas que el responsable ja existeixi) i prem el botó "Acceptar".	
	17. El sistema guarda aquest responsable i mostra la llista dels responsables de l'alumne.

18. L'actor repeteix el pas 12 fins que ha acabat d'introduir els responsables que vol.	
19. L'actor selecciona la pestanya "horari escolar" en cas que vulgui afegir participacions per l'alumne.	
	20. El sistema mostra la vista arbre on apareixen les participacions del alumne.
21. L'actor prem el botó "Crear nova entrada".	
	22. El sistema mostra la vista formulari per a una nova participació.
23. L'actor selecciona el dia el qual vol fer la participació.	
24. L'actor omple les altres dades referents a la participació (horari, acollida, menjador i assistència).	
25. L'actor prem el botó "Guardar i tancar".	
	26. El sistema guarda la participació creada i mostra de nou la llista de participacions de l'alumne.
27. L'actor repeteix el pas 21 fins que ha acabat d'introduir les participacions que vol.	
28. Quan l'actor ha acabat de donar d'alta l'alumne, prem el botó "Desar".	
	29. El sistema guarda l'alumne introduït a la base de dades amb els responsables registrats.
Curs alternatiu d'esdeveniments:	
<p>5. El sistema ens retornarà error en cas que no s'introdueixin les dades obligatòries.</p> <p>7. El sistema ens retornarà error en el cas que l'aula a la que assignem l'alumne estigui plena.</p> <p>12. Si no s'introdueix cap responsable, el sistema ens retornarà un error i no ens deixarà continuar.</p> <p>23. El sistema retornarà error si no s'omple el camp "dia" de la participació que es vol crear.</p> <p>23. El sistema no crearà la participació del dia indicat, en cas que aquesta ja existeixi.</p>	

Contractes del sistema:

- nouAlumne(a:alumne)
 - Precondició:
 - No hi ha cap alumne actiu.
 - L'alumne (a) no existeix en el sistema.
 - Postcondició:
 - Hi ha un alumne (a) actiu en el sistema.

- afegirDades (dades, aula)
 - Precondició:
 - Hi ha un alumne (a) actiu en el sistema.
 - Les dades que es volen introduir en el sistema són vàlides per a l'alumne (a).
 - Postcondició:
 - Les dades introduïdes s'han associat a l'alumne (a) actiu.
 - L'alumne (a) s'ha assignat a l'aula escollida.
 - Es manté l'alumne (a) actiu en el sistema.

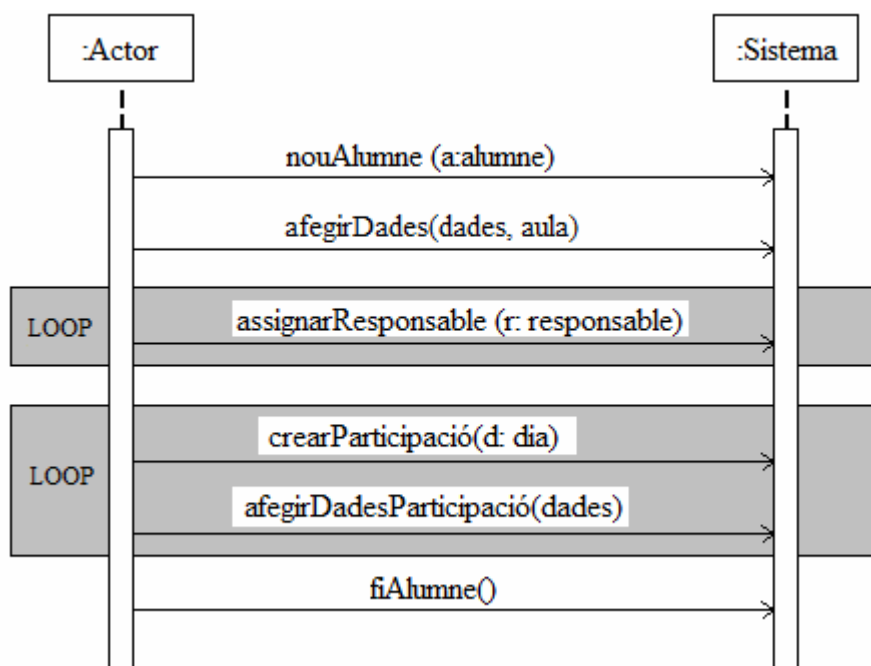
- assignarResponsable (r:responsable)
 - Precondició:
 - Hi ha un alumne (a) actiu en el sistema.
 - El responsable (r) no existeix o no està assignat a l'alumne (a) actiu.
 - Postcondició:
 - Hi ha un nou responsable (r) associat a l'alumne (a) actiu.
 - Es manté l'alumne (a) actiu en el sistema.

- crearParticipació (d:dia)
 - Precondició:
 - Hi ha un alumne (a) actiu en el sistema.
 - No hi ha cap participació activa.
 - No hi ha cap participació amb dia (d) per l'alumne (a) actiu.
 - Postcondició:
 - Hi ha un nova participació amb dia (d) associada a l'alumne (a) actiu.
 - Hi ha una participació (p) activa en el sistema.
 - Es manté l'alumne (a) actiu en el sistema.

- afegirDadesParticipació (dades)
 - Precondició:
 - Hi ha un alumne (a) actiu en el sistema.
 - Hi ha una participació (p) activa en el sistema.
 - Les dades que es volen introduir en el sistema són vàlides per la participació activa.
 - Postcondició:
 - Les dades introduïdes s'han associat a la participació (p) activa.
 - Es manté l'alumne (a) actiu en el sistema.
 - No hi ha cap participació activa.

- fiAlumne()
 - Precondició:
 - Hi ha un alumne (a) actiu en el sistema.
 - Postcondició:
 - El sistema conté un nou alumne (a) amb els responsables (r) associats i les participacions (p) creades.
 - No hi ha cap alumne actiu.

Diagrama de seqüència:



Il·lustració 8 - Crear Nou Alumne

5.3.2.2 Modificar Educadora

Cas d'ús:	Modificar educadora
Actors:	Directora
Descripció:	La finalitat d'aquest cas d'ús és modificar les dades de una educadora ja existent.
Requeriments:	R.4.2 Modificar una educadora R.4.3 Consultar una educadora
Curs normal d'esdeveniments:	
Accions dels actors	Sistema
1. L'actor prem el botó "Educadores".	
	2. El sistema mostra la vista arbre de totes les educadores registrades.
3. L'actor selecciona una educadora de la llista fent doble click.	
	4. El sistema mostra la vista formulari per a l'educadora seleccionada.
5. L'actor modifica les dades desitjades del formulari si ho necessita.	
6. En cas de que l'actor vulgui modifica l'aula assignada de l'educadora, aquest seleccionarà el botó "Aula assignada".	
	7. El sistema mostra la vista arbre de les aules existents.
8. L'actor selecciona l'aula adient.	
9. En cas de que l'actor vulgui modifica el càrrec assignat de l'educadora, aquest seleccionarà el botó "Càrrec".	
	10. El sistema mostra la vista arbre dels càrrecs disponibles.
11. L'actor selecciona el càrrec adient.	
12. Quan l'actor ha acabat de modificar les dades de l'educadora, prem el botó "Desar".	

	13. El sistema guarda les dades modificades per l'educadora seleccionada.
Curs alternatiu d'esdeveniments:	
3. Si l'educadora no existeix, s'haurà de donar d'alta.	
11. Si el càrrec no existeix, s'haurà de crear.	

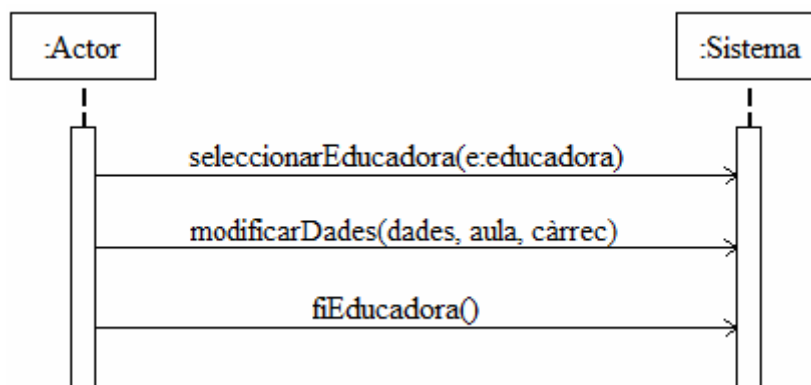
Contractes del sistema:

- seleccionarEducadora (e:educadora)
 - Precondició:
 - No hi ha cap educadora activa.
 - L'educadora(e) existeix en el sistema.
 - Postcondició:
 - Hi ha una educadora (e) activa en el sistema.

- modificarDades (dades, aula, càrrec)
 - Precondició:
 - Hi ha una educadora (e) activa en el sistema.
 - Les dades que es volen introduir en el sistema són vàlides per a l'educadora (e).
 - Postcondició:
 - Les dades introduïdes així com l'aula i el càrrec seleccionat, s'han associat a l'educadora (e) activa.
 - Es manté l'educadora (e) activa en el sistema.

- fiEducadora()
 - Precondició:
 - Hi ha una educadora (e) activa en el sistema.
 - Postcondició:
 - El sistema conté l'educadora (e) amb les noves dades modificades.
 - No hi ha cap educadora activa.

Diagrama de seqüència:



Il·lustració 9 - Modificar educadora

5.3.2.3 Esborrar Material

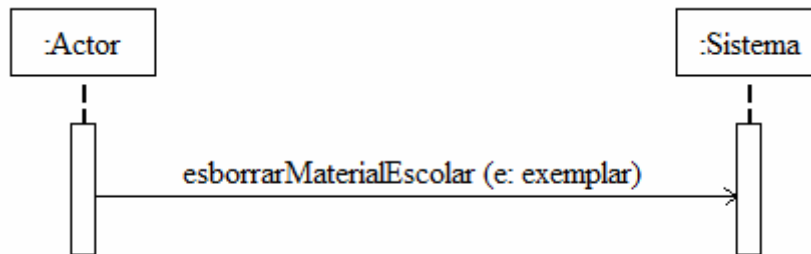
Cas d'ús:	Esborrar material
Actors:	Directora, Educadora/Educadora de suport
Descripció:	La finalitat d'aquest cas d'ús és d'esborrar del sistema un exemplar de material escolar.
Requeriments:	R.6.5 Esborrar un exemplar de material escolar.
Curs normal d'esdeveniments:	
Accions dels actors	Sistema
1. L'actor prem el botó "Material Escolar".	
	2. El sistema mostra la vista arbre de tot el material escolar registrat.
3. L'actor selecciona l'exemplar de material escolar de la llista.	
4. L'actor prem el botó "Esborrar".	
	5. El sistema mostra un missatge de confirmació per si realment es vol esborrar l'exemplar seleccionat.
6. L'actor confirma prement el botó "Sí".	

7. El sistema esborra de la base de dades l'exemplar de material escolar seleccionat.

Contractes del sistema:

- esborrarMaterial (e:exemplar)
 - Precondició:
 - L'exemplar(e) existeix en el sistema.
 - Postcondició:
 - No existeix en el sistema l'exemplar (e).

Diagrama de seqüència:



Il·lustració 10 - Esborrar Material Escolar

5.3.2.4 Assistent Consultar Alumnes Menjador

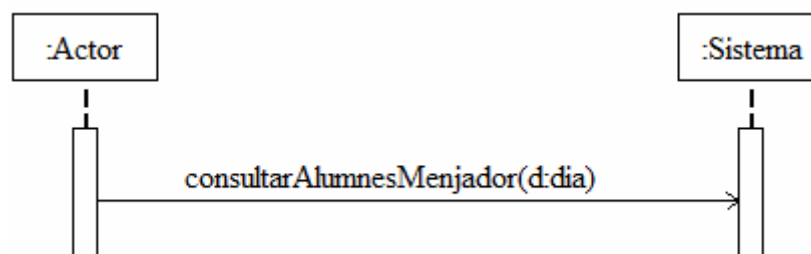
Cas d'ús:	Consultar alumnes menjador
Actors:	Directora, Educadora/Educadora de suport, Personal de cuina.
Descripció:	La finalitat d'aquest cas d'ús és crear un llistat dels alumnes que es queden a dinar a l'escola pel dia seleccionat (també ens serveix pels assistents consultar assistència i acollida).
Requeriments:	R.3.8 Consultar assistència pel dia seleccionat. R.3.9 Consultar menjador pel dia seleccionat. R.3.10 Consultar acollida pel dia seleccionat.
Curs normal d'esdeveniments:	

Accions dels actors	Sistema
1. L'actor prem el botó "Consultar alumnes menjador pel dia...".	
	2. El sistema mostra la vista de l'assistent on s'ha de escollir el dia.
3. L'actor seleccionar el dia en que vol consultar els alumnes que es queden a dinar.	
4. L'actor prem el botó "Acceptar".	
	5. El sistema mostra un llistat amb tots els alumnes que es queden a dinar pel dia seleccionat.
Curs alternatiu d'esdeveniments:	
3. El sistema retornarà error si s'introdueix un dia no vàlid.	

Contractes del sistema:

- consultarAlumnesMenjador(d:dia)
 - Precondició:
 - El dia (d) és vàlid pel sistema.
 - Postcondició:
 - El sistema mostra tots els alumnes actius del dia (d) apuntats al servei de menjador i que tenen previst assistir.

Diagrama de seqüència:



Il·lustració 11 - Assistent Consultar Alumnes Menjador

5.3.2.5 Assistent Crear Factura

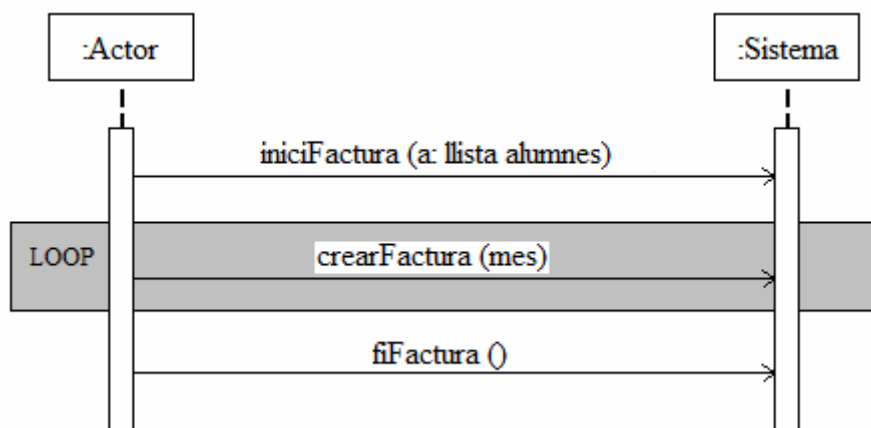
Cas d'ús:	Crear Factura
Actors:	Directora
Descripció:	La finalitat d'aquest cas d'ús es crear la factura pels alumnes seleccionats.
Requeriments:	R.9.1 Crear factura.
Curs normal d'esdeveniments:	
Accions dels actors	Sistema
1. L'actor prem el botó "Alumnes".	
	2. El sistema mostra la vista arbre de tots els alumnes actius a l'escola.
3. L'actor selecciona els alumnes els quals vol crear la factura.	
4. L'actor seleccionar l'assistent "Crear Factura" del menú accions.	
	5. El sistema obre una finestra per seleccionar el mes el qual es vol crear la factura.
6. L'actor selecciona qualsevol dia del mes el qual es vol facturar.	
	7. El sistema automàticament crea les factures per cadascun dels alumnes seleccionats, un cop creades, les mostra per pantalla.
8. Quan l'actor finalitza de verificar les factures, aquest prem el botó desar.	
	9. El sistema guarda les factures creades pels alumnes seleccionats.
Curs alternatiu d'esdeveniments:	
6. El sistema retorna error si el mes escollit encara no ha finalitzat.	
6. El sistema retorna error si l'alumne escollit no té una adreça de facturació.	
7. Si ja existeix la factura del mes indicat per l'alumne seleccionat, el sistema no fa res.	

Contractes del sistema:

- iniciFactura (a: llista alumnes)
 - Precondició:
 - No hi ha cap alumne actiu.
 - Els alumnes de la llista (a) existeixen en el sistema.
 - Postcondició:
 - Hi han alumnes (a) actius en el sistema.
- crearFactura (mes)
 - Precondició:
 - Hi han alumnes (a) actius en el sistema.
 - El mes que s'introdueix és vàlid.
 - Postcondició:
 - El sistema crea la factura pel mes introduït i l'associa a un alumne de la llista (a).
 - Es mantenen els alumnes (a) actius en el sistema.
- fiFactura()
 - Precondició:
 - Hi han alumnes (a) actius en el sistema.
 - Postcondició:
 - El sistema conté les factures creades pels alumnes (a) actius.
 - No hi ha cap alumne actiu.

Diagrama de seqüència:

El loop que trobem en aquest diagrama de seqüència, ens servirà per crear la factura de cadascun dels alumnes que hi haurà a la llista d'alumnes.



Il·lustració 12 - Assistent Crear Factura

5.3.2.6 Assistent Crear Participacions

Cas d'ús:	Crear Participacions periòdiques.
Actors:	Directora, Educadora/Educadora de suport.
Descripció:	La finalitat d'aquest cas d'ús és de crear participacions entre el període indicat, per un o més alumnes.
Requeriments:	R.3.2 Crear participacions durant un interval de temps per l'alumne seleccionat.
Curs normal d'esdeveniments:	
Accions dels actors	Sistema
1. L'actor prem el botó "Alumnes".	
	2. El sistema mostra la vista arbre de tots els alumnes actius a l'escola.
3. L'actor selecciona els alumnes els quals vol crear participacions.	
4. L'actor seleccionar l'assistent "Crear Participacions Periòdiques" del menú accions.	
	5. El sistema obre la vista formulari d'aquest assistent.
6. L'actor seleccionar la data d'inici i la data final del període que vol introduir.	
7. L'actor introdueix les altres dades com ara els dies de la setmana dels quals vol crear participacions, l'horari, l'ús de l'hora d'acollida i l'ús del menjador.	
8. L'actor prem el botó "Afegir".	
	9. El sistema automàticament crea les participacions introduïdes entre l'interval de temps, per cadascun dels alumnes seleccionats.
	10. El sistema mostra de nou la vista arbre de tots els alumnes actius a l'escola.

11. L'actor prem el botó "Desar".	
	12. El sistema guarda les participacions creades pels alumnes seleccionats.
Curs alternatiu d'esdeveniments:	
6. El sistema retorna error si la data d'inici és més gran que la data final.	
9. En cas que un alumne ja tingui una participació per un dia introduït, el sistema no crearà aquesta participació.	
9. En cas que el dia de la participació sigui un dia de vacances o cap de setmana, el sistema no crearà aquesta participació.	

Contractes del sistema:

- iniciParticipacions (a: llista alumnes)
 - Precondició:
 - No hi ha cap alumne actiu.
 - Els alumnes de la llista (a) existeixen en el sistema.
 - Postcondició:
 - Hi han alumnes (a) actius en el sistema.

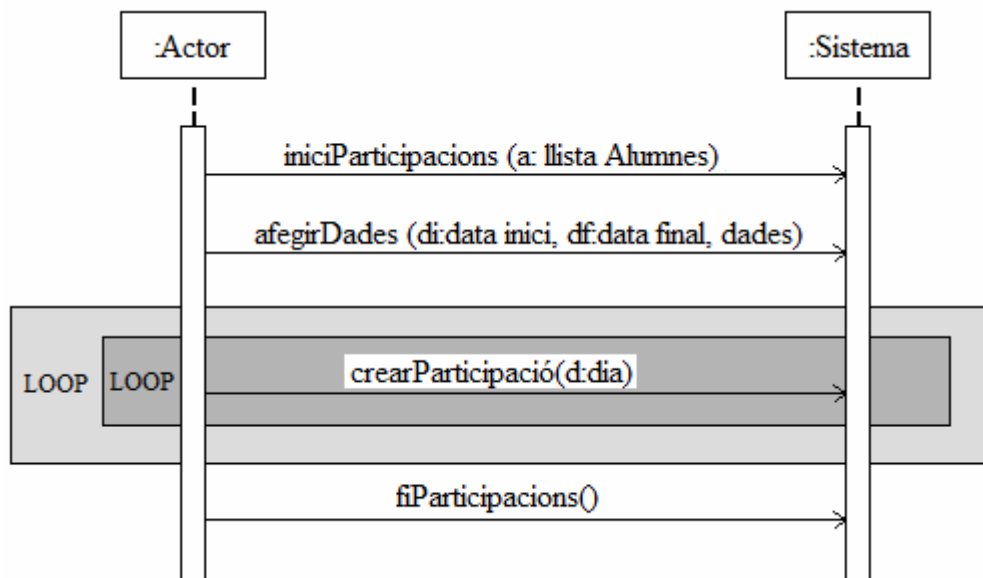
- afegirDades (di:data inici, df:data final, dades)
 - Precondició:
 - Hi han alumnes (a) actius en el sistema.
 - La data inici i la data final així com les altres dades que s'introdueix, són vàlides.
 - Postcondició:
 - Hi han unes dades en memòria que s'associaran a les participacions que es crearan pels alumnes (a) actius.
 - Es mantenen els alumnes (a) actius en el sistema.

- crearParticipació(d:dia)
 - Precondició:
 - Hi han alumnes (a) actius en el sistema.
 - Postcondició:
 - El sistema conté una nova participació amb dia (d) dins l'interval data inici (di) i data final (df). Aquesta nova participació conté les dades introduïdes i està lligada a un alumne de la llista (a).
 - Es mantenen els alumnes (a) actius en el sistema.

- fiParticipacions()
 - Precondició:
 - Hi han alumnes (a) actius en el sistema.
 - Postcondició:
 - El sistema conté les participacions dins l'interval de temps introduït per a tots els alumnes (a) actius.
 - No hi ha cap alumne actiu.

Diagrama de seqüència:

Com podem veure en aquest diagrama de seqüència, ens trobem dos loops seguits, el primer d'ells ens serveix per recórrer la llista d'alumnes, i el segon, per cadascun d'aquests alumnes crearem les participacions introduïdes.



Il·lustració 13 - Assistent Crear Participacions Periòdiques

6. Disseny

OpenERP incorpora per defecte uns 120 mòduls oficials aproximadament per gestionar una empresa, a més, se li han de sumar més de 350 mòduls extra que aporten més funcionalitats. D'aquests mòduls, s'ha de fer un estudi per veure si alguns d'ells es poden fer servir pel nostre propòsit. Segons les nostres necessitats, ens podem trobar en alguna d'aquestes situacions:

- Aprofitament del mòdul. Aquest cas tindrà lloc quan les nostres necessitats del sistema es garanteixin totalment amb algun dels mòduls oficials que ofereix OpenERP.
- Herència del mòdul. Quan les nostres necessitats del sistema no es compleixin totalment pels mòduls oficials d'OpenERP, podem adaptar-lo a les nostres necessitats mitjançant l'herència. D'aquesta manera, podrem afegir o treure el que necessitem pel nostre propòsit.
- Creació d'un nou mòdul. Quan cap mòdul satisfà les nostres necessitats, haurem de crear un nou mòdul.

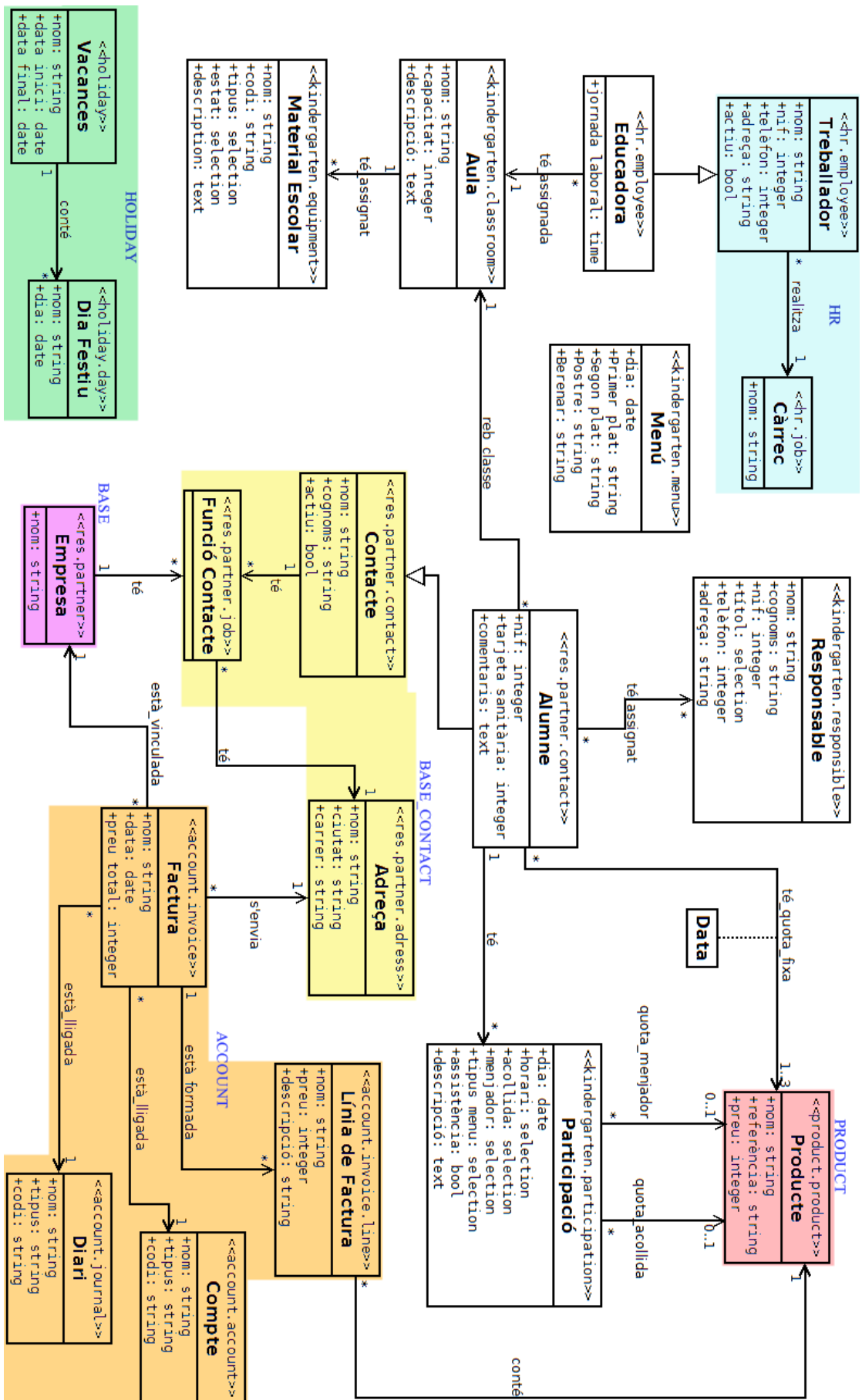
6.1 Aprofitament dels mòduls oficials i extres

Durant la fase de l'estudi de la tecnologia s'estudien els diferents mòduls dels que disposa OpenERP, fet això, podrem saber quins mòduls es poden ajustar a les nostres necessitats. Fruit d'aquesta fase i de l'anàlisi del problema, s'extreuen aquestes conclusions:

- El mòdul *base* d'openERP, el qual s'instal·la per defecte, aporta moltes possibilitats per a la gestió d'empreses i organitzacions. Es poden aprofitar algunes de les seves possibilitats com ara la gestió d'empreses, on per nosaltres, una empresa serà la família del alumne.
- A l'hora de gestionar les educadores, aprofitarem parcialment el mòdul *hr*, que serveix per la gestió de recursos humans. En el nostre cas, ens permet fer la gestió de les educadores, així com l'assignació de càrrecs dins l'escola.
- Per gestionar els alumnes, utilitzarem el mòdul *base_contact*, que serveix entre d'altres coses, per gestionar contactes i funcions. Nosaltres heretarem parcialment aquest mòdul, el qual aprofitarem per tractar els alumnes com contactes, i l'ampliarem per tal de adequar-lo a les nostres necessitats.

- Per gestionar el material escolar, en un principi es volia aprofitar parcialment els mòduls *product* i *stock*, que serveixen per gestionar productes i els seus stocks. Però aquests estan més pensats per empreses, i no ens permetia tractar cada producte per separat, sinó que tractava els productes per categories. Fet l'anàlisi del problema, es va veure que no era adequat pel nostre propòsit, així que finalment, s'ha creat de nou la gestió del material escolar i l'assignació d'aquests a les aules.
- Els mòduls *account* i *hr_timesheet_invoice*, ens ofereix la possibilitat de gestionar tota la part de comptabilitat i financera. D'aquesta manera, els podrem usar parcialment per tal de poder desenvolupar la facturació de l'escola.
- Per desenvolupar la facturació, també farem ús del mòdul *product*. Aquest mòdul ens servirà per desenvolupar les quotes, ja que aquestes són productes de tipus servei.
- Per gestionar les vacances, aprofitarem totalment el mòdul *holiday*, el qual ens permet gestionar les vacances escolars.
- Per gestionar les participacions del alumne es crearà un mòdul nou. D'aquesta manera, es crearà un nou objecte participació que ens servirà per controlar l'horari, menjador i acollida del alumne per un dia concret. A més, també ens servirà per controlar l'assistència dels alumnes.
- A l'hora de relacionar-ho tot i gestionar algunes parts pendents com ara les aules i els responsables dels alumnes, crearem un nou mòdul específic que serà el que englobarà tota la gestió de l'escola. Aquest mòdul s'anomenarà *kindergarten*.

A continuació, veurem un esquema que mostra en colors, tots els conceptes introduïts pèls mòduls d'OpenERP que s'aprofiten parcialment o totalment. La resta sense color, s'introdueixen de nou.



Il·lustració 14 - Esquema que mostra en color, els conceptes introduïts pels mòduls d'OpenERP

6.2 Diagrames de col·laboració

Els diagrames de col·laboració indiquen gràficament les relacions entre els diferents components per aconseguir els objectius de cada cas d'ús. Cada cas d'ús té una o més seqüències, i per cadascuna d'aquestes seqüències s'ha de fer el corresponent diagrama de col·laboració.

Com ja s'ha comentat a l'especificació, la majoria de diagrames de seqüència són força trivials, i es gestionen de manera semblant. A més, cal dir que OpenERP ja gestiona amb els seus propis mecanismes aquest tipus de relacions entre objectes.

S'ha de tenir en compte que tenim relacions entre objectes ja existents en el sistema, i que nosaltres hi estem afegint de noves amb el mòdul que hem creat. Dit això, mostrarem a continuació els diagrames de col·laboració d'alguns casos d'ús especificats a l'apartat anterior

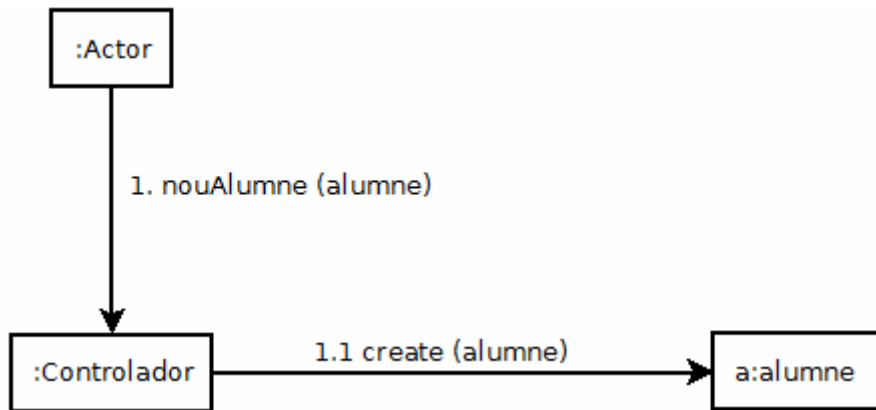
Abans de començar a fer el disseny, cal explicar la figura del controlador. Un dels principis bàsics del disseny és el principi GRASP (Patrons Generals de Software per Assignació de Responsabilitats), aquest ens diu entre d'altres coses, que en el sistema ha d'aparèixer un component que s'encarregui de capturar qualsevol esdeveniment del sistema, i delegar la responsabilitat al component que li pertorqui. És a dir, ha de tenir coneixement de tot el sistema.

En el nostre cas, OpenERP ja té definit el seu controlador intern per delegar les nostres peticions, però per tal de fer el disseny i representar les relacions en els diagrames de col·laboració, crearem un controlador (anomenat "controlador") per tal de delegar les responsabilitats. Com aquest controlador no apareix en el model conceptual, estem violant el principi d'espill, però d'aquesta manera evitem analitzar totes les relacions del model conceptual.

6.2.1 Cas d'ús: Crear nou alumne

6.2.1.1 *nouAlumne (alumne)*

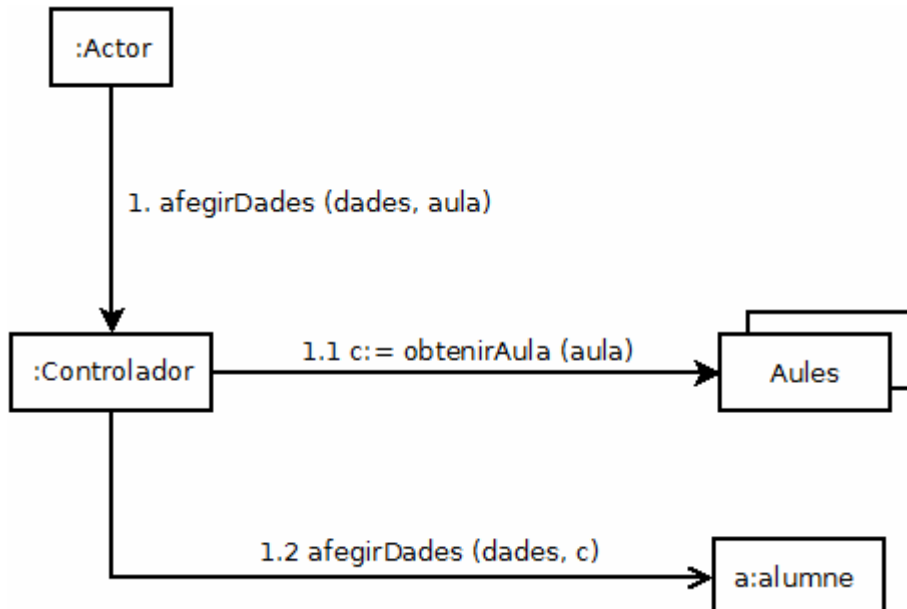
Aquest esdeveniment inicia el sistema per tal de poder donar d'alta un nou alumne amb el nom donat. El controlador rep la petició de l'actor, i aquest és l'encarregat de crear l'alumne, el qual es mantindrà actiu durant tot el cas d'ús.



Il·lustració 15 - Diagrama col·laboració nou alumne

6.2.1.2 *afegirDades (dades, aula)*

Aquest esdeveniment s'encarrega d'afegir les dades a l'alumne actiu, així com assignar-lo a una aula. El controlador rep la petició per part de l'actor, busca al repositori de aules, i un cop obtinguda l'aula, aquest és l'encarregat d'afegir les dades a l'alumne actiu i assignar-lo a l'aula obtinguda.

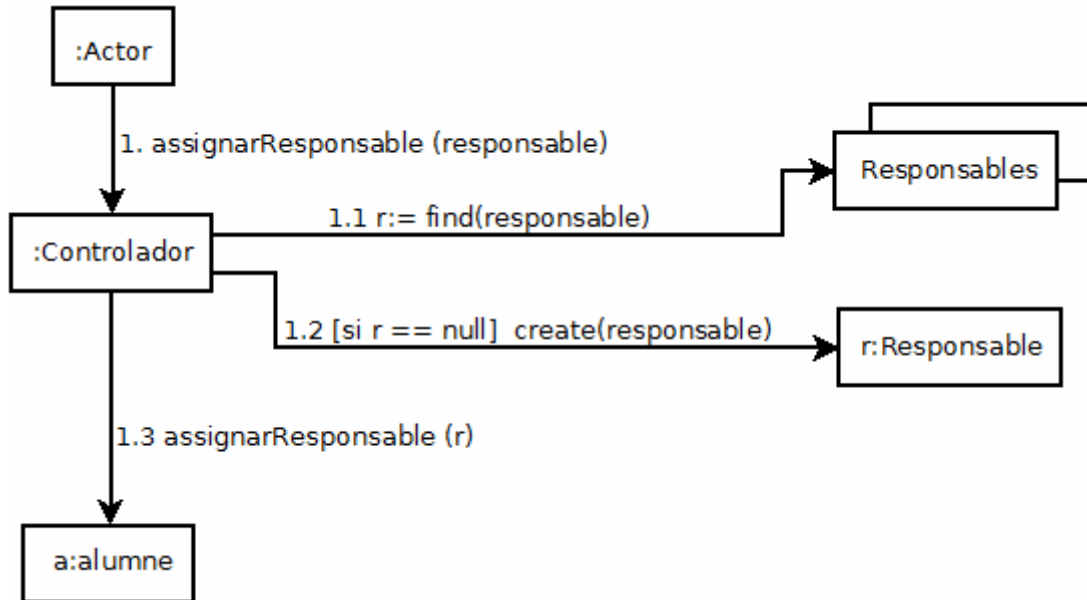


Il·lustració 16 - Diagrama col·laboració afegir dades

6.2.1.3 *assignarResponsable (responsable)*

En aquest esdeveniment assignem un responsable a l'alumne actiu. El controlador rep la petició d'assignar un responsable amb el nom donat, aquest controlador serà l'encarregat de buscar-lo al repositori de

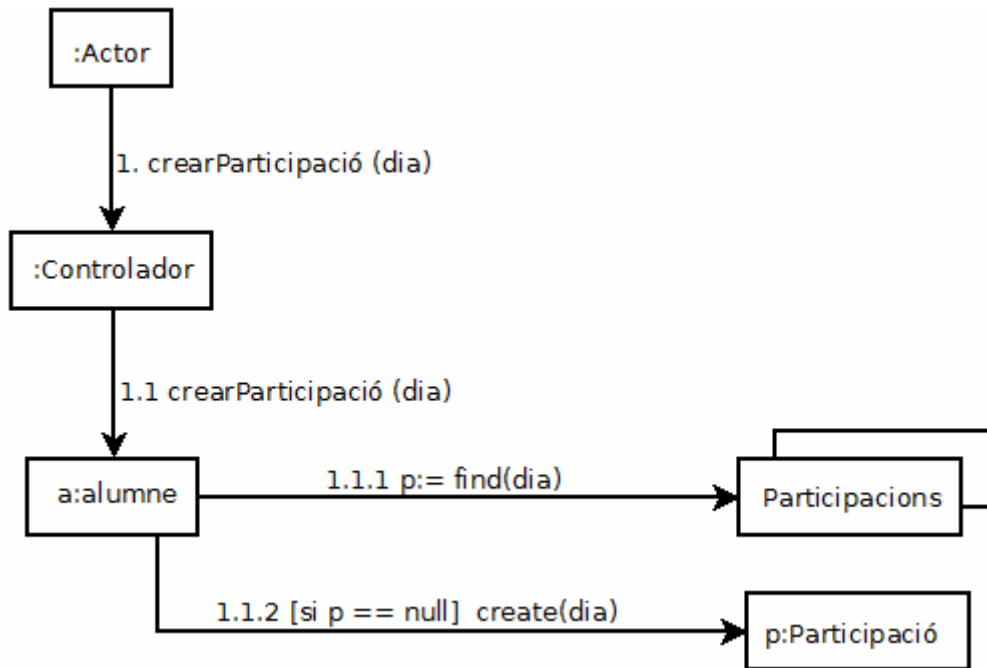
responsables, i en cas que aquest no existeixi, el mateix controlador crearà el responsable. Un cop obtingut, el controlador assignarà el responsable a l'alumne actiu. D'aquesta manera tindrem un repositori centralitzat de responsables, que s'ha de crear com a tard, en el moment de crear el controlador.



Il·lustració 17 - Diagrama col·laboració nou responsable

6.2.1.4 crearParticipació (dia)

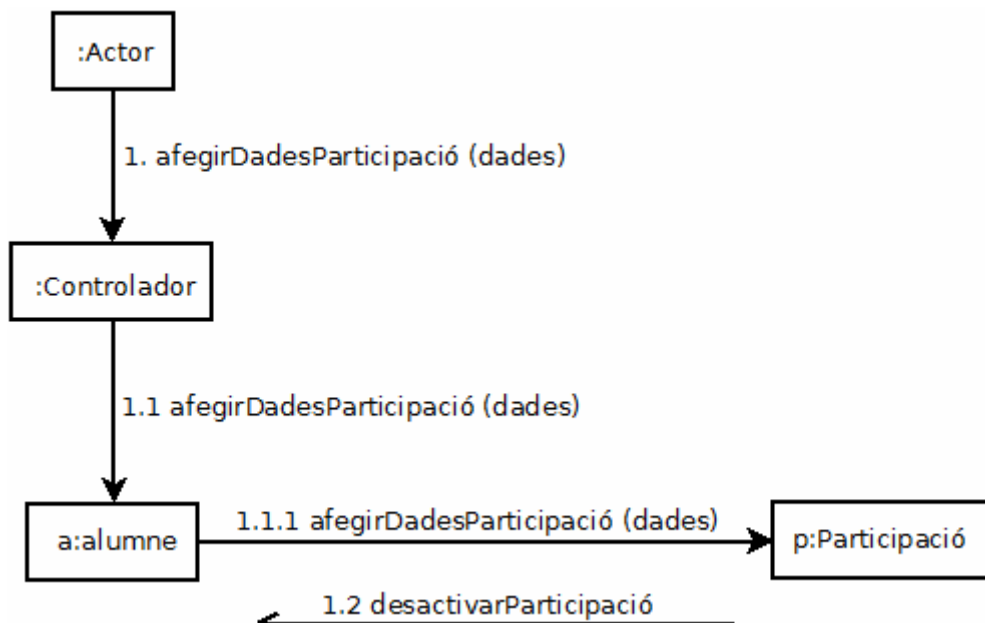
En aquest esdeveniment, crearem una participació per l'alumne actiu. El controlador rep la petició de crear una participació pel dia donat, i aquest, delega la responsabilitat a l'alumne actiu. L'alumne actiu serà doncs, l'encarregat de crear la participació, en cas que ja existeixi una participació pel dia donat, aquesta no es crearia. Tindrem doncs, un repositori de participacions fragmentat per alumnes, on cada alumne crearà el seu fragment de participacions. Aquesta nova participació creada es manté activa.



Il·lustració 18 - Diagrama col·laboració crear participació

6.2.1.5 *afegirDadesParticipació (dades)*

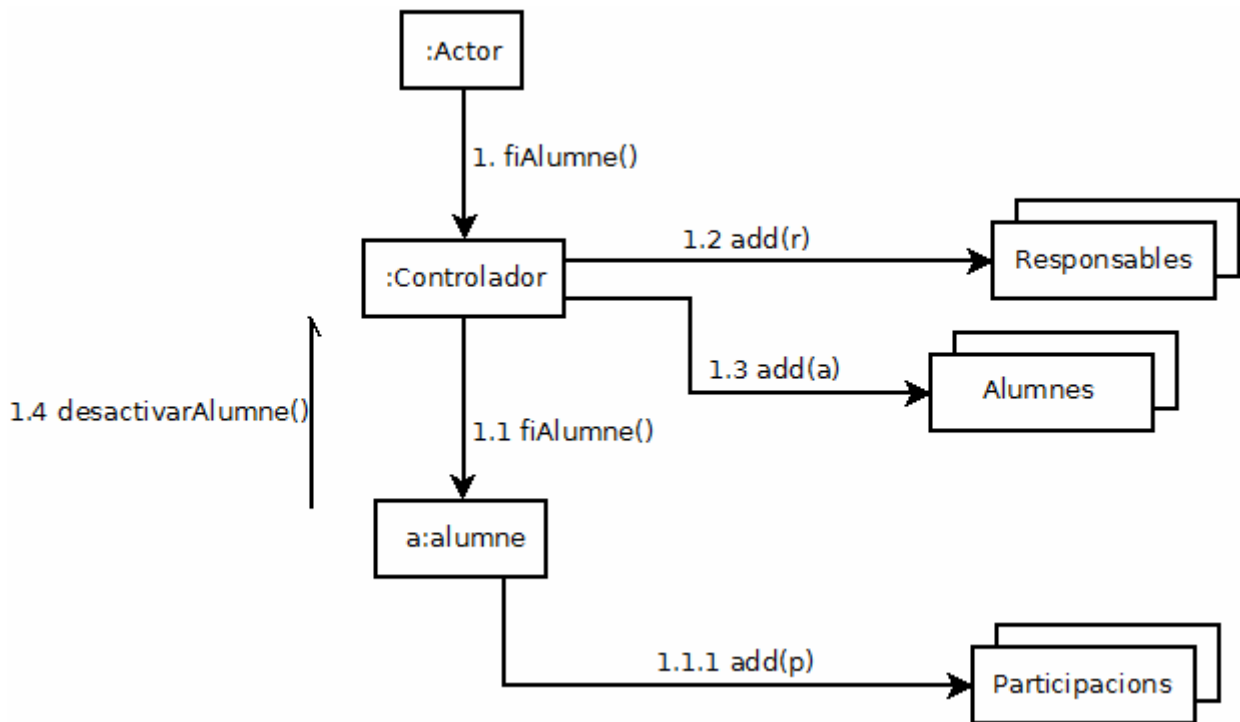
Aquest esdeveniment s'encarrega d'introduir les dades a la participació activa. El controlador rep la petició i delega la responsabilitat a l'alumne actiu, i aquest que n'és l'expert de participació, és l'encarregat d'afegir les dades a la participació activa. Un cop acabat, la participació deixa d'estar activa.



Il·lustració 19 - Diagrama col·laboració afegir dades participació

6.2.1.6 *fiAlumne ()*

Aquest esdeveniment, pel principi de la incorporació tardana, serà l'encarregat de fer tots els emmagatzematges pendents així com de finalitzar el cas d'ús. El controlador rep la petició, i primer, delega la responsabilitat a l'alumne actiu, que és l'encarregat de desar les participacions creades al sistema. Després serà el mateix controlador, que és l'expert de l'alumne i responsables, qui guarda la informació de l'alumne que ha estat actiu durant tot el cas d'ús, així com els responsables assignats a aquest alumne. Un cop guardat l'alumne actiu, aquest deixa d'estar actiu.

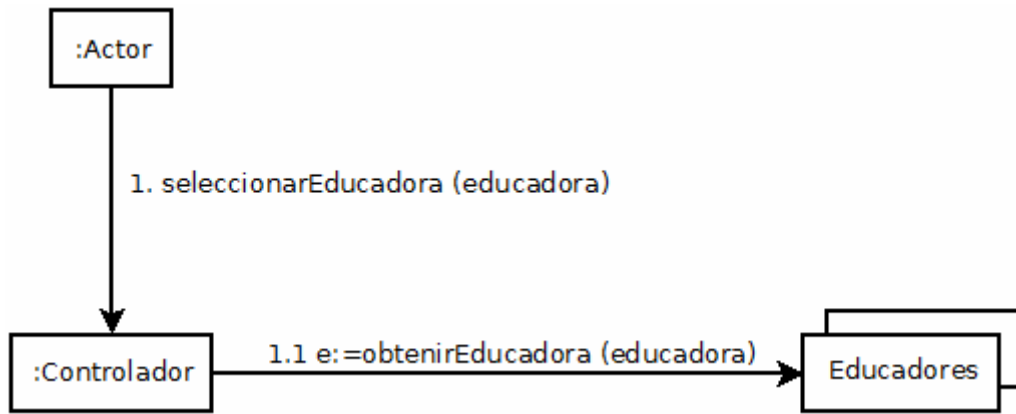


Il·lustració 20 - Diagrama col·laboració fi alumne

6.2.2 Cas d'ús: Modificar educadora

6.2.2.1 *seleccionarEducadora (educadora)*

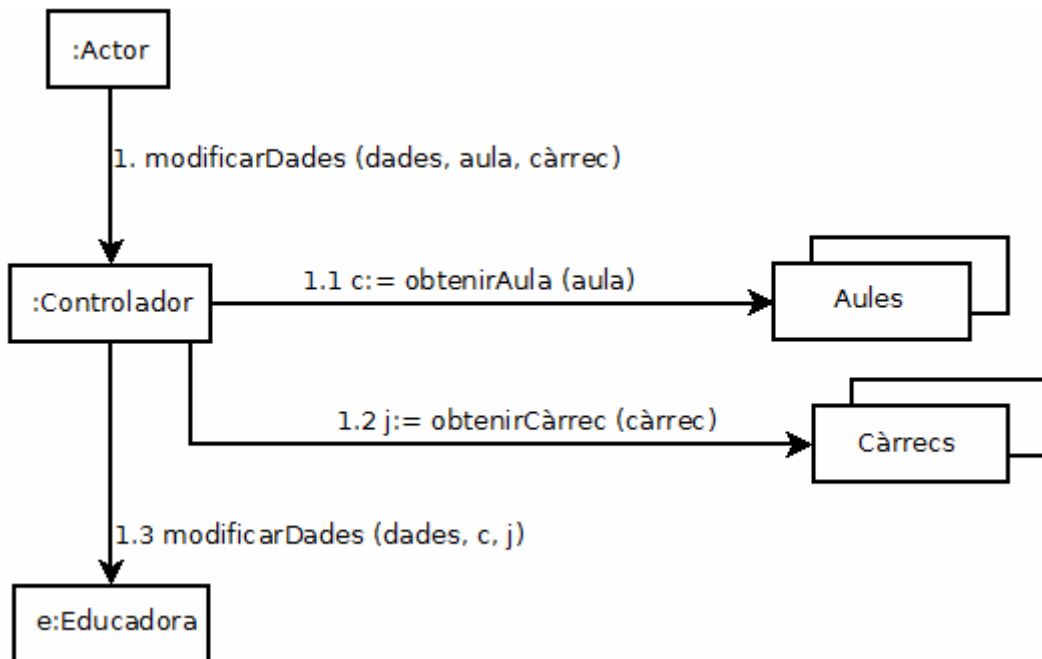
Aquest esdeveniment és l'iniciador del sistema a l'hora de modificar una educadora. El controlador rep la petició de l'actor, i ell mateix és l'encarregat de buscar l'educadora indicada. Aquesta educadora (e) obtinguda es mantindrà activa durant tot el cas d'ús.



Il·lustració 21 - Diagrama col·laboració seleccionar educadora

6.2.2.2 *modificarDades (dades, aula, càrrec)*

En aquest esdeveniment es rebran les noves dades, l'aula i/o el càrrec a modificar de l'educadora activa. El controlador rep la petició per part de l'actor, i aquest és l'encarregat d'obtenir l'aula i el càrrec dels repositoris pertinents. Un cop obtinguts, el controlador que n'és l'expert d'educadora, serà l'encarregat d'afegir les noves dades, aula i/o càrrec modificades a l'educadora activa.

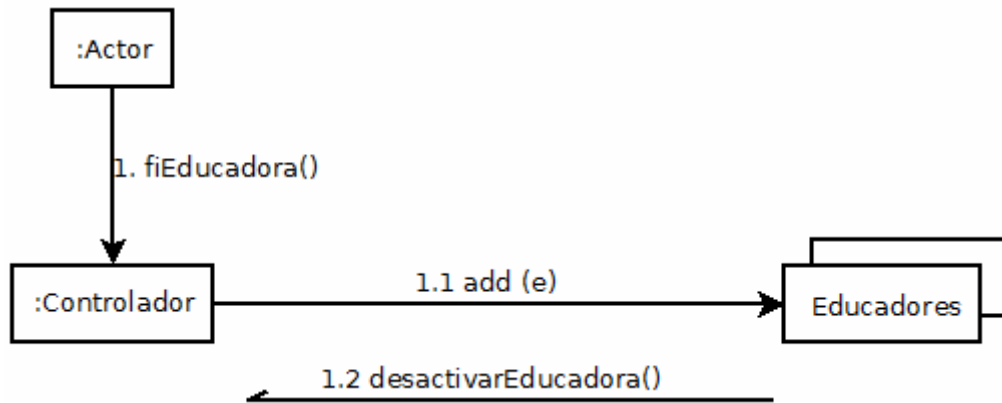


Il·lustració 22 - Diagrama col·laboració modificar dades

6.2.2.3 *fiEducadora ()*

Aquest esdeveniment s'encarregarà de finalitzar el cas d'ús. El controlador rep la petició per part de l'actor, i aquest que és l'expert de l'educadora, la desarà al sistema s'hagin o no modificat dades. Un cop guardada

l'educadora, aquesta es desactiva i deixà d'estar activa.

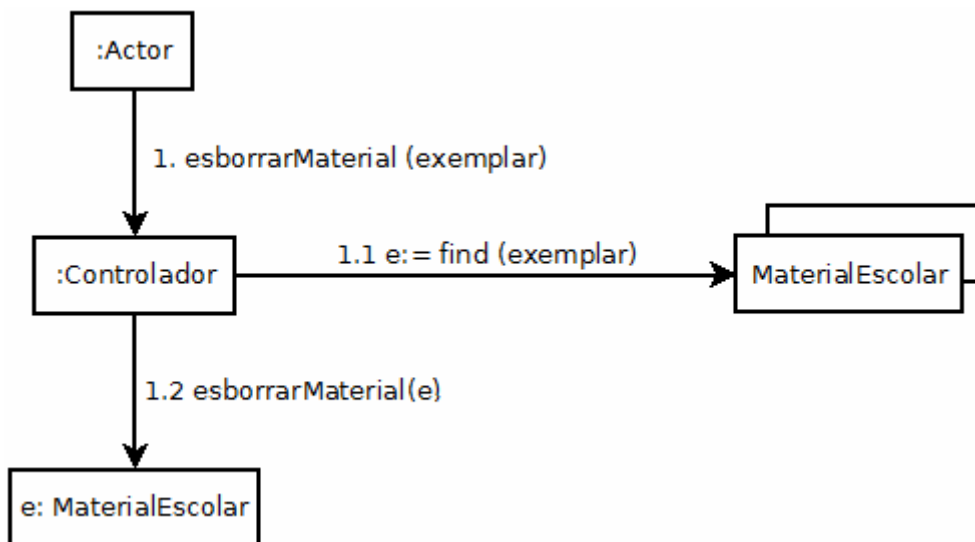


Il·lustració 23 - Diagrama col·laboració modificar dades

6.2.3 Cas d'ús: Esborrar Material

6.2.3.1 *esborrarMaterial (exemplar)*

Aquest esdeveniment s'encarrega d'eliminar l'exemplar de material escolar que l'actor hagi seleccionat. El controlador rep la petició per part de l'actor, i aquest és l'encarregat de buscar l'exemplar i eliminar-lo, tindrem doncs, que el controlador serà l'expert de material escolar.



Il·lustració 24 - Diagrama col·laboració esborrar material

6.3 Model de components

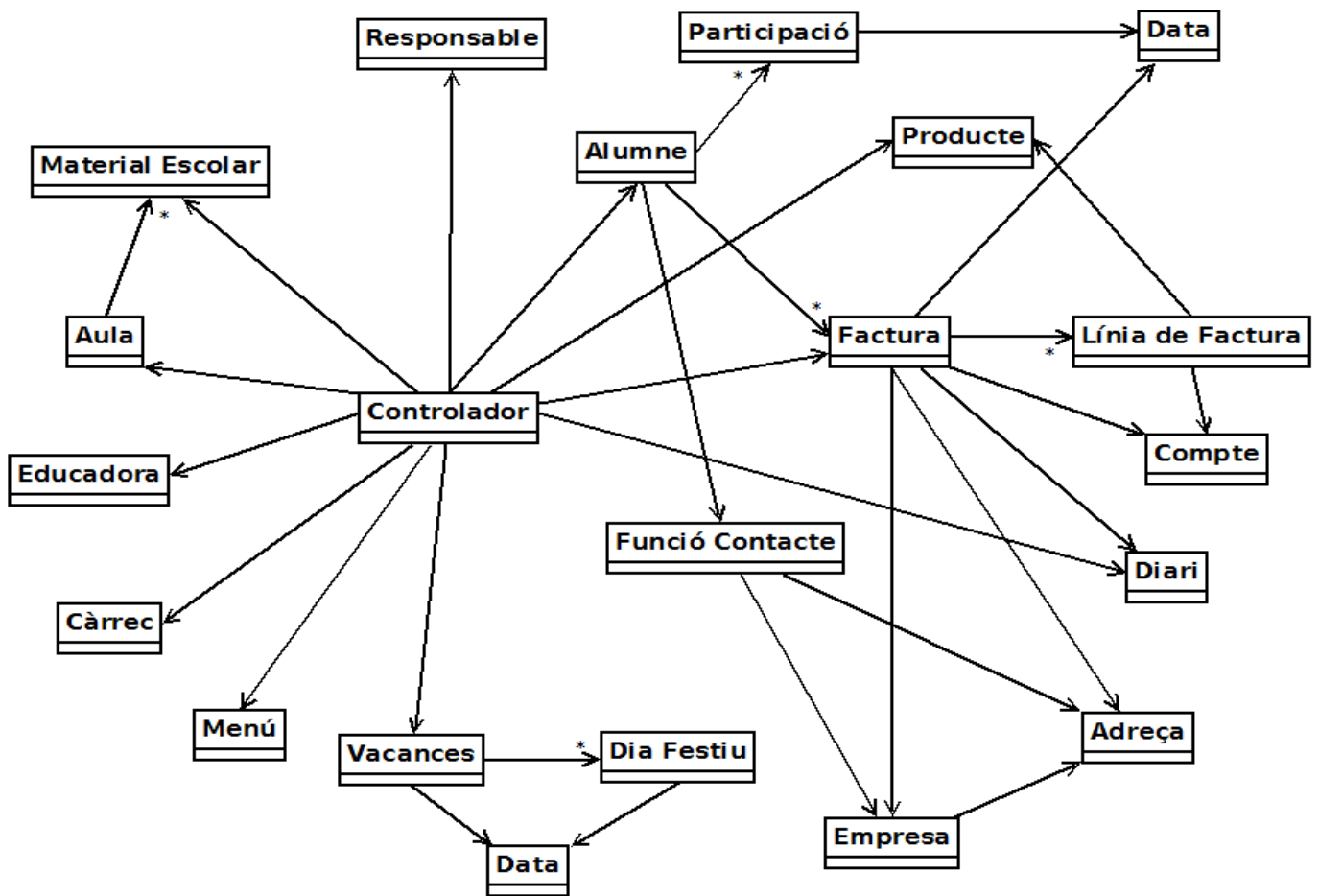
El model de components forma part del disseny i és la representació del model conceptual. En ell hi apareixen les representacions del model conceptual, així com les associacions importants per a la implementació de l'aplicació.

El primer que s'ha de fer és identificar les visibilitats entre components, és a dir, quins components es comuniquen entre ells. Per fer-ho més entenedor, dividirem el model de components en diferents parts, on en cadascuna d'elles veurem els diferents tipus de visibilitats possibles. A més, per millorar la claredat d'aquests models, s'ha omès la informació que conté cada objecte, informació que es pot trobar en el model conceptual del capítol 5.2 (il·lustració 6).

S'ha d'afegir que en algunes ocasions partim de mòduls ja creats, i les relacions que es poden heretar no es contemplen en aquest model.

6.3.1 Visibilitats d'atribut

En la següent il·lustració (25) veurem representades les visibilitats d'atribut entre els components del nostre sistema, on estan representades les visibilitats d'activació, recuperació i emmagatzematge. Les visibilitats marcades amb un asterisc, són aquelles que esdevenen en un repositori d'objectes.

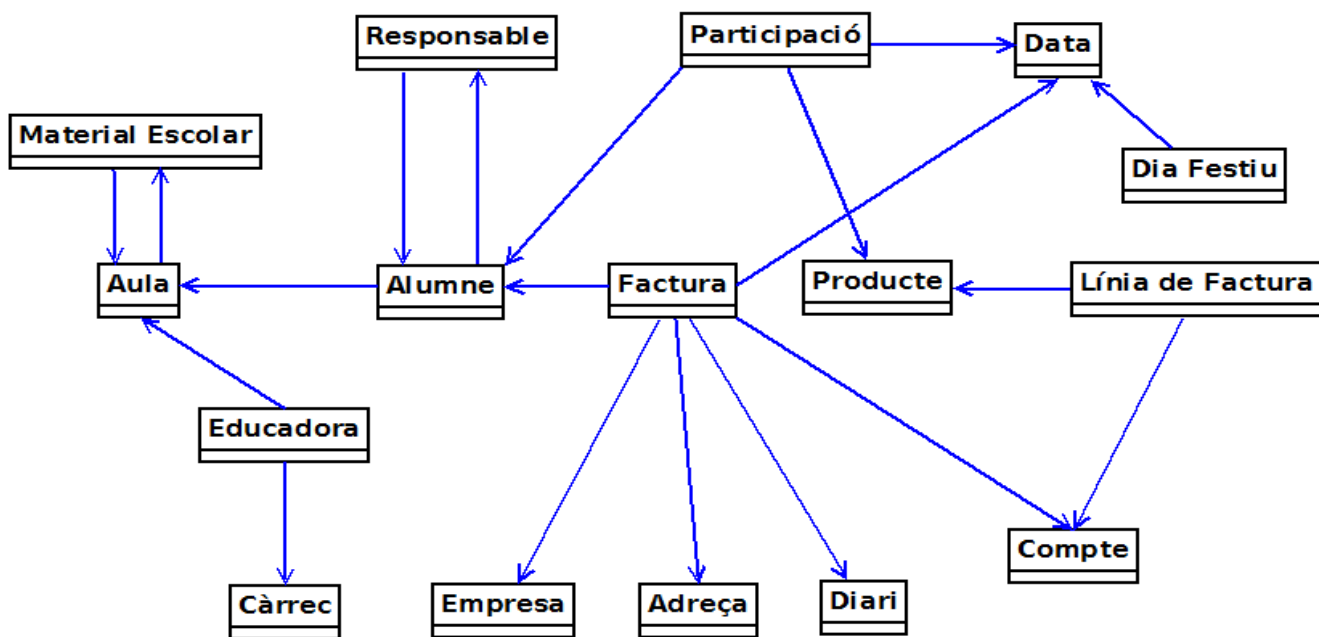


Il·lustració 25 - Visibilitats d'atribut

El component "Data" està repetit en aquest model de components per fer més entenedor l'esquema.

6.3.2 Visibilitats de paràmetre

En la següent il·lustració (26) podem veure representades les visibilitats de paràmetre, aquestes visibilitats neixen quan un objecte és passat a un altre com part d'un argument de l'operació, o bé quan és el valor de retorn de l'operació.



Il·lustració 26 - Visibilitats de paràmetre

6.3.3 Visibilitats locals

Pel últim existeixen les visibilitats locals, són aquelles on la seva validesa es limita a la realització d'una operació determinada, és a cada nou ús, quan es torna a crear de nou l'enllaç local. Pel que fa aquest tipus de visibilitat, no es crea cap en el nostre sistema.

7. Implementació

Una cop feta l'especificació en base als requeriments i dissenyada una solució al problema, comença l'etapa de la implementació. Aquesta s'ha de fer basant-nos en els conceptes especificats durant les etapes anteriors.

7.1 Versió d'OpenERP

Abans de començar aquesta etapa cal decidir amb quina versió d'OpenERP implementarem el projecte. OpenERP és un programa que està en continua actualització, encara que normalment són actualitzacions menors. A l'inici d'aquesta etapa em vaig trobar amb la decisió d'escollir entre la versió estable 5.0.16, en la que estava programat el mòdul *holiday* del que volia heretar les seves funcionalitats, o la nova versió 6.0 que acabava de sortir i que ens proporcionava moltes millores d'usabilitat. Però com a contra d'aquesta última versió, hi havien alguns canvis a nivell de codi respecte la versió 5.0, i per tant, caldria retocar i adequar el mòdul *holiday* a l'última versió.

Finalment i per recomanació del meu tutor, em vaig decidir per agafar la nova versió 6.0, que a més de les millores d'usabilitat, aquesta tindria un temps de suport més llarg de cara al futur.

Durant tot el projecte s'ha utilitzat la mateixa versió de Python i de PostgreSQL, en el cas de Python ha estat la versió 2.3 i en el cas de PostgreSQL ha estat la versió 8.3.

7.2 Mòduls desenvolupats

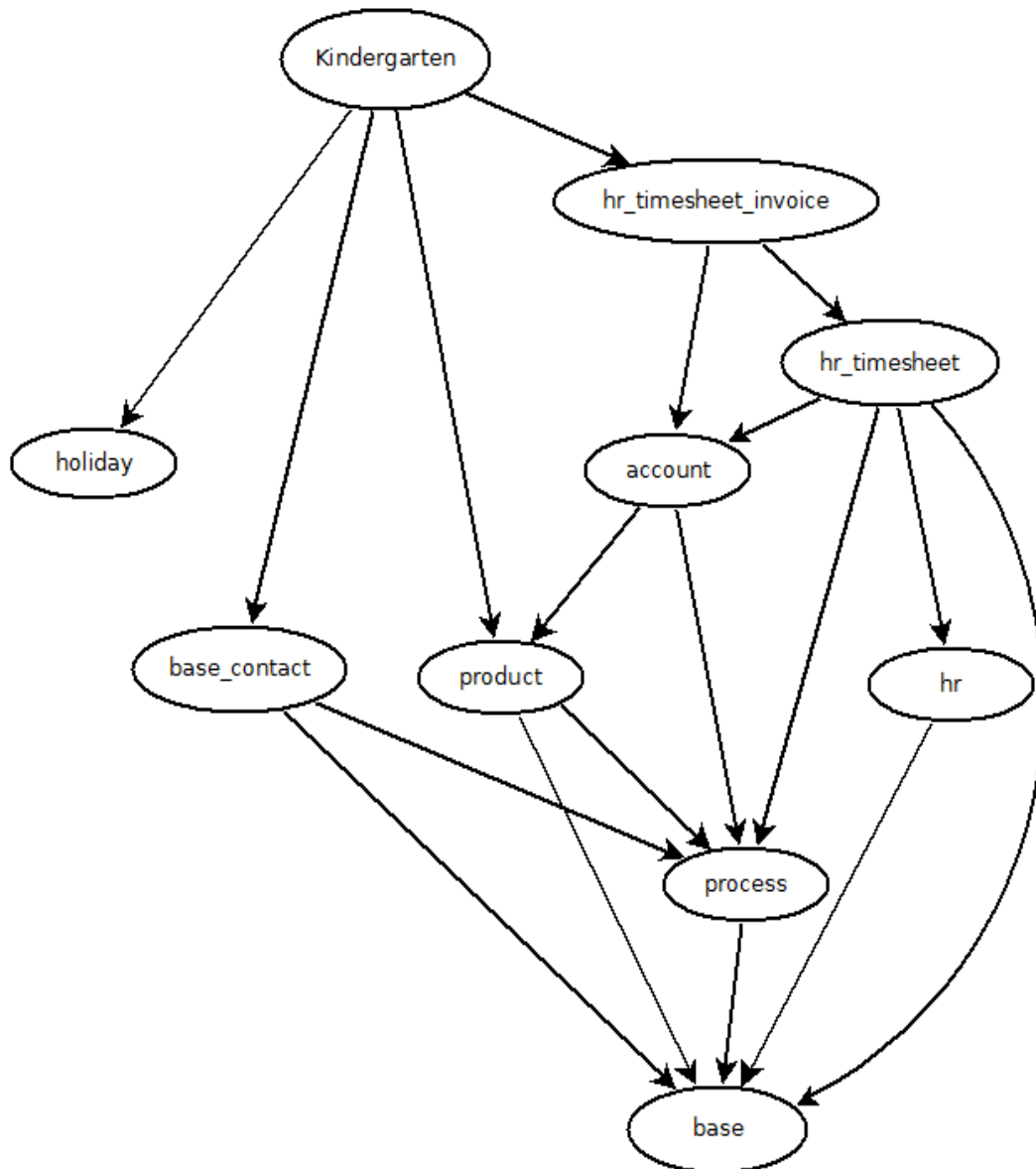
Durant el desenvolupament aprofitem codi que ja s'ha implementat en altres mòduls per reutilitzar-lo, aprofitant així el treball anterior i economitzant el temps. Ens trobem amb la situació que tots els mòduls que es té pensat heretar són els oficials i ja són compatibles amb la versió d'OpenERP 6.0, tots a excepció del mòdul *holiday* comentat abans. Així doncs, el primer que es va realitzar va ser l'adaptació a la versió 6.0 d'OpenERP del mòdul *holiday*, mòdul realitzat per un antic estudiant i del qual en podria heretar algunes de les seves característiques. Aquest mòdul es trobava fet amb la versió 5.0, i mitjançant alguns petits canvis de codi, es va fer compatible amb la versió 6.0.

7.2.1 Mòdul kindergarten

Aquest és el mòdul desenvolupat, i el que engloba tota la gestió de l'escola bressol. És on es farà la gestió dels alumnes, així com la informació que aquests contenen dels seus responsables, participacions i

facturació. Des d'aquest mòdul també es gestiona les educadores i els seus càrrecs, funció heretada i adaptada del mòdul hr. A més, també hi està afegit la gestió de les aules així com la del material escolar assignat a aquestes. Finalment, aquest mòdul disposa d'assistents a l'hora de fer consultes sobre els alumnes matriculats a l'escola, així com un assistent per fer la facturació d'aquests.

A continuació trobem una il·lustració on es pot observar els mòduls dels quals fem herència, i les seves dependències més importants.



Il·lustració 27 - Mòdul Kindergarten amb les seves dependències

Com s'ha explicat anteriorment en el capítol 3.3, OpenERP utilitza l'esquema MVC (Model Vista Controlador). A continuació veurem les definicions del MVC per a alguns dels objectes del nostre mòdul creat.

7.3 Model

En el model ens podem trobar diferents dades, bàsicament són de tres tipus possibles:

- Camps elementals. Són els camps bàsics com ara caràcters, enters, booleans, etc.
- Camps relacionals. Relaciona objectes del sistema. Aquests poden ser:
 - many2one: Relaciona un registre d'un objecte amb un registre d'un altre objecte.
 - one2many: Un registre d'un objecte es pot relacionar amb diferents registres d'un objecte. Necessita d'un many2one en l'objecte relacionat.
 - many2many: Relaciona diferents registres d'un objecte amb diferents registres d'un altre objecte.
- Camps funcionals. Aquests camps es calculen per valors d'altres camps i per defecte no s'emmagatzemen a la base de dades.

Vist això, veurem a continuació dos exemples de model, un d'ells és un objecte creat des de zero i l'altre un objecte heretat d'un altre mòdul.

7.3.1 Objecte Material Escolar

```
class kindergarten_equipment(osv.osv):
    #Objecte Material escolar del mòdul kindergarten
    _name = 'kindergarten.equipment'
    _description = 'Manage school equipment'
    _columns = {
        'name': fields.char('Name', size=30, required=True, help='Name or
title of the school equipment'),
        'code': fields.char('Code', size=30, required=True, help='Identified
code for this material'),
        'type': fields.selection([('children book', 'Children Book') ,
('music', 'Music') , ('toy', 'Toy') , ('plastic material', 'Plastic material') ,
('others', 'Others')] , 'Type' , required=True , help='Type of the school
material'),
        'state': fields.selection([('perfect', 'Perfect') , ('deteriorated',
'Deteriorated') , ('broken','Broken')] , 'State' , help='State of the school
equipment'),
        'description': fields.text('Description'),
        'classroom_id': fields.many2one('kindergarten.classroom' , 'Classroom
Assigned'),
    }
    _defaults = {
        'state': lambda *a: 'perfect',
    }
kindergarten_equipment()
```

Aquesta és la definició dels camps de l'objecte *kindergarten_equipment* que es troba al mòdul *kindergarten*, i que ens defineix el material escolar. En ell trobem definit el camp *_name*, que ens indica el nom de la taula que OpenERP mapejarà automàticament amb els camps introduïts. També hi ha el camp *_description*, on hi trobem una petita descripció de l'objecte.

El diccionari amb les dades és *_columns*, on es defineixen tots els valors de la taula creada, on s'especificarà cada camp que volem introduir dins d'una etiqueta. La definició d'un camp es realitza posant el nom del camp que volem definir entre cometes (*'nom_del_camp'*), seguit de dos punts i l'expressió *field.tipus*, on tipus és la definició del camp que volem introduir (camp elemental, relacional o funcional). Seguit del tipus de camp, s'introdueix la definició d'aquest camp entre parèntesis.

Per exemple el camp *'code'*, és de tipus *char* i a la interfície apareixerà amb el nom *'Code'*. Aquest té una llargada màxima de 30 caràcters i és obligatori introduir-lo per tal de crear el registre. A més, quan es seleccioni aquest camp amb el ratolí, apareixerà un missatge d'ajuda explicatiu d'aquest camp: *'Identified code for this material'*.

El camp *'type'* és un camp elemental de tipus *selection*, això significa que quan seleccionem aquest camp, ens apareixeran per escollir les opcions que hi ha definides, és a dir, llibre escolar, joguina, música, material plàstic i altres.

El camp *'classroom_id'* és un camp relacional que apunta a una altra taula de la base de dades, en aquest cas a *'kindergarten.classroom'*. Com és un camp many2one (molts a un), significa que aquest serà un camp on per un exemplar de material escolar, ens permetrà seleccionar l'aula a la que li volem assignar.

En aquest retall de codi, es poden apreciar alguns dels atributs que pot tenir una etiqueta. Per exemple l'atribut *required*, que serveix per indicar que el camp sigui obligatori i que calgui omplir-lo per tal de crear el registre. Si volem que sigui obligatori l'igualarem a *True*, en cas que no vulguem que sigui obligatori, no cal definir aquest atribut, ja que per defecte és *False*. En cas de l'atribut *readonly*, indiquem que el camp sigui o no només de lectura, i el funcionament és exactament igual que l'atribut *required*.

L'atribut *size*, ens serveix per definir la llargada dels camps de tipus *char*. I l'atribut *help*, ens serveix per definir una explicació per tal de que quan l'usuari apunti amb el ratolí a aquest camp, aparegui un missatge d'ajuda.

Per últim, dins el diccionari *_defaults*, es defineixen els valors per defecte que volem que apareguin quan es crea un nou registre. En el nostre exemple, quan vulguem crear un exemplar de material escolar, el camp

'type' ens apareixerà amb el valor 'Perfect' per defecte.

7.3.2 Objecte Alumne

```
class student(osv.osv):
    #Objecte Alumne heretat del objecte res.partner.contact del mòdul base_contact
    _description = 'Manage students'
    _inherit = 'res.partner.contact'
    _columns = {
        'classroom_id':fields.many2one('kindergarten.classroom', 'Classroom'),
        'nif': fields.char('Nif', size=9, help='National identity card'),
        'sanity_card': fields.char('Sanity Card', size=14, help='Sanity Card
Number'),
        'responsible_ids': fields.many2many('kindergarten.responsible'
,
'student_responsible_rel' , 'responsible_id' , 'student_id' , 'Responsible',
required=True),
        'participations_ids':fields.one2many('kindergarten.participation'
,
'student_id' , 'Participations'),
        'deposit':fields.boolean('Deposit paid?', help='Mark this option if the
deposit has been paid'),
        'student':fields.boolean('Student', help='Check this box if the contact is
a student'),
    }
student()
```

Aquesta és la definició de l'objecte alumne definit al mòdul *kindergarten*. Aquest hereta de l'objecte *res.partner.contact* que es troba al mòdul *base_contact*. Per indicar que és un objecte heretat, ho definim amb el camp *_inherit*, ficant-hi el nom de la taula que volem heretar. Això significa que nosaltres tindrem disponibles tots els camps definits a la taula *res.partner.contact*, més els camps afegits al diccionari *_columns* de l'objecte alumne. D'aquesta manera complementem els camps que manquen per satisfer les nostres necessitats.

En aquest exemple també podem trobar altres casos de camps relacionals, com ara el camp *'participations_ids'* definit com a *one2many*. Aquest apuntarà a una altra taula de la base de dades, en aquest cas a *'kindergarten.participation'*, i ens permetrà veure les participacions assignades a l'alumne que estigui actiu.

7.4 Controlador

El controlador com hem dit anteriorment, és la part lògica del mòdul, i comparteix arxiu amb el model de dades. Aquest respon a les funcions iniciades per l'usuari o bé pel sistema, ja sigui actualitzant les dades del model o realitzant les tasques pertinents. Quan creem una classe, aquesta ja té unes classes

predefinides, com pot ser *create*, *write*, *unlink*, etc. Però aquestes classes es poden redefinir per tal de adaptar-les a les nostres necessitats, o bé afegir alguna funció més.

Veiem doncs un exemple de controlador, en aquest cas, d'un objecte creat des de zero.

7.4.1 Objecte Participació

```
class participation(osv.osv):
    #Objecte Participació del mòdul kindergarten
    def _extra_hour_get(self, cr, uid, context={}):
        obj = self.pool.get('product.product')
        ids = obj.search(cr, uid, [('default_code','=', 'ACOLLIDA')])
        res = obj.read(cr, uid, ids, ['default_code', 'name'], context)
        return [(r['default_code'], r['name']) for r in res] + [('no', 'No')]

    def _dining_get(self, cr, uid, context={}):
        obj = self.pool.get('product.product')
        ids = obj.search(cr, uid, [('default_code','=', 'MENJADOR')])
        res = obj.read(cr, uid, ids, ['default_code', 'name'], context)
        return [(r['default_code'], r['name']) for r in res] + [('no', 'No')]

    _name = 'kindergarten.participation'
    _description = 'Manage participations, including schedule and use of the dining'

    _columns = {
        'day':fields.date('Day', required=True),
        'schedule': fields.selection([('morning','Morning: 8:45h a 12:00h') , ('morning
and midday','Morning and Midday: 8:45h a 15:00h') , ('all day','All Day: 8:45h a
17:00h')]) , 'Schedule', help='Student schedule'),
        'extra_hour':fields.selection(_extra_hour_get, 'Extra hours', method=True,
help='Used to know if the student has assisted to the first hour'),
        'dining':fields.selection(_dining_get, 'Extra services', method=True, help='Used
to know if the student has assisted to the dining'),
        'menu_type': fields.selection([(' ',' ') , ('normal','Normal') , ('trituated',
'Trituated') , ('diet', 'Diet') , ('muslim', 'Muslim (without pig)')]) , 'Menu type',
help='Type of the menu'),
        'assistance':fields.boolean('Assistance', help='Used to know if the student has
been assisted to the kindergarten'),
        'description':fields.text('Description'),
        'student_id': fields.many2one('res.partner.contact', 'Student', ondelete=
'cascade'),
    }
    _defaults = {
        'assistance' : lambda *a: True,
        'extra_hour' : lambda *a: 'no',
        'dining' : lambda *a: 'no',
        'menu_type' : lambda *a: ' ',
    }
    _order = "day"

    def create(self, cr, uid, vals, context=None):
        rep = self.search(cr, uid, [('student_id' , '=', vals['student_id']] ,
('day', '=', vals['day']]))
        if len(rep) == 0:
            return super(osv.osv,self).create(cr, uid, vals, context)
```

```

def write(self, cr, uid, ids, vals, context=None):
    stu_id = self.read(cr, uid, ids[0], ['student_id']) ['student_id'][0]
    part_ids = self.search(cr, uid, [('student_id','=', stu_id), ('day', '=',
vals['day'])])

    p = True
    for x in self.read(cr, uid, part_ids, ['day']):
        if vals['day'] == x['day']:
            p = False

    if p:
        return super(osv.osv,self).write(cr, uid, ids, vals, context)
    else:
        return super(osv.osv,self).write(cr, uid, ids, {
            'schedule':vals['schedule'],
            'extra_hour':vals['extra_hour'],
            'dining':vals['dining'],
            'menu_type':vals['menu_type'],
            'assistance':vals['assistance'],
            'description':vals['description'],
        }, context)

participation()

```

En aquest retall de codi hi tenim l'objecte participació, i com es pot veure, model i controlador comparteixen el mateix arxiu. Per explicar el controlador en aquest cas, no s'ha extret el codi de la part del model per tal que es pugui veure d'on prové cada camp. En el codi hi trobem quatre funcions:

- Funció `'_extra_hour_get'`. Aquesta funció està creada des de zero per satisfer les necessitats d'aquest objecte. Aquesta ens serveix per quan nosaltres a la vista participació seleccionem l'opció hores extra (`'extra_hour'`), on ens apareixerà un desplegable amb el producte amb codi `'ACOLLIDA'`. El sistema està definit de tal manera que aquest producte aparegui per defecte quan s'instal·li el mòdul kindergarten, i a més, sigui impossible crear un producte amb el mateix codi o bé, eliminar aquest producte. De tal manera, que en aquests desplegable, sempre ens apareixerà només l'opció d'acollida o bé, l'opció `'No'` en cas que no vulguem escollir aquest servei.
- Funció `'_dining_get'`. Aquesta funció també està creada des de zero, i el funcionament és exactament igual que la funció esmentada abans, amb la diferència que aquesta ens servirà per escollir el servei de menjador.
- Funció `'create'`. Aquesta és una funció predefinida que es crida quan es crea un nou objecte, aquesta la podem redefinir ficant-hi el codi que volem que s'executi en el moment de la creació. En el nostre exemple, el que fem es buscar de l'alumne actiu, les participacions que té pel dia que hem introduït, i les guardem a la llista `rep`. En cas, que aquesta llista contingui 0 elements (es a dir, no tingui participació), es crida al

super(osv.osv,self).create(cr, uid, vals, context) que el que fa, és cridar a la funció 'create' per defecte. Si es dóna el cas que la llista *rep* ja conté un element, el sistema no fa res, ja que només hi pot haver una participació per dia.

- Funció 'write'. Aquesta també és una funció predefinida que es crida quan es modifica un objecte, i com en el cas anterior, la podem redefinir. En el nostre exemple el que fem és semblant a l'anterior funció, quan es modifica el camp dia, es busca si existeix una participació pel nou dia introduït, en el cas que no existeixi es crida a la funció 'write' per defecte. Però si es dóna el cas que el dia modificat ja existeix, no es modificarà el camp dia de l'objecte participació, i només es modificaran les altres dades de la participació que s'hagin modificat (horari, acollida, menjador, etc.).

7.5 Vista

Les vistes en OpenERP estan definides en un arxiu XML. Aquest arxiu, mitjançant el llenguatge de les marques extensibles, es relaciona amb l'arxiu Python que conté la lògica i les dades que hem vist anteriorment. A les vistes es defineixen els elements del model que es vol que es mostrin, això es fa mitjançant etiquetes, ficant-hi una al començament i una altra al final de la definició. A més, es poden posar-hi atributs com ara "*required=true*", que significa que aquest camp és obligatori i es senyalitza amb color blau a la vista del usuari.

OpenERP disposa de cinc tipus de vistes diferents, aquestes són: Formulari (form), llista (tree), calendari (calendar), gràfic (graph) i cerca (search). Aquestes es defineix indicant el tipus de vista enmig de l'etiqueta "*<field name='type'> </field>*".

En aquest projecte s'han utilitzat les vistes arbre, formulari i cerca. A continuació mostrem uns exemples d'aquestes vistes així com d'altres elements característics.

7.5.1 Vista arbre

```
<record model="ir.ui.view" id="view_kindergarten_student_tree">
  <field name="name">res.partner.contact.tree</field>
  <field name="model">res.partner.contact</field>
  <field name="type">tree</field>
  <field name="inherit_id" ref="base_contact.view_partner_contact_tree"/>
  <field name="arch" type="xml">
    <data>
      <field name="first_name" position="after">
```

```

        <field name="classroom_id"/>
        <field name="nif"/>
        <field name="sanity_card"/>
    </field>
    <field name="mobile" position="replace"/>
    <field name="email" position="replace"/>
    <field name="partner_id" position="replace"/>
    <field name="function" position="replace"/>
</data>
</field>
</record>

```

Aquest fragment de codi correspon a la vista llista (arbre) de l'objecte alumne. A la primera línia indiquem l'identificador d'aquesta vista el qual ha de ser únic. Tot seguit hi ha unes línies que serveixen per definir aquesta vista, on definim el nom que nosaltres li donem a la vista indicat per l'etiqueta *"name"*. A continuació a l'etiqueta *"model"* definim el nom de la taula de la base de dades, i a l'etiqueta *"type"* definim el tipus de vista.

En aquest exemple, la vista és heretada i això ho indiquem amb l'etiqueta *"inherit"* seguit del lloc on es troba aquesta vista, especificat amb el nom del mòdul i l'identificador que té la vista que volem heretar.

La definició dels camps s'escriu tot seguit de la línia `<field name="arch" type="xml">`, i com és una vista heretada haurem d'especificar on volem afegir els camps, això vindrà indicat per l'atribut *position*. En el nostre cas, estem dient que després del camp *first_name* (`position="after"`), volem afegir els camps *classroom_id*, *nif* i *sanity card*.

Al ser una vista heretada, es pot donar el cas que hi hagin camps que no volem que apareguin, això ho indicarem ficant-hi l'atribut `position="replace"` darrera del nom dels camps que volem treure.

A continuació veiem com es veu aquesta vista arbre:

Cognoms	Primer nom	Aula	Nif	Targeta Sanitària	Idioma
Checa Galera	Sara	P0 - Timbals	61734256J	CHGA 123412344	Catalan / Català
Dupont	Christopher	P0 - Timbals	33462345L	DUCH 454545454	Catalan / Català
Lossius Millet	Victor	P0 - Timbals	46243159A	LOMI 232323232	Catalan / Català
Perez Coll	Monica	P1 - Clarinets	83527425H	PECO 676767677	Catalan / Català

Il·lustració 28 - Vista arbre de l'objecte alumne

7.5.2 Vista formulari

```
<record model="ir.ui.view" id="view_kindergarten_classroom_form">
  <field name="name">kindergarten.classroom.form</field>
  <field name="model">kindergarten.classroom</field>
  <field name="type">form</field>
  <field name="arch" type="xml">
    <form string="kindergarten.classroom">
      <field name="name"/>
      <field name="square_metres"/>
      <field name="capacity"/>
      <newline/>
      <notebook colspan="4">
        <page string="Equipment assigned">
          <field name="equipment_ids" nolabel="1">
            <tree string="School Equipment">
              <field name="name"/>
              <field name="code"/>
              <field name="type"/>
              <field name="state"/>
            </tree>
            <form string="School Equipment">
              <separator string="Material Information" colspan="2"/>
              <newline/>
              <field name="name"/>
              <field name="code"/>
              <newline/>
              <field name="type"/>
              <field name="state"/>
              <newline/>
              <separator string="Extra information" colspan="4"/>
              <field name="description" colspan="4" nolabel="1"/>
            </form>
          </field>
        </page>
        <page string="Extra information">
          <field name="description" nolabel="1"/>
        </page>
      </notebook>
    </form>
  </field>
</record>
```

Aquest fragment de codi correspon a la vista formulari de l'objecte aula, objecte el qual ha estat creat des de zero. Les primeres línies són per definir l'identificador, nom, model i el tipus de vista, igual que en el cas anterior de la vista arbre.

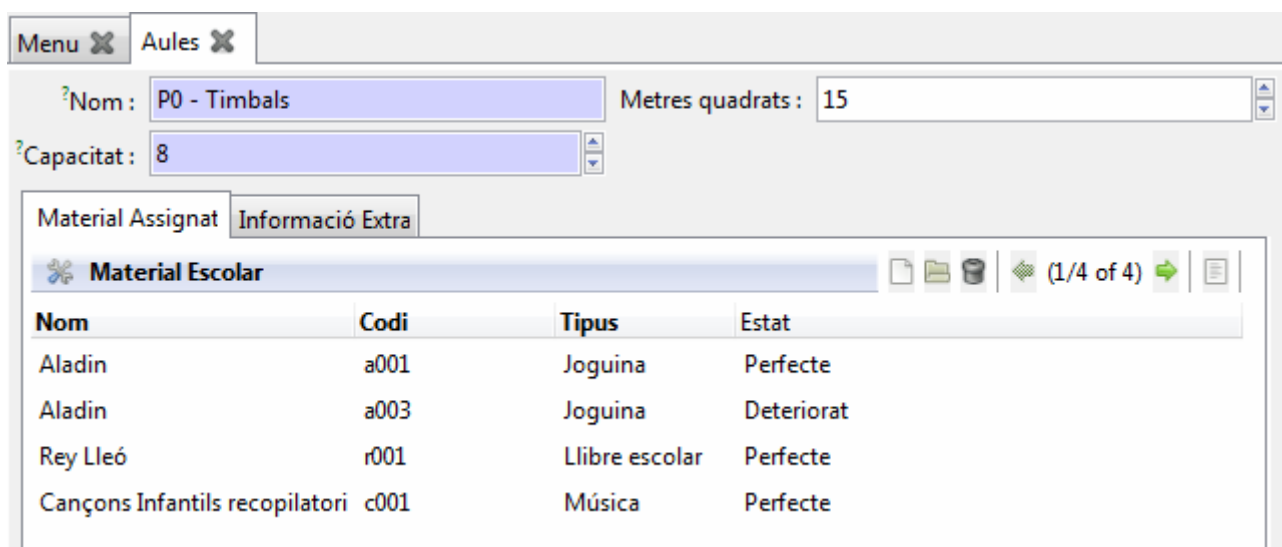
Tot seguit es defineixen els camps que es volen mostrar de la vista formulari. OpenERP per defecte organitza l'estructura de la vista en quatre columnes, cada camp per defecte ocupa dos columnes, un per l'etiqueta i un altra pel camp en si. Així que en cas que nosaltres no especifiquem la estructura que volem mostrar, per defecte ens apareixeran dos camps per línia de la següent manera: etiqueta1 + camp1 + etiqueta2 + camp2.

Com en el cas de la vista arbre, els camps vindran definits després de la línia `<field name="arch" type="xml">`, i on especificarem cada camp que volem mostrar dins d'una etiqueta. En aquest cas, a més de l'etiqueta per definir els camps ens hi trobem uns altres tipus de etiquetes. Com ara l'etiqueta *newline*, que ens serveix per fer un salt de línia, o l'etiqueta *notebook*, que ens serveix per obrir un contenidor de pestanyes, on cada pestanya vindrà definida per l'etiqueta *page*.

En aquest cas (`<page string="Equipment assigned">`), definim la pestanya material assignat, dins la qual afegim un camp one2many anomenat *equipment_ids*, que conté dues vistes, una vista arbre i una formulari. Això ho definim amb les etiquetes *tree* i *form* respectivament, dins les quals afegirem els camps que vulguem mostrar com s'ha explicat fins ara.

En aquest fragment de codi apareixen altres atributs de etiquetes, com a la línia `<field name="description" colspan="4" nolabel="1"/>`, on amb l'atribut *colspan* estem dient el número de columnes que volem que ocupi aquest camp; i amb l'atribut *nolabel*, quan és igual a 1 estem dient que no volem que aparegui el nom d'aquest camp, només ens apareixerà el camp en si.

La vista formulari la podem personalitzar amb diferents elements, per exemple amb l'etiqueta `<separator string="Extra information" colspan="4"/>`, la qual ens servirà per crear una línia que juntament amb els atributs *string* i *colspan*, podrem definir el nom que volem que aparegui sobre aquesta línia i la llargada d'aquesta respectivament.



Nom	Codi	Tipus	Estat
Aladin	a001	Joguina	Perfecte
Aladin	a003	Joguina	Deteriorat
Rey Lleó	r001	Llibre escolar	Perfecte
Cançons Infantils recopilatori	c001	Música	Perfecte

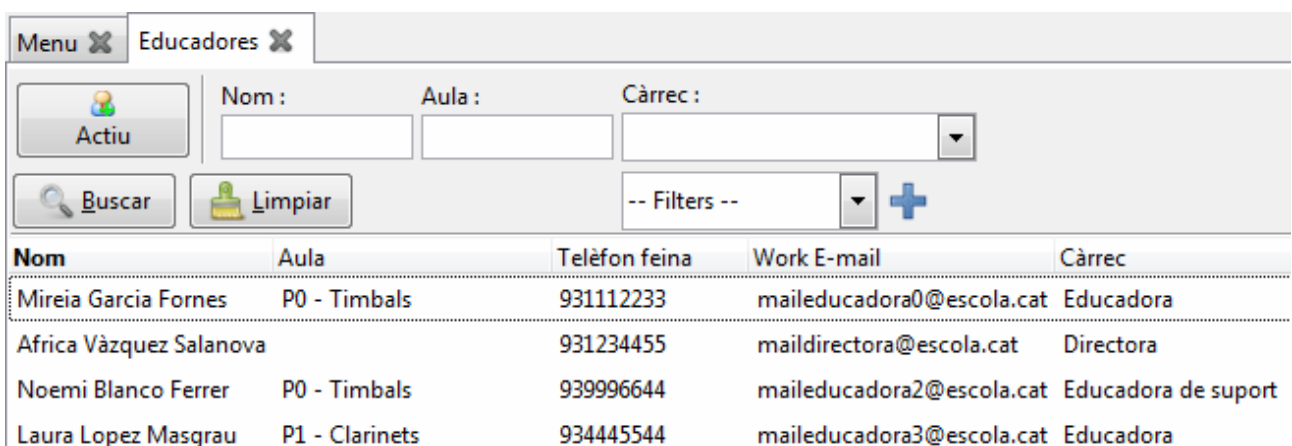
Il·lustració 29 - Vista formulari de l'objecte aula

7.5.3 Vista cerca

```
<record model="ir.ui.view" id="view_kindergarten_educator_filter">
  <field name="name">Employees</field>
  <field name="model">hr.employee</field>
  <field name="type">search</field>
  <field name="inherit_id" ref="hr.view_employee_filter"/>
  <field name="arch" type="xml">
    <data>
      <field name="name" position="after">
        <field name="classroom_id"/>
      </field>
      <field name="department_id" position="replace"/>
      <field name="parent_id" position="replace"/>
      <group string="Group By..." position="replace"/>
    </data>
  </field>
</record>
```

Amb la versió 6.0 d'OpenERP, apareix una nova opció que ens permet modificar la vista cerca, que amb anteriors versions no existia i estava molt més limitada la cerca de registres. El funcionament és el mateix que amb la vista arbre o formulari, però en aquest cas, definim els camps que volem que apareguin a la vista cerca, i aquesta es situa sobre la vista arbre del mateix objecte.

Aquest fragment de codi correspon a la vista cerca de l'objecte educadora, heretat de la taula *employee* del mòdul *hr*. Afegirem el camp de cerca *classroom_id* i al ser una vista heretada, traurem els camps que no volem que apareguin. Podem veure com queda aquesta vista cerca a la següent il·lustració, situat sobre la vista arbre:



Nom	Aula	Telèfon feina	Work E-mail	Càrrec
Mireia Garcia Fornes	P0 - Timbals	931112233	maileducadora0@escola.cat	Educadora
Africa Vázquez Salanova		931234455	maildirectora@escola.cat	Directora
Noemi Blanco Ferrer	P0 - Timbals	939996644	maileducadora2@escola.cat	Educadora de suport
Laura Lopez Masgrau	P1 - Clarinets	934445544	maileducadora3@escola.cat	Educadora

Il·lustració 30 - Vista cerca de l'objecte educadora

7.5.4 Accions

```
<record model="ir.actions.act_window" id="action_kindergarten_classroom">
  <field name="name">Classrooms</field>
  <field name="res_model">kindergarten.classroom</field>
  <field name="view_type">form</field>
  <field name="view_mode">tree,form</field>
</record>
```

Aquest codi correspon al menú acció de l'objecte aula.

Les accions ens serveixen per definir quines són les vistes disponibles d'un objecte. En aquest cas, amb l'etiqueta *view_mode* estem dient que per l'objecte aula, hi ha dos tipus de vistes: arbre i formulari. Al accedir per defecte, ho farem a la vista arbre, ja que és la primera que s'ha afegit com a valor de l'etiqueta.

7.5.5 Menús

```
<menuitem name="Students Management" id="menu_configuration_students"
parent="kindergarten"/>

<menuitem name="Students" id="menu_kindergarten_student"
parent="menu_configuration_students" action="action_kindergarten_student"/>
```

Aquest fragment de codi correspon als menús de l'aplicació. Amb l'etiqueta *menuitem* especifiquem que és un menú i llavors hem de definir el nom i l'identificador amb els atributs *name* i *id* respectivament. També podem especificar dins de quina carpeta pot estar aquest menú amb l'atribut *parent*, i amb l'atribut *action*, especifiquem quina acció donarà lloc quan es seleccioni aquest menú.

En el cas d'aquest fragment de codi, estem dient que existeix una carpeta anomenada "Escola Bressol", que aquesta carpeta conté dins la carpeta "Gestió d'alumnes", i dins la qual hi ha el menú "Alumnes".

7.5.6 Dreceres

```
<act_window domain=" [('classroom_id','=',active_id)]" id="act_classroom_student"
name="Students assigned" res_model="res.partner.contact"
src_model="kindergarten.classroom"/>
```

Les dreceres ens serveixen per accedir ràpidament a les dades definides en aquesta. S'especifica que és una drecera amb l'etiqueta *act_window*. Com en els casos anteriors, aquestes també tenen un nom i identificador, i en aquest cas, es filtren les dades de la drecera amb l'atribut *domain*. Amb l'atribut *res_model* especifiquem

quin és el model de dades que mostrarà la drecera, i amb l'atribut *src_model* especifiquem el model on volem que aparegui la drecera.

En aquest fragment de codi, estem dient que hi ha una drecera a l'objecte aula que ens servirà per mostrar els alumnes assignats a l'aula actual que estiguem consultant.

7.6 Implementació d'altres recursos

A més del model, vista i controlador implementats pel nostre mòdul, també s'han desenvolupat altres tipus de funcionalitats integrades dins d'aquestes. A continuació hi veurem uns exemples.

7.6.1 Assistents (wizards)

Un assistent o wizard ens permetrà executar una tasca concreta que hem programat sobre un o varis objectes del nostre mòdul. A l'hora de dissenyar un assistent, es procedeix de la mateixa forma que en un model estàndard d'OpenERP però, a diferència d'aquests, OpenERP no crearà taules ni camps a la base de dades.

En el nostre mòdul s'han definit els següents assistents:

- Afegir participacions periòdiques (*create_participations_wizard*). Ens serveix per crear participacions per un o més alumnes dins l'interval de temps introduït.
- Crear Factura (*create_invoice_wizard*). Ens serveix per crear les factures per un o més alumnes de mes introduït.
- Consultar assistència pel dia... (*consult_participation_wizard*). Ens serveix per consultar els alumnes que vindran a l'escola pel dia seleccionat.
- Consultar acollida pel dia... (*consult_dining_wizard*). Ens serveix per consultar els alumnes que faran ús del servei d'acollida pel dia seleccionat.
- Consultar menjador pel dia... (*consult_extra_hour_wizard*). Ens serveix per consultar els alumnes que faran ús del servei de menjador pel dia seleccionat.

7.6.2 Introducció de dades automàticament

Existeixen dos tipus d'introducció de dades automàtiques, encara que els dos funcionen exactament igual. En el primer cas es defineix un arxiu xml amb nomenclatura *nom_del_modul_data.xml*, on dins d'ell es defineixen les dades necessàries per fer funcionar el mòdul. Dins d'aquest arxiu també podem introduir les dades que considerem estàndards, per tal de que qualsevol escola bressol que instal·li aquest mòdul, pugui fer ús d'aquestes dades. En el nostre mòdul, s'ha definit en aquest arxiu les dades referents als productes de tipus quota necessaris per fer la facturació, així com les dades referents als càrrecs en una escola bressol, on creiem que totes funcionen de manera semblant.

En el segon cas, es defineix un arxiu xml amb nomenclatura *nom_del_modul_demo.xml*, i dins d'ell, s'introdueixen dades demostratives que es poden fer servir per veure el funcionament del mòdul. Aquestes dades, a l'hora de la instal·lació del mòdul, es poden escollir si s'instal·laran o no.

```
<data nouupdate="1">
  <record id="kindergarten_classroom_p0" model="kindergarten.classroom">
    <field name="name">P0 - Timbals</field>
    <field eval="8" name="capacity"/>
    <field eval="15" name="square_metres"/>
    <field name="description">Aula per a nens de 0 a 1 any</field>
  </record>
</data>
```

En el nostre mòdul, s'han definit dades de mostra d' alumnes i educadores, així com les dades referents a les aules com es pot veure en el petit retall de codi. El funcionament es exactament igual a l'explicat en anteriors apartats de les vistes, però en aquest cas, amb l'etiqueta *record* indicarem que volem crear un registre, i en ella definirem un identificador i el model al que es refereix. Dins d'aquesta etiqueta, definirem línia a línia tots els camps que volem omplir, cadascun d'ells en la seva etiqueta corresponent.

7.6.3 Seguretat i permisos

Un usuari que no sigui l'administrador d'OpenERP té l'accés restringit als mòduls instal·lats, cal definir doncs, els perfils d'usuaris juntament amb les seves regles d'accés . Primer de tot s'han de crear els perfils. Aquests hauran d'estar definits en un arxiu xml amb nomenclatura *nom_del_modul_security.xml*. Després, s'han de definir les regles d'accés a l'arxiu *ir.model.access.csv*, situat a la carpeta *security* del nostre mòdul. Dins d'aquest arxiu, podem definir per a cadascun dels perfils creats, l'accés a cadascuna de les parts del mòdul, així com del tipus d'accés que tenen (lectura, escriptura, crear i esborrar).

En el nostre mòdul s'han definit els perfils de directora, educadora i personal de cuina, cadascun d'ells amb els accessos pertinents que s'han definit anteriorment al capítol 5.3.1. Aquests accessos també es poden modificar des del mateix programa si s'entra com administrador del sistema; d'aquesta manera no cal que l'usuari final tingui accés al codi. Per veure els permisos de cada perfil en detall, consulteu el capítol 1.5 de l'Annex I: Manual de l'administrador (il·lustració 46).

7.6.4 Traduccions

Els desenvolupadors d'OpenERP sempre utilitzant l'anglès a l'hora de crear un mòdul, i un cop fet, el tradueixen als diferents idiomes amb les eines que ofereix OpenERP. És per això que el nostre mòdul s'ha programat íntegrament amb anglès (a excepció de les dades de mostra), i un cop finalitzat el mòdul, s'ha procedit a la traducció al català.

OpenERP facilita molt la traducció als diferents idiomes, ja que disposa d'una opció per exportar traducció, on només hem d'escollir els mòduls a traduir i escollir el format en el que volem extreure la plantilla. Per traduir el nostre mòdul, es va tenir que exportar la plantilla en format *.po*, i línia a línia traduir el mòdul.

7.7 Test i jocs de proves

Aquesta etapa s'ha anat realitzant durant el transcurs del projecte, a mesura que s'anaven afegint funcionalitats especificades als requeriments del sistema. S'ha fet servir el que en anglès es diu *Unit Testing* (prova unitària), que consisteix en provar el correcte funcionament d'una part del codi, d'aquesta manera ens assegurem de que cada part del codi funcioni per separat. Posteriorment amb les proves d'integració, es pot assegurar el correcte funcionament de tot el sistema. El testeig de cadascuna de les diferents parts del nostre mòdul va consistir en provar tota la casuística possible de les dades, inserint-hi dades extremes si era necessari forçar el sistema. També es va testejar l'aplicació amb diferents perfils d'usuari, per tal de trobar els màxims errors possibles durant aquesta fase.

En el nostre cas, primer es va realitzar pas a pas el model i vista de tot el sistema; es va començar realitzant la gestió d'aules, seguit de la gestió d'educadores i acabant amb els alumnes. D'aquestes tres grans parts, es va realitzar el testeig pertinent de cadascuna d'elles, i fins que cadascuna d'elles no funcionava correctament, no s'avançava a la següent. Un cop es va assegurar el correcte funcionament del model i vista del sistema, es va procedir a programar la lògica del mòdul. I, com abans, quan s'aplicaven noves funcionalitats es procedia

al testeig d'aquestes. Finalment, l'últim pas, va ser la programació dels assistents, possiblement la part més complexa i que va requerir més temps de implementació i de testeig.

Per exemple, posem el cas de l'assistent per crear factures. Per començar es va desenvolupar el model posant-hi les dades que hauria de tenir la factura resultant, i a la vegada, es va treballar amb la vista de l'assistent així com la de la factura resultant. Un cop fet i corregits els errors, es va treballar amb la lògica d'aquest, però primer, només implementant la factura amb la quota fixa. Seguidament, havent-hi corregits els errors, es va procedir a implementar la factura en la seva totalitat, introduint-hi les quotes d'acollida i menjador, segons si aquests serveis havien estat utilitzats o no. Per últim, havent testejat en la seva totalitat l'assistent, es va procedir a introduir les excepcions d'aquest, per exemple, com que no es podia facturar un mes que encara no havia finalitzat o que no es podia facturar un alumne si aquest no tenia dades de facturació.

D'aquesta manera hem anat desenvolupant el mòdul pas a pas, i un cop finalitzat, hem pogut assegurar el correcte funcionament de tot el sistema.

8. Planificació final i anàlisi econòmica

8.1 Planificació final

Per la realització d'aquest projecte final de carrera s'ha dedicat unes 30 hores setmanals, el que aproximadament ens dona una mitja de 4 hores diàries. En total es calcula que s'hauran dedicat unes 600 hores per la realització total del projecte des del seu inici fins al final. El total d'aquestes hores es reparteixen de la següent manera:

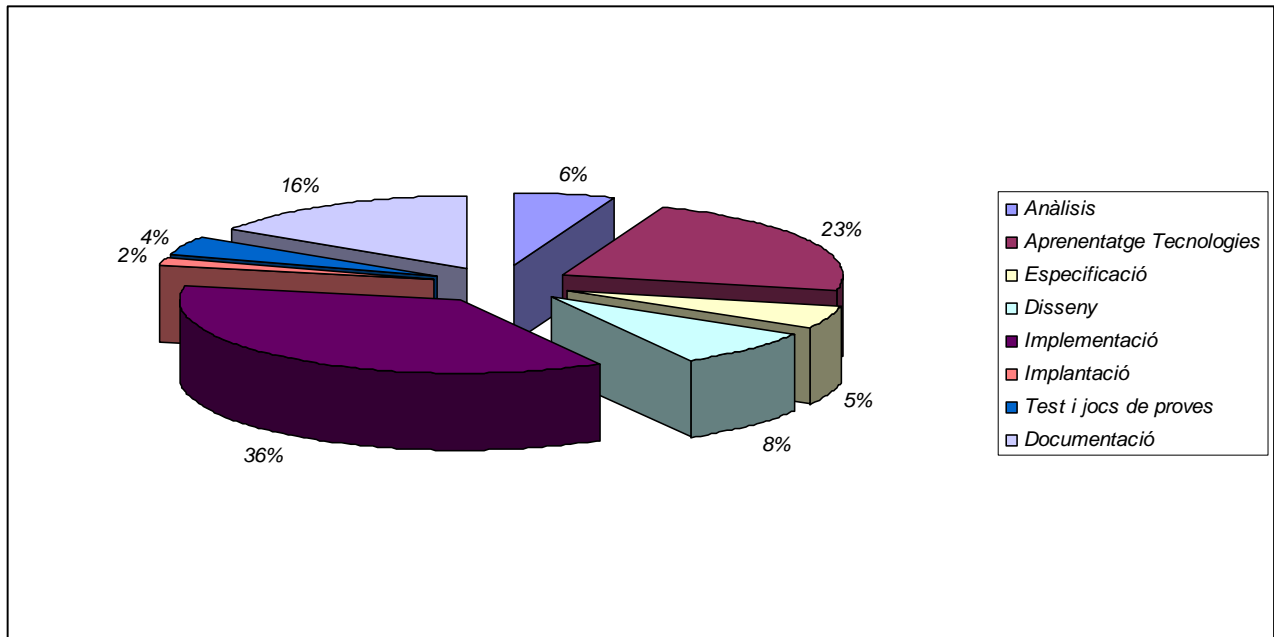
- **Anàlisi.** Durant aquesta etapa es van establir els passos a seguir durant la realització del projecte, així com la descripció d'aquest. Va portar més temps del previst degut a la situació de confusió que comporta un inici de projecte final de carrera.
- **Estudi de les tecnologies i aprenentatge d'OpenERP.** Aquesta va ser la primera etapa un cop establerts els passos a seguir durant el projecte, es va escollir la tecnologia a usar i es va realitzar l'aprenentatge d'aquesta. Aquesta etapa va durar bastant més del previst degut a la complicació que comporta introduir-se per primer cop al món d'OpenERP, i al desconeixement de les tecnologies que això comporta (Python, Xml i Postgresql). A més, se li ha de sumar la escassa informació que es troba més enllà de la web oficial d'OpenERP i els seus fòrums.
- **Especificació.** Aquesta etapa va durar menys del previst degut que des d'un bon principi es van definir uns bons requisits durant la fase d'anàlisi.
- **Disseny.** Un cop fet l'anàlisi i l'especificació, s'estudia quins objectes formaran part de la solució software final, a la qual es defineixen els objectes que podem heretar i els que hauréem de crear de nou.
- **Implementació.** Aquesta etapa encara que ha estat la més llarga, ha portat menys temps del previst. Això és degut principalment a que treballar amb OpenERP comporta una complicada corba d'aprenentatge, però un cop s'ha adquirit l'experiència necessària, es fa molt més fàcil treballar-hi.
- **Implantació.** En aquesta fase no prevista és on s'instal·la i configurar el servidor i els clients d'OpenERP. Degut a que han anat apareixen noves versions d'OpenERP, s'ha anat actualitzant el programa i corregits els errors que han anat sorgint.

- **Test i jocs de proves.** Per comprovar el correcte funcionament i possibles millores es realitzen un conjunt de proves al sistema. D'aquestes proves han sorgit modificacions que després s'han tingut que implementar. Aquesta etapa ha sigut paral·lela a la implementació, d'aquesta manera quan es finalitzava una part del mòdul, es feien les proves pertinents. Un cop s'ha finalitzat el mòdul, s'ha fet un testeig global del sistema amb la interacció de l'usuari final, i la introducció de dades reals al sistema.
- **Documentació.** Aquesta fase esdevé la memòria del projecte així com els pertinents manuals del sistema. És una tasca que ha ocupat un elevat nombre d'hores i més del que s'esperava. S'ha anat realitzant a mesura que s'ha avançat en les diferents parts del projecte, per evitar en gran mesura, tenir que fer-ho tot de cop les últimes setmanes.

A continuació veiem el desglossament de les hores dedicades en cada apartat, així com el percentatge d'hores que ha suposat cadascuna d'elles.

<i>Etapa</i>	<i>Temps (hores)</i>	<i>%</i>
<i>Anàlisi</i>	35	5,8
<i>Aprentatge tecnologies</i>	135	22,5
<i>Especificació</i>	30	5
<i>Disseny</i>	50	8,3
<i>Implementació</i>	220	36,6
<i>Implantació</i>	10	1,6
<i>Test i jocs de proves</i>	25	4,1
<i>Documentació</i>	95	15,8
<i>Total</i>	600	100

Il·lustració 31 - Desglossament d'etapes en hores de treball

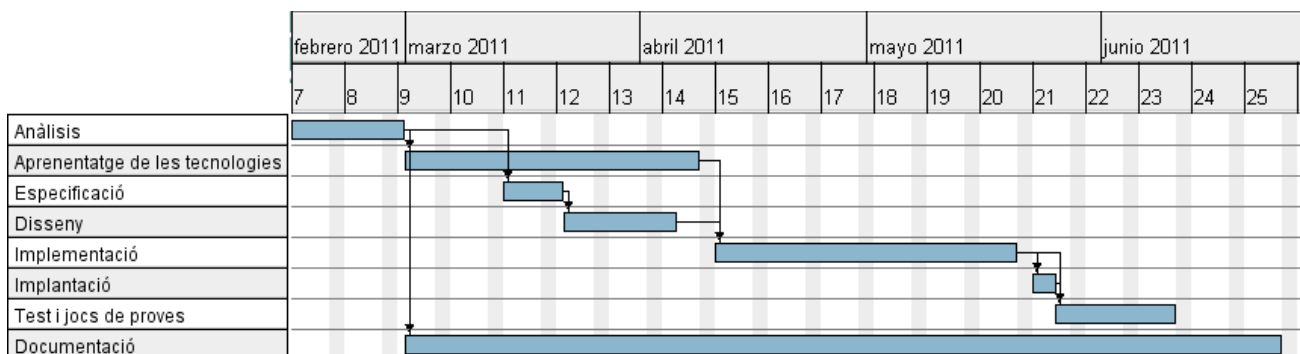


Il·lustració 32 - Desglossament gràfic de les etapes

Com es pot apreciar en les il·lustracions 31 i 32, l'etapa que més temps s'ha invertit és la implementació amb un 36% del temps total, seguit de prop per l'aprenentatge de les tecnologies amb un 23%. Això ha estat per la falta d'experiència amb el llenguatge de programació Python, el qual no coneixia i s'ha invertit força temps a l'aprenentatge d'aquest.

A continuació veiem amb un 16% la documentació, tasca que s'ha anat realitzant per parts durant la realització del projecte, per tal de documentar tot el que es fa. Després trobem les tasques d'anàlisi, especificació i disseny amb un 6%, 5% i 8% respectivament, les quals també se'ls ha dedicat força temps. Per últim, el testeig amb un 4% i la implantació amb un 1,6% són les tasques que menys temps s'hi han dedicat.

La inversió de temps així com l'evolució del projecte vist en un diagrama de Gantt:



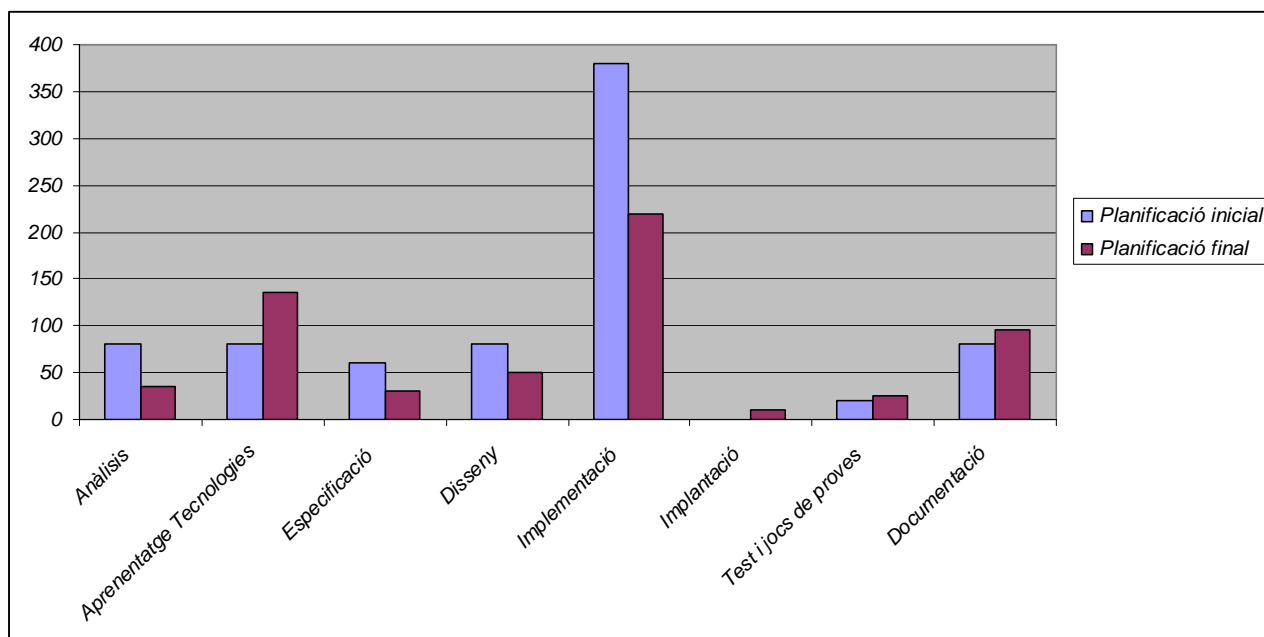
Il·lustració 33 - Diagrama de GANTT amb la planificació final

8.1.1 Comparativa entre planificació inicial i resultat final

Al capítol 4 s'ha mostrat la previsió inicial per fer aquest projecte. A continuació podem veure la taula i el gràfic de temps comparatiu, entre el temps que estava previst dedicar-li i el resultat final.

<i>Etapa</i>	<i>Planificació inicial</i>	<i>Planificació final</i>
<i>Anàlisis</i>	80	35
<i>Aprenentatge Tecnologies</i>	80	135
<i>Especificació</i>	60	30
<i>Disseny</i>	80	50
<i>Implementació</i>	380	220
<i>Implantació</i>	0	10
<i>Test i jocs de proves</i>	20	25
<i>Documentació</i>	80	95
<i>Total</i>	780	600

Il·lustració 34 - Desglossament d'etapes en hores entre la previsió i el resultat final



Il·lustració 35 - Gràfic comparatiu entre les hores planificades inicialment i el resultat final

Amb aquestes dades es poden extreure les conclusions següents:

- Es tenia previst dedicar-li al projecte aproximadament unes 780 hores, i finalment se li han dedicat 600 hores, un 23,07% menys del planificat inicialment.
- Degut al desconeixement de la tecnologia i la poca informació que es troba a la xarxa, se li han dedicat més hores del previst a l'aprenentatge.

- En la implementació hi ha hagut una diferència molt gran respecte al temps previst, ja que aquest ha disminuït considerablement. Això és degut a la gran quantitat d'hores dedicades de l'aprenentatge, així com l'existència de mòduls que eliminen temps de programació, i la facilitat que atorga OpenERP per fer la interfície.
- Pel que fa l'anàlisi, especificació i disseny, s'ha disminuït bastant el temps segurament perquè a l'inici del projecte no ho tenia molt clar com arribaria a la solució software final. Un cop fet l'aprenentatge de la tecnologia, he començat a tenir les idees més clares de com hi podia arribar-hi.
- Ha aparegut una nova tasca que no estava prevista inicialment, degut a que OpenERP està contínuament en actualització i s'ha tingut que anar actualitzant el programa així com la correcció dels errors que han sorgit per aquest tema.
- El testeig ha augmentat lleugerament, però la unió de les tasques de correcció d'errors amb la implementació, ha estat clau per no augmentar molt més les hores d'aquesta etapa.
- Respecte la documentació, s'ha elevat lleugerament el nombre d'hores però dins del que estava previst.

8.2 Anàlisi econòmica

En aquest apartat, destaquem el cost econòmic que podria tenir aquest projecte tenint en compte les hores que s'han invertit per la seva realització. Es considera que l'equip físic per la realització d'aquest projecte (PCs i altre material d'oficina) ja es té, i a més, el software utilitzat és lliure i no s'ha de pagar llicències (OpenERP, PostgreSQL, gedit, OpenOffice,...). Per tant, l'únic camp que esdevé despesa econòmica és en el de recursos humans.

Per calcular el cost total del projecte, s'ha assignat cada tasca a un rol professional, on cadascun d'aquests té un sou i un preu hora. Veiem a continuació quins rols hem hagut d'assumir:

- **Analista.** Rol que s'encarrega de dur a terme les etapes d'anàlisi del problema, així com del disseny de l'aplicació. El sou s'estima que és de 30.000 euros bruts l'any. Preu hora: 22,5€.
- **Programador.** Rol encarregat de la implementació de l'aplicació així com de la part del

testeig. El seu sou s'estima a 20.000 euros bruts l'any. Preu hora: 15€.

- **Tècnic Informàtic.** S'encarrega de la migració de dades així com de la implantació del programa. Sou estimat en 15.000 euros bruts l'any. Preu hora: 11,25€.
- **Redactor.** Rol encarregat de la redacció de la memòria i els manuals d'usuari. Sou estimat en 15.000 euros l'any. Preu hora: 11,25€.

Per calcular el preu hora s'ha fet servir la fórmula següent:

Preu hora = (sou brut anual * increment seguretat social) / (setmanes treballades * hores treballades setmana)

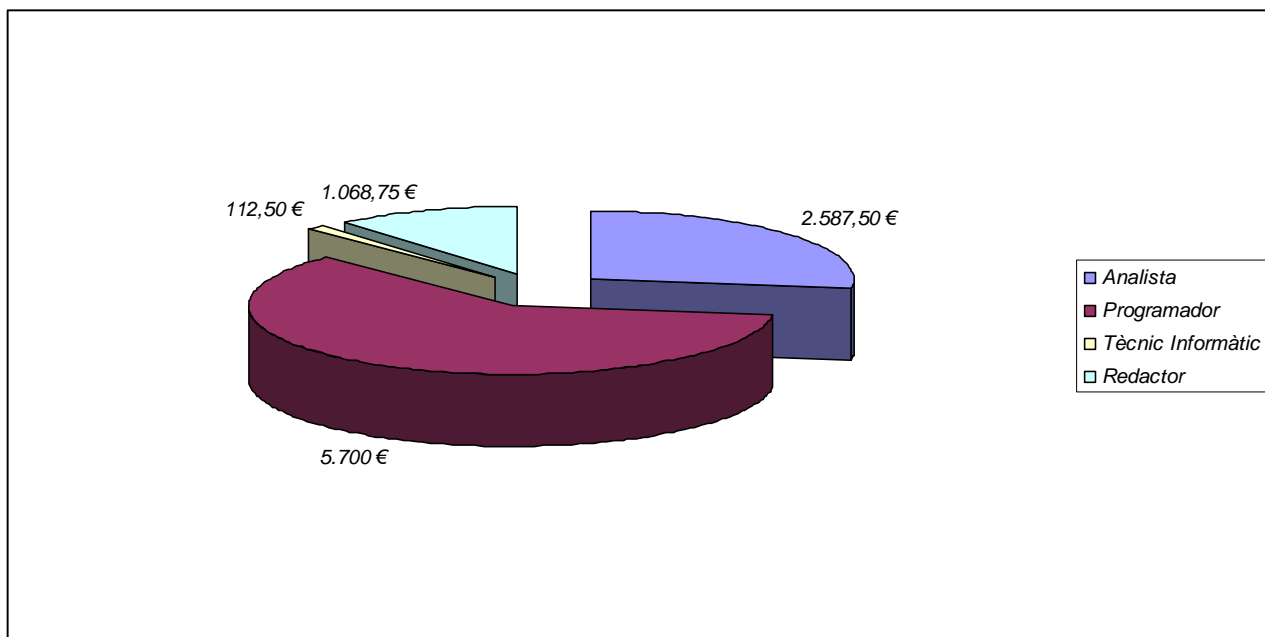
On l'increment de la seguretat social té un valor de 1,35, es calcula que hi ha 45 setmanes treballades a l'any i 40 són el nombre d'hores treballades per setmana.

Amb aquestes dades, obtenim el preu total del projecte desglossat en les diferents tasques.

<i>Etapa</i>	<i>Temps (hores)</i>	<i>Rol professional</i>	<i>Cost</i>
<i>Anàlisi</i>	35	<i>Analista</i>	787,50 €
<i>Aprenentatge Tecnologies</i>	135	<i>Programador</i>	2.025 €
<i>Especificació</i>	30	<i>Analista</i>	675 €
<i>Disseny</i>	50	<i>Analista</i>	1.125 €
<i>Implementació</i>	220	<i>Programador</i>	3.300 €
<i>Implantació</i>	10	<i>Tècnic informàtic</i>	112,50 €
<i>Test i jocs de proves</i>	25	<i>Programador</i>	375 €
<i>Documentació</i>	95	<i>Redactor</i>	1.068,75 €
<i>Total</i>	600		9.468,75 €

Il·lustració 36 - Desglossament del preu de les etapes segons hores imputades

A continuació es mostra un gràfic amb el cost total per cadascun dels rols dels treballadors, on podem veure com el major cost és el del programador, degut a que les tasques de aprenentatge i implementació són les que han requerit més temps.



Il·lustració 37 - Desglossament del preu de les etapes segons hores imputades

Finalment, el cost total del projecte hagués estat de 9.468 €. S'ha de tenir en compte que aquest projecte s'ha desenvolupat des del total desconeixement de les tecnologies emprades, per tant, aquest cost total és un valor fictici. El projecte tindria un cost al mercat bastant inferior, degut principalment a l'experiència prèvia dels treballadors, per tant, es suprimiria l'etapa d'aprenentatge de les tecnologies, i es reduiria lleugerament el temps de les etapes d'especificació, disseny i implementació. Fruit d'aquesta reducció de temps, el cost total del projecte en una empresa real hagués baixat considerablement.

Cal afegir, que encara que en una empresa hagués baixat el cost total, en aquesta s'hi podria afegir l'etapa de formació del personal que ha d'usar l'aplicació, en cas que l'empresa considerés que els manuals del sistema no fossin suficients.

9. Conclusions

9.1 Conclusions personals

- Ha estat el primer cop com a informàtic on m'he trobat en la situació de trobar solució a un problema real, assumint el paper de diferents rols durant tot el desenvolupament del projecte.
- La utilitat de treballar directament amb el client i les reunions amb aquest. Encara que aquest projecte no és aplicable, si que ha estat basat en una situació real on hi hagut una persona que ha adquirit el rol de client, i s'ha pogut comprovar de primera mà la utilitat d'aquestes reunions de cara a obtenir la informació necessària per desenvolupar el projecte.
- He comprés la necessitat d'entendre perfectament el problema que s'ha de resoldre i saber quina és la solució a la que es vol arribar; d'aquesta manera es redueix el temps de cara a les etapes de l'anàlisi i el disseny.
- Entendre molt més àmpliament la importància dels ERP en el món dels negocis, ja que aplicant un ERP correctament a un negoci , es facilita i incrementa la productivitat d'aquest.

9.2 Conclusions tècniques

- Els objectius principals d'aquest projecte per gestionar una escola bressol comportant tota la gestió d'alumnes i educadores, així com la de les aules i el material escolar, han estat completament assolits. Ja que s'ha dissenyat i implementat un sistema complet per gestionar totes aquestes relacions.
- El gran potencial que té desenvolupar amb OpenERP, i la senzillesa que aquest ens atorga gràcies al patró Model-Vista-Controlador i la clara divisió d'aquestes capes, així com les relacions entre elles.
- La inclinada corba d'aprenentatge que suposa treballar per primer cop amb aquesta tecnologia. S'ha de dedicar molt de temps per aquesta fase ja que la formació pot semblar complicada, però un cop ja has assolit aquesta fase, la implementació és més senzilla del que en un principi ens pot semblar. Gràcies sobretot a la gran quantitat de mòduls disponibles per OpenERP, dels quals podem heretar les funcionalitats que nosaltres creiem necessàries.
- La participació de la comunitat afegint noves funcions en forma de mòduls a la plataforma Launchpad. La comunitat d'OpenERP actualitza diàriament i aporta mòduls i traduccions, a més de l'ajuda que et poden donar en cas que facis preguntes tècniques dins d'aquesta plataforma.

- La separació del client-servidor que fa OpenERP no suposa cap complicació afegida, sinó que ens facilita les coses permetent-nos utilitzar el client en sistemes operatius diferents.
- La interfície gràfica d'Openerp és limitada, ja que només ens permet ajustar la col·locació dels camps. Encara que això pugui semblar un aspecte negatiu, és molt positiu respecte al temps que ens estalvia per fer el disseny gràfic.
- Com s'ha utilitzat la nova versió 6.0 d'OpenERP, aquesta encara no disposa de la totalitat traducció dels mòduls al nostre idioma. És per això que en el nostre projecte, es troben vistes heretades encara no traduïdes.
- Existeix molt poca documentació d'OpenERP a la xarxa, i casi tota aquesta informació es troba en anglès. Pràcticament tota la informació es troba concentrada als fòrums de la web oficial d'OpenERP i a la plataforma Launchpad, però s'ha de disposar d'un bon nivell d'anglès.
- La realització del projecte s'ha pogut realitzar en la seva totalitat amb eines lliures, exceptuant alguns programes de pagament els quals ja es disposava de llicència abans de començar el projecte. A l'annex III, es poden consultar quines han sigut les eines utilitzades.

9.3 Possibles ampliacions

Un cop finalitzat el projecte i assolits els objectius que s'havien proposat, s'han pensat una sèrie de millores i ampliacions. Aquestes són:

- Per la gestió de les educadores, es podria aprofitar el mòdul *hr_attendance* per tal de controlar les baixes temporals així com els retards de les educadores. Aquesta opció no ha estat contemplada en el nostre projecte i es podria fer, aprofitant aquest mòdul el qual ja s'instal·la per defecte.
- Pel que fa les educadores també es podria afegir el pagament de les nòmines. Es tindria que estudiar si es poden utilitzar els mateixos mòduls que es fan servir per la facturació dels alumnes, o utilitzar els mòduls de recursos humans destinats al pagament de les nòmines.
- Millorar la gestió dels menús amb la creació automàtica d'aquests i possibilitat de creació d'informes.

9.4 Aspectes mediambientals

En aquest tipus de projectes sembla difícil trobar aspectes que suposin una millora pel medi ambient. Tot i

així, podem comentar el següent:

- La incorporació d'aquesta eina en una escola bressol pot suposar una disminució en la despesa de paper. Al tenir tota la informació disponible a l'ordinador i poder consultar la informació que es vulgui, s'estalvia paper al fer-ho tot electrònicament.
- Tota la informació pot estar centralitzada en un mateix servidor, per tant és possible una reducció de la despesa elèctrica ja que no es necessiten diversos ordinadors.

9.5 Anàlisi d'accessibilitat

OpenERP no ens permet la modificació de la interfície gràfica o la forma de comunicació que té envers els seus usuaris, tot i així es pot recomanar:

- Usar un contrast de colors adient per l'ull humà modificant-ho a través de la pantalla de l'ordinador.
- Utilitzar les eines que ens proporcionen els sistemes operatius com l'amplificador de text o el disposar d'un teclat en pantalla.

Annex I: Manual d'administrador

1.1 Instal·lació del servidor OpenERP

Per instal·lar el servidor OpenERP haurem d'accedir a la web oficial de descàrregues d'OpenERP des del següent link:

<http://www.openerp.com/downloads>

Un cop accedit al link, hem de descarregar la versió OpenERP 6.0 (en el moment d'escriure aquesta memòria, l'última versió descarregada i comprovat el seu correcte funcionament ha estat la v6.0.1). Ens donaran l'opció de descarregar el servidor (server) o el conjunt de servidor i client (all-in-one). En cas que vulguem accedir al sistema com usuari en l'ordinador que estem instal·lant el servidor, ens baixarem el All-in-one. Cal dir, que el servidor haurà d'estar instal·lat en Windows o GNU/Linux.

1.1.1 Instal·lació del servidor en Windows

Un cop baixat l'arxiu en versió Windows, executarem aquest arxiu i seguirem les instruccions de l'assistent:

1. Quan ens preguntin per l'idioma de la instal·lació, escollirem l'opció que vulguem i polsarem el botó "Next".
2. Ens donarà a escollir la ruta de la instal·lació, l'escollim o deixem la que està per defecte, i polsem el botó "Next".
3. A partir d'aquí comença la instal·lació i la resta serà tot automàtic.

1.1.2 Instal·lació del servidor en Linux

Pel correcte funcionament del servidor OpenERP, abans necessitarem instal·lar una sèrie de paquets addicionals, així com realitzar algunes configuracions bàsiques. Els passos a fer a través del terminal seran els següents:

1. Primer cal instal·lar el servidor i les eines de PostgreSQL, la base de dades que utilitza OpenERP. Per fer això, introduïrem la comanda:

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

2. Fet això, caldrà editar el fitxer de configuració d'accés a PostgreSQL. Accedim al fitxer:

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

3. Busquem la línia "***local all all ident sameuser***" quasi al final del fitxer i la modifiquem per:

“local all all md5”

4. Reiniciem el servidor PostgreSQL:

sudo /etc/init.d/postgresql-8.3 restart

5. Ara hem de crear un usuari per a PostgreSQL per tal de poder crear la base de dades, introduint les següents comandes:

- *sudo su postgres*
- *createuser --pwprompt usuari* (on usuari és el nom que desitgem)
- Introduïrem la contrasenya dos cops.
- Retornarem a la línia de comandes polsant [Ctrl] + D

6. Instal·lem la resta de paquets restants necessaris per la instal·lació:

sudo apt-get install python python-xml python-libxml2 python-libxslt1 python-lxml python-psycopg python-psycopg2 python-imaging python-reportlab

En cas de trobar-se algun problema addicional alhora de instal·lar el servidor, consulteu el següent link:

<http://www.aulaerp.com/aula/course/view.php?id=7>

1.2 Execució de servidor OpenERP

1.2.1 Windows

Per executar el servidor, només cal executar l'accés directe que s'ha creat a l'escriptori. En cas que no aparegui l'accés directe, també es pot executar el servidor anant a Inicío -> Programes -> OpenERP Server.

1.2.2 Linux

Per executar el servidor, ens dirigim al directori on tinguem instal·lat el servidor, entrem a la carpeta *bin* i introduïm la següent comanda:

/openerp-server.py

Per poder crear la base de dades, caldrà abans iniciar el servidor amb l'usuari i contrasenya que hem introduït abans:

./openerp-server.py -r nom_usuari -w contrasenya

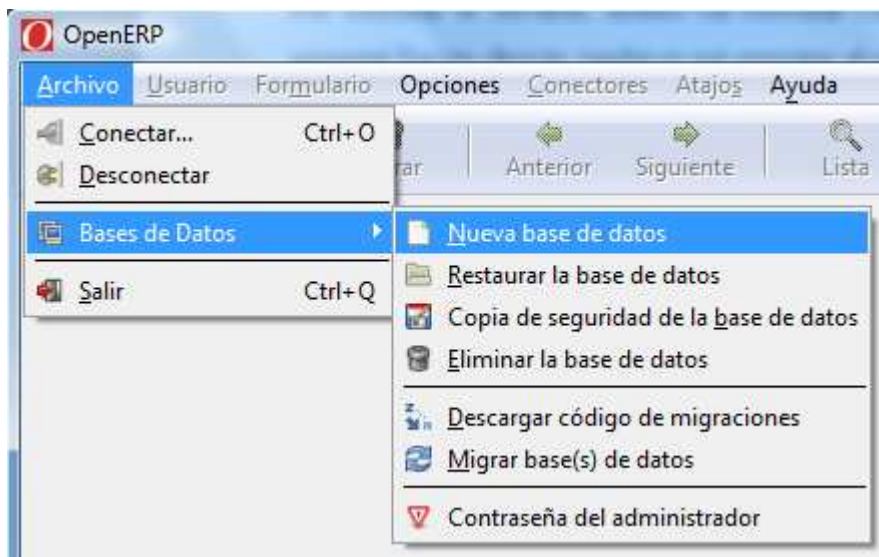
Aquesta acció la podem simplificar per properes vegades, afegint-hi "-s" al final, per tal de que es guardi la configuració d'aquesta instrucció i properes vegades només haurem d'executar el servidor.

Si els mòduls es modifiquen i volem que s'apliquin als clients, escriurem la següent comanda:

```
./openerp-server.py -update all
```

1.3 Creació d'una base de dades

Un cop executat el servidor, ja podem iniciar el client. Per crear la base de dades ens dirigim a Archivo -> Base de Datos -> Nueva base de datos.



Il·lustració 38 - Menú creació d'una base de dades

Quan pensem sobre "Nueva base de datos" ens apareixerà el següent formulari:

The image shows a screenshot of a dialog box titled 'Crear una nueva base de datos'. The form contains several fields and controls: 'OpenERP Server' with the value 'socket://localhost:8070' and a 'Cambiar' button; 'Super Administrator Password' with masked characters and a note '(admin, por defecto)'; 'New Database Name' with an empty text box; 'Load Demonstration Data' with a checked checkbox; 'Default Language' with a dropdown menu set to 'Catalan / Català'; 'Administrator Password' and 'Confirm Password' with empty text boxes. At the bottom, there are 'Cancelar' and 'Aceptar' buttons.

Il·lustració 39 - Formulari creació d'una base de dades

On:

1. És la direcció del servidor d'OpenERP que volem crear, per defecte ja ens ve donada, però si la volem modificar hem de pulsar a "Cambiar".
2. La contrasenya del super-administrador, per defecte és admin.
3. Nom de la base de dades (no pot portar espais).
4. Opció per instal·lar dades de mostra per tal de poder fer proves, etc.
5. Idioma.
6. Contrasenya de l'administrador d'aquesta base de dades.
7. Confirmació de la contrasenya de l'administrador.

Un cop definit tots els camps, pulsarem el botó "Acceptar" i es començarà a crear la base de dades. Aquest procés pot trigar una mica.



Il·lustració 40 - Assistent configuració de base de dades

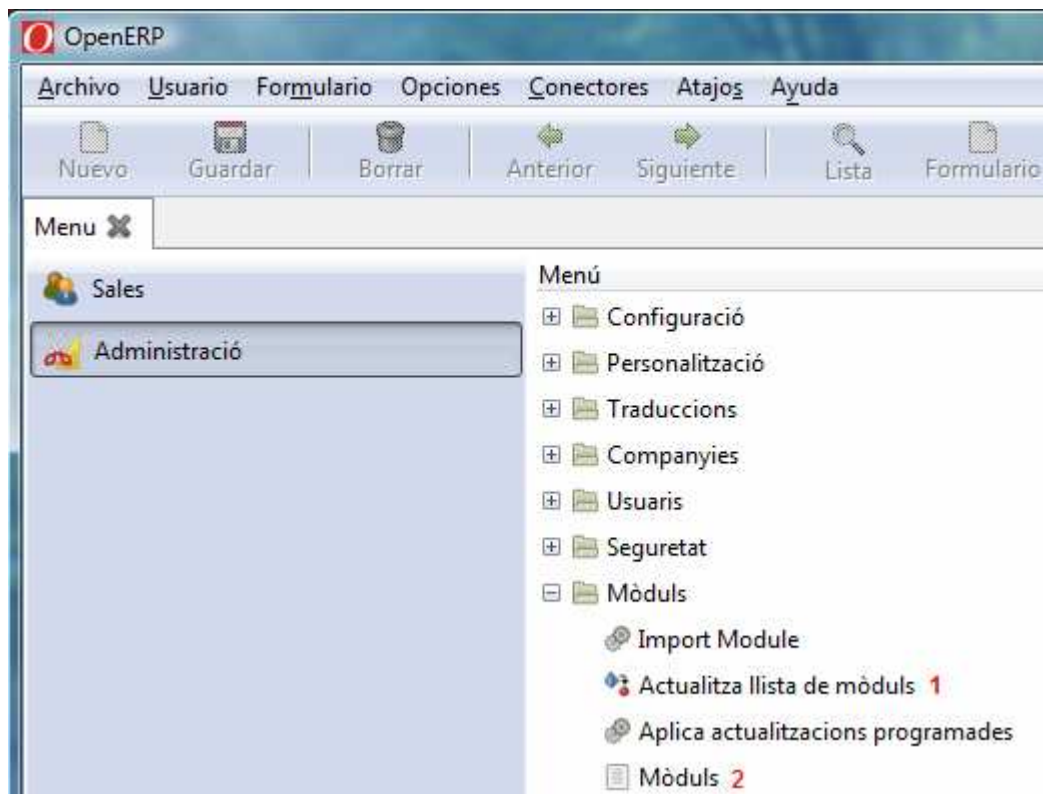
Un cop creada, posem "Inicia Configuració" per tal d'iniciar un nou assistent per definir la base de dades creada:

- El primer que ens demana serà quin tipus de perfil volem; haurem de seleccionar "Estesa" ja que sinó no es permetrà instal·lar els mòduls que nosaltres vulguem.
- A continuació ens demanarà la informació de la nostre empresa, en aquest cas, de l'escola bressol. Un cop omplert tots els camps posem "Següent".
- Per últim, ens preguntarà si volem instal·lar alguns mòduls per l'aplicació, pulsarem "Omet" ja que en el nostre cas a l'hora d'instal·lar el mòdul de l'escola bressol, es farà tot automàticament.

1.4 Instal·lació dels mòduls

Un cop creada la base de dades, ja podem començar a instal·lar els mòduls. Les funcionalitats d'OpenERP s'afegeixen mitjançant mòduls, aquests es troben al directori *OpenERP/server/addon*. És en aquest directori on haurem d'afegir els mòduls que desitgem, en el nostre cas, afegirem el mòdul creat *kindergarten* i el mòdul extra *holiday*.

Fet això, haurem d'actualitzar la llista de mòduls, anant a Administració -> Mòduls -> Actualitza llista de mòduls (número 1).



Il·lustració 41 - Menú mòduls

Ens apareixerà un assistent per actualitzar la llista de mòduls, polsem a "Actualitzar" i un cop actualitzat polsem "Acceptar". Aquest assistent el que fa és buscar els possibles nous mòduls que s'hagin afegit a la carpeta "addons".

Un cop actualitzat, ja podem instal·lar el nostre mòdul, per això haurem d'anar a Administració -> Mòduls -> Mòduls (Il·lustració 41 - número 2). Ens apareixerà una llista amb tots els mòduls disponibles, buscarem el nostre mòdul *kindergarten* i farem doble click en ell. El sistema ens mostrarà una nova finestra:

The screenshot shows the 'kindergarten' module configuration form in OpenERP. The form includes the following fields and sections:

- Form Fields:**
 - Nom: kindergarten
 - Certificat de qualitat: (empty)
 - Descripció breu: kindergarten
 - Categoria: Escola Bressol
 - Dades d'exemple:
- Dependències Tab:**
 - Autor:** Victor Lossius Millet
 - Llicència:** Affero GPL-3
 - Web de l'autor:** http://openerp.com
 - Versió:**
 - Última versió: 6.0.1.0
 - Versió instal·lada: (empty)
 - Versió publicada: (empty)
- Descripció:** Gestió d'una escola bressol
- Stat:** No instal·lat
- Buttons:** Programa per instal·lació

On the right side of the interface, there are two buttons: 'Guia tècnica' and 'Aplica actualitzacions programades'.

Il·lustració 42 - Vista formulari kindergarten

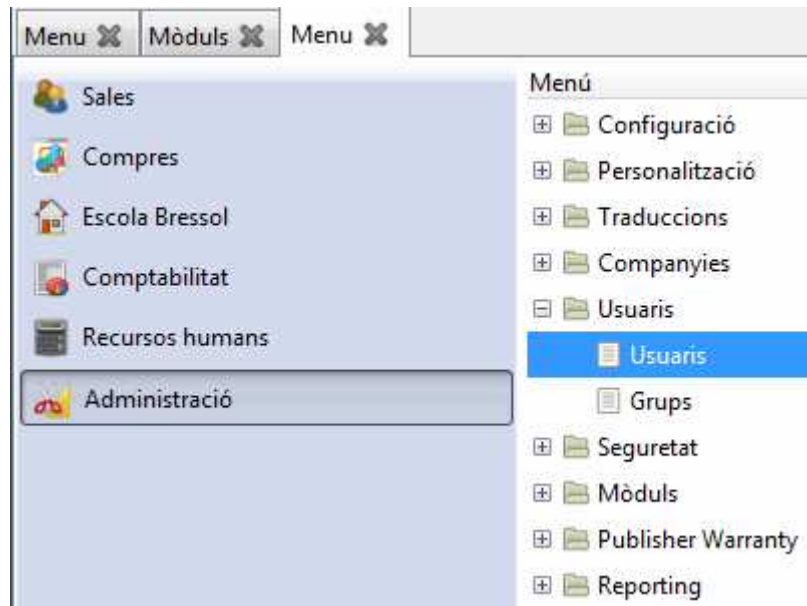
En aquesta vista podrem veure el nom del mòdul, l'autor, descripció... així com altres camps informatius. A la pestanya "Dependències" podrem observar els mòduls necessaris per fer funcionar aquest mòdul i que s'instal·laran automàticament.

Per tant, només farà falta que polsem el botó "Programa per instal·lació" i seleccionem després l'assistent "Aplica actualitzacions programades" situat a la dreta.

Aquest procés trigarà una estona, i una vegada complert, s'iniciarà un assistent per configurar el mòdul *hr_timesheet_invoice* del qual fem herència; en el nostre cas els ometrem ja que el mòdul ja bé configurat per defecte.

1.5 Creació d'usuaris

Per crear usuaris ens dirigirem al menú "Administració", seleccionarem la carpeta "Usuaris" i polsarem sobre l'opció "Usuaris".



II·lustració 43 - Menú creació usuaris

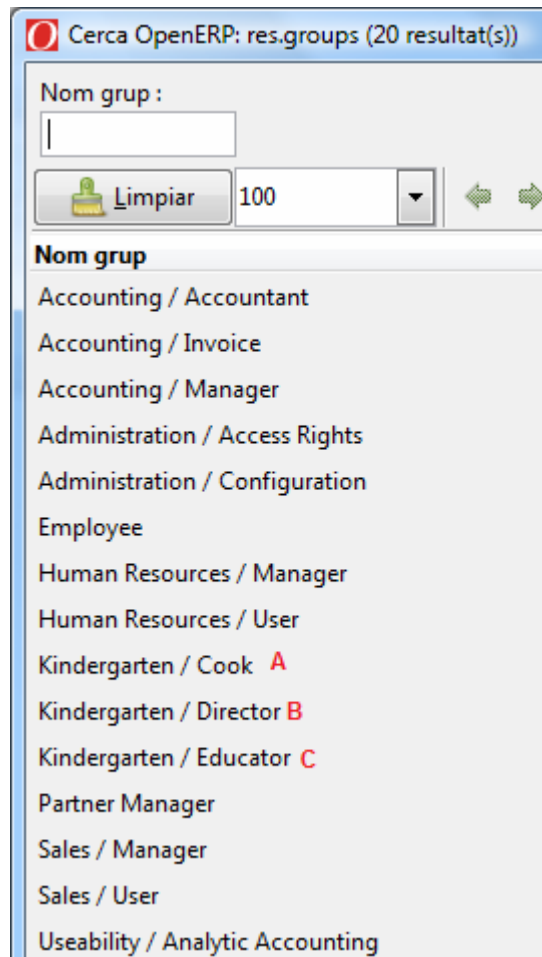
Ens apareixerà la vista arbre de tots els usuaris creats. Com és la primera vegada que accedim només estarà creat l'usuari administrador i l'usuari demo. Per crear un de nou, polsarem en "Nuevo" i ens apareixerà la vista formulari d'usuari:

II·lustració 44 - Vista formulari creació usuaris

On:

1. Nom de la persona. (obligatori)
2. Nom de l'usuari amb el qual accedirà al sistema. (obligatori)

3. Estat de l'usuari, si està actiu o no.
4. Contrasenya de l'usuari.
5. Pestanya Usuari, on podrem afegir dades per aquest contacte com ara l'adreça, mail, idioma, etc.
6. A l'apartat Grups és on seleccionarem les autoritzacions, haurem de polsar "Añadir" i ens apareixerà una llista amb totes les autoritzacions disponibles.



Il·lustració 45 - Vista arbre perfils d'autorització

On:

- A. És el perfil d'usuari pel personal de cuina.
- B. És el perfil d'usuari per la directora.
- C. És el perfil d'usuari per les educadores.

Aquests perfils ja venen creats per defecte al instal·lar el mòdul, en cas que nosaltres vulguem modificar les autoritzacions d'aquests perfils, haurem d'anar a Administració -> Usuaris -> Grups (Il·lustració 45). Des d'aquesta opció podrem seleccionar el perfil que vulguem modificar, i afegir o treure les autoritzacions pertinents.

A continuació, mostrem una taula amb les autoritzacions per defecte de cadascun dels perfil d'usuaris. En ella es pot veure totes les vistes disponibles, i els tipus de permisos que hi ha: lectura, escriptura, crear i esborrar.

Vista	Perfils											
	Directora				Educadora				Personal cuina			
	Lectura	Esctiptura	Crear	Esborrar	Lectura	Esctiptura	Crear	Esborrar	Lectura	Esctiptura	Crear	Esborrar
Alumne	x	x	x	x	x	x			x			
Responsable	x	x	x	x	x	x	x					
Participació	x	x	x	x	x	x	x	x	x			
Facturació	x	x	x	x	x							
Educadora	x	x	x	x	x							
Càrrec	x	x	x	x								
Aula	x	x	x	x	x							
Material Escolar	x	x	x	x	x	x	x	x				
Vacances	x	x	x	x	x				x			
Menús	x	x	x	x	x				x			

Il·lustració 46 - Autoritzacions per cada perfil

Annex II: Manual d'usuari

1.1 Instal·lació del client OpenERP

Per instal·lar el client OpenERP haurem d'accedir a la web oficial de descàrregues d'OpenERP des del següent link:

<http://www.openerp.com/downloads>

Un cop accedit al link, hem de descarregar el client de la versió OpenERP 6.0.

1.1.1 Instal·lació del client en Windows

Un cop baixat el client en versió Windows, executarem aquest arxiu i seguirem les instruccions de l'assistent:

1. Quan ens preguntin per l'idioma de la instal·lació, escollirem l'opció que vulguem i polsarem el botó "Next".
2. Ens donarà a escollir la ruta de la instal·lació, l'escollim o deixem la que està per defecte, i polsem el botó "Next".
3. A partir d'aquí comença la instal·lació i la resta serà tot automàtic.

1.1.2 Instal·lació del client en Linux

Un cop baixat el client en versió Linux, s'hauran d'instal·lar unes llibreries necessàries amb la següent comanda:

```
sudo apt-get install python python-xml python-libxml2 python-libxslt1 python-lxml python-psycopg  
python-psycopg2 python-imaging python-reportlab python-pyparsing python-pydot  
python-matplotlib python-numpy python-pychart python-egenix-mxdatetime python-vobject
```

Fet això, es pot procedir a instal·lar el client amb la comanda:

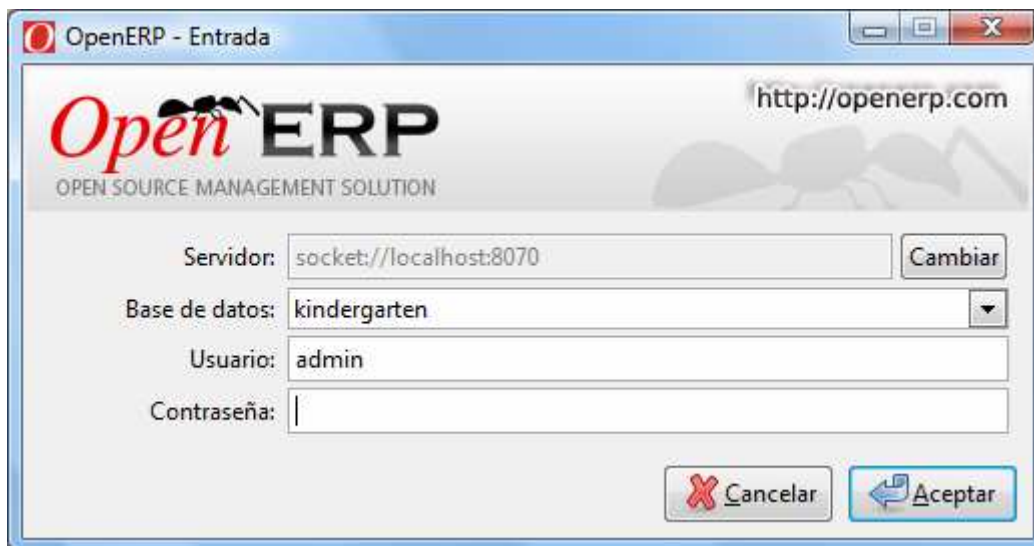
```
sudo apt-get install openerp-client
```

1.2 Execució del client

Per iniciar el programa, executem l'aplicació:

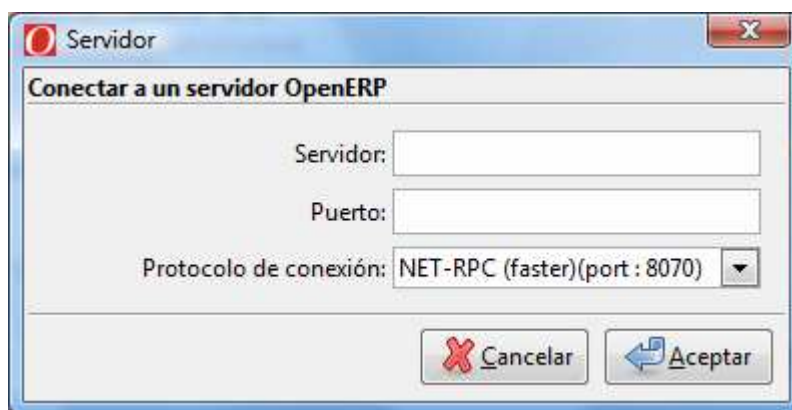
- Windows: Executem l'accés directe creat a l'escriptori, o al menú inici, programes, OpenERP Client.
- Linux: L'executem anant al menú Aplicacions, Internet, OpenERP Client.

Un cop executat, ens apareix una finestra on hem d'introduir el servidor, la base de dades, usuari i contrasenya (aquestes dades han de ser proporcionades per l'administrador del sistema).



Il·lustració 47 - Finestra client per accedir al sistema

Ens apareixerà la casella "Servidor" omplerta per defecte. Si les dades no són correctes polsem el botó "Cambiar" per modificar-les. D'aquesta manera ens apareixerà una altra finestra on podem indicar la direcció del servidor, el port, i el protocol de connexió.



Il·lustració 48 - Configuració connexió servidor

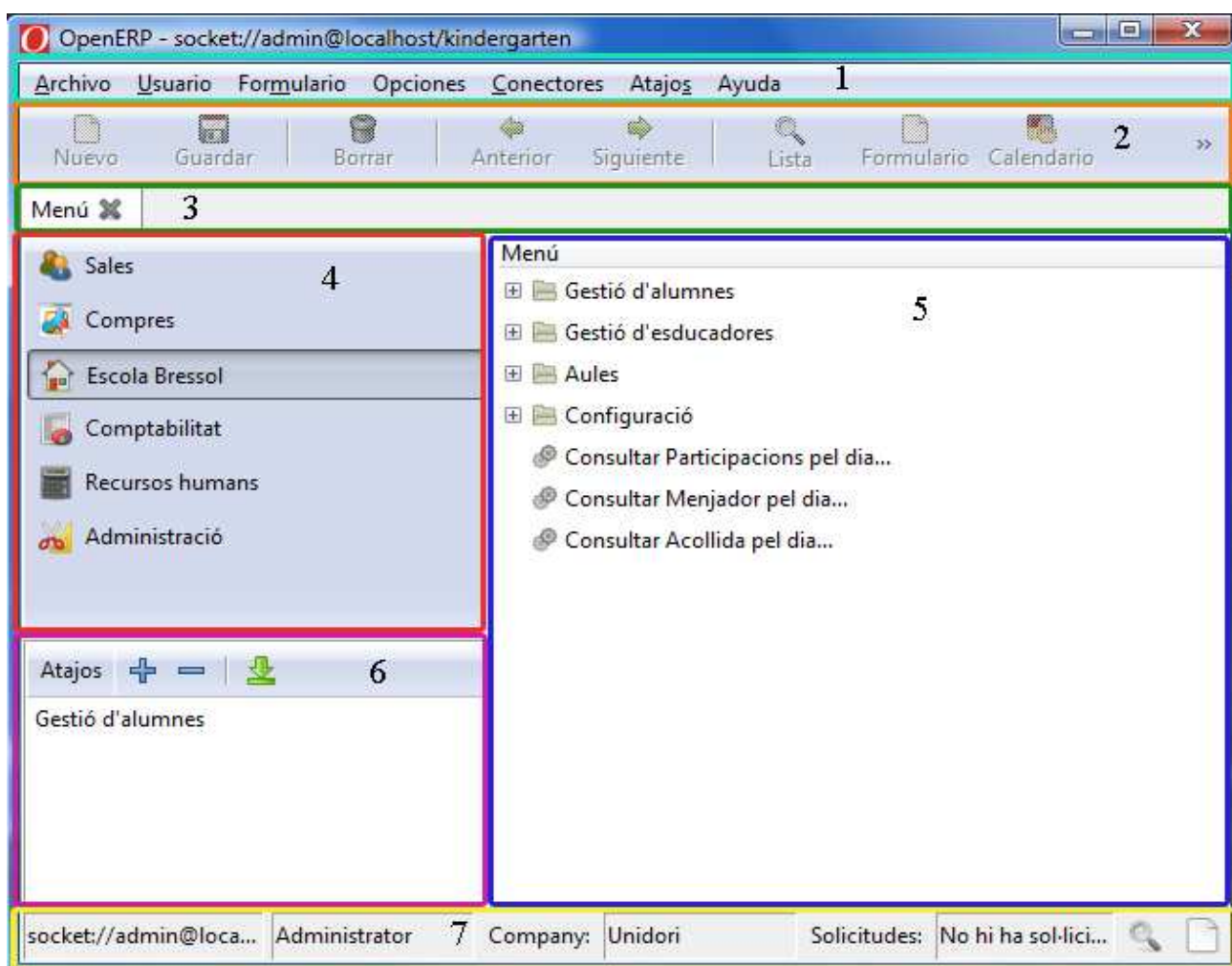
Introduïdes les dades tornarem a la finestra anterior (Il·lustració 47), i si la direcció del servidor és correcta podrem escollir la base de dades associada a aquest servidor. Si el missatge que ens surt és "Could not

connect to Server", significa que no ha trobat el servidor OpenERP. Això pot ser degut a que les dades introduïdes són incorrectes, el servidor ha caigut o que existeix algun tallafoc que impedeixi la connexió. Si apareix un missatge que indica que no hi ha una base de dades creada, caldrà comentar-ho amb l'administrador del sistema.

Un cop escollit el servidor i la base de dades, només caldrà accedir amb el nostre nom d'usuari i contrasenya, i pulsar el botó "Acceptar".

1.3 Interfície d'usuari

Quan accedim a l'aplicació ens apareixerà la següent finestra:



Il·lustració 49 - Menú principal client OpenERP

Aquest és el menú principal del client d'OpenERP, en ell podem distingir:

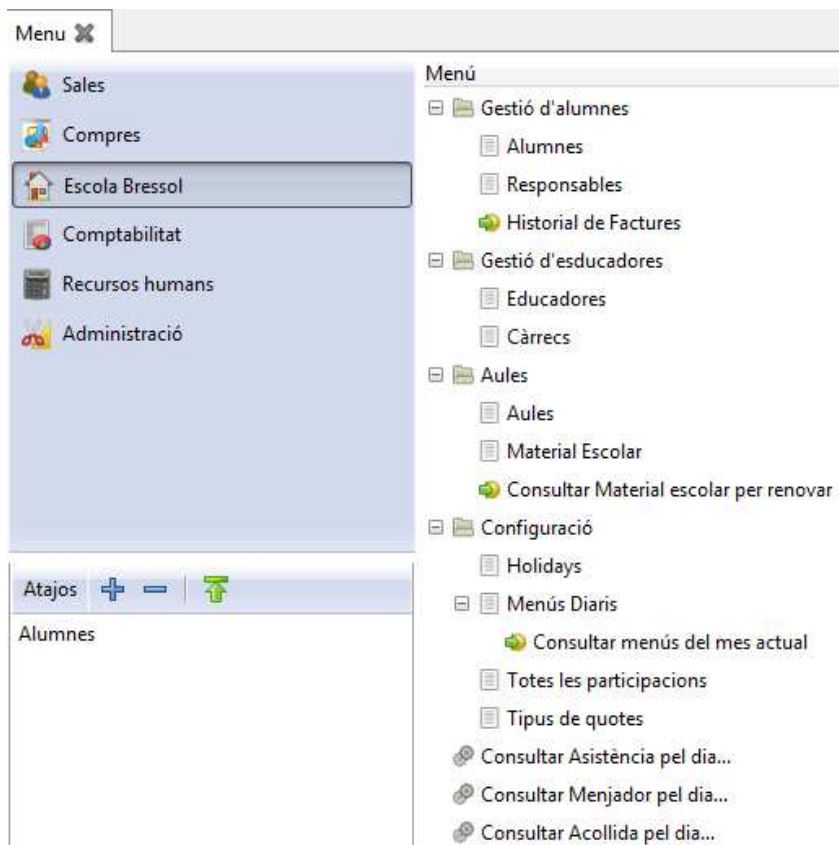
1. És el menú de l'aplicació. Aquest ens ofereix diferents opcions del programa com ara connectar/desconnectar, canviar contrasenya i altres opcions de l'aplicació.
2. Aquí trobem les accions més habituals d'OpenERP: Nou registre, desar, esborrar, etc.
3. Pestanyes. Cadascuna d'elles ens mostra una finestra diferent i ens serveix per moure'ns fàcilment per les diferents vistes.
4. És el menú dels mòduls, on es mostren els mòduls instal·lats per poder accedir a ells. Quan seleccionem un d'ells, la finestra de la dreta canviarà per mostrar-nos el menú del mòdul escollit.
5. Secció principal. Es mostren els menús del mòdul escollit, i és on es carreguen les vistes formulari, arbre, etc.
6. Accessos directes. Ens permet afegir accessos directes a les opcions que més utilitzem.
7. Informació del servidor on estem connectats i amb quin usuari hem accedit, així com poder veure les sol·licituds d'usuari (missatges interns entre usuaris).

1.4 Ús general

En aquest capítol s'explica com utilitzar les diferents parts del mòdul creat. Primer seleccionarem el mòdul *kindergarten* i a la secció principal ens apareixerà els menús del mòdul (il·lustració 50).

Les captures de pantalla que es mostren a continuació estan fetes accedint al sistema amb perfil d'administrador, per tal de mostrar totes les opcions que existeixen. És possible que algunes de les vistes es vegin lleugerament modificades i no apareguin totes les opcions segons el perfil d'usuari amb el qual s'accedeixi al sistema. Per consultar els permisos d'usuari de cada perfil, consulteu l'Annex I - capítol 1.5 - il·lustració 46).

S'ha de tenir en compte que aquest manual està fet sobre una versió primerenca d'OpenERP 6.0, i tant el programa com els mòduls oficials encara no estan traduïts al 100% al català, així que és possible que es vegin paraules en castellà o inclòs en anglès a les captures de pantalla següents.



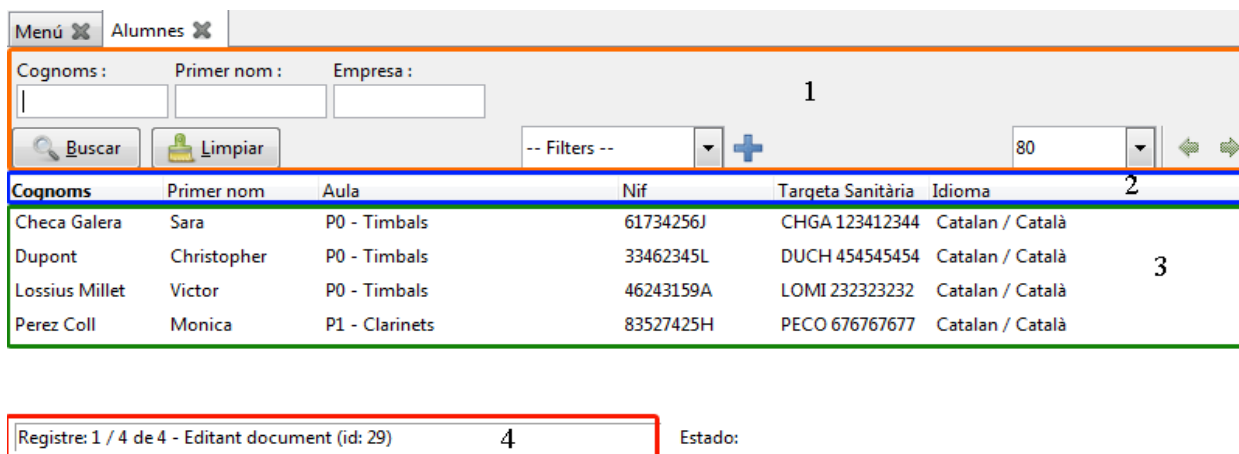
Il·lustració 50 - Menú principal client OpenERP

A continuació, mostrem cadascuna de les opcions disponibles en el menú del mòdul.

1.4.1 Gestió d'alumnes

Dins la carpeta "Gestió d'alumnes", ens apareixeran tres opcions: alumnes, responsables, i historial de factures.

1.4.1.1 Alumnes



Il·lustració 51 - Menú alumnes

Al seleccionar l'opció "Alumnes" ens apareixerà la vista arbre d'alumnes (il·lustració 51), en aquesta vista es poden distingir els següents elements:

1. Cercador. Des d'aquí podem fer una cerca de registres amb els filtres que nosaltres vulguem ficar. Si es prem el símbol "+" que hi ha, apareixeran més opcions de cerca. Un cop ficades les dades de la cerca, s'ha de prémer el botó "Buscar", i si volem tornar a la llista inicial amb tots els recursos, s'ha de prémer el botó "Netejar".
2. Columnes amb la diferent informació que mostra cada registre. Si es prem sobre el títol d'una columna, els registres s'ordenen segons el valor d'aquella columna.
3. Llistat de tots els registres disponibles.
4. Informació sobre els registres. Ens indica el número de registres totals i l'identificador del recurs seleccionat.

Des d'aquesta vista podem crear, donar de baixa, consultar o modificar alumnes. Per crear un nou alumne haurem de prémer el botó "Nou"; en cas que vulguem consultar/modificar un alumne, el seleccionarem de la llista i s'obrirà la vista formulari d'aquest:

The screenshot shows a web application interface for managing students. The main form is titled "Alumnes" and contains the following fields and sections:

- Search and Filter:** "Cognoms: Lossius Millet", "Primer nom: Victor", "Aula: P0 - Timbals".
- Navigation Tabs:** "General", "Informació extra", "Responsables", "Horari Escolar", "Notes".
- Personal Information:** "Nif: 46243159A", "Targeta Sanitària: LOMI 232323232".
- Jobs and Addresses:** A table with columns: Seq., Funció a l'empresa, Empresa, Adreça, Telèfon, Fa. The first row shows: 0, Carlos Lossius, Premia de Mar 8 08320, El Masnou.

The sidebar on the right contains the following options:

- Assistent participacions periòdiques
- Crear factura 2
- Futures Participacions
- Participacions del mes actual
- 3

The interface is annotated with three colored boxes and numbers:

- Red box (1):** Encloses the main form area.
- Blue box (2):** Encloses the sidebar options.
- Green box (3):** Encloses the sidebar options.

At the bottom, the status bar shows: "Registre: 3 / 4 de 4 - Editant document (id: 27) Estado:"

Il·lustració 52 - Menú formulari alumne

En aquesta vista podem distingir tres zones, en color vermell (1) veurem la informació d'aquest registre, en color blau (2) els assistents i en color verd (3) les dreceres.

1.4.1.1.1 Informació del registre (il·lustració 52 - Zona 1)

En aquesta il·lustració, veiem que conté el nom, cognoms i l'aula assignada, així com una sèrie de pestanyes. Els camps en blau, significa que són camps obligatoris, com ara els cognoms. En cas que vulguem assignar l'alumne a una aula o bé modificar la que ja té, ho fem seleccionant la lupa que hi ha al costat d'aquest camp; ens apareixerà el llistat d'aules que hi ha, i haurem d'escollir una d'elles. Si es dona el cas que l'aula està plena, l'aplicació ens retornarà un error.

Nom	Capacitat	Metres quadrats	Descripció
P0 - Timbals	8	15	Aula per a nens de 0 a 1 any
P1 - Clarinets	13	20	Aula per a nens de 1 a 2 anys
P1 - Fagots	13	20	Aula per a nens de 1 a 2 anys
P2 - Violins	20	35	Aula per a nens de 2 a 3 anys
P2 - Violoncels	20	35	Aula per a nens de 2 a 3 anys

Il·lustració 53 - Menú formulari alumne - escollir aula

Aquesta vista també conté una sèrie de pestanyes: a la primera (general) que es veu a la il·lustració 52, es podran afegir dades de l'alumne, així com a la següent pestanya (Informació extra) on hi apareixeran altres dades extra del alumne. A la tercera pestanya "Responsables", si la seleccionem ens apareixerà el llistat de responsables assignats a l'alumne actual:

Cognoms	Nom	Títol	Telèfon	Mòbil
Lossius	Carlos	Pare	931112211	619987654
Millet	Carina	Mare	931112211	619987654

Il·lustració 54 - Menú formulari alumne - pestanya responsables

En cas que vulguem treure un responsable, només haurem de seleccionar un d'ells i prémer el botó "Quitar",

però aquest llistat ha de contenir com a mínim un responsable, ja que sinó no ens deixarà desar-ho en el sistema. En cas que hi vulguem afegir un, premem el botó "Añadir", i ens apareixerà la següent finestra:

Cognoms	Nom	Títol	Telèfon	Mòbil
Checa	Paco	Pare	931111111	619222222
Coll	Cecilia	Mare	932222222	619444444
Dupont	Cristobal	Avi	931231231	619090909
Dupont	Christian	Pare	931119911	619123456
Galera	Helena	Mare	93333333	619555555
Lossius	Carlos	Pare	931112211	619987654
Millet	Carina	Mare	931112211	619987654

II·lustració 55 - Menú formulari alumne - cerca responsables

Aquí podem associar un responsable del llistat a l'alumne actual, en cas que no hi aparegui i vulguem donar d'alta un nou responsable, seleccionem "Nuevo" i ens apareixerà la vista formulari d'un nou responsable:

II·lustració 56 - Menú formulari alumne - alta responsable

Podrem afegir el nom, NIF i títol del responsable, on títol és la categoria d'aquest, ja sigui pare, mare, tutor, avi, etc. A més, podrem afegir la informació personal i la informació del treball, per tal de facilitar les possibilitats de contactar amb aquest. A la pestanya informació extra, podrem afegir la informació extra que nosaltres creguem necessària. Un cop finalitzem, premem el botó "Acceptar" per tal de desar la informació, i tornarem a la vista formulari de l'alumne (il·lustració 52).

La següent pestanya "Horari Escolar" de la vista de l'alumne, és on es poden registrar les participacions de l'alumne o modificar les existents, i la última pestanya "Notes", podrem afegir la informació extra que es cregui necessària, com ara posar a quin responsable contactar primer en cas de necessitar-ho.

En cas que vulguem donar de baixa un alumne, desmarcarem la casella "Actiu" situada a la pestanya "Informació extra", però mai esborrarem un alumne registrat en el sistema per tal de mantenir un històric.

1.4.1.1.2 Assistent participacions periòdiques (il·lustració 52 - Zona 2)

Aquest assistent situat a la zona en vermell, serveix per crear participacions automàticament entre un període de temps. D'aquesta manera, podrem tenir registrades totes les dades per cadascun dels dies de l'alumne. Quan seleccionem aquest assistent ens apareixerà la següent finestra:

OpenERP - Assistent per crear participacions periòdiques

Període

Data Inici : 01/05/2011 Data Final : 31/05/2011

Dies de la setmana

Dilluns : Dimarts :
Dimecres : Dijous :
Divendres :

Horari Escolar

Horari : Mati: 8:45h a 12:00h Hores Extra : No
Serveis extra : No Tipus de Menú :

Cancel·lar Afegir

Il·lustració 57 - Menú formulari alumne - assistent participacions periòdiques

Aquí tenim tres seccions diferenciades:

- Secció "Període": Podrem seleccionar la data inici i la data final d'aquest. Aquests camps estan marcats en blau, la qual cosa significa que són obligatoris. Podem afegir-los manualment, o seleccionar la lupa que hi ha al costat perquè ens aparegui un calendari. Si s'introdueixen dies que no són vàlids, el sistema retornarà error.
- Secció "Dies de la setmana": Aquí podem seleccionar els dies de la setmana els quals volem crear les participacions; per defecte venen tots marcats.
- Secció "Horari Escolar": Aquí podem seleccionar l'horari, si fa ús del servei d'acollida, si fa ús del menjador, i el tipus menú de l'alumne en cas que aquest es quedi al menjador.

Per exemple, en aquest cas (il·lustració 57), creem les participacions entre el 1 de maig i el 31 de maig, assignant-li l'horari de matí.

Si anem a la pestanya "Horari Escolar" de l'alumne, veurem el llistat de les participacions que hem creat:

General	Informació extra	Responsables	Horari Escolar	Notes																																																																																																		
<div style="display: flex; justify-content: space-between; align-items: center;"> Participacions (1/22 of 22) </div> <table border="1"> <thead> <tr> <th>Dia</th> <th>Horari</th> <th>Hores Extra</th> <th>Serveis extra</th> <th>Tipus de Menú</th> <th>Assistència</th> <th>Comentaris</th> </tr> </thead> <tbody> <tr><td>02/05/2011</td><td>Matí: 8:45h a 12:00h</td><td>No</td><td>No</td><td></td><td><input checked="" type="checkbox"/></td><td></td></tr> <tr><td>03/05/2011</td><td>Matí: 8:45h a 12:00h</td><td>No</td><td>No</td><td></td><td><input checked="" type="checkbox"/></td><td></td></tr> <tr><td>04/05/2011</td><td>Matí: 8:45h a 12:00h</td><td>No</td><td>No</td><td></td><td><input checked="" type="checkbox"/></td><td></td></tr> <tr><td>05/05/2011</td><td>Matí: 8:45h a 12:00h</td><td>No</td><td>No</td><td></td><td><input checked="" type="checkbox"/></td><td></td></tr> <tr><td>06/05/2011</td><td>Matí: 8:45h a 12:00h</td><td>No</td><td>No</td><td></td><td><input checked="" type="checkbox"/></td><td></td></tr> <tr><td>09/05/2011</td><td>Matí: 8:45h a 12:00h</td><td>No</td><td>No</td><td></td><td><input checked="" type="checkbox"/></td><td></td></tr> <tr><td>10/05/2011</td><td>Matí: 8:45h a 12:00h</td><td>No</td><td>No</td><td></td><td><input checked="" type="checkbox"/></td><td></td></tr> <tr><td>11/05/2011</td><td>Matí: 8:45h a 12:00h</td><td>No</td><td>No</td><td></td><td><input checked="" type="checkbox"/></td><td></td></tr> <tr><td>12/05/2011</td><td>Matí: 8:45h a 12:00h</td><td>No</td><td>No</td><td></td><td><input checked="" type="checkbox"/></td><td></td></tr> <tr><td>13/05/2011</td><td>Matí: 8:45h a 12:00h</td><td>No</td><td>No</td><td></td><td><input checked="" type="checkbox"/></td><td></td></tr> <tr><td>16/05/2011</td><td>Matí: 8:45h a 12:00h</td><td>No</td><td>No</td><td></td><td><input checked="" type="checkbox"/></td><td></td></tr> <tr><td>17/05/2011</td><td>Matí: 8:45h a 12:00h</td><td>No</td><td>No</td><td></td><td><input checked="" type="checkbox"/></td><td></td></tr> <tr><td>18/05/2011</td><td>Matí: 8:45h a 12:00h</td><td>No</td><td>No</td><td></td><td><input checked="" type="checkbox"/></td><td></td></tr> </tbody> </table>					Dia	Horari	Hores Extra	Serveis extra	Tipus de Menú	Assistència	Comentaris	02/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>		03/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>		04/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>		05/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>		06/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>		09/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>		10/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>		11/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>		12/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>		13/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>		16/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>		17/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>		18/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>	
Dia	Horari	Hores Extra	Serveis extra	Tipus de Menú	Assistència	Comentaris																																																																																																
02/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>																																																																																																	
03/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>																																																																																																	
04/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>																																																																																																	
05/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>																																																																																																	
06/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>																																																																																																	
09/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>																																																																																																	
10/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>																																																																																																	
11/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>																																																																																																	
12/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>																																																																																																	
13/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>																																																																																																	
16/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>																																																																																																	
17/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>																																																																																																	
18/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>																																																																																																	

Il·lustració 58 - Menú formulari alumne - pestanya horari escolar

Quan fem ús del assistent, s'exclouen del interval de temps els caps de setmana i els dies festius, és per això que no hi apareixen tots els dies del període de temps introduït.

Si volem modificar alguna participació, només hem de seleccionar-la i s'obrirà la finestra de la participació:

II·lustració 59 - Menú formulari alumne - participació

En el nostre exemple, modificarem la participació per indicar que l'alumne farà ús de l'acollida i el menjador, només el 9 de maig. Per desar totes les dades fetes fins ara, haurem de desar el registre seleccionant "Guardar" en el menú principal de l'aplicació.

Aquest assistent es pot fer servir per més d'un alumne a la vegada. Per fer això anirem a la vista arbre d'alumnes i marcarem els que vulguem; després seleccionarem l'opció "Accions" del menú principal de l'aplicació i escollirem l'assistent pertinent.

1.4.1.1.3 Assistent crear factura (il·lustració 52 - Zona 2)

Aquest assistent ens permet crear la factura de l'alumne seleccionat. Un cop cridem a l'assistent ens apareixerà una finestra que ens demanarà el mes que volem facturar:

II·lustració 60 - Menú formulari alumne - assistent crear factura

Un cop introduït el mes i si aquest ja ha finalitzat, es crearà la factura quan premem el botó "Acceptar". S'obrirà la pestanya de factures creades on ens apareixerà la nova factura creada:

Data factura	Número	Client	Descripció	Salesman	Data vencim S
31/05/2011		Carlos Lossius	5/2011 - Victor Lossius Millet	Administrator	
Residual Amount: 0,00		Base imposable: 130,70		Import total: 130,70	

II-lustració 61 - Menú formulari alumne - menú factura

On hi podem observar la data de factura, la persona a la que va dirigida la factura, la descripció on hi figura el mes i l'alumne, i per últim, el preu a pagar d'aquesta factura.

Si seleccionem la factura, la veurem desglossada en línies:

The screenshot shows the 'Created Invoices' menu with the following details:

- Diari: Escola Journal (EUR)
- Número: [Empty]
- Divisa: EUR (€)
- Proveïdor: Carlos Lossius
- Adreça factura: Premia de Mar 8 08:
- Posició fiscal: [Empty]
- Data factura: 31/05/2011

The invoice summary shows:

- Referència lliure: [Empty]
- Data venciment: __/__/__
- Total: 0,00

The 'Línies de factura' table is as follows:

Producte	Compte	Imposto:	Quantitat	Preu un.	Subtotal	Descripció
[QUOTAFIXA] Quota fixa escola bressol	X11002 Debtors - (test) (0)	(0)	1,00	116,20	116,20	Quota fixa escola bressol - Lossius Millet
[ACOLLIDA] Acollida: 7:45h a 8:45h	X11002 Debtors - (test) (0)	(0)	1,00	7,00	7,00	Acollida: 7:45h a 8:45h - Lossius Millet
[MENJADOR] Menjador	X11002 Debtors - (test) (0)	(0)	1,00	7,50	7,50	Menjador - Lossius Millet

Below the table, there are fields for 'Impostos' (1,2 E+3), 'Base: 130,70', 'Impost: 0,00', 'Total: 130,70', 'Pendent: 0,00', and buttons for 'Cancel·la' and 'Approve'.

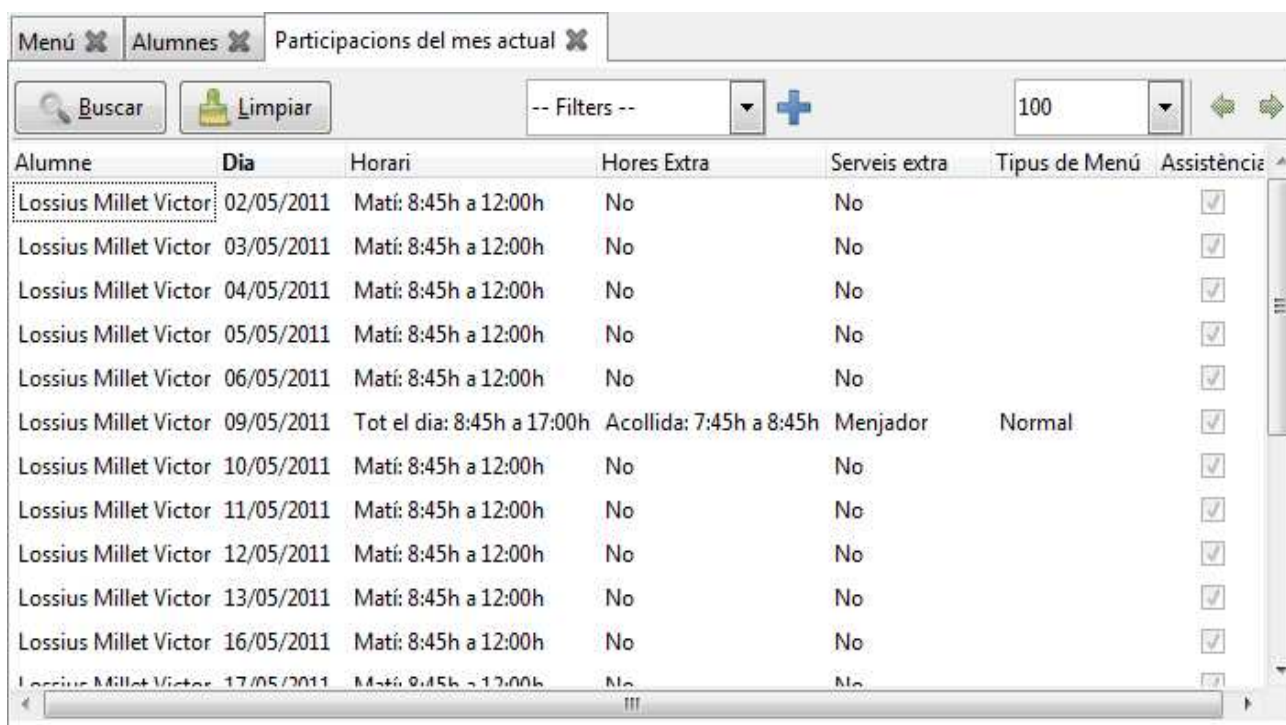
II-lustració 62 - Menú formulari alumne - menú factura - línies de factura

Aquest cas és del exemple fet de les participacions anteriorment, on hi podem veure tres línies: una per la quota fixa, una altra per l'ús del servei d'acollida que es va fer ús el 9 de maig, i l'última pel servei de menjador que també es va fer ús el 9 de maig. Cadascuna d'aquestes línies té el seu preu associat, i el preu final es pot veure al camp "Base". Un cop assegurat que és correcte, podem prémer el botó "Validar".

1.4.1.1.4 Dreceres (il·lustració 52 - Zona 3)

En aquesta zona hi trobem les dreceres, que serveixen per filtrar les dades de l'alumne actual. En aquest cas hi trobem dues, una ens serveix per filtrar les participacions futures, i la segona per filtrar les participacions

només del mes actual. En el nostre exemple, aquesta és la vista resultant de seleccionar la drecera "Participacions del mes actual".



Alumne	Dia	Horari	Hores Extra	Serveis extra	Tipus de Menú	Assistència
Lossius Millet Victor	02/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>
Lossius Millet Victor	03/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>
Lossius Millet Victor	04/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>
Lossius Millet Victor	05/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>
Lossius Millet Victor	06/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>
Lossius Millet Victor	09/05/2011	Tot el dia: 8:45h a 17:00h	Acollida: 7:45h a 8:45h	Menjador	Normal	<input checked="" type="checkbox"/>
Lossius Millet Victor	10/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>
Lossius Millet Victor	11/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>
Lossius Millet Victor	12/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>
Lossius Millet Victor	13/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>
Lossius Millet Victor	16/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>
Lossius Millet Victor	17/05/2011	Matí: 8:45h a 12:00h	No	No		<input checked="" type="checkbox"/>

Il·lustració 63 - Menú formulari alumne - drecera

1.4.1.2 Responsables

L'opció "Responsables" ens serveix per veure tots els responsables registrats al sistema i interactuar amb ells: podem crear, consultar, modificar o esborrar. D'aquesta manera podem treballar amb ells sense la necessitat d'accedir des de la vista d'alumnes. Quan accedim, ens apareixerà el llistat de tots els responsables registrats sense tenir en compte de quin o quins alumnes són responsables. Si seleccionem un d'ells, ens apareixerà la vista formulari com la que hem vist anteriorment (il·lustració 56), i podem treballar-hi de la mateixa manera que l'explicada abans.

1.4.1.3 Historial de Factures

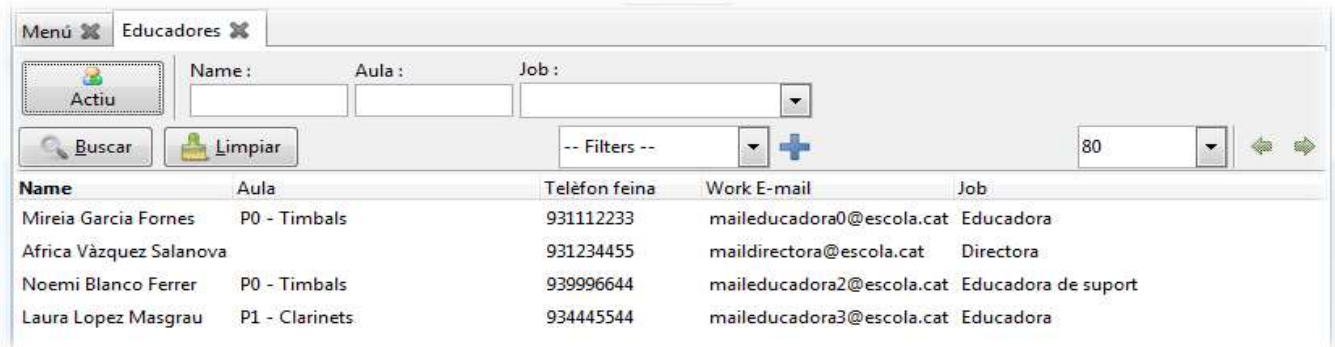
Aquesta opció és una drecera que ens porta directament a la vista de factures comentada anteriorment (il·lustració 61), on podem veure totes les factures del sistema.

1.4.2 Gestió d'educadores

Dins la carpeta "Gestió d'alumnes", ens apareixeran dues opcions: educadores i càrrecs.

1.4.2.1 Educadores

Amb aquesta opció podrem treballar-hi amb les educadores, ja sigui donar d'alta, consultar/modificar o donar de baixa. Quan seleccionem l'opció ens apareixerà el llistat d'educadores actives en el sistema:

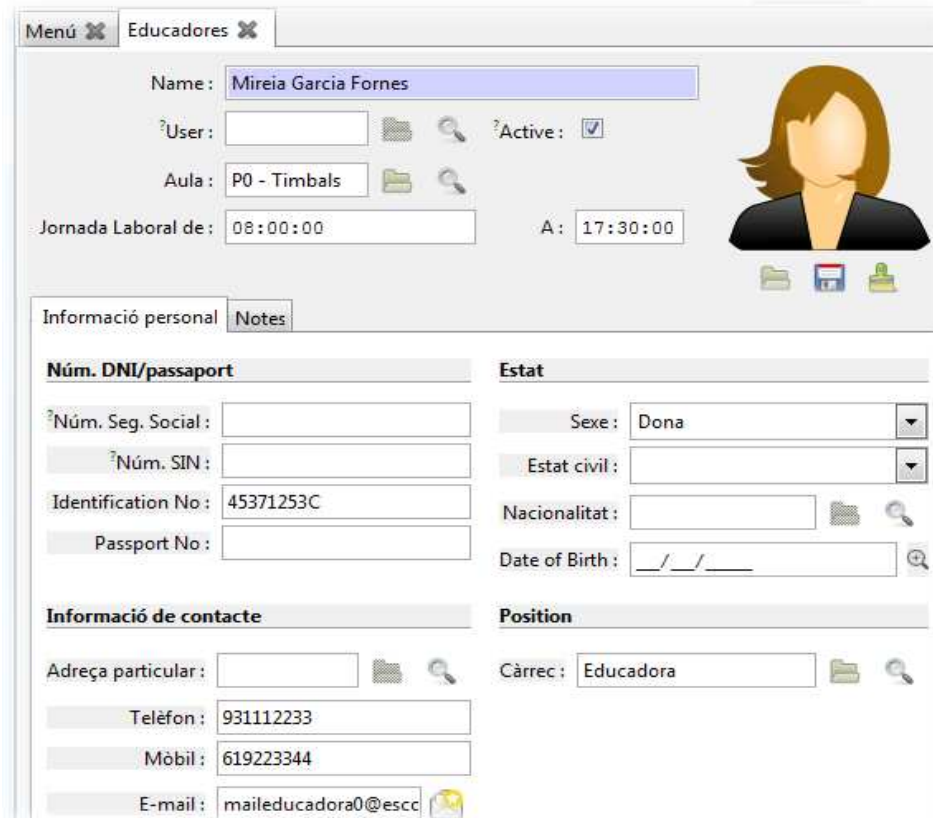


The screenshot shows a web application interface for managing educators. At the top, there is a 'Menú' and 'Educadores' tab. Below the tab, there are search and filter options, including 'Actiu', 'Buscar', 'Limpiar', and a filter dropdown set to '80'. The main content is a table with the following data:

Name	Aula	Telèfon feina	Work E-mail	Job
Mireia Garcia Fornes	P0 - Timbals	931112233	maileducadora0@escola.cat	Educadora
Africa Vázquez Salanova		931234455	maildirectora@escola.cat	Directora
Noemi Blanco Ferrer	P0 - Timbals	939996644	maileducadora2@escola.cat	Educadora de suport
Laura Lopez Masgrau	P1 - Clarinets	934445544	maileducadora3@escola.cat	Educadora

II·lustració 64 - Menú educadora

Per donar d'alta una educadora premem el botó "Nou" i ens apareixerà la finestra de nou alumne. Si volem consultar o modificar seleccionarem una educadora de la llista.



The screenshot shows the 'Educadores' menu with the 'Mireia Garcia Fornes' record selected. The form displays the following information:

Name: Mireia Garcia Fornes
User: [] ?Active:
Aula: P0 - Timbals
Jornada Laboral de: 08:00:00 A: 17:30:00

Informació personal

Núm. DNI/passaport: [] Estat: []
Núm. Seg. Social: [] Sexe: Dona
Núm. SIN: [] Estat civil: []
Identification No: 45371253C Nacionalitat: []
Passport No: [] Date of Birth: []

Informació de contacte

Adreça particular: [] Càrrec: Educadora
Telèfon: 931112233
Mòbil: 619223344
E-mail: maileducadora0@esc

II·lustració 65 - Menú formulari educadora

En aquesta finestra podrem afegir totes les dades necessàries d'una educadora, així com l'usuari que li atorguem per accedir al sistema. Quan seleccionem el camp aula, s'obrirà una finestra per escollir l'aula que volem assignar-li, i el mateix amb el camp càrrec, on ens donarà a escollir el càrrec que li assignem a l'educadora.

Per donar de baixa una educadora desmarcarem la casella "Actiu" de la vista formulari d'aquesta educadora. Mai s'esborrarà ja que volem mantenir l'històric d'educadores.

1.4.2.2 Càrrecs

Amb aquesta opció podrem crear els càrrecs que utilitzarem per assignar a les educadores. Quan seleccionem aquesta opció ens apareixerà la llista de càrrecs:

Job Name	Expected Employees	No of Employee	Expected in Recruitment	State
Directora	1	1,00	0,00	In Position
Educadora	5	2,00	3,00	In Position
Educadora de suport	5	1,00	4,00	In Position
Educadora en pràctiques	1	0,00	1,00	In Position
Personal de cuina	1	0,00	1,00	In Position

Il·lustració 66 - Menú càrrecs

Quan accedim per primer cop hi trobarem aquests càrrecs estàndards per defecte; en cas que vulguem crear o esborrar, ho farem amb les opcions del menú principal d'OpenERP. Per consultar o modificar algun d'aquests només l'hauré de seleccionar i s'obrirà la vista formulari:

Name	Aula	Telèfon	E-mail	Càrrec
Mireia Garcia Fornes	P0 - Timbals	931112233	maileducadora0@escola.cat	Educadora
Laura Lopez Masgrau	P1 - Clarinets	934445544	maileducadora3@escola.cat	Educadora

Il·lustració 67 - Menú formulari càrrec

En la vista podrem indicar el nombre d'empleats que ha de tenir aquest càrrec, i automàticament ens apareix el nombre d'empleats que falta per contractar. A la pestanya "Empleat" podrem veure els empleats assignats a aquest càrrec.

En aquest exemple (il·lustració 67), hi podem veure com pel càrrec "Educatora" hi ha assignades dues educadores.

1.4.3 Aules

Dins la carpeta "Aules", trobarem les opcions per gestionar aquestes. Hi apareixen tres opcions: aules, material escolar i consultar el material per renovar

1.4.3.1 Aules

Al accedir en aquesta opció ens apareixerà el llistat d'aules, i la manera de funcionar és l'explicada fins ara. Seleccionem una aula per tal de consultar o modificar les dades i s'obrirà aquesta finestra:

The screenshot shows a software interface for managing classrooms. At the top, there are two tabs: 'Menú' and 'Aules'. Below the tabs, there are three input fields: 'Nom:' with the value 'P0 - Timbals', 'Capacitat:' with the value '8', and 'Metres quadrats:' with the value '15'. To the right of these fields are two buttons: 'Educatores Assignades' and 'Alumnes Assignats'. Below the input fields, there are two tabs: 'Material Assignat' and 'Informació Extra'. The 'Material Assignat' tab is active, showing a table of school materials. The table has four columns: 'Nom', 'Codi', 'Tipus', and 'Estat'. There are four rows of data in the table. Below the table, there are several icons and a page indicator '(1/4 of 4)'.

Nom	Codi	Tipus	Estat
Aladin	a001	Joguina	Perfecte
Aladin	a003	Joguina	Deteriorat
Rey Lleó	r001	Llibre escolar	Perfecte
Cançons Infantils recopilatori	c001	Música	Perfecte

Il·lustració 68 - Menú formulari aula

En ella podrem veure el nom de l'aula així com la capacitat d'aquesta. Aquesta vista conté dues dreceres referents a l'aula que tinguem oberta, la primera ens servirà per veure les educadores que imparteixen classe a l'aula, i la segona drecera, serveix per veure els alumnes assignats a l'aula.

A la pestanya "Material Assignat" podrem veurem el material assignat a aquesta, i a la dreta del nom del camp es poden veure una sèrie de botons:



II·lustració 69 - Barra eines material escolar

Aquests botons ens permeten:

1. Crear un nou registre que s'afegirà a la pestanya de "material assignat".
2. Obrir el registre seleccionat en una nova finestra per veure'l amb més detall o editar-lo.
3. Esborrar el registre seleccionat.
4. Informació sobre el nombre de registres totals i quin és el seleccionat.
5. Passar de la vista arbre a la vista formulari o viceversa.

Quan creem o seleccionem un exemplar de material escolar, s'obrirà aquesta finestra:

II·lustració 70 - Menú formulari aula - pestanya material escolar

En aquesta, podem assignar-hi les dades referents al exemplar de material escolar. Cal dir que el camp "Codi" és un camp obligatori i únic, per tant, si es fica un codi ja assignat a un altre exemplar, el sistema ens retornarà un error.

Aquesta vista aula està pensada per tal de que cada educadora porti la gestió de la seva aula assignada. Cada educadora responsable d'una aula serà l'encarregada del material assignat a la seva aula, i si per exemple, una joguina es trenca o es deteriora, l'educadora ho pot indicar a l'aplicació.

1.4.3.2 Material Escolar

Des d'aquesta opció, podem veure tot el material registrat a l'aplicació independentment de l'aula a la que estigui assignada; d'aquesta manera no caldrà veure el material accedint des de l'aula. Quan accedim,

apareixerà el llistat de material assignat. Si seleccionem algun exemplar, ens apareixerà una finestra semblant a la vista anteriorment, però en aquest cas, hi ha un camp que ens indica l'aula assignada.

Menú ✕ Material Escolar ✕

Informació Exemplar

?Nom : Aladin ?Codi : a001
 ?Tipus : Joguina ?Estat : Perfecte

Aula Assignada

P0 - Timbals

Informació Extra

Ninot aladin. Tamany 12x20

Il·lustració 71 - Menú formulari material escolar

1.4.3.3 Consultar material escolar per renovar

Aquesta opció és una drecera feta per facilitar la feina per tal de veure el material que es troba en estat deteriorat o trencat, per tal de saber quin s'ha de renovar. Quan la seleccionem, apareixerà un llista de tot el material escolar deteriorat independentment de l'aula a la que estigui assignada.

Menú ✕ Material per renovar ✕

Tipus : Nom :

Buscar Limpiar -- Filters -- 100

Nom	Codi	Tipus	Estat	Aula Assignada
Aladin	a003	Joguina	Deteriorat	P0 - Timbals
Cançons Disney: Canta conmigo	c002	Música	Trencat	P1 - Clarinets

Il·lustració 72 - Drecera material escolar per renovar

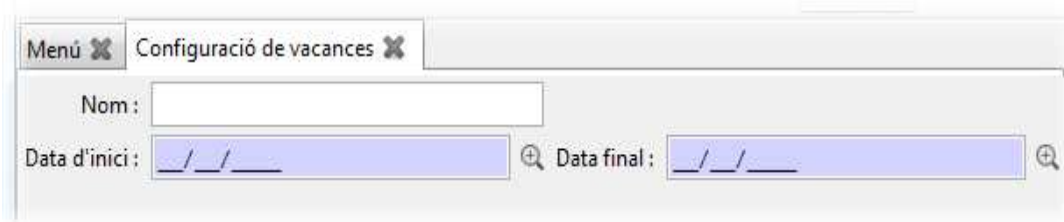
1.4.4 Configuració

Dins la carpeta "Configuració", trobarem les opcions de configuració de l'aplicació. Hi apareixen diferents opcions: vacances, menús diaris, tipus de quotes i totes les participacions

1.4.4.1 Vacances

Amb aquesta opció podrem crear períodes de vacances. Això vol dir que durant aquests dies de vacances no es crearan participacions, ja que els alumnes no assistiran a l'escola.

Com sempre, primer ens apareixerà el llistat de períodes creats, i si volem crear un de nou, premem el botó "Nou" i s'obrirà la finestra formulari.

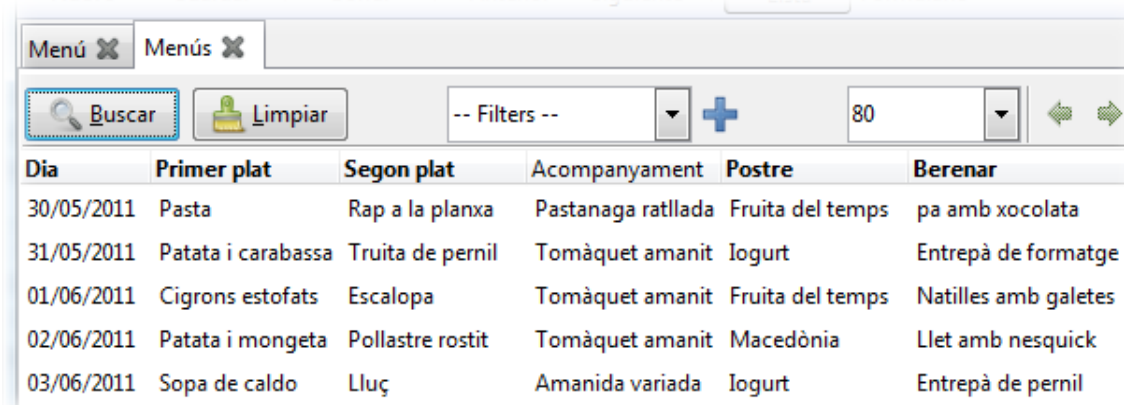


Il·lustració 73 - Menú formulari vacances

Els períodes de vacances no es poden solapar, ja que l'aplicació ens retornarà un error quan es doni el cas.

1.4.4.2 Menús diaris

Des d'aquesta opció podrem crear els menús diaris amb la informació que aquests contenen. En cas que creem dos menús pel mateix dia, el sistema ens retornarà un error.



Dia	Primer plat	Segon plat	Acompanyament	Postre	Berengar
30/05/2011	Pasta	Rap a la planxa	Pastanaga ratllada	Fruita del temps	pa amb xocolata
31/05/2011	Patata i carabassa	Truita de pernil	Tomàquet amanit	Iogurt	Entrepà de formatge
01/06/2011	Cigrans estofats	Escalopa	Tomàquet amanit	Fruita del temps	Natilles amb galetes
02/06/2011	Patata i mongeta	Pollastre rostit	Tomàquet amanit	Macedònia	Llet amb nesquick
03/06/2011	Sopa de caldo	Lluç	Amanida variada	Iogurt	Entrepà de pernil

Il·lustració 74 - Menú menús

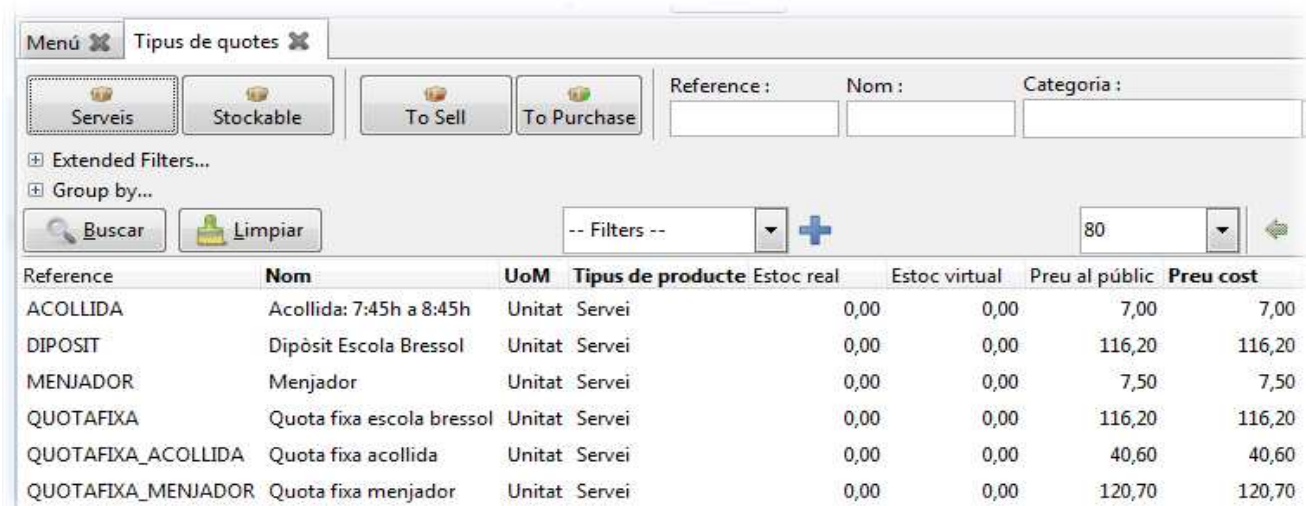
Si seleccionem la drecera "Veure menús del mes actual", es filtraran tots els menús del mes en que ens trobem per tal de que sigui més fàcil la lectura de menús.

1.4.4.3 Totes les participacions

Des d'aquesta opció veurem totes les participacions creades independentment de l'alumne a qui estan assignades. A més, ens ho permet veure en format calendari.

1.4.4.4 Tipus de quotes

Aquí podem veure els tipus de quotes que té l'escola. Aquestes es creen per defecte quan s'instal·la el mòdul, i ens permetrà modificar les dades d'aquestes, per exemple quan el preu de les quotes es vegi modificat.



Reference	Nom	UoM	Tipus de producte	Estoc real	Estoc virtual	Preu al públic	Preu cost
ACOLLIDA	Acollida: 7:45h a 8:45h	Unitat	Servei	0,00	0,00	7,00	7,00
DIPOSIT	Dipòsit Escola Bressol	Unitat	Servei	0,00	0,00	116,20	116,20
MENJADOR	Menjador	Unitat	Servei	0,00	0,00	7,50	7,50
QUOTAFIXA	Quota fixa escola bressol	Unitat	Servei	0,00	0,00	116,20	116,20
QUOTAFIXA_ACOLLIDA	Quota fixa acollida	Unitat	Servei	0,00	0,00	40,60	40,60
QUOTAFIXA_MENJADOR	Quota fixa menjador	Unitat	Servei	0,00	0,00	120,70	120,70

Il·lustració 75 - Menú quotes

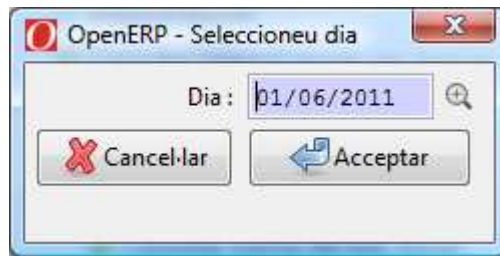
1.4.5 Assistents consulta

Existeixen tres assistents per consultar dades:

- Consultar assistència pel dia...: Ens serveix per consultar els alumnes que tenen previst assistir a l'escola el dia seleccionat.
- Consultar menjador pel dia...: Ens serveix per consultar els alumnes que tenen previst fer ús del menjador el dia seleccionat.
- Consultar acollida pel dia...: Ens serveix per consultar els alumnes que tenen previst fer ús del servei d'acollida pel dia seleccionat.

Els tres funcionen de la mateixa manera així que només explicarem un d'ells.

Agafem per exemple l'assistent per consultar l'acollida, quan el seleccionem s'obrirà una finestra demanant el dia el qual volem fer la consulta, per defecte ens apareixerà el dia actual:



Il·lustració 76 - Assistent consultar acollida pel dia...

Un cop seleccionat, apareixerà el llistat dels alumnes que fan ús del servei d'acollida.

Alumne	Dia	Horari	Hores Extra	Serveis extra	Tipus de Menú	Assistèn
Lossius Millet Victor	01/06/2011	Matí: 8:45h a 12:00h	Acollida: 7:45h a 8:45h	No		<input checked="" type="checkbox"/>
Dupont Christopher	01/06/2011	Matí i Migdia: 8:45h a 15:00h	Acollida: 7:45h a 8:45h	Menjador	Normal	<input checked="" type="checkbox"/>
Perez Coll Monica	01/06/2011	Tot el dia: 8:45h a 17:00h	Acollida: 7:45h a 8:45h	Menjador	Triturat	<input checked="" type="checkbox"/>

Il·lustració 77 - Assistent consultar acollida pel dia...

Annex III: Programes usats

Durant la creació d'aquest projecte s'han usat els següents programes o tecnologies:

- **Sistemes operatius**

- Microsoft Windows vista Professional 64 bits: Sistema operatiu de pagament de Microsoft.
<http://windows.microsoft.com/es-ES/windows-vista/products/home>
- Microsoft Windows 7 Professional 32 bits: Sistema operatiu de pagament de Microsoft.
<http://windows.microsoft.com/es-ES/windows-7?os=win7>
- Ubuntu 9.10: Versió del sistema linux amb escriptori GNOME. Versió de 32 bits.
<http://www.ubuntu.com/>

- **Creació de diagrames**

- Dia: programa gratuït per crear diagrames de diferents classes. En aquest projecte s'ha utilitzat per crear el model conceptual i els diagrames de col·laboració.
<http://projects.gnome.org/dia/>
- GanttProject: programa gratuït per crear diagrames de Gantt.
<http://www.ganttproject.biz/>
- Microsoft Office Excel: Eina de Microsoft de pagament que s'ha utilitzat per crear alguns dels gràfics d'aquest PFC.

- **Programació**

- Gedit: Editor de textos lliure, s'ha usat la versió 2.30.1 de Microsoft.
<http://ftp.gnome.org/pub/GNOME/binaries/win32/gedit/2.30/>
- Eclipse: IDE de programació molt completa.
<http://www.eclipse.org/>

- **Edició Gràfica**

- Paint: Eina que incorpora windows per defecte que s'ha fet servir per retocar alguna imatge o diagrama.
<http://windows.microsoft.com/en-us/windows7/products/features/paint>
- Adobe Photoshop 8.0.1: Editor gràfic de pagament que s'ha fet servir per retocar imatges.
<http://www.adobe.com/products/photoshop.html>

- **Llenguatges de programació usats**

- Python: Llenguatge de programació que usa OpenERP.

<http://www.python.org/>

- XML: Llenguatge d'etiquetes que usa OpenERP.

<http://www.w3.org/XML/>

- **Bases de dades**

- PostgreSQL: Gestor de bases de dades que usa OpenERP.

<http://www.postgresql.org/>

- **Administrador de la base de dades**

- pgAdmin: Eina per administrar les bases de dades PostgreSQL.

<http://www.pgadmin.org/>

- **Ofimàtica**

- Microsoft Office 2002: Suit ofimàtica de Microsoft, de pagament. Eina utilitzada per la redacció de la memòria.

- OpenOffice: Suit ofimàtica lliure. Eina utilitzada per la redacció de la memòria.

<http://www.openoffice.org/>

Bibliografia

A continuació podem veure les diferents fonts d'informació consultades per la creació d'aquest PFC:

- Web EPSEVG.
<http://www.epsevg.upc.edu/>
- Wikipedia, per la realització de diferents consultes.
<http://es.wikipedia.org/wiki/Wikipedia:Portada>
- Organització i planificació d'un PFC.
http://www.lsi.upc.edu/~bcasas/docencia/pfc/manual_PFC2.pdf
- Web d'OpenERP.
<http://www.openerp.cat/>
- Web Compiere.
<http://www.compiere.com/>
- Web Openbravo
<http://www.openbravo.com/es/>
- Web FacturaLux
<http://factualux.org>
- Aules OpenERP, per l'aprenentatge amb diferents tutorials.
<http://www.openerp.cat/aula/>
- Diapositives OpenERP, l'ERP lliure.
<http://www.slideshare.net/raimonesteve/openerp-lerp-lliure>
- Fòrum OpenERP.
<http://www.openerp.com/forum/>
- Llibre d'OpenERP, Open Object Developer Book.
<http://brochures.sisalp.fr/openobject-developer.pdf>
- OpenERP Technical Memento (Resum).
http://www.openerp.com/download/OpenERP_Technical_Memento_v0.6.3.pdf
- Launchpad. Per la consulta dels mòduls extra.
<https://launchpad.net/>
- Projecte OpenERP-School.
<https://launchpad.net/school-base-openerp-module>
- Python documentation.
<http://docs.python.org/contents.html>
- Guia d'estil del codi Python.
<http://mundogeek.net/traduccion/es/guia-estilo-python.htm>

- Web PostgreSQL
<http://www.postgresql.org/>
- Biblioteca EPSEVG.
<http://biblioteca.upc.edu/VILANOVA/>
- Web PFC Biblioteca UPC.
<http://biblioteca.upc.es/e-portals/PFC/>
- Apunts ESO1 (Enginyeria del Software: Disseny) de J.M. Merenciano.
- Llibret informatiu sobre escoles bressol municipals de Badalona.

