

# *Fester Manager*

**MEMORIA PRESENTADA PER:**  
*Rubén Cerdà Roig*

GRAU D'ENGINYERIA INFORMÀTICA

**Direcció:** Pau Micó

**Convocatoria de defensa:** *Setembre de 2015*



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA  
CAMPUS D'ALCOI

## INDEX

0. Acrònims.....	7
1. Introducció.....	11
1.1 Antecedents i motius de l'elecció.....	11
1.2 Escenari.....	11
1.3 Proposta.....	11
1.4 Objectius.....	11
1.5 Estructura del document.....	12
2. Avantprojecte.....	13
2.1 Estat de l'art.....	13
2.1.1 Control de versions.....	13
2.1.2 Entorns de Desenvolupament.....	13
2.1.3 Llenguatges de programació.....	14
2.1.4 Aplicacions Web.....	16
2.1.5 Servidors de Bases de dades.....	18
2.1.6 Servidors Web.....	18
2.2 Estudi de propostes.....	19
2.2.1 Proposta 1.....	19
2.2.2 Proposta 2.....	20
2.3 Justificació.....	21
3. Especificació de requeriments.....	22
3.1 Requeriments funcionals.....	22
3.2 Requeriments no funcionals.....	23
4. Planificació.....	24
4.1 Disseny centrat en l'usuari (DCU).....	24
4.2 Participants.....	25
4.3 Diagrama de Gantt.....	25
5. Anàlisi i disseny del sistema.....	27
5.1 Usuaris.....	27
5.2 Disseny conceptual.....	27
5.2.1 Escenaris d'ús.....	27
5.2.2 Diagrama de casos d'ús.....	28
5.2.2.1 Mànager.....	28
5.2.2.2 Usuari.....	31
5.3 Disseny de model de dades.....	33
5.4 Disseny de les vistes.....	34
5.4.1 Sketches.....	34
6. Implementació.....	40
6.1 Arquitectura de l'aplicació.....	44
6.1.1 Lògica de l'aplicació.....	48
6.1.2 Vistes.....	51
6.1.3 Permisos i seguretat.....	52
6.1.4 Internacionalització.....	53
6.1.5 Persistència de dades.....	53
7. Proves.....	55
7.1 Proves unitàries.....	55

7.2 Proves d'integració.....	56
8. Instal·lació i manual d'usuari.....	57
8.1 Instal·lació i posada en funcionament d'Odoo.....	57
8.2 Manual de Manager.....	69
8.2.1 Login.....	69
8.2.2 Preferències d'usuari.....	69
8.2.3 Socis i altres.....	71
8.2.3.1 Socis.....	71
8.2.3.2 Xiquets.....	72
8.2.3.3 Ex-Socis.....	72
8.2.4 Administració.....	72
8.2.4.1 Anys.....	72
8.2.4.2 Categories.....	75
8.2.4.3 Assentament Comptable.....	76
8.2.4.4 Usuaris.....	77
8.2.4.5 Grups.....	77
8.3 Comptabilitat.....	79
8.3.1 Pagaments actuals.....	79
8.3.2 Pagaments passats.....	81
8.3.3 Deure i Haver.....	81
8.4 Manual d'Usuari.....	82
9. Futures ampliacions.....	84
10. Conclusions.....	85
11. Bibliografia.....	86

## INDEX DE FIGURES

Figura 1: Arquitectura MVC Java EE.....	19
Figura 2: Capes de l'arquitectura MVC.....	19
Figura 3: Arquitectura Odoo.....	20
Figura 4: DCU.....	25
Figura 5: DCU àgil.....	25
Figura 6: Diagrama de Gantt.....	26
Figura 7: Cas d'ús de Socis.....	29
Figura 8: Cas d'ús de Xiquets.....	29
Figura 9: Cas d'ús de Pagaments.....	30
Figura 10: Cas d'ús de Deure/Haver.....	30
Figura 11: Cas d'ús d'Any.....	30
Figura 12: Cas d'ús d'Assentament comptable.....	31
Figura 13: Cas d'ús de Categoria.....	31
Figura 14: Cas d'ús d'Usuaris.....	31
Figura 15: Cas d'ús de Grups.....	32
Figura 16: Cas d'ús de Socis per a usuari.....	32
Figura 17: Cas d'ús de Pagaments per a usuari.....	32
Figura 18: Cas d'ús d'Usuaris per a usuari.....	33
Figura 19: Model de base de dades.....	37
Figura 20: Sketch estructura principal.....	38
Figura 21: Sketch Socis.....	40
Figura 22: Sketch Xiquets.....	41
Figura 23: Sketch Pagament.....	41
Figura 24: Sketch Deure i Haver.....	42
Figura 25: Sketch Any.....	42
Figura 26: Sketch Categoria.....	43
Figura 27: Sketch Assentament Comptable.....	43
Figura 28: Sketch Usuari.....	44
Figura 29: Sketch Grup.....	47
Figura 30: Implementació al núvol, Binami.....	51
Figura 31: Implementació al núvol, Binami.....	51
Figura 32: Implementació al núvol, Binami.....	52
Figura 33: Arquitectura Odoo.....	53
Figura 34: Model Vista Controlador.....	54
Figura 35: Arbre del mòdul.....	55
Figura 36: Exemple de fitxer XML del mòdul.....	59
Figura 37: Fitxer ir.model.access.csv amb permisos.....	60
Figura 38: Fitxer openerp-server.conf.....	62
Figura 39: Configuració amb PGAdmin 3.....	62
Figura 40: Configuració PGAdmin 3.....	78
Figura 41: Creació Base de dades.....	79
Figura 42: Pantalla de Login.....	82
Figura 43: Permisos d'usuari Administrador.....	82
Figura 44: Instal·lació Mòdul.....	83
Figura 45: Instal·lació Mòdul.....	83

Figura 46: Carregar Idiomes.....	84
Figura 47: Carregar Idiomes.....	84
Figura 48: Instal·lació Mòdul Xarxa Social.....	85
Figura 49: Configuració Servidor de Correu d'eixida.....	85
Figura 50: Creació Usuari Manager.....	86
Figura 51: Correu d'invitació a Odoo.....	87
Figura 52: Afegir usuari a grup Manager.....	88
Figura 53: Pantalla Login.....	88
Figura 54: Duplicar base de dades.....	89
Figura 55: Eliminar base de dades.....	89
Figura 56: Còpia de seguretat de la base de dades.....	90
Figura 57: Restaurar base de dades.....	90
Figura 58: Canviar contrassenya.....	90
Figura 59: Pantalla Login.....	91
Figura 60: Menú preferències d'usuari.....	91
Figura 61: Preferències d'usuari.....	92
Figura 62: Creació de Soci.....	93
Figura 63: Assignació de Categoria i Usuari.....	93
Figura 64: Formulari Xiquets.....	94
Figura 65: Formulari Any.....	94
Figura 66: Botons d'any.....	95
Figura 67: Vista amb any actiu.....	95
Figura 68: Pop-Up crear assentament.....	96
Figura 69: Assentament creat.....	96
Figura 70: Carregar Assentament.....	97
Figura 71: Formulari Categoria.....	97
Figura 72: Formulari Categoria.....	98
Figura 73: Formulari Assentament Comptable.....	98
Figura 74: Creació Usuari.....	99
Figura 75: Canviar contrassenya a usuari.....	99
Figura 76: Tree de grups.....	99
Figura 77: Afegir usuaris a grup.....	100
Figura 78: Afegir usuaris a grup.....	100
Figura 79: Pagaments actuals.....	101
Figura 80: Detall pagament.....	101
Figura 81: Afegir deure.....	102
Figura 82: Afegir Haver.....	102
Figura 83: Eliminar Assentament de deure o haver.....	103
Figura 84: Formulari Soci per a usuari.....	104
Figura 85: Formulari Usuari per a usuari.....	105

## 0. Acrònims

- **AGPL:** Llicència pública general de Affero, és una llicència copyleft derivada de la llicència Pública General de GNU dissenyada específicament per a assegurar la cooperació amb la comunitat en el cas de software que corre en servidors de xarxa.
- **API:** Interfície de Programació d'Aplicacions, és el conjunt de subrutines, funcions i procediments que ofereix certa biblioteca per a ser utilitzat per altre software com una capa d'abstracció.
- **BSD:** Llicència software otorgada principalment per als sistemes BSD (Berkeley Software Distribution).
- **C++:** Llenguatge de programació dissenyat a mitat dels anys 80.
- **C:** Llenguatge de programació originalment desenvolupat per Dennis M. Ritchie entre l'any 69 i 72, com a evolució del anterior llenguatge B.
- **CDDL:** Llicència Comú de Desenvolupament i Distribució, és una llicència de codi obert i lliure, produïda per Sun Microsystems, basada en la Mozilla Public License, versió 1.1.
- **CSS:** Fulla d'estil en cascada, és un llenguatge utilitzat per a definir i crear la presentació d'un document estructurat escrit en HTML o XML.
- **CSV:** Tipus de document en format obert senzill per a representar dades en forma de taula, en les que les columnes se separen per comes i les files per bots de línia.
- **DCU:** Disseny Centrat en l'Usuari, és una filosofia de disseny que té per objecte la creació de productes que resolguen unes necessitats concretes dels seus usuaris finals, conseguint la major satisfacció i millor experiència d'ús possible amb el mínim esforç per la seua part.
- **EJB:** Enterprise JavaBeans, són una de les interfícies de programació d'aplicacions que formen part de l'estàndar de construcció d'aplicacions empresarials.
- **ERP:** Sistema de planificació de recursos empresarials.
- **GIT:** Software de control de versions dissenyat per Linus Torvalds.

- **GNU:** Sistema Operatiu de tipus Unix.
- **GNU GPL:** Llicència Pública General de GNU, és la llicència més àmpliament utilitzada en el món del software i garanteix als usuaris finals la llibertat d'utilitzar, estudiar, compartir i modificar el software.
- **GTK+:** GIMP ToolKit, és un conjunt de biblioteques multiplataforma per a desenvolupar interfícies gràfiques d'usuari (GUI), principalment per als entorns gràfics GNOME, XFCE i ROX.
- **HTTP:** Hyper Text Transfer Protocol, és el protocol utilitzat en cada transacció de la World Wide Web.
- **HTTPS:** És la versió segura de HTTP.
- **IDE:** Ambient de Desenvolupament Integrat, en anglès Integrated Development Environment. Aplicació informàtica que proporciona serveis integrals per a facilitar-li al desenvolupador o programador el desenvolupament del software.
- **J2SE:** Col·lecció d'API's del llenguatge de programació Java.
- **JAVA:** es un llenguatge de programació de propòsit general, concurrent, orientat a objectes que fou dissenyat específicament per a tindre tant poques dependències com fora possible.
- **JAVA EE:** Java Platform, Enterprise Edition, es una plataforma de programació per a desenvolupar i executar software d'aplicacions en el llenguatge de programació Java.
- **JDBC:** Java Database Connectivity, és una API que permet l'execució d'operacions sota bases de dades des del llenguatge de programació Java.
- **JMS:** Java Message Service, és la solució creada per Sun Microsystems per a l'ús de coles de missatges.
- **JOOMLA:** Sistema de gestió de continguts que permet desenvolupar llocs web dinàmics i interactius.
- **JSF:** JavaServer Faces, és una tecnologia i framework per a aplicacions Java basades en web.
- **JSP:** JavaServer Pages és una tecnologia que ajuda als desenvolupadors de software a crear pàgines web dinàmiques basades en HTML, XML, etc.



- **JPA:** Java Persistence API, és l'API de persistència desenvolupada per a la plataforma Java EE.
- **MVC:** Model Vista Controlador, és un patró d'arquitectura de software que separa les dades i la lògica de negoci d'una aplicació de l'interfície d'usuari i el mòdul encarregat de gestionar els events i les comunicacions.
- **Odoo:** és un sistema ERP integrat de codi obert.
- **OpenERP:** és un sistema ERP integrat de codi obert, la versió anterior a Odoo.
- **ORM:** Mapeig Objecte-Relacional, és una tècnica de programació per a convertir dades entre el sistema de tipus utilitzat en un llenguatge de programació orientat a objectes i l'utilització d'una base de dades relacional com a motor de persistència.
- **PHP:** Llenguatge de programació d'ús general de codi del costat del servidor originalment desenvolupat per al desenvolupament web de contingut dinàmic.
- **RDBMS:** Sistema de Gestió de Bases de dades basat en el model relacional.
- **RMI:** Java remote Method Invocation, és un mecanisme ofert per Java per a invocar un mètode de forma remota.
- **SaaS:** Software com a Serveï, és un model de distribució de software on el suport es lògic i les dades que controla s'allotgen en servidor d'una companyia de tecnologies d'informació i comunicació, als que s'accedeix vïa internet des d'un client.
- **SMTP:** Protocol per a transferència simple de correu, és un protocol de red utilitzat per al intercanvi de missatges de correu electrònic entre computadores o altres dispositius.
- **SQL:** Llenguatge de consulta estructurat, és un llenguatge declaratiu d'accés a bases de dades relacionals que permet especificar diversos tipus d'operacions amb elles.
- **UNIX:** Sistema Operatiu portable, multitasca i multiusuari.
- **XML:** de l'anglès eXtensible Markup Language, es un llenguatge de marques desenvolupat per el World Wide Web Consortium utilitzat per a emmagatzemar dades en forma llegible.



- **XML-RPC:** És un protocol de cridada a procediment remot que utilitza RML per a codificar les dades i HTTP com a protocol de transmissió de missatges.
- **YML:** Format de serialització de dades llegible per humans inspirat en llenguatges com XML, C, Python o Perl.

## **1. Introducció**

### ***1.1 Antecedents i motius de l'elecció***

El projecte actual s'engloba dins de l'àrea de desenvolupament d'aplicacions web. De les diferents propostes de l'equip docent, s'ha escollit l'opció del "Fester Manager", treball proposat i dirigit pel docent Pau Micó.

El principal motiu pel que he decidit realitzar aquest treball de final de grau és el de millorar la meua formació en aplicacions web, endinsant-me un poc més en una àrea que m'interessa molt i que té molt de mercat, i molt de marge de creixement. A banda, professionalment m'ajudarà a millorar en l'empresa on estic realitzant les pràctiques.

### ***1.2 Escenari***

La proposta del "Fester Manager" naix d'una filà de Moros i Cristians. En la majoria de filaes el control econòmic del montepio es realitza de forma manual, com s'ha fet sempre, mitjançant un "cartonet" on setmanalment, l'encarregat apunta el que cada fester ingressa, això sempre que el fester puga passar-se pel local. A banda del handicap físic que suposa el no poder anar al local a actualitzar o consultar les dades, està el tenir que escriure per a cada fester les mateixes dades i realitzar els mateixos càlculs. Es una tasca engorrosa per a l'encarregat de la filà i incòmoda per al fester que no sempre pot arribar-se a la filà.

### ***1.3 Proposta***

En aquest projecte es proposa el desenvolupament d'una aplicació web on el fester puga accedir mitjançant usuari i contrassenya a la base de dades del montepio des del seu navegador web , i d'aquesta manera consulte el total i el detall del montepio sense la necessitat de desplaçar-se a cap lloc. Tant mateix, que la tasca de l'encarregat de la filà siga reduïda de forma considerable, facilitant-li el control amb uns senzills moviments.

### ***1.4 Objectius***

Un dels principals objectius establerts per a aquest TFG resideix en saber aplicar al mon real els conceptes d'anàlisi, disseny i desenvolupament de software despresos durant el grau. Resulta de gran importància la col·laboració del client per a poder portar endavant el treball.

La principal diferència amb les pràctiques realitzades durant el grau resideix en saber atendre les necessitats d'un client (sense agafar-se únicament a l'enunciat d'una pràctica),

i a partir d'una idea, ser capaç de polir-la i desenvolupar-la fins aconseguir el que el client necessita.

- Elaboració d'un estat de l'art al respecte d'aplicacions semblants per a gestió econòmica.
- Analitzar els requeriments, realitzar el disseny i la planificació que siga capaç de satisfer-los a la vegada que permeta ampliar i mantenir en un futur l'aplicació.
- Adoptar una solució i implementar-la.
- Distribució real de l'aplicació en l'entorn d'una filà.

### **1.5 Estructura del document**

- **Avantprojecte:** Estudi de les eines existents.
- **Requeriments:** Recull el procés de presa de requeriments.
- **Planificació:** Recull la planificació amb les diferents fases del projecte i participants del mateix.
- **Anàlisi i disseny:** Recull el process d'anàlisi dels requeriments i del disseny de l'aplicació.
- **Implementació:** Es detallen les decisions de disseny com la arquitectura i tecnologies empleades. Detalls d'implementació i casuística.
- **Instal·lació i manual d'usuari:** descriu quins passos hi ha que realitzar per a la correcta posada en producció del producte i el manual d'usuari per a formar-lo en l'ús de l'aplicació.
- **Proves:** descriu les proves que s'han realitzat.
- **Futures ampliacions:** Explica quina funcionalitat es podria incorporar en futures versions de l'aplicació.
- **Conclusions:** després de la realització del projecte.

## 2. Avantprojecte

### 2.1 Estat de l'art

#### 2.1.1 Control de versions

##### **GIT**

*GIT* es un software de control de versions dissenyat per Linus Torvalds, pensant en l'eficiència i la confiabilitat del manteniment de versions d'aplicacions quan aquestes tenen un gran nombre de fitxers de codi font.

Destaca la seua gestió distribuïda. *GIT* li dona a cada programador una còpia local de l'historial del desenvolupament sencer, i els canvis es propaguen entre els repositoris locals. Els canvis s'importen com branques addicionals i poden ser fusionats en la mateixa forma que es fa amb la branca local.

Actualment nombrosos desenvolupaments col·laboratius son gestionats amb el sistema *GIT*.

En aquest projecte s'utilitzarà un sistema de control de versions online per a previndre possibles problemes que puguen sortir a les màquines locals així com poder revertir errades que puguen provocar-se durant el desenvolupament de noves funcionalitats.

#### 2.1.2 Entorns de Desenvolupament

##### **Netbeans**

És un entorn de desenvolupament integrat lliure, fet principalment per al llenguatge de programació *Java*. Netbeans es un producte lliure i gratuït sense restriccions d'ús.

El *Netbeans IDE* suporta el desenvolupament de tots els tipus d'aplicació *Java* (*J2SE*, web, *EJB* i aplicacions mòbils).

##### **Eclipse**

Es una plataforma de desenvolupament de codi obert en *Java*. En sí mateix, *Eclipse* es un marc i un conjunt de serveis per a construir un entorn de desenvolupament a partir de components connectats, plugins...

## **Sublime Text**

Es un editor de text i editor de codi font, està escrit en *C++ i Python* per als plugins. Desenvolupat originalment com una extensió de Vim, amb el temps anà creant una identitat pròpia.

Es distribueix de forma gratuïta, però no es software lliure o de codi obert, es pot obtenir una llicència per al seu ús il·limitat, però el no disposar d'aquesta no genera ninguna limitació més enllà d'una alerta cada cert temps. Suporta multitud de llenguatges, entre els que destaquen *C, C++, Java, JavaScript, Python, PHP, XML...*

### **2.1.3 Llenguatges de programació**

#### **Java**

Java es un llenguatge de programació orientat a objectes que va aconseguir la seua maduresa amb la popularització d'internet i que es, en certa manera, l'hereder legítim de *C++*. L'expansió d'aquest llenguatge entre la comunitat de programadors ha sigut vertiginosa i s'ha imposat com el paradigma dels llenguatges de programació orientat a objectes. En l'entorn acadèmic i d'investigació, l'aprenentatge de Java ha reemplaçat a l'aprenentatge de llenguatges de programació estructurada com Pascal i inclús C, que sempre s'han considerat llenguatges d'elecció per a la introducció a la programació.

#### **Python**

Python es un llenguatge de programació interpretat i la seua filosofia fa especial èmfasi en una sintaxis que afavoreixca un codi llegible. Es tracta d'un llenguatge de programació multiparadigma, ja que suporta orientació a objectes, programació imperativa i, en menor mesura, programació funcional. Es un llenguatge interpretat, utilitza tipat dinàmic i es multiplataforma.

Es administrat per la Python Software Foundation. Té una llicència de codi obert, anomenada Python Software Foundation License, que es compatible amb la Llicència pública general de *GNU* a partir de la versió 2.1.1, i incompatible en certes versions anteriors.

Es un llenguatge de programació multiparadigma. Açò significa que més que forçar als programadors a adoptar un estil particular de programació, permet diferents estils:

programació orientada a objectes, imperativa i funcional.

Python utilitza tipat dinàmic i conteig de referències per a la administració de memòria.

- **Propòsit general:** Es poden crear tot tipus de programes. No es un llenguatge creat específicament per a la *web*, encara que entre les seues possibilitats es troba el desenvolupament de pàgines.
- **Multiplataforma:** Hi ha versions disponibles de *Python* en molts sistemes informàtics diferents. Originalment es va crear per a *Unix*, encara que qualsevol sistema es compatible amb el llenguatge sempre i quant existeixca un intèrpret programat per a ell.
- **Interpretat:** No s'ha de compilar el codi abans de la seua execució. En realitat sí que es realitza una compilació, però aquesta es realitza de forma transparent per al programador. En certs casos, quan s'executa per primera vegada un codi, es produeixen uns bytcodes que es guarden en el sistema i que serveixen per a accelerar la compilació implícita que realitza l'interpret cada vegada que s'executa el mateix codi.
- **Interactiu:** *Python* disposa d'un interpret per línia de comandament amb el que es poden introduir sentències. Cada sentència s'executa i produeix un resultat visible, que pot ajudar-nos a entendre millor el llenguatge i provar els resultats de l'execució de troços de codi ràpidament.
- **Orientat a objectes:** La programació orientada a objectes està soportada en *Python* i ofereix en molts casos una forma senzilla de crear programes amb components reutilitzables.
- **Funcions i llibreríes:** Disposa de moltes funcions incorporades en el propi llenguatge, per al tractament de strings, números, fitxers, etc. A més, existeixen moltes llibreríes que podem importar en els programes per a tractar temes específics com la programació de finestres o sistemes en xarxa, o coses tant interessants com crear fitxers comprimits en *.zip*.
- **Sintaxi clara:** Destacar que *Python* té una sintaxi molt visual, gràcies a una notació indentada (amb marges) d'obligat compliment. En molts llenguatges, per a separar troços de codi, s'utilitzen elements com les claus o paraules clau *begin* i *end*. Per a separar els troços de codi en *Python* es deu tabular cap a dins, col·locant un marge

al codi que anirà dins d'una funció o un bucle. Açò ajuda a que tots els programadors adopten les mateixes notacions i que els programes de qualsevol persona tinguen una aparença molt pareguda.

### 2.1.4 Aplicacions Web

#### Java EE

*Java Platform, Enterprise Edition* o *Java EE*, es una plataforma de programació per a desenvolupar i executar software d'aplicacions en el llenguatge de programació Java. Permet utilitzar arquitectures de N capes distribuïdes i es recolza ampliament en components de software modulars executant-se sota un servidor d'aplicacions. La plataforma *Java EE* està definida per una especificació. Similar a altres especificacions del *Java Community Process*, *Java EE* es també considerat informalment com un estàndar degut a que els proveïdors deuen complir certs requeriments de conformitat per a declarar que els seus productes son conformes a *Java EE*; estandaritzat per *The Java Community Process / JCP*.

*Java EE* té varies especificacions de *API*, tals com *JDBC*, *RMI*, *e-mail*, *JMS*, *Servicis Web*, *XML*, etc, i defineix com coordinar-los. *Java EE* també configura algunes especificacions úniques per a *Java EE* per a components. Aquestes inclouen *Enterprise JavaBeans*, *servlets*, *portlets* (seguint la especificació de *Portlets Java*), *JavaServer Pages (JSP)* i varies tecnologies de serveis web. Això permet al desenvolupador crear una Aplicació d'Empresa portable entre plataformes i escalable, a la vegada que integrable amb tecnologies anteriors. Altres beneficis afegits son, per exemple, que el servidor d'aplicacions pugua controlar transaccions, la seguretat, escalabilitat, concurrència i gestió dels components desplecats, significat que els desenvolupadors poden concentrar-se més en la lògica de negoci dels components en lloc de en tasques de manteniment de baix nivell.

En aquest projecte s'estudia l'utilització de *Java EE* com a proposta primera de solució.



## Odoo 8

*Odoo* (conegut anteriorment com *OpenERP*) es un sistema d'*ERP* integrat de codi obert actualment produït per l'empresa belga *Odoo S.A.* El fabricant declara el seu producte com una alternativa de codi obert a *SAP ERP* i *Microsoft Dynamics*.

Els mòduls de *Odoo*, en la seua majoria, estan coberts per la llicència *AGPL* i algunes parts utilitzen una derivada de la llicència *Mozilla Public License*. Com a conseqüència directa, *Odoo* no requereix cap tipus de pagament de llicències per a ser utilitzat, a diferència dels software més utilitzats del mercat. Això també implica que, mentres es respecten els terminis de la llicència, la modificació directa del programa es possible.

Està escrit en *Python*. La seua arquitectura es client-servidor, té components client-servidor separats. El servidor s'executa independentment del client i controla la lògica de negoci i es comunica amb la aplicació de base de dades. El client presenta la informació als usuaris i els permet interactuar amb el servidor.

El mòdul del servidor està escrit en el llenguatge *Python*. El client es comunica amb aquest a través d'interfaces *XML-RPC*. La funcionalitat del negoci s'organitza en mòduls. Els mòduls no son més que carpetes amb una estructura predefinida, amb codi en *Python* i fitxers *XML* en el seu interior. Un mòdul defineix la estructura de les dades, formularis, informes, menús, procediments, fluxos de treball, etc. Els mòduls es defineixen mitjançant una sintaxis independent del client, de tal forma que afegir nous objectes, com menús i formularis els fa disponibles a qualsevol client.

Els clients son lleugers perque no contenen la lògica del negoci. Es dona suport a dos aplicacions oficialment:

- Una aplicació web implementada com un servidor *HTTP* que permet als usuaris connectar-se mitjançant un navegador d'internet.
- Una aplicació d'escriptori escrita en *Python* utilitzant el kit d'eines *GTK+*

Per a l'administració, control i gestió de les dades, s'utilitza el motor de *PostgreSQL* que en l'actualitat, és un dels gestors de bases de dades *Open Source* més potents que existeixen al mercat.

El codi font d'*Odoo* s'allotja en *GitHub*, utilitzant el sistema de control de versions *Git*.

El desenvolupament de mòduls es realitza editant fitxers *Python* i *XML*. No hi ha un editor oficial. En aquest cas s'utilitzarà *Sublime Text*.

En aquest projecte s'estudia l'utilització d'*Odoo* com a proposta de solució.

### 2.1.5 Servidors de Bases de dades

#### MySQL

Es la base de dades de codi obert de major acceptació mundial, ja que permet una creació assequible i fiable d'aplicacions de bases de dades integrades basades en web d'alt rendiment i fàcilment ampliables.

#### PostgreSQL

Es un sistema de gestió de bases de dades objecte-relacional, distribuït baix la llicència *BSD* i amb el seu codi font disponible lliurement. Es el sistema de gestió de bases de dades de codi obert més potent del mercat i en les seues últimes versions no té res que envejar-li a altres bases de dades comercials.

Utilitza un model client/servidor i utilitza multiprocessos en compte de multifils per a garantir l'estabilitat del sistema. Una errada en un dels processos no afectarà a la resta i el sistema continuarà funcionant.

### 2.1.6 Servidors Web

#### Glassfish

Es un servidor d'aplicacions de software lliure desenvolupat per *Sun Microsystems*, companyia adquirida per *Oracle Corporation*, que implementa les tecnologies definides en la plataforma *Java EE* i permet executar aplicacions que segueixen aquesta especificació. Es gratuït, de codi lliure i es distribueix baix un llicenciamnt dual a través de la llicència *CDDL* i la *GNU GPL*. La versió comercial es anomenada *Oracle GlassFish Enterprise Server*.

*Glassfish* té com a base al servidor *Sun Java System Application Server* de *Oracle Corporation*, un derivat d'*Apache Tomcat*.

#### Apache

Es un servidor web *HTTP* de codi obert, per a plataformes *UNIX*, *Microsoft Windows*, *Macintosh* i altres. Es gratuït.

## 2.2 Estudi de propostes

### 2.2.1 Proposta 1

Desenvolupar l'aplicació seguint un paradigma *MVC* (Model-Vista-Controlador) i utilitzant *Java Enterprise Edition (JEE)*. La part de les vistes es pot desenvolupar amb *JSF (JavaServer Faces)* utilitzant llenguatge *XHTML* i *Bootstrap*. La part del model es desacopla de la vista a partir dels *Java Beans*, la integració amb la base de dades es fa amb *JPA (Java Persistence API)* i per a la lògica de l'aplicació s'utilitzen *EJBs (Enterprise JavaBeans)*.

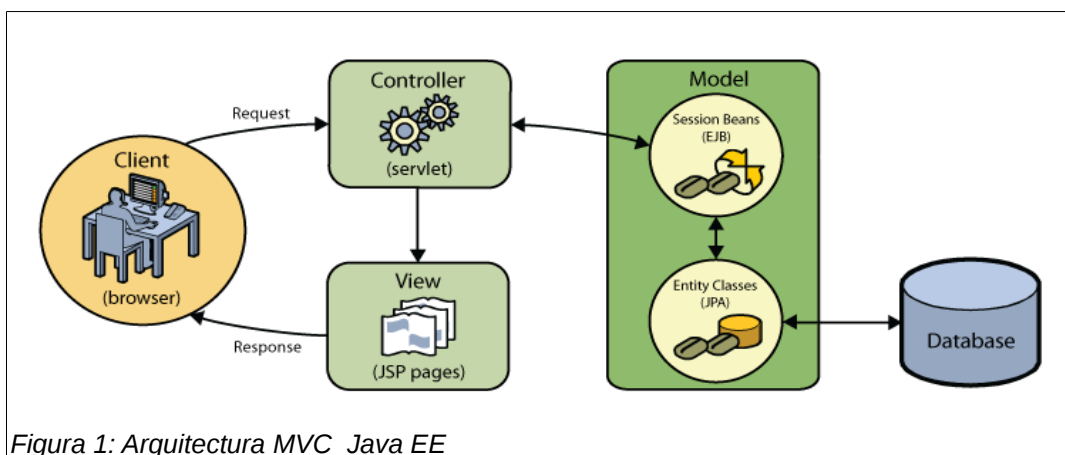


Figura 1: Arquitectura MVC Java EE

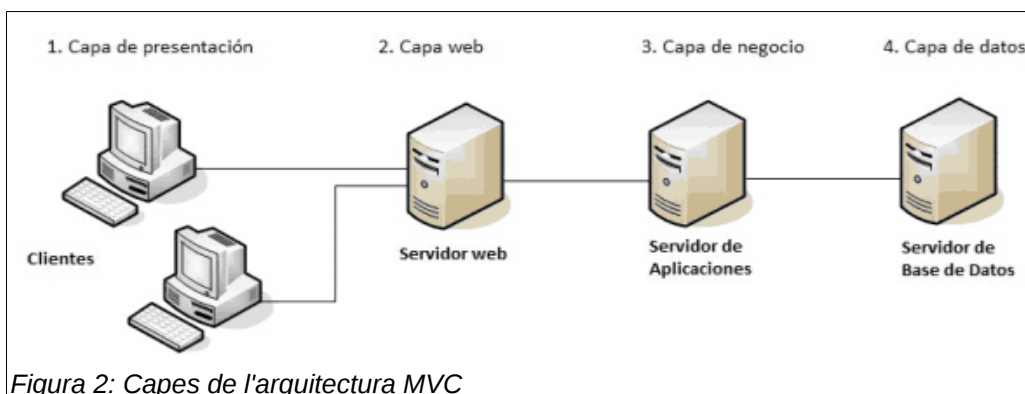


Figura 2: Capes de l'arquitectura MVC

### 2.2.2 Proposta 2

Desenvolupament d'un modul amb *Odoo*, seguint l'arquitectura client-servidor.

Un sistema d'*Odoo* està format per tres components principals:

- El servidor de base de dades *PostgreSQL*, el qual conté totes les bases de dades, cada una conté totes les dades i la majoria d'elements de configuració relacionats amb el sistema *Odoo*.
- L'aplicació del servidor d'*Odoo*, el qual conté tota la lògica empresarial i assegura que *Odoo* còrrega òptimament.
- El servidor web, una aplicació web separada anomenada *Open Object*, el qual permet que et connectes a *Odoo* des d'un navegador web estàndar.

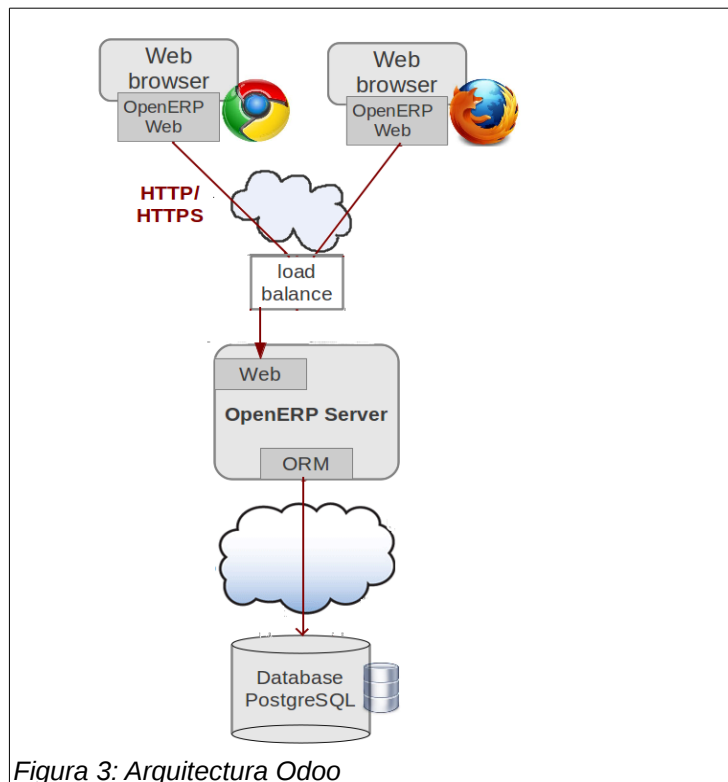


Figura 3: Arquitectura Odoo

## 2.3 Justificació

Les dos propostes son igualment vàlides, però s'ha triat l'opció d'Odoo per diferents motius. Ademés de les següents característiques que detallaré i que reforcen l'elecció d'Odoo, ha sigut una bona oportunitat per a mí de millorar professionalment en el desenvolupament d'Odoo, ja que és l'eina de treball amb la qual porte a cap les pràctiques d'empresa. L'empresa on em trobe realitzant les pràctiques fins ara ha desenvolupat amb OpenERP, la versió anterior a Odoo, i en un futur proper està prevista la migració a Odoo de gran part del que hi ha desenvolupat, per la qual cosa em servirà d'aprenentatge, formació i millora de cara al futur en l'empresa.

- **Llibertat:** *Odoo* com producte no perteneix a cap dels seus distribuïdors, te llibertat per a escollir al proveïdor que més li convinga segons les seues necessitats.
- **Filosofia Open:** Es pot contractar únicament allò que necessite. Lo habitual és terceritzar tots els processos de la implantació, així i tot l'empresa pot ser que ja disponga d'alguns recursos, per tant requeriria el desenvolupament d'algun mòdul específic o formació/suport tècnic de terminat mòdul.
- **Codi Obert:** Al ser software lliure, es disposa del codi per a realitzar qualsevol millora sota moduls ja existents, o crear un de nou adaptat a les necessitats, com es el nostre cas.
- **Connectivitat amb altres productes:** Visualització d'informes en *Adobe PDF*, importació/exportació amb *Microsoft Office* o *OpenOffice*, *Google Maps*, *Mozilla Thunderbird*, *Magento*, *Joomla*, i molts altres, amb la possibilitat de connexió amb quasi qualsevol tecnologia utilitzant *Jripple*.
- **Flexibilitat:** *Odoo* disposa de més de 500 mòduls, molts d'ells específics per a determinats sectors. Es posible començar a utilitzar tant sols uns pocs mòduls i anar integrant més mòduls posteriorment.
- **Gratuit:** *Odoo* es un producte que no te costos de llicències.

A més, amb *Odoo* es preté anar més enllà d'un *ERP*, però mantenint sempre el model *opensource*. Per a això, reuneix moltes solucions des d'una mateixa aplicació, es a dir, és un sistema unificat. Aporta recursos empresarials en forma d'aplicacions per a: Crear i dissenyar la pàgina web de la empresa/organització (també comerç electrònic i blogs). Potenciar el marketing (vía email, chat, events, creació de comunitats, etc...).

### 3. Especificació de requeriments

L'elecció dels treball porta una serie de requeriments funcionals i no funcionals que van a presentar-se a continuació.

#### 3.1 *Requeriments funcionals*

En base a l'escenari plantejat en el projecte, es defineixen els següents requeriments que ajudaran al plantejament del model de dades, maquetes de les vistes i funcionament de l'aplicació final.

- Possibilitat d'alta de nous usuaris amb accés al sistema per usuari/contrassenya.
- Dos nivells de privilegi (o formes d'accedir a l'aplicació):
  - Administrador: que tindrà privilegis d'accés a la llista completa d'usuaris i a tota la informació, a més a més, podrà afegir/borrar dades i crear/eliminar usuaris.
  - Usuari sense privilegis: sols amb permisos per a realitzar consultes únicament sobre la seua informació (accés privat).
- Seguretat en la forma de com viatja la informació personal per la xarxa.
- Possibilitat de còpia de seguretat de les dades des del nivell administrador.
- Internacionalització de l'aplicació web en valencià/castellà/anglés.
- Tres formes d'interactuar amb els pagaments:
  - Manual: l'administrador 'pica' manualment cada pagament que es fa en mà a la filà.
  - Remeses mensuals: hi ha usuaris que han autoritzat que se'ls passe un càrrec mensual de determinat import. Com que açò suposa autorització bancària, etc, el que es pot fer es simular-ho i que:
    - Aparega mensualment (en la data determinada per l'administrador) un pagament per a cada usuari que ha habilitat el mode 'remesa'.
    - Que l'administrador pugua 'llançar' manualment la remesa en el moment que ell vulga (apreta el botó cargar remesa) i apareix el pagament a tots els usuaris que han habilitat el mode 'remesa'.
  - Transferència bancaria: Per a això, l'aplicació hauria de poder llegir la informació des d'un full d'excel (que generarem des del comptar bancari de la filà).

- Llistat de socis.
- Cada soci podrà tindre un usuari de l'aplicació (no obligatòriament).
- Cada soci estarà al carrec dels pagaments dels seus fills.
- Cada soci tindrà únicament un pagament (per exemple any 2015) al any, que reunirà tots els pagaments (deure i haver) que es produeixquen en el mateix any.
- Possibilitat de renovar cada any els assentaments comptables.
- Possibilitat de tancar l'any per a tots els pagaments.
- Possibilitat de carregar a l'any següent, de forma automàtica, el romanent de l'any que s'acaba de tancar, siga a deure o a haver, depenent el cas.
- Possibilitat de carregar desde l'assentament a tots els socis.

### 3.2 *Requeriments no funcionals*

#### Entorn de desenvolupament

El desenvolupament de l'aplicació es portarà a cap utilitzant *Sublime Text 3* com a entorn de desenvolupament amb els plugins necessaris per treballar amb els fitxers *python*, *xml*, *po*, i *csv*.

- **Control de versions:** S'utilitzarà un sistema de control de versions per al codi font de l'aplicació de tipus *GIT*. L'escollit es *GitLab*.
- **Persistència de dades:** Les dades de l'aplicació deuen guardar-se en un sistema, per això es farà ús d'un backend en el que es guardaran els objectes necessaris.
- **Llenguatge de programació:** Per a desenvolupar de forma nativa per a *Odoo* s'utilitzarà el llenguatge de programació *Python*, requeriment imprescindible per a aquest fi.
- **Entorn d'execució:** S'utilitzarà el navegador *Chrome* per accedir a la aplicació web. La aplicació s'executarà en el servidor.
- **Proves:** Es realitzaran les proves que resulten necessàries.
- **Documentació:** Es proporcionarà una memòria final.

## 4. Planificació

### 4.1 Disseny centrat en l'usuari (DCU)

La metodologia que s'utilitzarà per al desenvolupament del projecte és el disseny centrat en l'usuari (DCU).

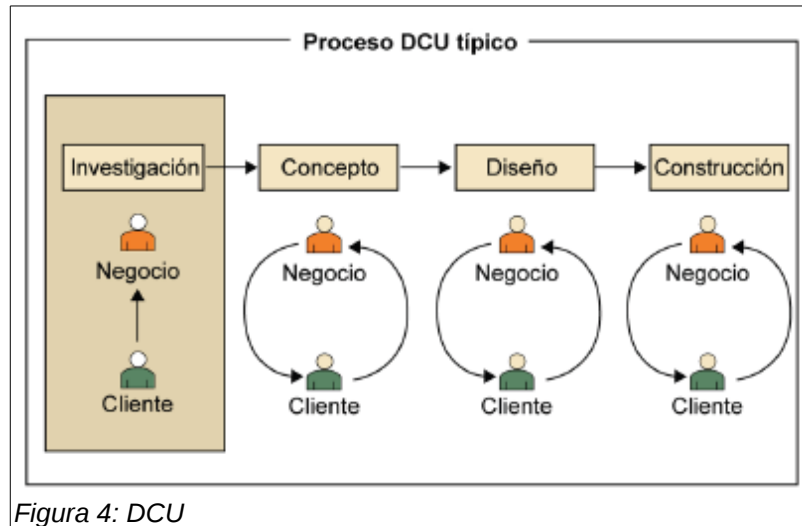


Figura 4: DCU

En particular aquest projecte utilitzarà una variació del tipus DCU àgil. Aquesta metodologia combina el procés típic del disseny centrat en l'usuari i Agile Software Development.

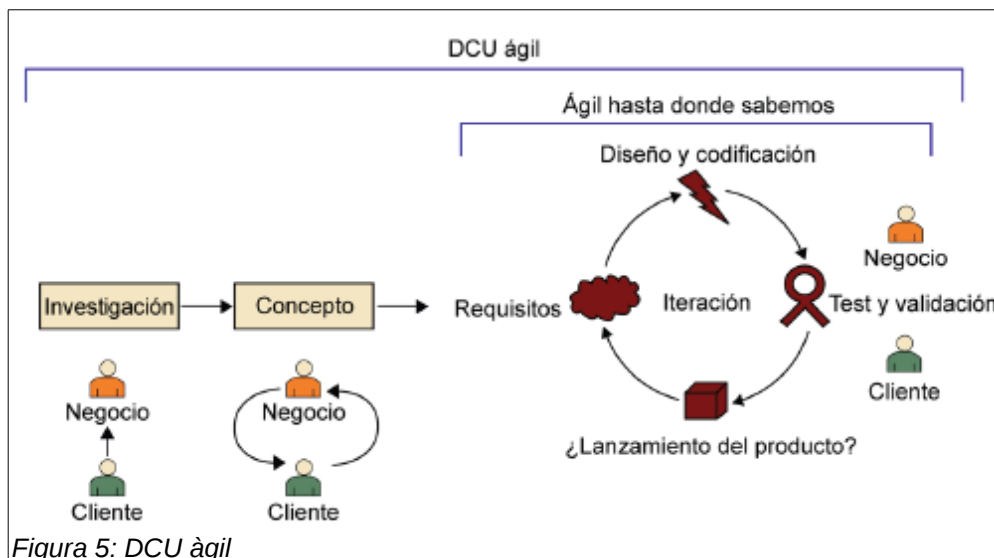


Figura 5: DCU àgil

La proposta és utilitzar el model DCU clàssic a les fases d'investigació, concepte i disseny, deixant la metodologia àgil exclusivament per a la fase de codificació de la aplicació.



Actualment l'experiència de l'usuari i l'utilitat son imprescindibles per a l'èxit de qualsevol tipus de producte o software.

### ¿Què es el disseny centrat en l'usuari?

El disseny centrat en l'usuari es una aproximació al disseny de productes i aplicacions que posa a l'usuari en mig de tot el procés. Així, podem entendre el *DCU* com una filosofia en la que el seu deure principal es que, per a garantir l'èxit d'un producte, hi ha que tindre en compte a l'usuari en totes les fases del disseny. A més, també podem entendre el *DCU* com una metodologia de desenvolupament: una forma de planificar projectes i un conjunt de mètodes que es poden utilitzar en cada una de les principals fases.

## 4.2 Participants

**Director del projecte:** *Pau Micó* serà l'encarregat de la direcció del projecte.

**Dissenyador:** L'alumne serà l'encarregat del disseny del projecte.

**Programador:** L'alumne serà l'encarregat de la programació i implementació del projecte.

**Clients:** Considerem clients als usuaris finals de l'aplicació com podrien ser els testers o consultors del projecte.

## 4.3 Diagrama de Gantt

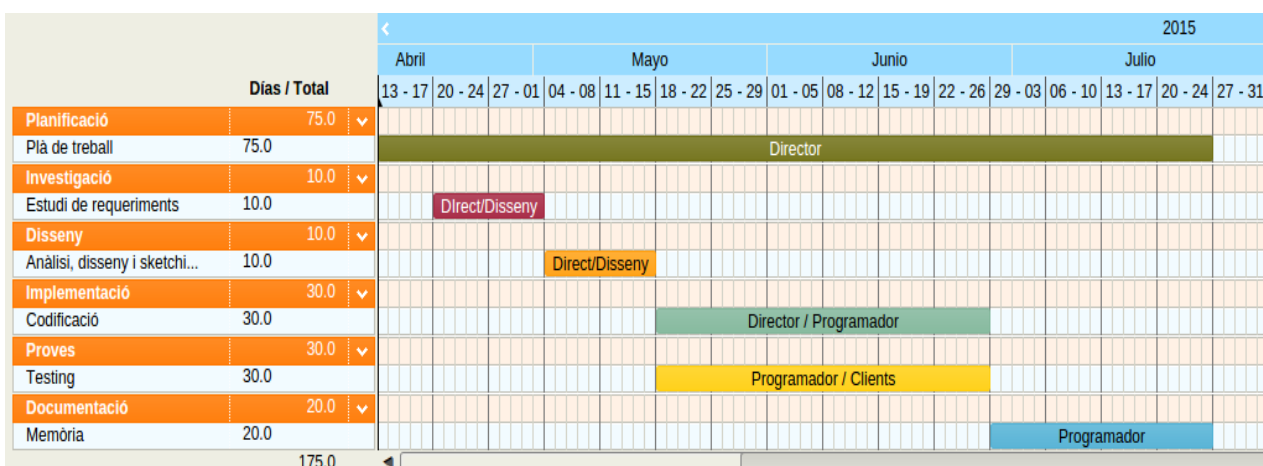


Figura 6: Diagrama de Gantt

A continuació es detallen de forma resumida els objectius de cada fase.

- **Plà de treball:** Definir el projecte i realitzar la planificació inicial. El plà de treball servirà com a guia per a la resta del desenvolupament.
- **Estudi de requeriments:** Definició i estudi dels requeriments dels usuaris i de



l'aplicació.

- **Anàlisi, disseny i sketching:** Realitzar l'anàlisi de tasques, disseny i sketching.
- **Codificació:** Implementar la solució del projecte.
- **Testing:** Fase lligada a la codificació, d'acord amb la metodologia àgil, que ens aportarà nous requeriments i millores sota la nostra implementació.
- **Memòria:** Documentació de la memòria final del projecte.

## 5. Anàlisi i disseny del sistema

### 5.1 Usuaris

A l'aplicació hi ha tres perfils d'usuari que hi ha que considerar:

- **Administrador:** És l'usuari que pot accedir a tots els apartats de l'aplicació, i a més a més, es l'encarregat del manteniment de la mateixa, és qui instal·la i qui parametritza l'aplicació.
- **Mànager:** És l'usuari que serà l'encarregat a la filà de controlar l'aplicació. No confondre amb l'administrador.
- **Usuari:** És l'usuari estàndard de l'aplicació, tant sols podrà accedir a consultar dades.

### 5.2 Disseny conceptual

#### 5.2.1 Escenaris d'ús

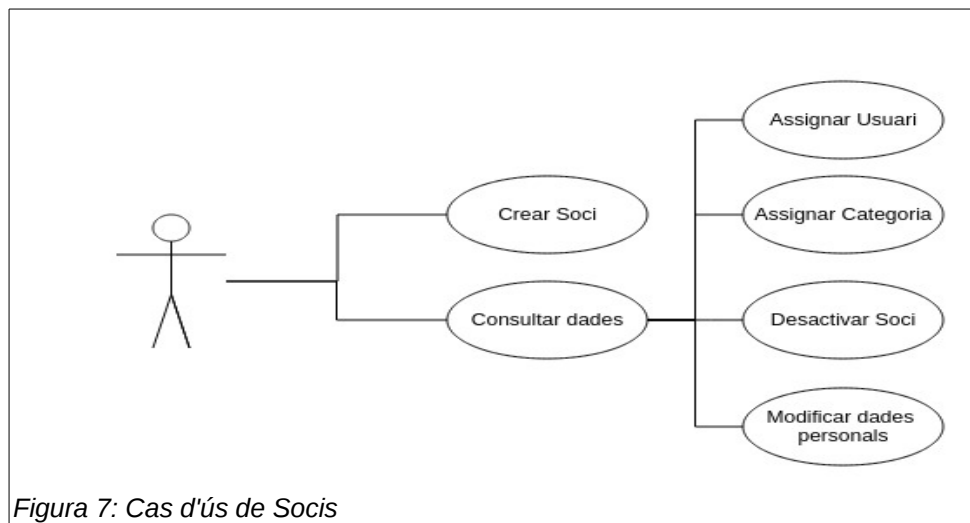
- **Administrador:** Per a aquest usuari no es valora el context, ja que pot accedir a tota l'aplicació.
- **Mànager:**
  - L'usuari accedeix al formulari de login entrant a la direcció de l'aplicació. Accedeix als diferents menus per a crear o modificar socis, infants, anys, assentaments comptables, pagaments, categories, usuaris, etc
  - L'usuari accedeix al formulari de login entrant a la direcció de l'aplicació. Accedeix al pagament per a afegir algun deure o haver en concret.
  - L'usuari accedeix al formulari de login entrant a la direcció de l'aplicació. Accedeix al assentament comptable per a carregar-lo massivament als socis.
  - L'usuari accedeix al formulari de login entrant a la direcció de l'aplicació. Accedeix a l'any per a tancar-lo.
  - L'usuari accedeix al formulari de login entrant a la direcció de l'aplicació. Accedeix únicament per a consulta de dades.
- **Usuari:**
  - L'usuari accedeix al formulari de login entrant a la direcció de l'aplicació. Accedeix únicament per a consulta de dades.

## 5.2.2 Diagrama de casos d'ús

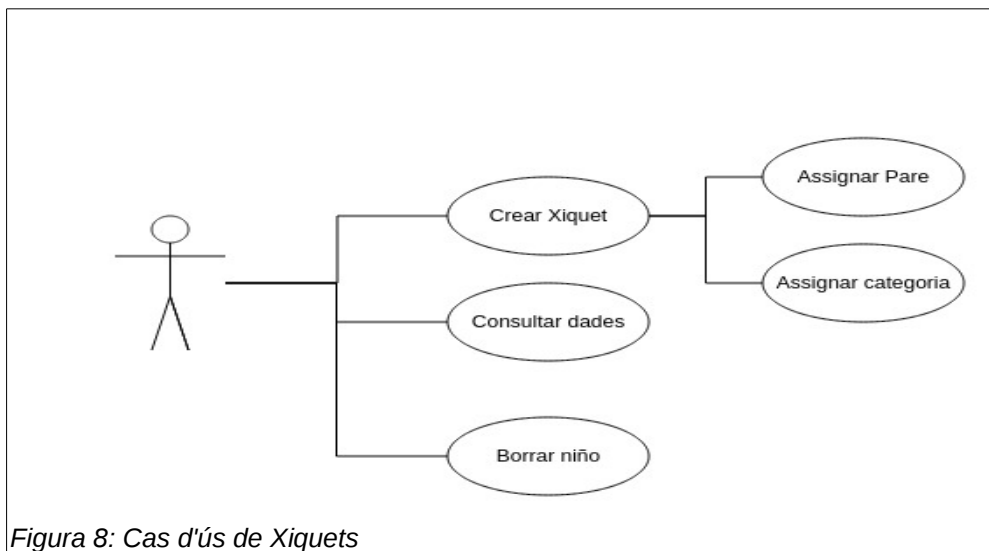
A continuació vegem els diferents diagrames de casos d'ús que es poden donar en cada apartat/model de l'aplicació. Distinguem entre Màner i Usuari.

### 5.2.2.1 Màner

#### Socis



#### Xiquets



## Pagaments

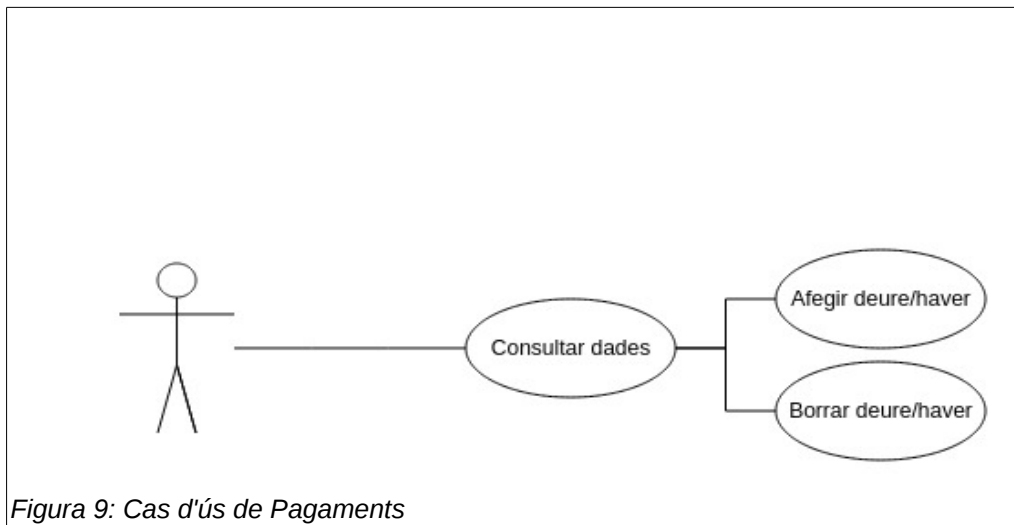


Figura 9: Cas d'ús de Pagaments

## Deure/Haver

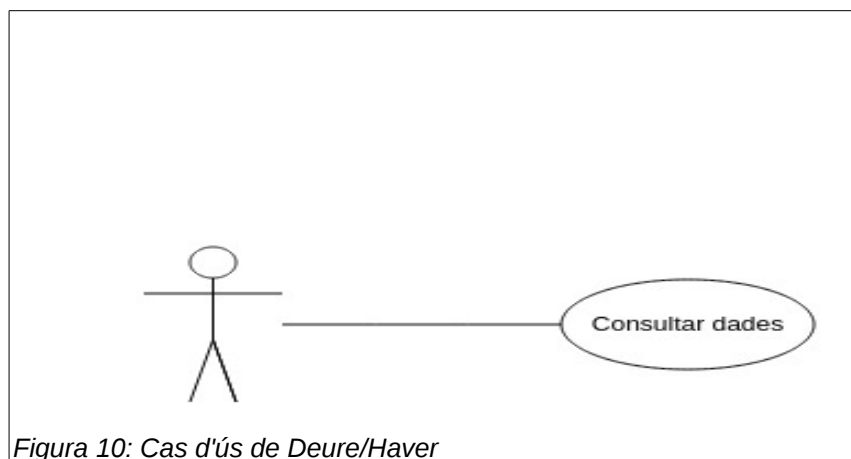


Figura 10: Cas d'ús de Deure/Haver

## Any

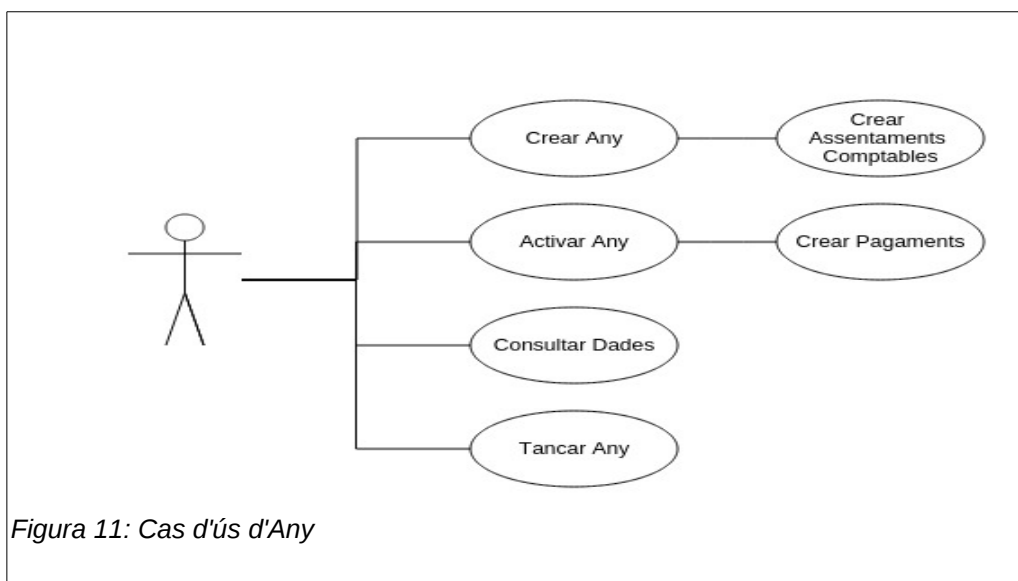


Figura 11: Cas d'ús d'Any

## Assentament Comptable

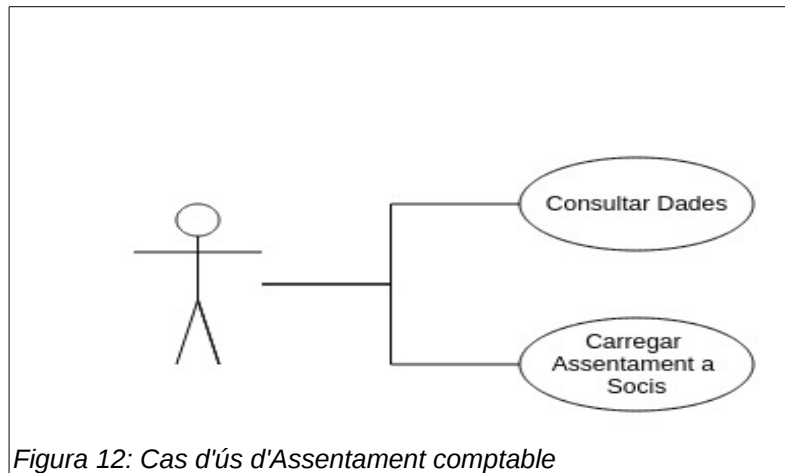


Figura 12: Cas d'ús d'Assentament comptable

## Categoria

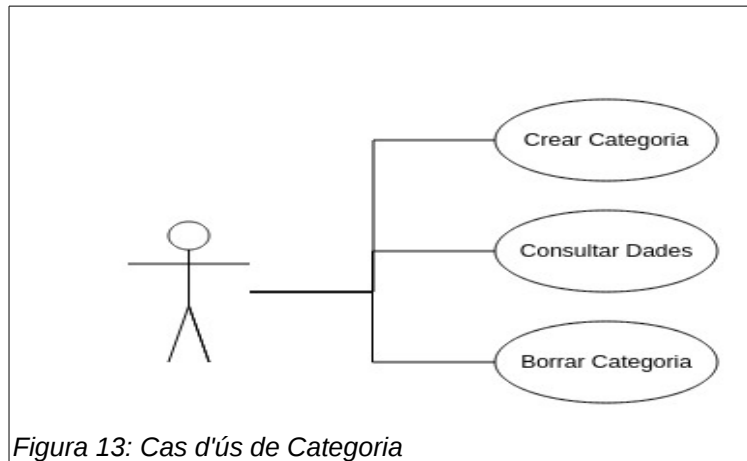


Figura 13: Cas d'ús de Categoria

## Usuaris

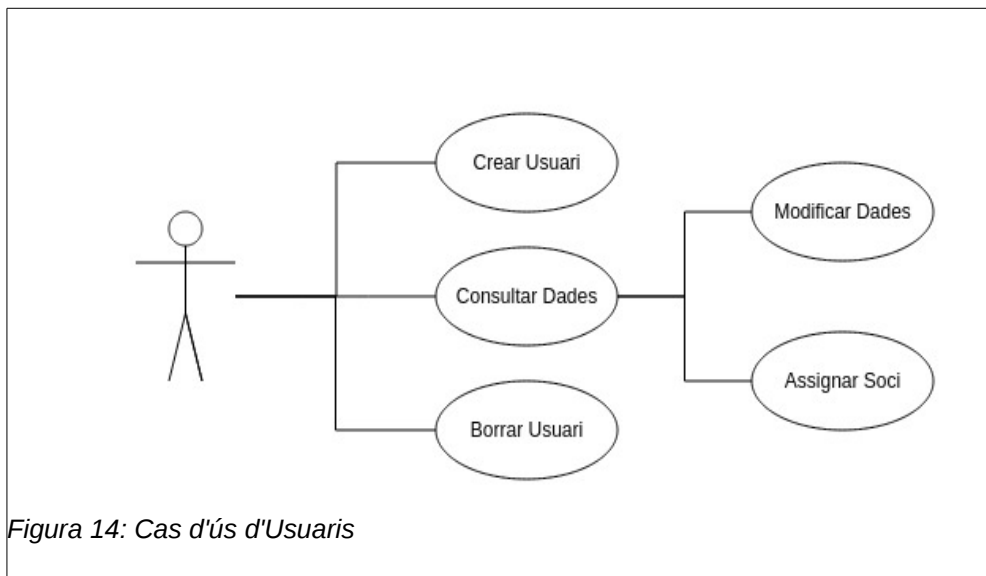


Figura 14: Cas d'ús d'Usuaris

## Grups

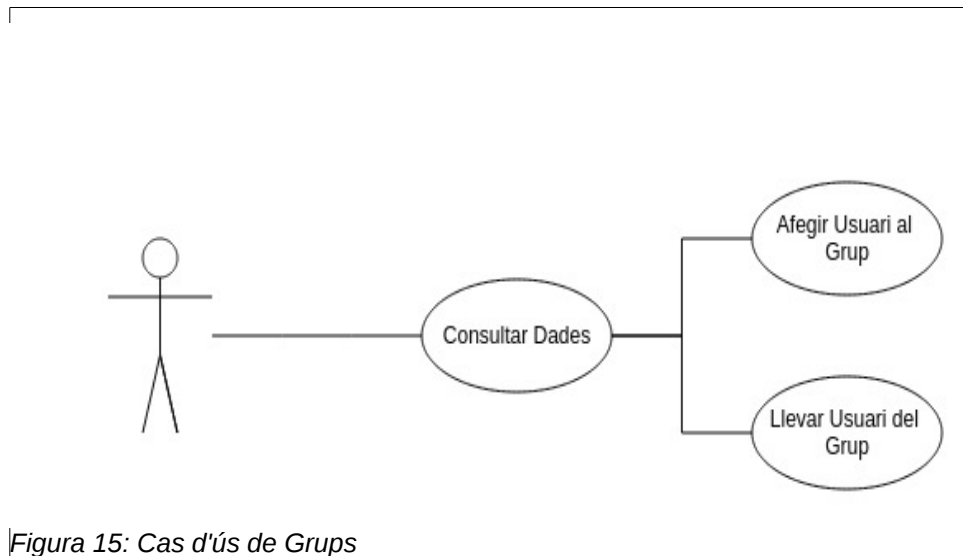


Figura 15: Cas d'ús de Grups

### 5.2.2.2 Usuari

## Socis

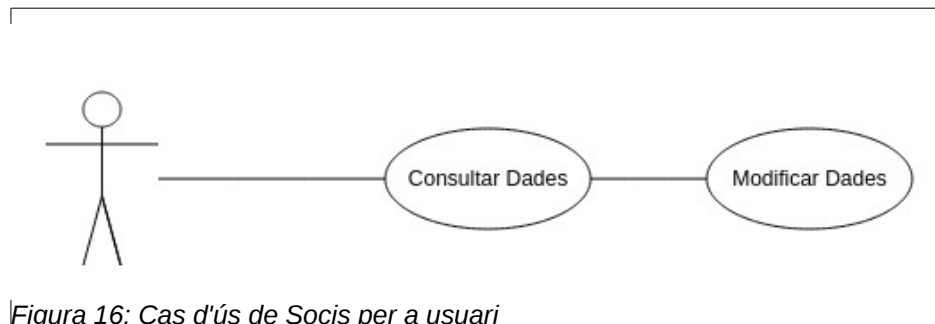


Figura 16: Cas d'ús de Socis per a usuari

## Pagaments

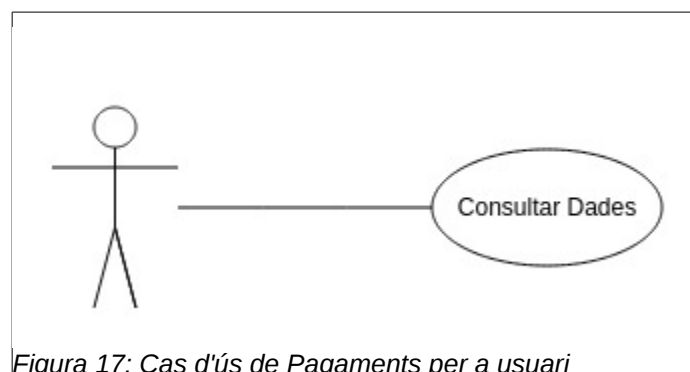
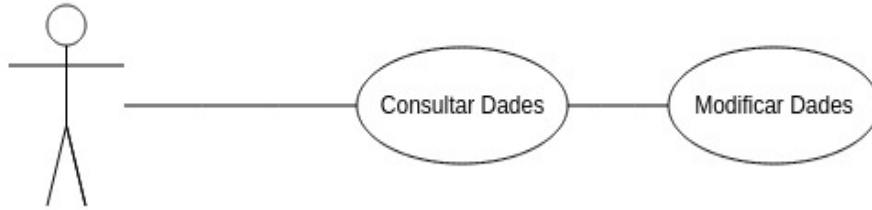


Figura 17: Cas d'ús de Pagaments per a usuari

## Usuaris



*Figura 18: Cas d'ús d'Usuaris per a usuari*



### 5.3 Disseny de model de dades

D'acord amb els requeriments s'ha dissenyat el següent model de dades:

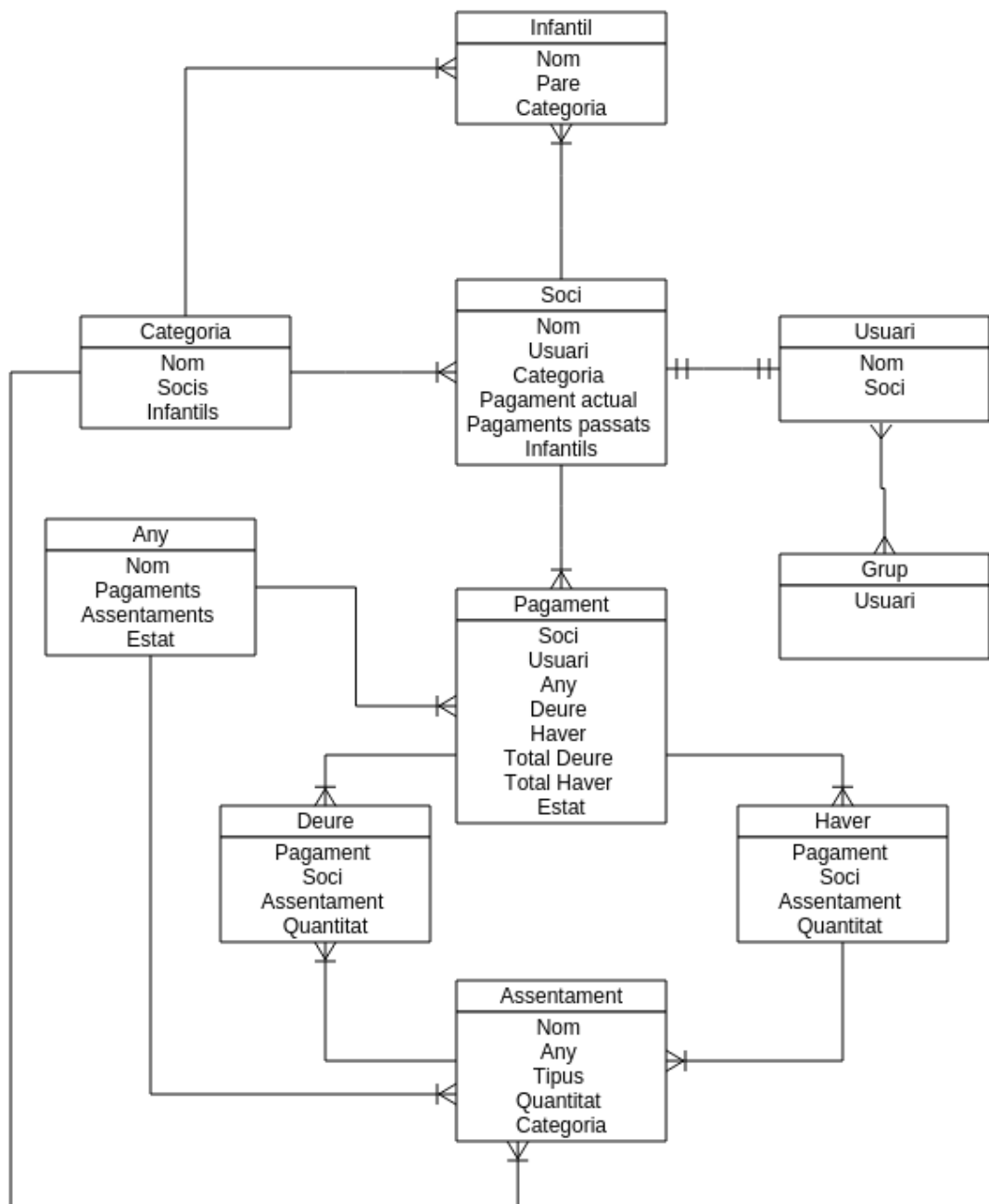


Figura 19: Model de base de dades

## 5.4 Disseny de les vistes

### 5.4.1 Sketches

Odoo fa ús d'una vista estàndard que segueix una estructura pròpia. El disseny constarà de vistes *tree* i vistes *formulari*. Per a la qual cosa es centrarem en la vista formulari que si que serà del nostre interès saber com estructurarem la informació i els diferents botons que la componen. Les vistes *tree* es limiten a mostrar diferents camps del model, per això no es necessari dissenyar l'*Sketch*.

A continuació vegem el disseny del que serà l'estructura principal de la vista de l'aplicació:

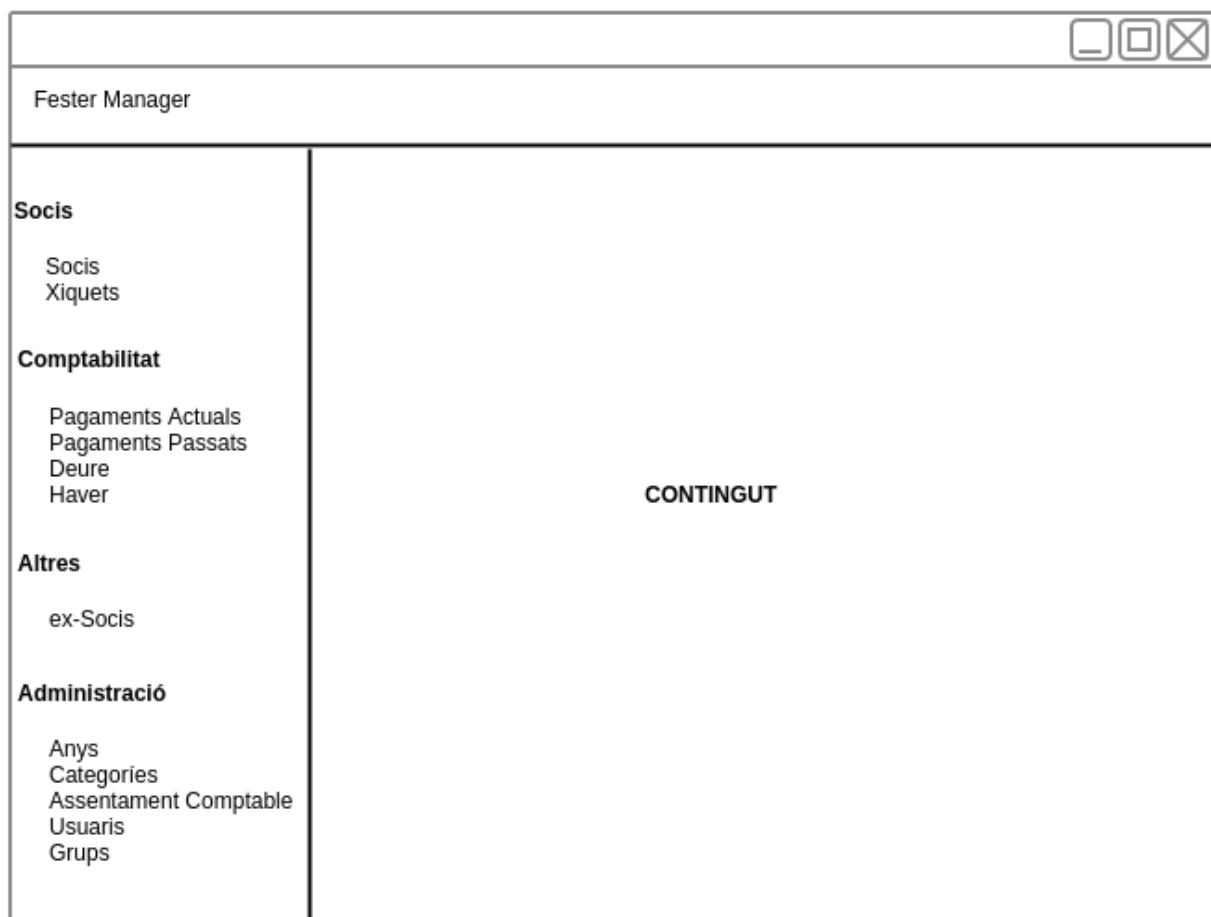



Figura 20: Sketch estructura principal

El contingut, com s'ha nomenat ja, estarà format per dos tipus de vista, *tree* i *formulari*. A continuació i per a cada model, es mostrarà el disseny de la vista formulari.

## Formulari Socis

Edita Crea



**Nom**

Adreça

Lloc Web

Actiu

Categoria

Usuari

Any

Romanent Any Anterior

Telèfon

Mòbil

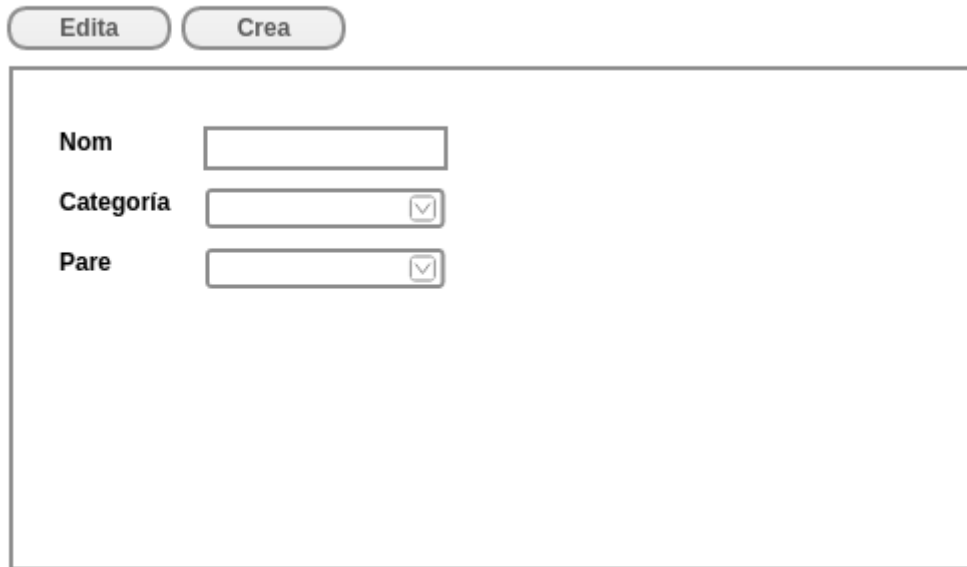
Correu Electrònic

Recargar Pagaments

Notes Internes	Pagament Actual	Pagaments Passats	Xiquets

Figura 21: Sketch Socis

## Formulari Xiquets



Edita Crea

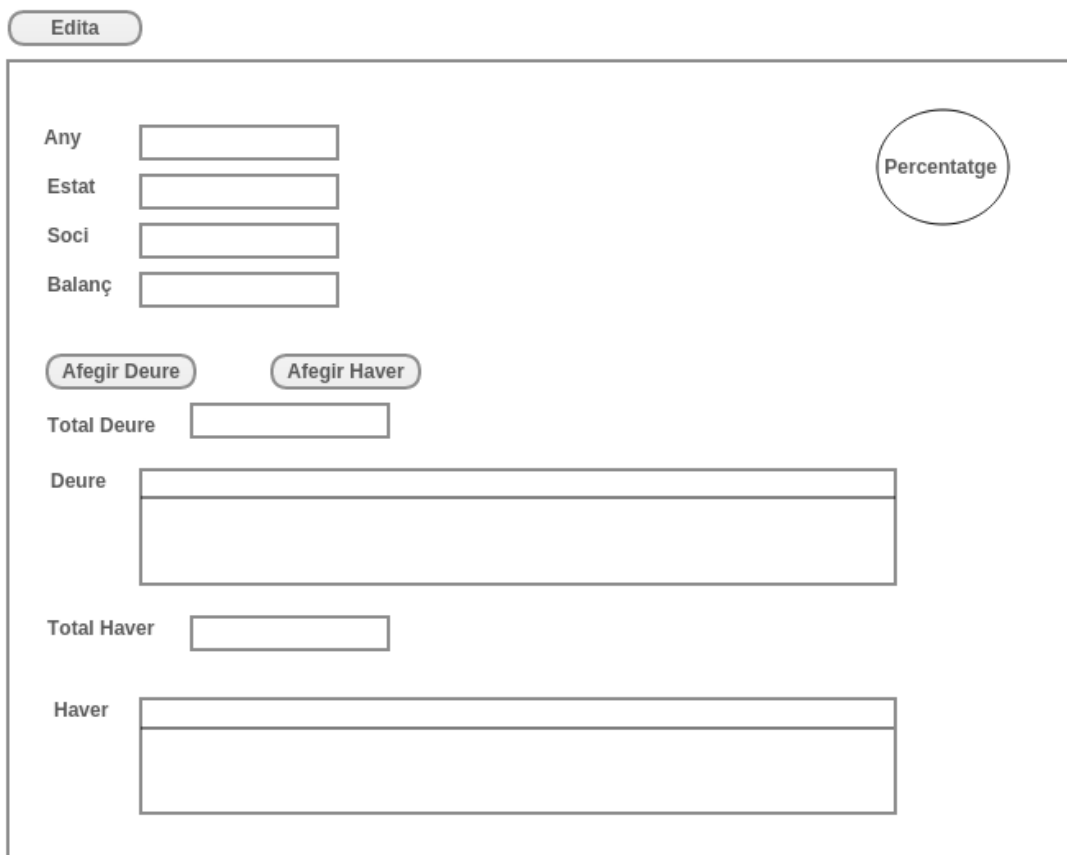
Nom

Categoria

Pare

Figura 22: Sketch Xiquets

## Formulari Pagament



Edita

Any

Estat

Soci

Balanç

Percentatge

Afegir Deure Afegir Haver

Total Deure

Deure 

--	--

Total Haver

Haver 

--	--

Figura 23: Sketch Pagament

## Formulari Deure I Haver

Any	<input type="text"/>
Soci	<input type="text"/>
Ref	<input type="text"/>
Concepte	<input type="text"/>
Quantitat	<input type="text"/>
Data	<input type="text"/>

Figura 24: Sketch Deure i Haver

## Formulari Any

Edita    Crea

Any	<input type="text"/>
Estat	<input type="text"/>

Activar/Tancar Any

Crear Pagament a Socs      Crear Assentament

Assentaments


Figura 25: Sketch Any  
Fester Manager – Rubén Cerdà Roig

## Formulari Categoria

Categoria

Major

Socis


Figura 26: Sketch Categoria

## Formulari Assentament Comptable

Ref	<input type="text"/>
Concepte	<input type="text"/>
Tipus	<input type="text"/>
Categoria	<input type="text"/>
Any	<input type="text"/>
Quantitat	<input type="text"/>

Figura 27: Sketch Assentament Comptable

## Formulari Usuari

The sketch shows a user form interface. At the top, there are two buttons: 'Edita' and 'Crea'. Below these buttons is a large rectangular frame containing the form fields. On the left side of the frame, there is a square icon with an 'X' inside. To the right of this icon, the label 'Nom' is positioned above the label 'Adreça Electrònica'. Below the 'Adreça Electrònica' label, there are four form elements: a checkbox labeled 'És soci', a dropdown menu labeled 'Soci Relacionat', and two text input fields labeled 'Telèfon' and 'Mòbil'.

Figura 28: Sketch Usuari

## Formulari Grup

The sketch shows a group form interface. At the top, there is a button labeled 'Edita'. Below the button is a large rectangular frame containing the form fields. At the top of the frame, there are two text input fields: one labeled 'Aplicació' and one labeled 'Nom'. Below these fields, the label 'Usuaris' is positioned above a large, empty rectangular area, which likely represents a list or table of users.

Figura 29: Sketch Grup

## 6. Implementació

Amb *Odoo* les opcions d'instal·lació poden ser:

- *Odoo SaaS (Software as a Service) o Cloud computing*, en el Núvol (Modalitat *SaaS* compartit o modalitat *SaaS* privat).
- instal·lació de l'aplicació en prestador de serveis de hosting remot (també al Núvol).
- instal·lació en servidor propi de l'empresa.

### Odoo SaaS

*SaaS* es un paquet complet allotjat en un proveïdor, que inclou els següents serveis: hardware de servidor, hosting de la solució genèrica, instal·lació i configuració inicial, redundància de l'arquitectura, backups, manteniment del sistema i suport. També conegut com *On-Demand*. Es presenta en forma d'una subscripció mensual amb un preu fixe per usuari.

Els paquets *SaaS* no permeten desenvolupar mòduls específics a les nostres necessitats. Per el contrari ofereixen un servei a un preu fixe basat en mòduls software estàndar que contenen pocs riscos de migració. Proveïdors *SaaS* es limiten generalment als mòduls certificats i validats per l'autor original i director del projecte.

Aquestes son els principals avantatges de una solució *SaaS d'Odoo*:

- Rendiment inmillorable de l'inversió (cost d'implementació: 0, cost de llicències: 0)
- Costos que son controlats i sense sorpreses (l'oferta inclueix el manteniment, migracions freqüents i suport)
- Una solució clau en mà, instal·lat en menys de 24 hores
- Paquets adaptats i preconfigurats per a diferents sectors de l'activitat
- Una arquitectura molt sòlida garantida per a tindre un accés constant i permanent, accessible des de qualsevol lloc

Aquesta opció es recomanable per a organitzacions xicotetes que no necessiten del desenvolupament de moduls adaptats a les seues necessitats.



## Hosting per proveïdor

A primera vista, un sistema allotjat d'*Odoo* pareix similar a *SaaS*: proporciona *Odoo* des d'una instal·lació remota a través d'un navegador web. Però, en general, les similituds s'acaven ací.

Per a comparar-lo amb un paquet de *SaaS*, hi ha que comprovar si l'oferta d'allotjament inclueix correctament els següents elements:

- Hardware de servidor
- Hosting
- Manteniment
- Migracions futures
- Còpies de seguretat
- Redundància de servidor
- Suport telefònic i per correu
- Actualitzacions freqüents als mòduls

També ademés:

- La versió d'*Odoo* proposada
- Els costos d'implementació (configuració, càrrega de dades, formació)
- Els cost de configuració (si és proposat)
- La tecnologia i el procediment utilitzat per a la seguretat de la base de dades
- La tecnologia i el procediment per a la prevenció i errades del sistema
- La tecnologia i el procediment per a restaurar un sistema defectuós
- Limitacions en el nombre d'usuaris, el nombre d'usuaris simultanis, i el tamany de la base de dades
- El nivell de suport i el seu cost
- El procediment utilitzat per a actualitzar *Odoo*
- El procediment adoptat per a actualitzacions d'*Odoo*

Cridar a aquestos proveïdors pot ser una bona solució si estàs disposat a confiar totes les especificacions tècniques per al funcionament d'*Odoo* a ells, inclús si tens que utilitzar mòduls personalitzats o extensions que no estan a la versió lliberada estable.

## Instal·lació interna

Les grans i mitjanes empreses solen instal·lar *Odoo* utilitzant els seus propis recursos interns de l'empresa. Generalment, prefereixen tindre el seu propi servei informàtic a càrrec del manteniment.

Aquestes empreses poden fer funcionar l'aplicació internament, o recórrer a un soci *Odoo* que farà el treball d'implementació de *ERP* o ajudar-los amb ell. En general, les empreses prefereixen adoptar una solució intermitja que consisteix en:

- Arrancada de la implementació inicial a un soci per a limitar riscos i retards en la integració. Que els permet ser controlats per experts i obtenir una configuració d'alta qualitat.
- Fer-se càrrec de les necessitats simples per sí mateix una vegada que el software ha sigut implementat. És molt més convenient per a que siguen capaços de modificar les tables de bases de dades, formularis, plantilles i fluxos de treball interns que depenen de forma rutinaria en un proveïdor.

Una instal·lació interna probablement resultarà més costosa que un paquet *SaaS* o servei allotjat. Inclús si vosté es posa a càrrec de tot, es necessitarà un poc de temps per a aprendre a gestionar la aplicació. Açò representa un risc significatiu.

Llavors, una implementació interna pot ser particularment interessant quan:

- Volem mantenir les dades en la nostra companyia.
- Pensem que voldrem modificar el nostre software.
- Volem un paquet específic de mòduls.

Donada la naturalesa del cas, tenim que descartar l'opció de l'oferta *SaaS* ja que és una solució genèrica i no ens ofereix la possibilitat de desenvolupar nous mòduls.

En aquest cas s'instalarà a un proveïdor de hosting ([www.bitnami.com](http://www.bitnami.com)), la segona opció, que ens proporcionarà una instal·lació de servidor *Odoo*, que ens ofereix la possibilitat de desenvolupament del nostre mòdul. També barallem la possibilitat d'instal·lació interna, per això s'indicarà pas per pas com s'instal·la el servidor i el mòdul a l'apartat d'instal·lació. Les dues instal·lacions seran sota *Linux*.

## Hosting per proveïdor Bitnami

En aquest cas les imatges seran d'una demo gratuïta d'una hora de duració que proporciona *Bitnami*:

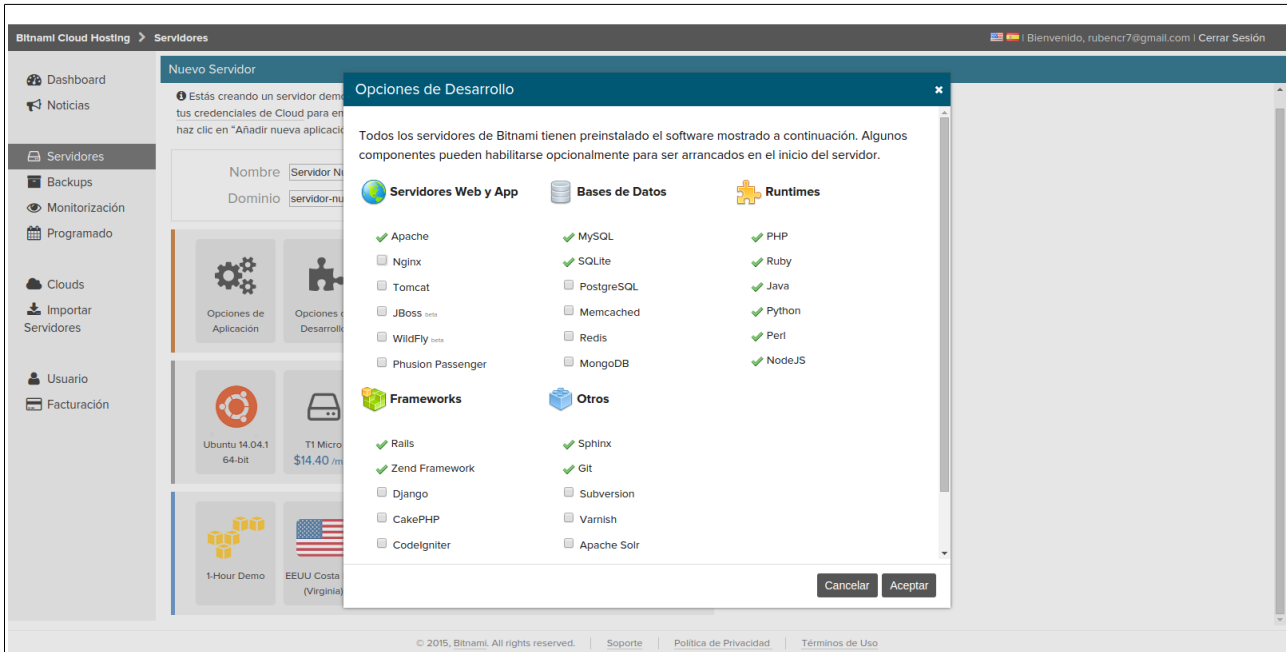


Figura 30: Implementació al núvol, Binami

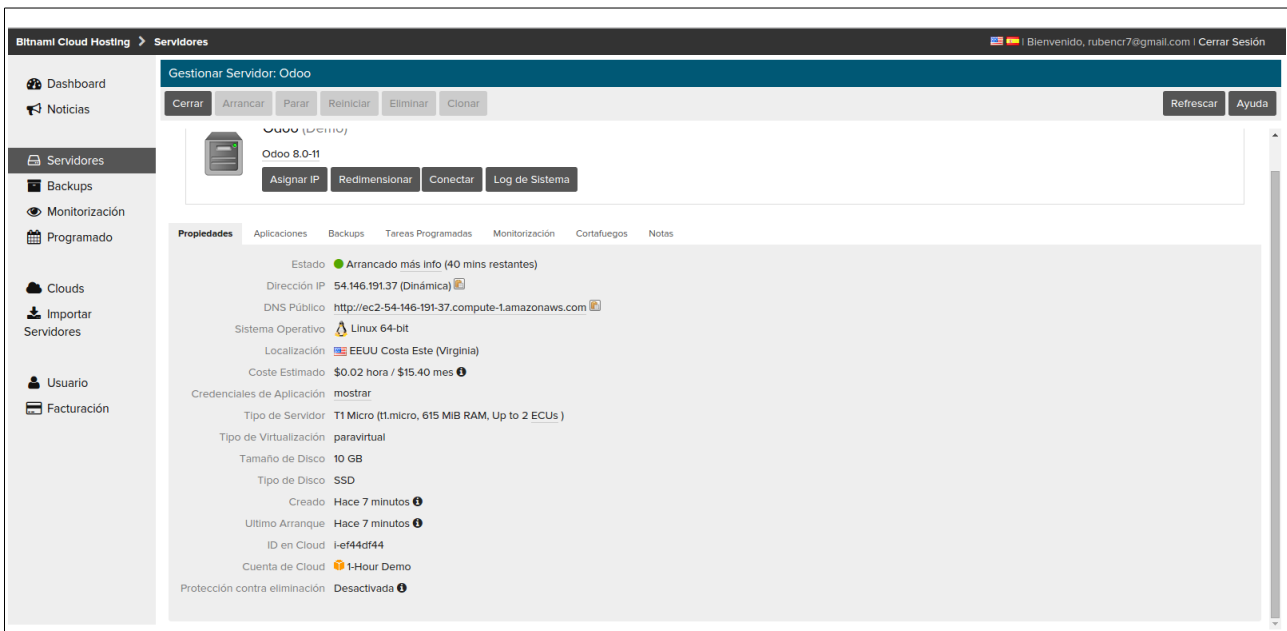


Figura 31: Implementació al núvol, Binami

Ens proporciona un túnel SSH per a poder afegir els nostres mòduls.

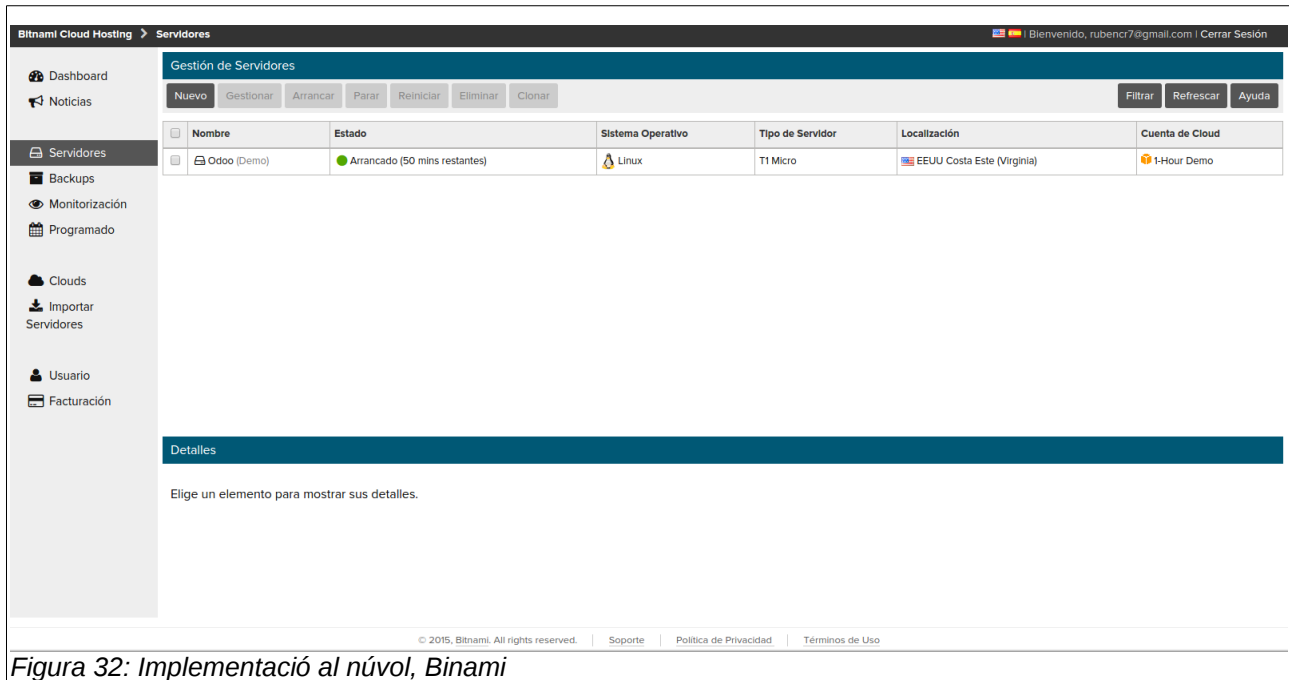


Figura 32: Implementació al núvol, Binami

## 6.1 Arquitectura de l'aplicació

Odooo controla una arquitectura de Client-Servidor permetent que els components de client i del servidor romanguen separats, per tant el servidor s'executa amb independència del client i es l'encarregat de la lògica de negoci i la comunicació amb la base de dades, mentre que el client interactúa amb l'usuari proporcionant-li informació i permetent-li que aquestos interactúen o operen amb el servidor.

### Servidor

La part del servidor està escrita amb el llenguatge de programació *Python*, el qual proporciona un marc complet de desenvolupament, donant un gran marge de característiques per a crear aplicacions. Les característiques més destacables son: Flexibilitat *ORM (Object Relational Mapping o Mapeig d'Objectes relacionals)*, una arquitectura *MVC (Model Vista Controlador)*, vistes i models de dades extensibles, diferents motors de reports, tot unit i coherent, marc d'accessibilitat de la xarxa.

### Mòduls

Les aplicacions de negoci estan organitzades dins de mòduls. Un mòdul es una carpeta

amb una estructura predefinida la qual conté codi python i fitxers xml, aquest defineix l'estructura de dades, formularis, reports, menús, processos, fluxos de treball, entre d'altres. A més, son definits de forma independent de la sintaxi del client, aleshores si afegim un menú o formularis, aquestos estaran disponibles per a qualsevol client.

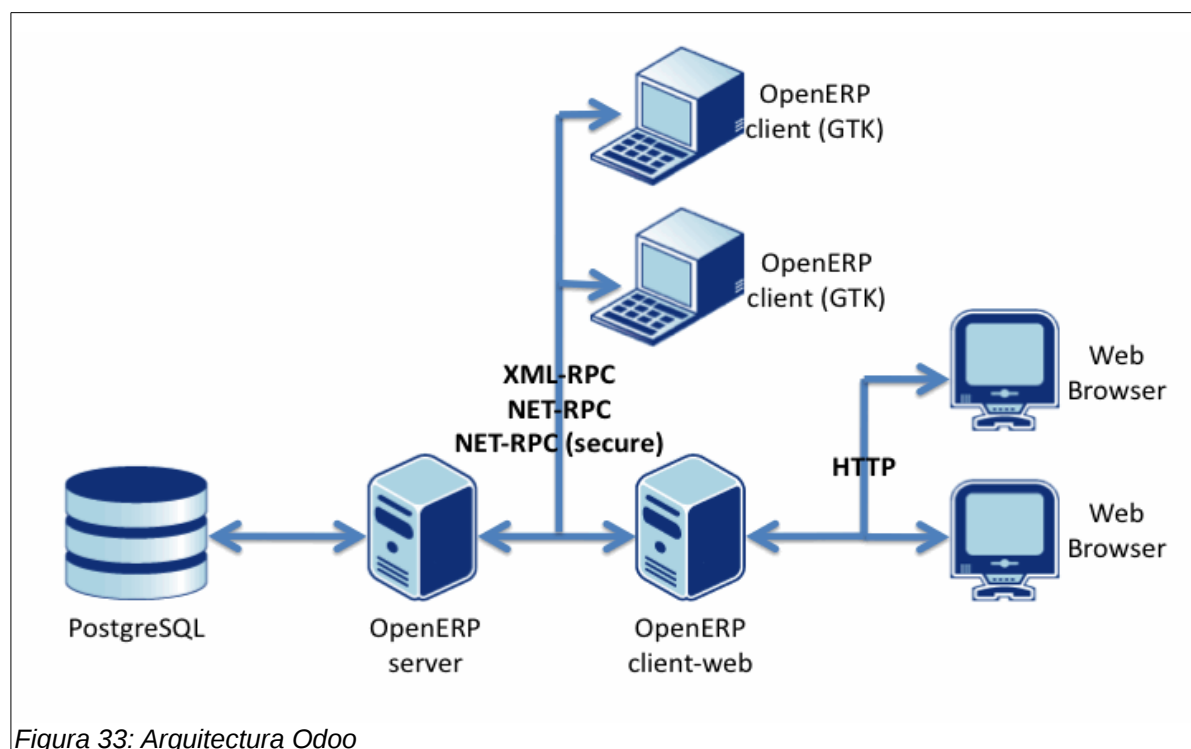
Els mòduls fan ús d'*ORM* d'*Odoo* per a la persistència de les seues dades en una base de dades relacional (*PostgreSQL*). Els mòduls poden insertar dades en la base de dades durant la instal·lació, que son proporcionats mitjançant fitxers *XML*, *CSV* o *YML*.

## Client

Les aplicacions client son lleugeres ja que no tenen la lògica de negoci. El client es comunica amb el servidor utilitzant les interfícies *XML-RPC* i/o *NET-RPC* depenent de la configuració del servidor. Odoo ha desenvolupat dos clients diferents:

- Aplicació WEB: Implementant com un servidor *HTTP* per a permetre als usuaris connectarse a través d'un navegador web.
- Aplicació d'Escriptori: Desenvolupat en *Python* amb el kit d'eines gràfiques ampliament utilitzat *GTK+*

En la següent figura es mostra l'arquitectura d'*Odoo*:



## Marc de Desenvolupament (Framework)

El marc de treball o desenvolupament d'*Odoo* es *OpenObject* i els seus mecanismes son: Servidor de Base de Dades relacional i *ORM* integrat en *OSV*, Model vista-controlador *MVC*, Generador de Reports, Fitxers de traduccions \*.po.

A continuació descrivim breument els principals mecanismes:

### Servidor de Base de dades relacional i *ORM* integrat en *OSV*

El nivell de base de dades d'*Odoo* es proporcionat per una base de dades relacional (*PostgreSQL*). Mentre s'executen consultes *SQL* directes des dels mòduls *Odoo*, l'accés de base de dades relacional es realitza mitjançant el mapeig d'objectes relacionals (*ORM*), el qual es troba integrat en *OSV (Object Service)*.

El model de dades es descrit en codi python i *Odoo* crea les taules. Tots els beneficis de *RDBMS* (Restriccions úniques, integritat relacional i consultes eficients) s'utilitzen quan es possible per la flexibilitat de *Python*.

### Model vista-controlador *MVC*

*MVC* es un patró estructural utilitzat en l'enginyeria de software, on es tenen dades separades (model), una interfície d'usuari i un component intermig que interactúa amb els que es el controlador.

Per tant en *Odoo* tenim els tres components presents:

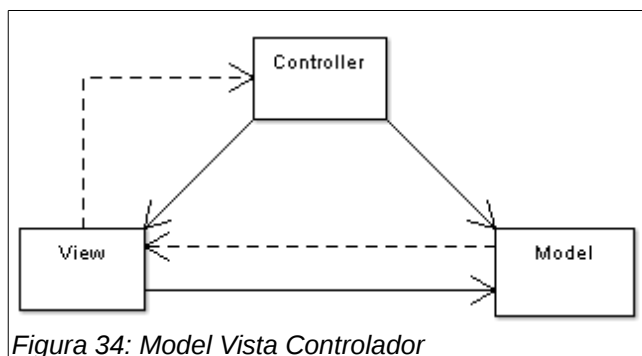


Figura 34: Model Vista Controlador

- **Model:** Les taules relacionals de *PostgreSQL*
- **Vistes:** Vistes definides en fitxers *XML*
- **Controlador:** Els objectes d'*Odoo*

## Objectes de negoci

- Declarats com a classes *Python*, aquests recursos son conservats automàticament per *Odoo* basat en la seua configuració. En aquestes classes es declaren els models de dades i la lògica de negoci de l'aplicació.

## Fitxers de dades

- Fitxers *XML* o *CSV* que declaren metadades, dades de configuració (per exemple seguretat) o vistes.

## Controladors Web

- Controlen les peticions dels navegadors web. Ho fa *Odoo* nativament.

## Dades web estàtiques

- Images, *CSS* o fitxers *Javascript* utilitzats per a l'interfície web o pàgina web.

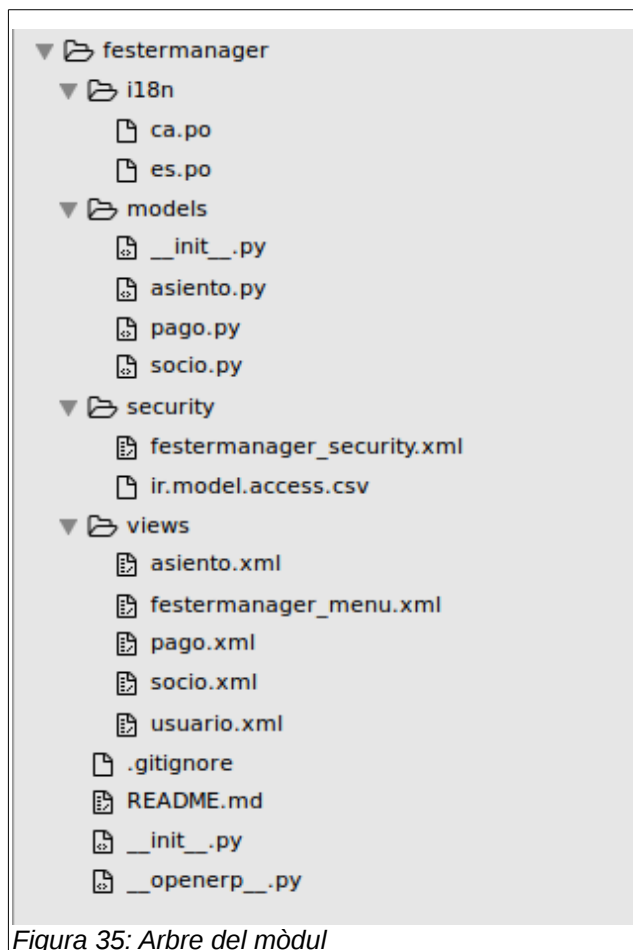


Figura 35: Arbre del mòdul

### 6.1.1 Lògica de l'aplicació

La lògica de l'aplicació es troba a la carpeta *models*, implementada als fitxers *Python*.

#### **res\_partner**

Aquesta classe es la que implementa els socis.

Casuística que s'ha hagut de controlar:

- Una funció per a recarregar a l'usuari els pagaments, si aquest acaba de ser enregistrat, de tal forma que quan faça login pugua vore els seus pagaments. De no ser així la unió de usuari-soci no tindria efecte en els pagaments ja creats.

#### **festermanager\_infantil**

Aquesta classe es la que implementa als socis infantils.

#### **res\_users**

Aquesta classe es la que implementa el model d'usuaris.

#### **res\_groups**

Aquesta classe implementa els grups d'usuaris.

#### **festermanager\_year**

Aquesta classe implementa el model d'anys. L'any podrà ser actiu i passat, i cada any tindrà nous assentaments comptables i pagaments.

Casuística que s'ha hagut de controlar:

- Cada any serà únic.
- Activar l'any.
- Tancar l'any.
- Poder afegir assentaments comptables a l'any actual.
- Crear pagaments de l'any actual als socis, la qual cosa implica:
  - No crear pagament als socis que ja hi el tenen creat, d'aquesta forma sempre que es polse el botó de crear pagaments no hi haurà marge al error de



l'administrador de la filà de duplicar pagaments.

- No crear pagament a socis inactius de la filà.
- Controlar els socis que tenen balanç positiu o negatiu de l'any anterior en quant a pagaments, per a carregar-li amb l'assentament que toque (deure o haver) a l'any que s'acaba de crear.

### **festermanager\_pago**

Classe que implementa el model de pagaments. En aquesta classe es veurà reflexat el deure i l'haver de cada soci, per tant el que li queda per pagar en tot moment.

Casuística que s'ha hagut de controlar:

- Poder afegir al deure i al haver del pagament del soci, directament des del pagament del soci mitjançant un parell de botons.

### **festermanager\_asiento**

Classe que implementa el model d'assentament comptable.

Casuística que s'ha hagut de controlar:

- Si l'assentament pertany a un any actiu o no, per a mostrar/ocultar botons en cada cas.
- Poder carregar la fulla si l'assentament té categoria (és de tipus fulla) o bé és un assentament normal.
- Poder carregar fulla al pagament d'eixe any de tots els socis que pertanyen a la categoria de l'assentament des de l'assentament comptable amb un botó, en cada cas:
  - Fulla de tipus Senior: Per a cada pagament de soci de la categoria de la fulla en qüestió, es controla si ja li ha sigut carregada la fulla, si no ha sigut carregada se li carrega.
  - Fulla de tipus Infantil: Per a cada pagament de soci se li controla el nombre de fulles infantils de la categoria de la fulla que té carregades i els infantils d'eixa categoria que té al càrrec eixe soci. Es compara i es carrega en cas de faltar-ne.
- Poder carregar assentament comptable normal al pagament de tots els socis de

l'any actual des del mateix assentament comptable amb un botó, a deure o haver segons el tipus d'assentament.

- Cada any els assentaments comptables es renoven i amb això el seu número de referència.

### **festermanager\_debe**

Classe que implementa el model del deure. Aquest model recull l'assentament comptable del seu tipus, amb ell el nom, quantitat econòmica, referència, soci, any,...

### **festermanager\_haber**

Classe que implementa el model de l'haver. Aquest model recull l'assentament comptable del seu tipus. Es diferencia del deure en què la quantitat de diners no ve automàticament de l'assentament, sino que s'edita manualment per l'administrador, ja que els abonaments son variables per part dels festers, al contrari que els deures, que són fixos.

## 6.1.2 Vistes

Les vistes estan implementades a la carpeta *views*, mitjançant els fitxers *XML* s'han estructurat les vistes de cada model que podran seran de tipus *tree* i *formulari*, així com algunes qüestions de lògica com son els casos d'ocultar funcionalitats depenent de cada situació i tipus d'usuari de l'aplicació. A continuació una part d'un fitxer *XML* de la nostra aplicació:

```

4 .....<record: model="ir.ui.view" id="asiento_form_view">
5 .....<field: name="name">asiento_form</field>
6 .....<field: name="model">festermanager.asiento</field>
7 .....<field: name="arch": type="xml">
8 .....<form: string="Accounting Entry" create="false" edit="false">
9 .....<sheet>
10 .....<sgroup>
11 .....<field: name="ref"/>
12 .....<field: name="name"/>
13 .....<field: name="tipo"/>
14 .....<field: name="categoria": options="{ 'no_create_edit': 1 }"/>
15 .....<field: name="year": readonly="1"/>
16 .....<field: name="precio"/>
17 .....<field: name="activo_asiento": invisible="1"/>
18 .....<field: name="activo_fulla": invisible="1"/>
19 .....<button: name="cargar_asiento":
20 ..... type="object":
21 ..... string="Load Entry to Partners":
22 ..... groups="festermanager.festermanager_group_manager":
23 ..... attrs="{ 'invisible': [ ('activo_asiento', '=', False)] }"/>
24 .....<button: name="cargar_fulla":
25 ..... type="object": string="Load Sheet to Partners":
26 ..... groups="festermanager.festermanager_group_manager":
27 ..... attrs="{ 'invisible': [ ('activo_fulla', '=', False)] }"/>
28 .....</group>
29 .....<footer: attrs="{ 'invisible': [ ('ref', '!=', False)] }">
30 .....<button: name="write": type="object": string="Save": class="oe_highlight"/> or <button: name="cancel": string="Discard": special="cancel": class="oe_link"/>
31 .....</footer>
32 .....</sheet>
33 .....</form>
34 .....</field>
35 .....</record>
36 .....
37 .....<record: model="ir.ui.view" id="asiento_tree_view">
38 .....<field: name="name">asiento_tree</field>
39 .....<field: name="model">festermanager.asiento</field>
40 .....<field: name="arch": type="xml">
41 .....<tree: string="Accounting Entries" create="false" edit="false">
42 .....<field: name="year"/>
43 .....<field: name="ref"/>
44 .....<field: name="name"/>
45 .....<field: name="tipo"/>
46 .....<field: name="categoria"/>
47 .....<field: name="precio"/>
48 .....</tree>
49 .....</field>
50 .....</record>
51 .....</record>

```

Figura 36: Exemple de fitxer XML del mòdul

Vistes a l'aplicació:

### fester\_manager.xml

Implementa el menú de l'aplicació. Ocultant en cada cas el menú a cada tipus d'usuari. Als demés fitxers xml de la carpeta *views* es troben les vistes *formulari* i *tree* de cada model de dades.

### 6.1.3 Permisos i seguretat

Un dels aspectes més importants d'*Odoo* és el mecanisme de seguretat que implementa. Existeixen 3 conceptes diferents que afecten a la seguretat:

- **Usuaris:** són les persones que utilitzaran l'*ERP*. El primer usuari serà l'administrador o súper-usuari del sistema, per tant aquest rol ha de ser assignat a algú amb un gran coneixement tècnic de l'aplicació o sinó a l'implantador.
- **Grups:** Es defineixen nivells d'accés i de seguretat que comparteixen els usuaris que pertanyen al grup. Si un usuari pertany a diferents grups tindrà permisos més elevats per a recurs determinat.
- **Regles de seguretat:** Són regles que s'assignen als grups que defineixen els permisos per determinats objectes de l'aplicació:
  - Es poden definir diferents nivells d'accés als recursos del sistema *ERP*: lectura, escriptura, creació, eliminació, etc.
  - Visualització i personalització de menús.
  - Control de la informació a la que s'accedeix, per exemple limitant la visió global.
  - Regles referents als fluxos de treball de l'aplicació.

La seguretat i els permisos de l'aplicació es troba a la carpeta *security*.

Dins del *festermanager\_security.xml* es troben declarats els dos grups d'usuaris que té l'aplicació: *manager* i *usuaris*.

Al *ir.model.access.csv* es troben declarats els permisos de cada un dels dos grups d'usuaris per a cada model de dades.

```
1 id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink
2 access_partner_manager,festermanager_partner_manager,model_res_partner,festermanager_group_manager,1,1,1,1
3 access_partner_user,festermanager_partner_user,model_res_partner,festermanager_group_user,1,1,0,0
4 access_users_manager,festermanager_users_manager,model_res_users,festermanager_group_manager,1,1,1,1
5 access_users_user,festermanager_users_user,model_res_users,festermanager_group_user,1,1,0,0
6 access_groups_manager,festermanager_groups_manager,model_res_groups,festermanager_group_manager,1,1,0,0
7 access_groups_user,festermanager_groups_user,model_res_groups,festermanager_group_user,0,0,0,0
8 access_infantil_manager,festermanager_infantil_manager,model_festermanager_infantil,festermanager_group_manager,1,1,1,1
9 access_infantil_user,festermanager_infantil_user,model_festermanager_infantil,festermanager_group_user,1,0,0,0
10 access_pago_manager,festermanager_pago_manager,model_festermanager_pago,festermanager_group_manager,1,1,1,0
11 access_pago_user,festermanager_pago_user,model_festermanager_pago,festermanager_group_user,1,0,0,0
12 access_year_manager,festermanager_year_manager,model_festermanager_year,festermanager_group_manager,1,1,1,0
13 access_year_user,festermanager_year_user,model_festermanager_year,festermanager_group_user,1,0,0,0
14 access_debe_manager,festermanager_debe_manager,model_festermanager_debe,festermanager_group_manager,1,1,1,1
15 access_debe_user,festermanager_debe_user,model_festermanager_debe,festermanager_group_user,1,0,0,0
16 access_haber_manager,festermanager_haber_manager,model_festermanager_haber,festermanager_group_manager,1,1,1,1
17 access_haber_user,festermanager_haber_user,model_festermanager_haber,festermanager_group_user,1,0,0,0
18 access_asiento_manager,festermanager_asiento_manager,model_festermanager_asiento,festermanager_group_manager,1,1,1,1
19 access_asiento_user,festermanager_asiento_user,model_festermanager_asiento,festermanager_group_user,1,0,0,0
20 access_categoria_manager,festermanager_categoria_manager,model_festermanager_categoria,festermanager_group_manager,1,1,1,1
21 access_categoria_user,festermanager_categoria_user,model_festermanager_categoria,festermanager_group_user,1,0,0,0
22
```

Figura 37: Fitxer *ir.model.access.csv* amb permisos

Taula amb els permisos per a cada grup:

Model	Lectura		Escriptura		Crear		Borrar	
	Manag	Usuari	Manag	Usuari	Manag	Usuari	Manag	Usuari
Socis	Si	Si	Si	Si	Si	No	Si	No
Usuaris	Si	Si	Si	Si	Si	No	Si	No
Grups	Si	No	Si	No	No	No	No	No
Infantil	Si	Si	Si	No	Si	No	Si	No
Pagament	Si	Si	Si	No	Si	No	No	No
Any	Si	Si	Si	No	Si	No	No	No
Deure	Si	Si	Si	No	Si	No	Si	No
Haver	Si	Si	Si	No	Si	No	Si	No
Assentament C.	Si	Si	Si	No	Si	No	Si	No
Categoria	Si	Si	Si	No	Si	No	Si	No

#### 6.1.4 Internacionalització

L'internacionalització de la aplicació, amb les seues traduccions en espanyol i català venen definides a la carpeta i18n mitjançant els fitxers es.po i ca.po

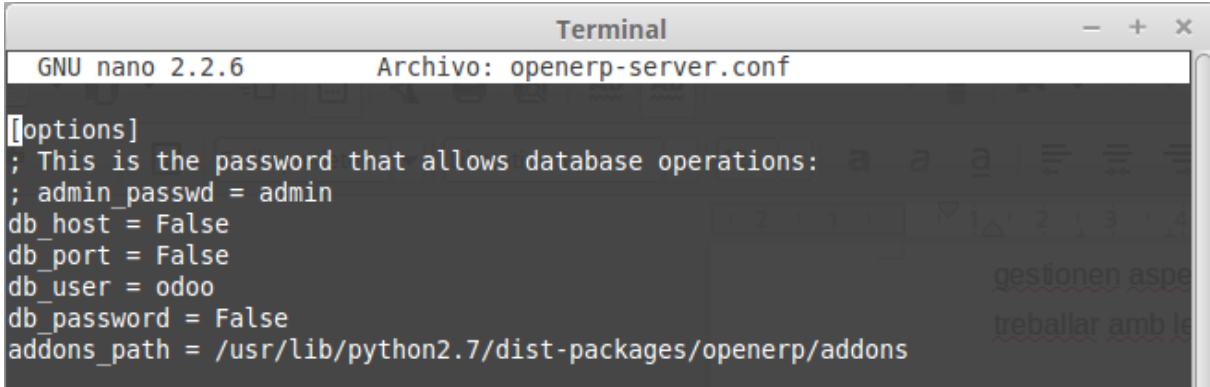
Nativament l'aplicació està en anglés. L'usuari podrà escollir entre l'anglés, l'espanyol i el català.

#### 6.1.5 Persistència de dades

Per defecte, *PostgreSQL* tant sols permet connexions sota sockets *UNIX* i connexions loopback (de "localhost", la mateixa maquina on el servidor PostgreSQL està instal·lat).

El socket *UNIX* es bò si volem executar *Odoo* i *PostgreSQL* en la mateixa màquina. Aquest es el nostre cas. *Odoo* es connecta al postgres local mitjançant un socket *UNIX* pel port 5432. Es configura per defecte seguint el manual d'instal·lació.

Fitxer de configuració a `~/odoo-dev/odoo/debian/openerp-server.conf`, si vulguerem canviar de port o d'altres configuracions s'afegiria a aquest fitxer:

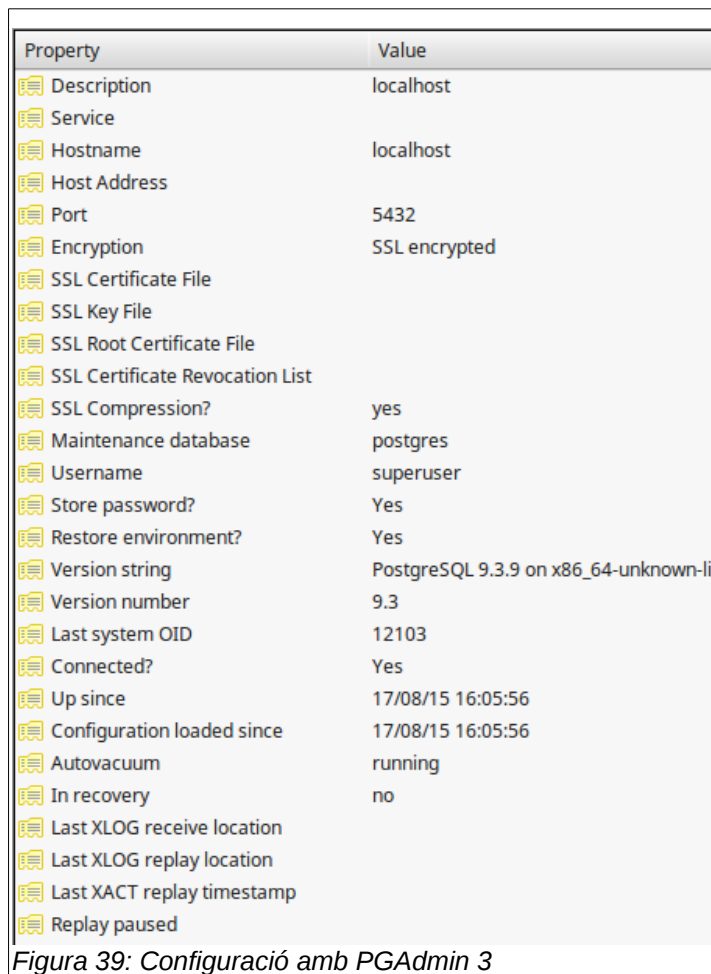


```
GNU nano 2.2.6      Archivo: openerp-server.conf

[options]
; This is the password that allows database operations:
; admin_passwd = admin
db_host = False
db_port = False
db_user = odoo
db_password = False
addons_path = /usr/lib/python2.7/dist-packages/openerp/addons
```

Figura 38: Fitxer `openerp-server.conf`

Configuració del PGAdmin, per a gestionar la base de dades, es pot observar el port pel qual s'accedeix, l'usuari per a administrar, etc:



Property	Value
Description	localhost
Service	
Hostname	localhost
Host Address	
Port	5432
Encryption	SSL encrypted
SSL Certificate File	
SSL Key File	
SSL Root Certificate File	
SSL Certificate Revocation List	
SSL Compression?	yes
Maintenance database	postgres
Username	superuser
Store password?	Yes
Restore environment?	Yes
Version string	PostgreSQL 9.3.9 on x86_64-unknown-li
Version number	9.3
Last system OID	12103
Connected?	Yes
Up since	17/08/15 16:05:56
Configuration loaded since	17/08/15 16:05:56
Autovacuum	running
In recovery	no
Last XLOG receive location	
Last XLOG replay location	
Last XACT replay timestamp	
Replay paused	

Figura 39: Configuració amb PGAdmin 3

## 7. Proves

La finalitat d'aquest apartat es aconseguir un correcte funcionament de l'aplicació. I per a garantir aquest correcte funcionament, s'ha sotmés a diferents tipus de proves per a detectar errors. Aquest punt es fundamental en el desenvolupament d'una aplicació d'aquestes característiques, ja que hem de aconseguir una correcta visualització i una correcta resposta als possibles errors davant l'usuari que l'utilitze.

Per a detectar aquestos errors, s'ha portat a cap tant proves unitaries com d'integració.

### 7.1 Proves unitaries

Cada vegada que finalitzava una "versió" del mòdul i abans de mostrar-la a l'usuari, era sotmés a una serie de proves amb la pitjor intenció possible per a trobar els possibles errors. Així com proves seguint les indicacions de l'aplicació per a veure que realitza el que te que realitzar.

Les diferents proves que s'han realitzat dins d'aquesta categoria son les següents:

- **Validació de camps buïts:** s'ha comprovat que per a continuar realitzant alguna acció determinada, deuen estar tots els camps obligatoris plens. I que aparega un missatge d'error indicant quins camps són els que falten per plenar.
- **Validació del format de les dades d'entrada:** s'ha comprovat que a part d'estar tots els camps obligats plens, així com s'indica en el punt anterior, també que aquestos camps tinguen el format correcte per a que no hi haja problemes a l'hora d'insertar les dades a la base de dades. I que un missatge d'error aparega indicant quines dades són les que tenen un format incorrecte.
- **Validació de la informació en la base de dades:** s'ha comprovat que al insertar, eliminar o modificar qualsevol dada en la base de dades, es faja correctament el que voliem que s'efectuara.
- **Correcta visualització:** s'ha comprovat que el mòdul en qüestió es visualitze be tant a Chrome, Mozilla Firefox, com a Internet Explorer.

## **7.2 Proves d'integració**

Aquestes proves son aquelles que es realitzen de forma conjunta una vegada tenim el mòdul desenvolupat.

- **Persistència de la sessió:** s'ha comprovat la persistència de la sessió de l'usuari, fins que l'usuari no la tanca.
- **Opcions dels diferents usuaris:** s'han creat diferents perfils d'usuari, per a poder comprovar que depenent del perfil d'usuari que es tracte, tinga accés a unes determinades opcions o altres.
- **Prova en remot:** Comprovar el funcionament en remot en cas de posar-se en producció.



## 8. Instal·lació i manual d'usuari

Com hem explicat, per al cas que es necessite o s'opte per fer una instal·lació interna, s'indicarà com posar en funcionament el servidor d'*Odoo* i la instal·lació del mòdul *FesterManager*.

Important recordar que aquest pas el realitzarà el superusuari de l'aplicació.

### 8.1 Instal·lació i posada en funcionament d'*Odoo*

Instal·lem *Odoo* en el sistema *Linux* (*Debian* per exemple):

```
$ sudo apt-get update && sudo apt-get upgrade
$ sudo apt-get install git
$ mkdir ~/odoo-dev
$ cd ~/odoo-dev
$ git clone https://github.com/odoo/odoo.git -b 8.0 (obté el codi font d'Odoo)
$ ./odoo/odoo.py setup_deps (instala les dependències del sistema d'Odoo)
$ ./odoo/odoo.py setup_pg (instala postgresSQL i superusuari de base de dades)
$ ~/odoo-dev/odoo/odoo.py (arrancar servidor Odoo)

#Crear Alias
$ alias odoo='~/odoo-dev/odoo/odoo.py'
$ echo 'alias odoo="~/odoo-dev/odoo/odoo.py"' >> ~/.bashrc && . ~/.bashrc
```

Instal·lem *PGAdmin 3* per a la gestió de la base de dades:

```
$ sudo apt-get install pgadmin3
```

Superusuari postgres per a *PGAdmin 3*:

```
$ sudo su - postgres
$ createuser --createdb --username postgres --no-createrole --superuser --pwprompt odoo
Enter password for new role: odoo
Enter it again: odoo
$ exit
```

Com podem observar a la imatge, l'usuari encarregat del manteniment de la base de dades serà l'usuari postgres que hem creat anteriorment.

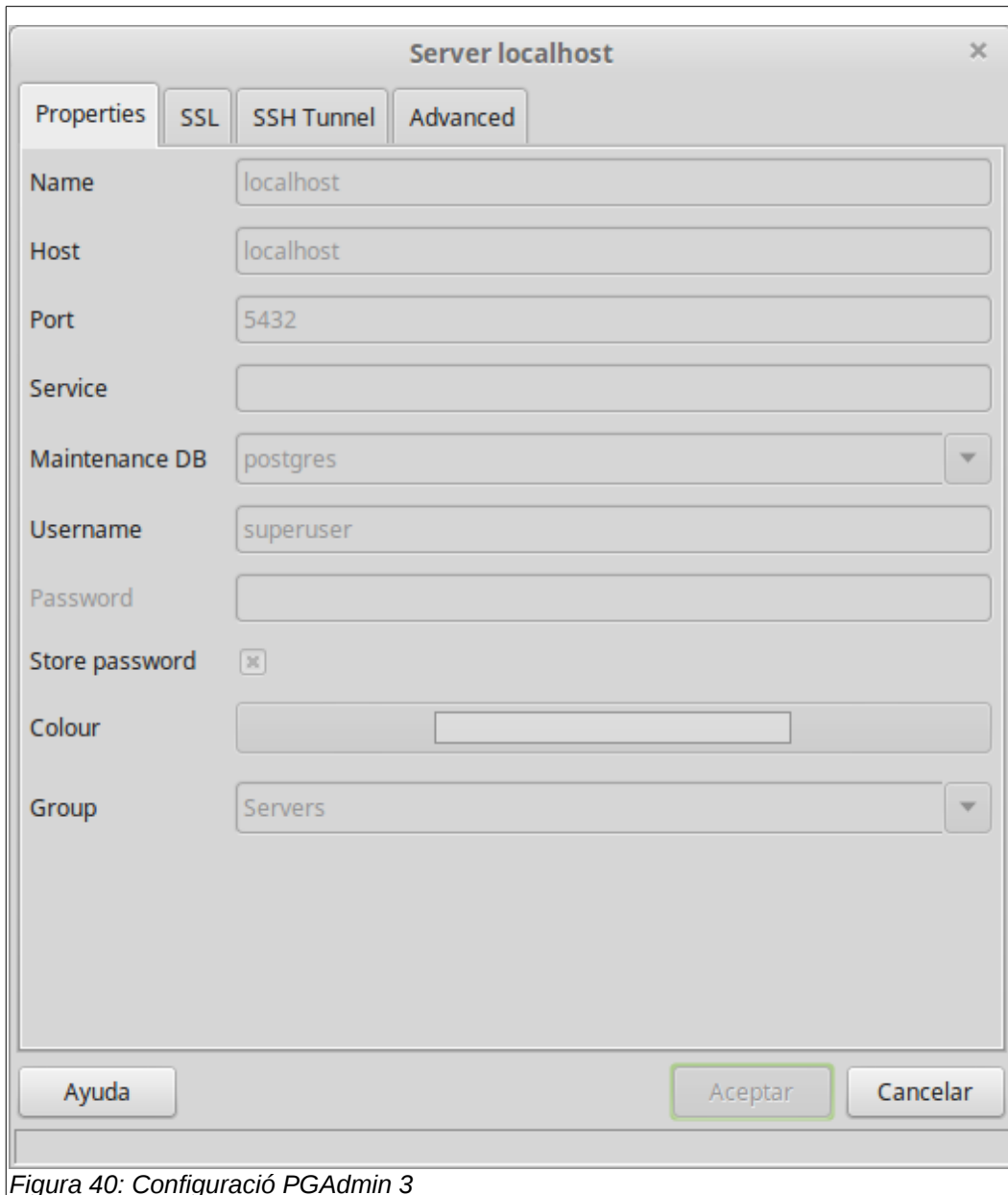


Figura 40: Configuració PGAdmin 3

Una vegada tenim *Odoo* i *PostgreSQL* instal·lat i configurat, procedim a la creació de la base de dades que utilitzarem per al nostre mòdul així com la instal·lació del mateix: Accedim vïa navegador web (en aquest cas *Chrome*) a la url <http://localhost:8069>, per a accedir a *Odoo*, el primer que es trobarem serà aquesta pantalla:

**odoo**

Administración base de datos

- Crear
- Duplicar
- Eliminar
- Copia de seguridad
- Restaurar
- Contraseña

### Crear una nueva base de datos

Rellene este formulario para crear una base de datos de Odoo. Puede crear bases de datos para diferentes compañías o para diferentes objetivos (pruebas, producción). Una vez esté la base de datos creada, podrá instalar la primera aplicación.

By default, the master password is 'admin'. This password is required to create, delete, dump or restore databases.

**Contraseña maestra**

**Seleccione un nombre de base de datos:**

**Cargar datos de demostración:**  Marque esta casilla para evaluar Odoo

**Idioma por defecto:**

**Elija una contraseña:**

**Confirmar contraseña:**

**Crear Base de Datos**

Figura 41: Creació Base de dades

En aquest punt crearem la base de dades. Ací introduïrem els següents camps:

- **Contrassenya mestra:** serà la contrassenya que utilitzarem per a gestionar la base de dades, es a dir, fer còpies, eliminar, duplicar i restaurar.
- **Nom:** aquest serà el nom de la base de dades. En aquest cas li diem 'festermanager'.
- **Idioma per defecte:** com el nom indica, serà l'idioma per defecte de la nostra instal·lació. En aquest cas triem el català.
- **Contrassenya:** com el seu nom també indica, serà la contrassenya que utilitzarem per a accedir a l'aplicació. Aquesta serà la contrassenya del superusuari junt amb el nom per defecte 'admin'.

Li donem a crear base de dades.

Una vegada creada la base de dades ja podrem accedir a la nostra instal·lació pel navegador. Encara que en el moment de crear-la entra directament, mostrem com es la pantalla de login d'Odoo:

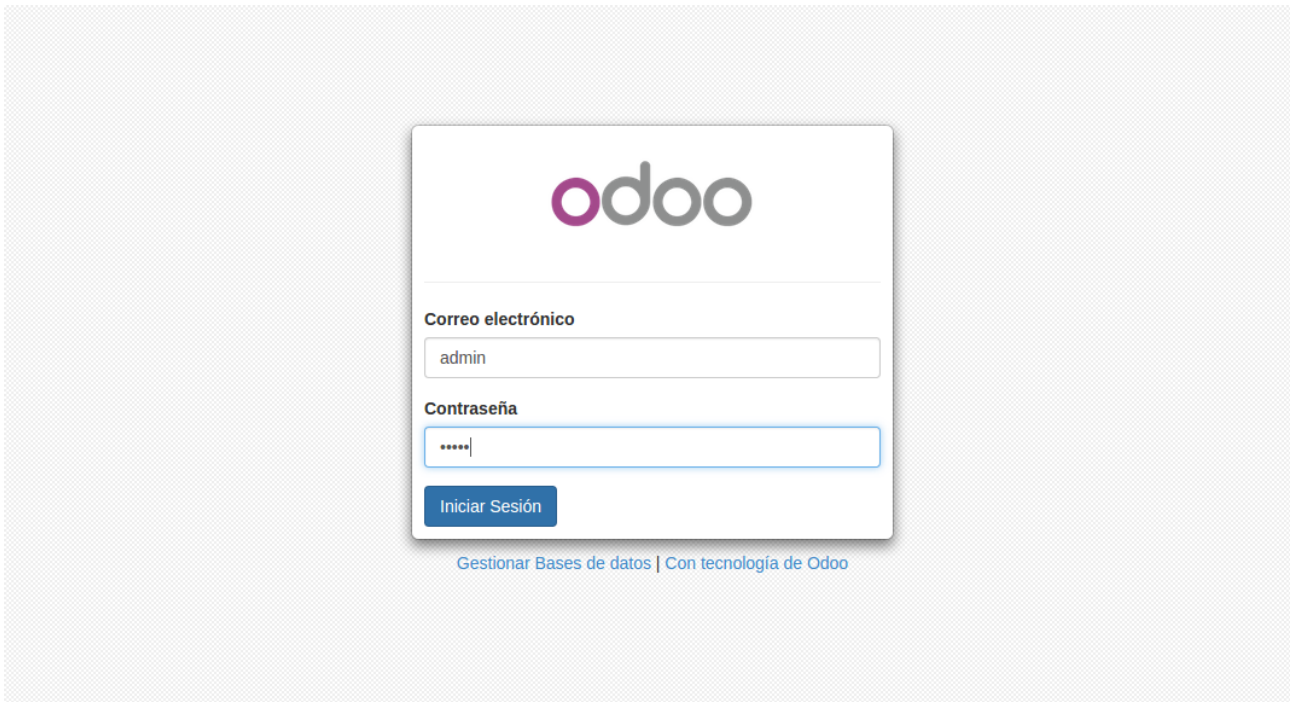


Figura 42: Pantalla de Login

El primer que farem serà accedir al nostre usuari i donar-li permís de “Característiques tècniques”:

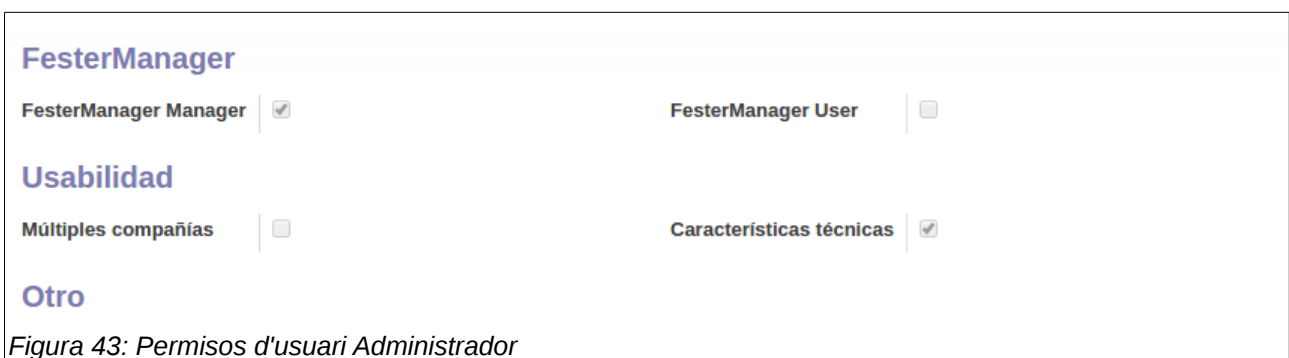


Figura 43: Permisos d'usuari Administrador

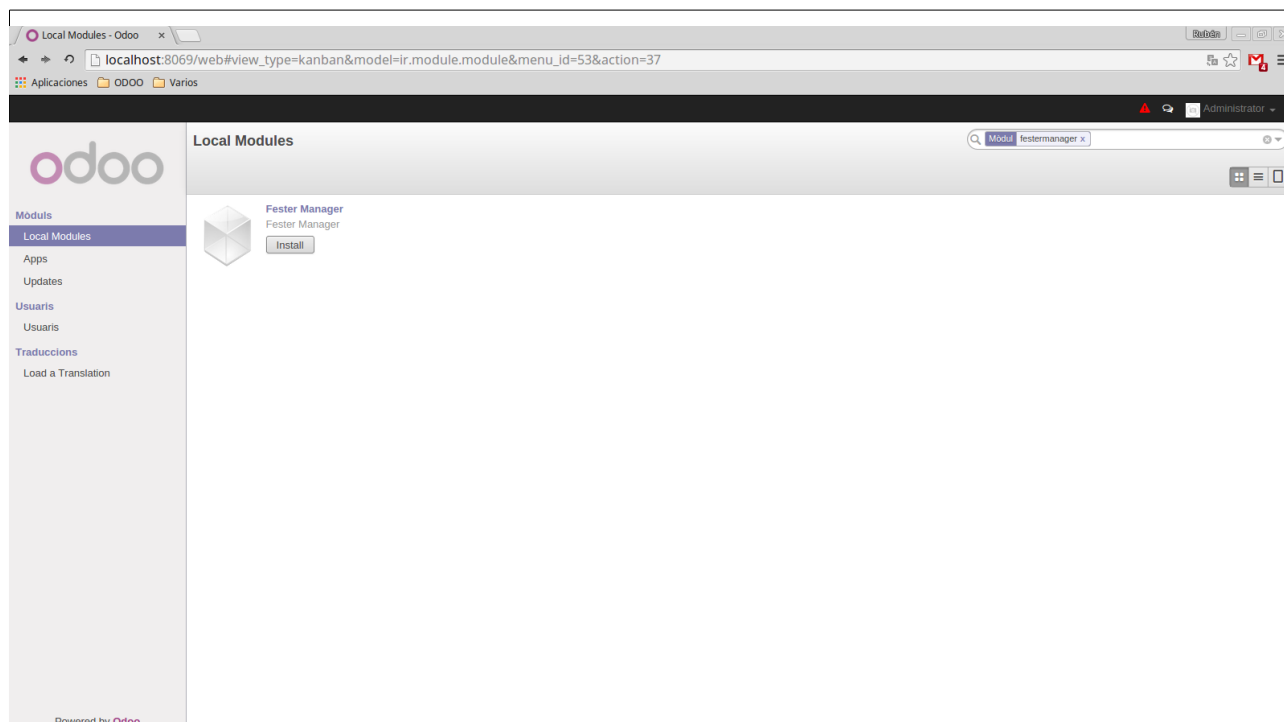


Figura 44: Instal·lació Mòdul

Una vegada accedim, busquem el nostre mòdul anomenat 'festermanager':

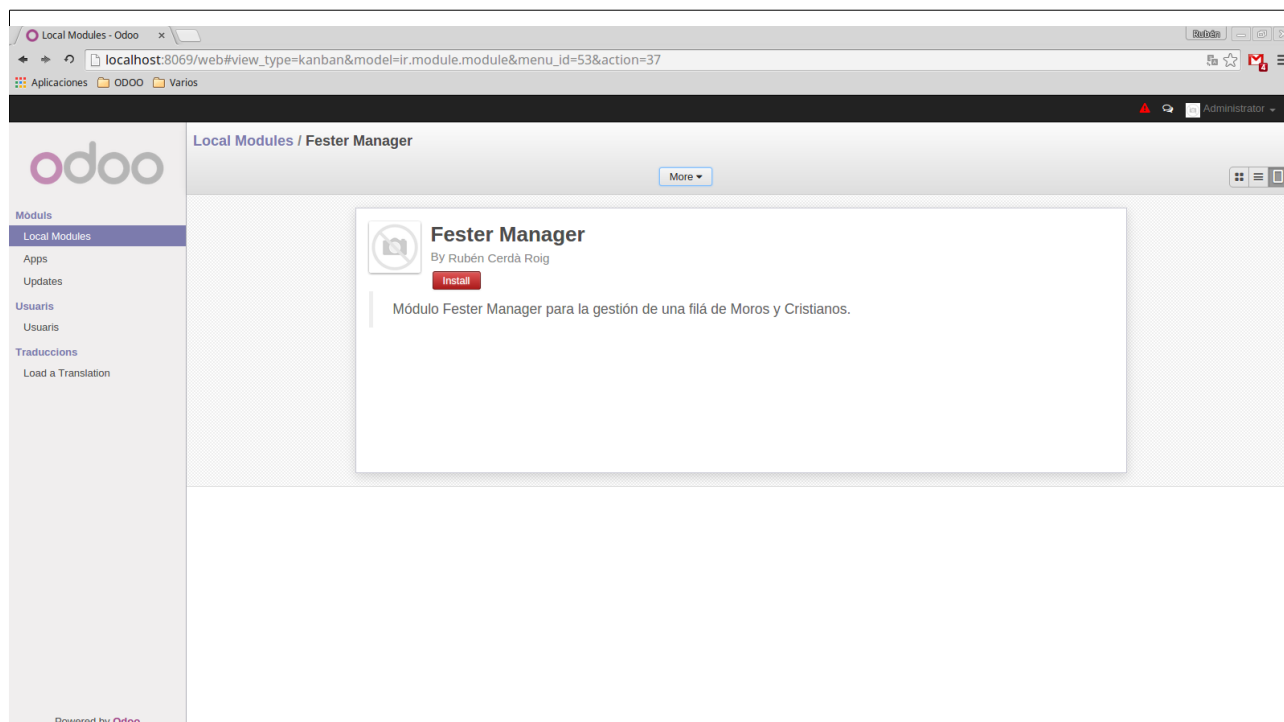


Figura 45: Instal·lació Mòdul

Li donem a instal·lar.

Una vegada instal·lat el mòdul, anirem a configuració per carregar l'altre idioma que ens falta, que es l'espanyol (l'anglès vé per defecte a *Odoo*). Dins de configuració, el menú de "Load a Translation":

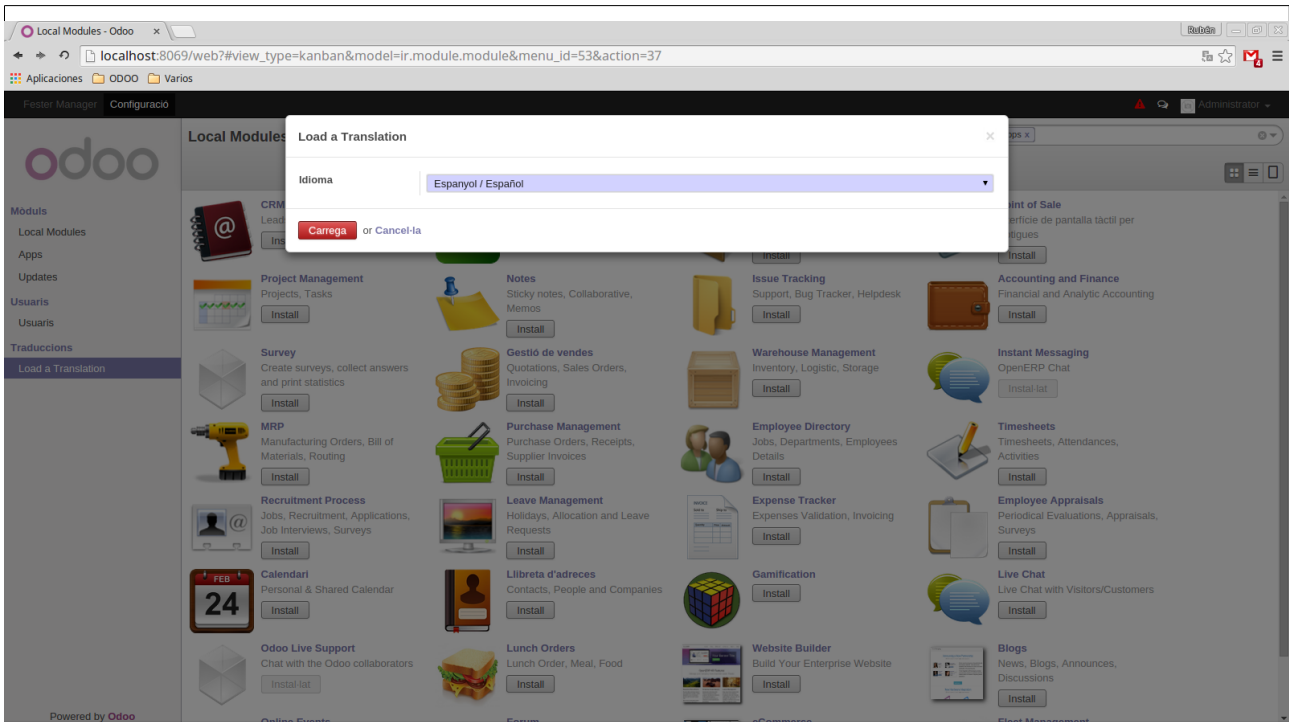


Figura 46: Carregar Idiomes



Figura 47: Carregar Idiomes

Opcionalment, si es vol utilitzar un servidor de correu d'eixida:

A continuació s'ha de configurar el servidor de correu d'eixida per a poder enviar correus automàtics quan es cree un usuari, per tant de que l'usuari puga configurar la seua contrassenya.

Primer que res instal·larem el mòdul d'Odoo anomenat "mail":

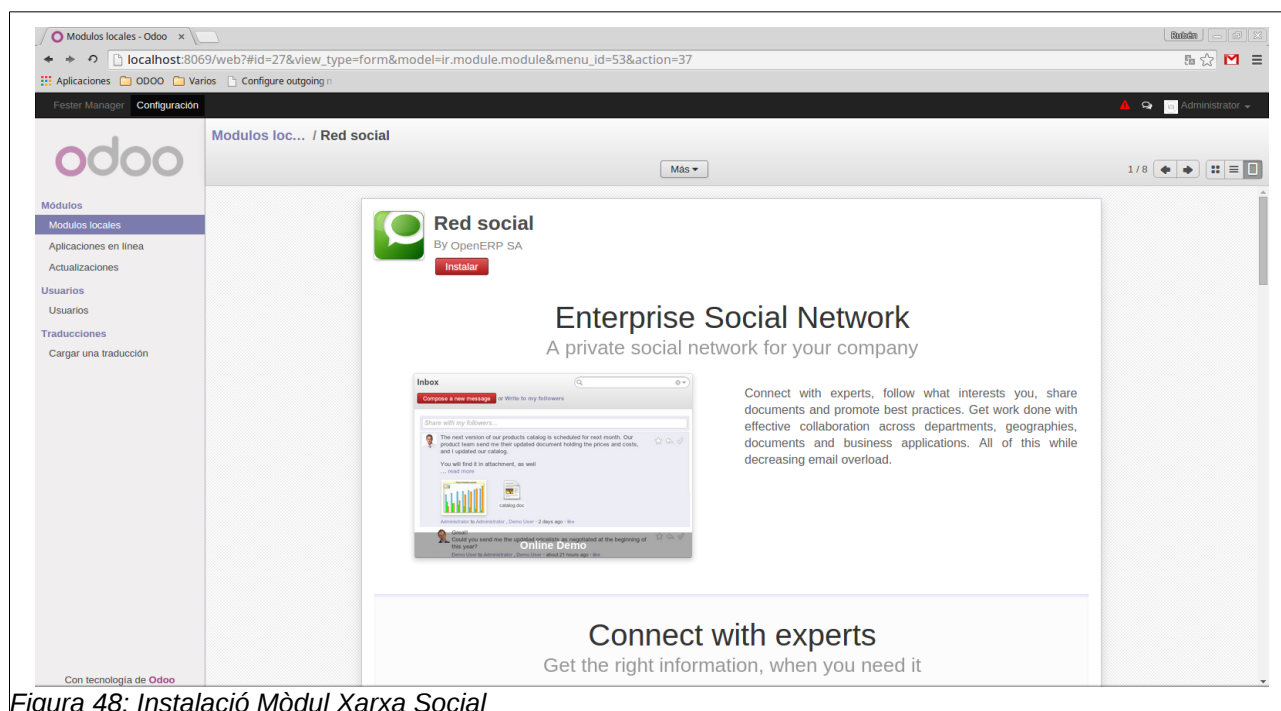


Figura 48: Instal·lació Mòdul Xarxa Social

A continuació anirem a *configuració/tècnic/email/servidors* de correu d'eixida i crearem i configurarem el servidor de correu d'eixida, amb les següents dades, en aquest cas correu *gmail*:

Descripción	Gmail	Prioridad	1
<b>Información de la conexión</b>			
Servidor SMTP	smtp.gmail.com	Puerto SMTP	465
Depurando	<input type="checkbox"/>		
<b>Seguridad y autenticación</b>			
Seguridad de la conexión	SSL/TLS		
Nombre de usuario	festermanager@gmail.com		
Contraseña	.....		
<input type="button" value="Probar conexión"/>			

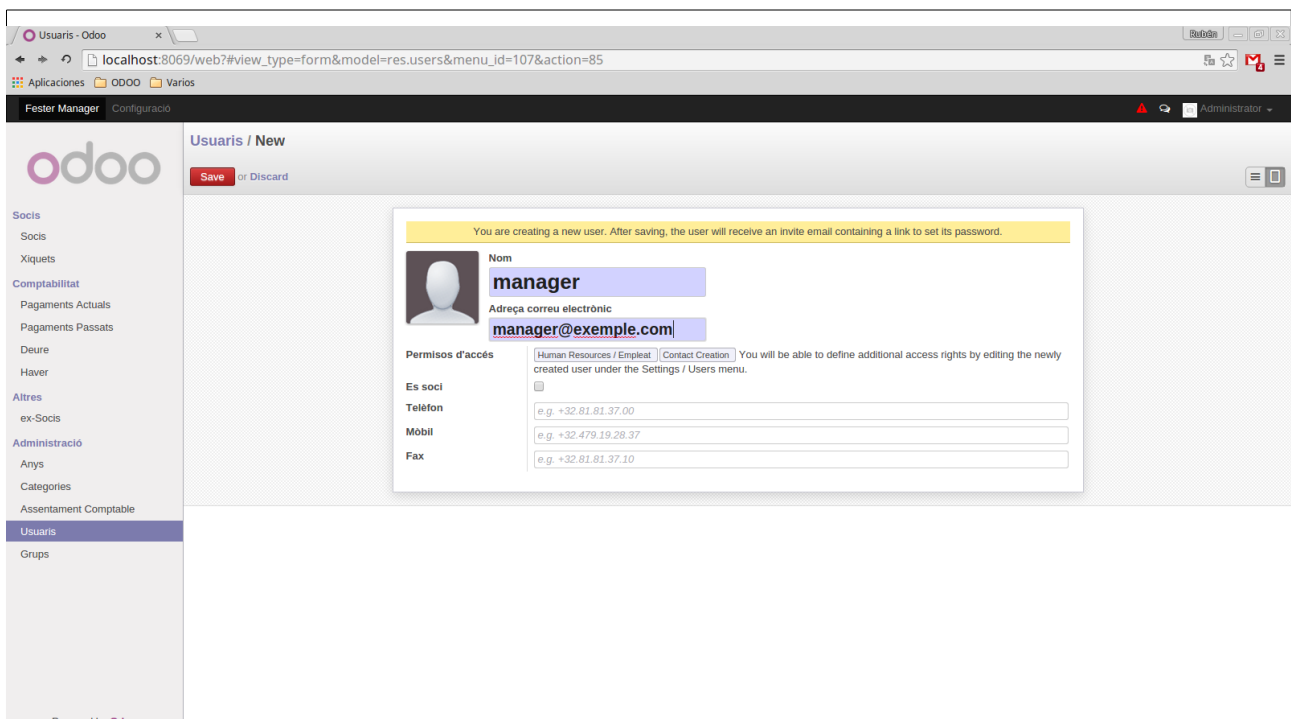
Figura 49: Configuració Servidor de Correu d'eixida

Trobem les següents dades:

- **Descripció:** Una descripció del correu d'eixida.
- **Prioritat:** La prioritat del servidor de correu. Quan més baix més prioritat. Significa que el servidor de correu amb el número més baix serà el més utilitzat.
- **Servidor SMTP:** La direcció del SMTP del servidor, en aquest cas *smtp.gmail.com*.
- **Port SMTP:** El port del SMTP del servidor, en aquest cas el 465 per a Gmail.
- **Seguretat de la connexió:** El tipus de seguretat. Deguem triar SSL/TLS per a Gmail. Les dades són enviades per *HTTPS*.
- **Nom d'usuari:** El compte de correu, en aquest exemple *festermanager@gmail.com*
- **Contrassenya:** La contrassenya del correu.

Una vegada fet este pas, ja tenim el servidor de correu d'eixida configurat per a poder enviar automàticament correus als usuaris.

Ara dins del menú de Fester Manager, accedirem a usuaris i crearem l'usuari que serà l'encarregat de la filà de controlar l'aplicació, que pertanyerà al grup "Manager".



The screenshot shows the Odoo web interface for creating a new user. The browser address bar indicates the URL is `localhost:8069/web?#view_type=form&model=res.users&menu_id=107&action=85`. The page title is 'Usuaris / New'. The form contains the following fields and options:

- Save** or **Discard** buttons.
- Message:** You are creating a new user. After saving, the user will receive an invite email containing a link to set its password.
- Avatar:** A placeholder image of a person.
- Nom:** `manager`
- Adreça correu electrònic:** `manager@exemple.com`
- Permisos d'accés:** `Human Resources / Empleat` and `Contact Creation`. A note states: "You will be able to define additional access rights by editing the newly created user under the Settings / Users menu."
- Es soci:**
- Telèfon:** `e.g. +32.81.81.37.00`
- Mòbil:** `e.g. +32.479.19.28.37`
- Fax:** `e.g. +32.81.81.37.10`

Figura 50: Creació Usuari Manager



En la següent figura vegem la invitació per correu per a connectar-se, en cas d'haver-se configurat, amb el mòdul de mail, un servidor de correu d'eixida. En cas de no haver-ho fet, la contrassenya es proporcionarà de forma manual a l'usuari. Tenim l'opció com a superusuari de canviar-li-la.



Figura 51: Correu d'invitació a Odoo

El següent pas serà el d'editar les dades de la filà, com el nom, direcció, etc.. Açò hem deixat en mans de l'administrador, per això farà clic sota el símbol d'*Odoo* dalt a l'esquerra, s'obrirà una finestra on podrà introduir les dades de la filà així com un logotip.

El següent pas i definitiu per a donar per posada en marxa l'aplicació serà el d'afegir a l'usuari manager al grup Manager, per a que puga accedir al menú de l'aplicació i les seues funcionalitats:

Des del menú *grups*, triem *FesterManager Manager* i editem per a afegir a l'usuari.

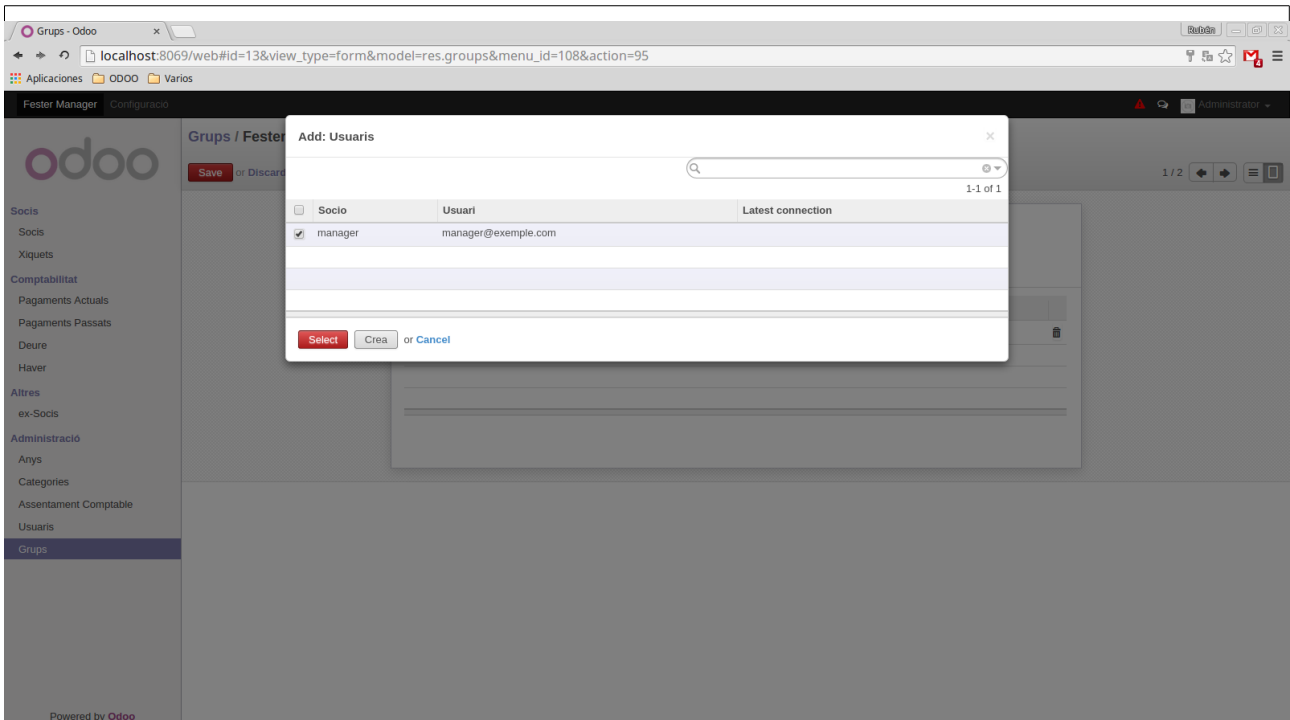


Figura 52: Afegir usuari a grup Manager

Amb açò l'encarregat de la filà ja està preparat per a poder accedir a l'aplicació.

Un punt important es el de guardar còpies i restaurar còpies de seguretat de la base de dades, per a això accedirem a <http://<adreça>:8069/> (en el nostre cas d'exemple l'adreça és *localhost*) i entrarem a "Gestionar bases de datos":



Figura 53: Pantalla Login

Una vegada dins podrem fer varies accions com es pot veure al menú de l'esquerra.

- **Duplicar:** com el propi nom indica, ens permet duplicar una base de dades. Després es podrà accedir a ella de la mateixa forma que amb l'original.

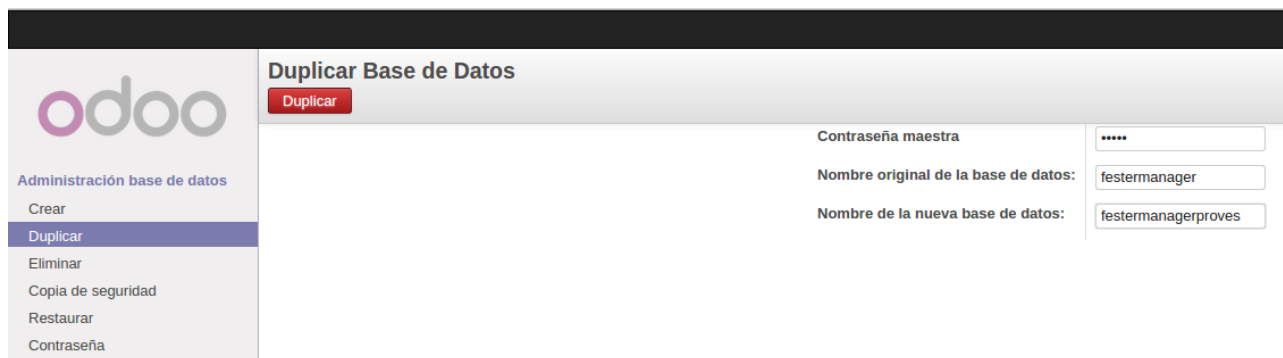


Figura 54: Duplicar base de dades

- **Eliminar Base de dades:** Elimina la base de dades que se li indica. Per a poder fer-ho necessitarem introduir la contrassenya d'administrador.




Figura 55: Eliminar base de dades

- **Còpia de seguretat:** Ens permet crear una còpia de seguretat de la nostra base de dades i guardar-la en un fitxer extern. Després es podrà restaurar aquesta base de dades des del menú de restaurar. Ho farem tal com està a la figura:

Figura 56: Còpia de seguretat de la base de dades

- **Restaurar:** Des d'ací podrem restaurar, com hem dit abans, una còpia de seguretat d'una base de dades anteriorment creada. Hem de restaurar-la a un nom que no existeixca, además necessitarem la contrassenya d'administrador per a poder restaurar. Deixem les opcions com es veu a la figura:

Figura 57: Restaurar base de dades

- **Contrassenya:** Ací es pot canviar la contrassenya de l'administrador:

Figura 58: Canviar contrassenya

## 8.2 Manual de Manager

### 8.2.1 Login

La primera cosa que farem serà autenticar-nos a l'aplicació amb el nostre correu i contrassenya. Per això accedirem a *http://<adreça>:8069/* (en aquest cas d'exemple *localhost*).

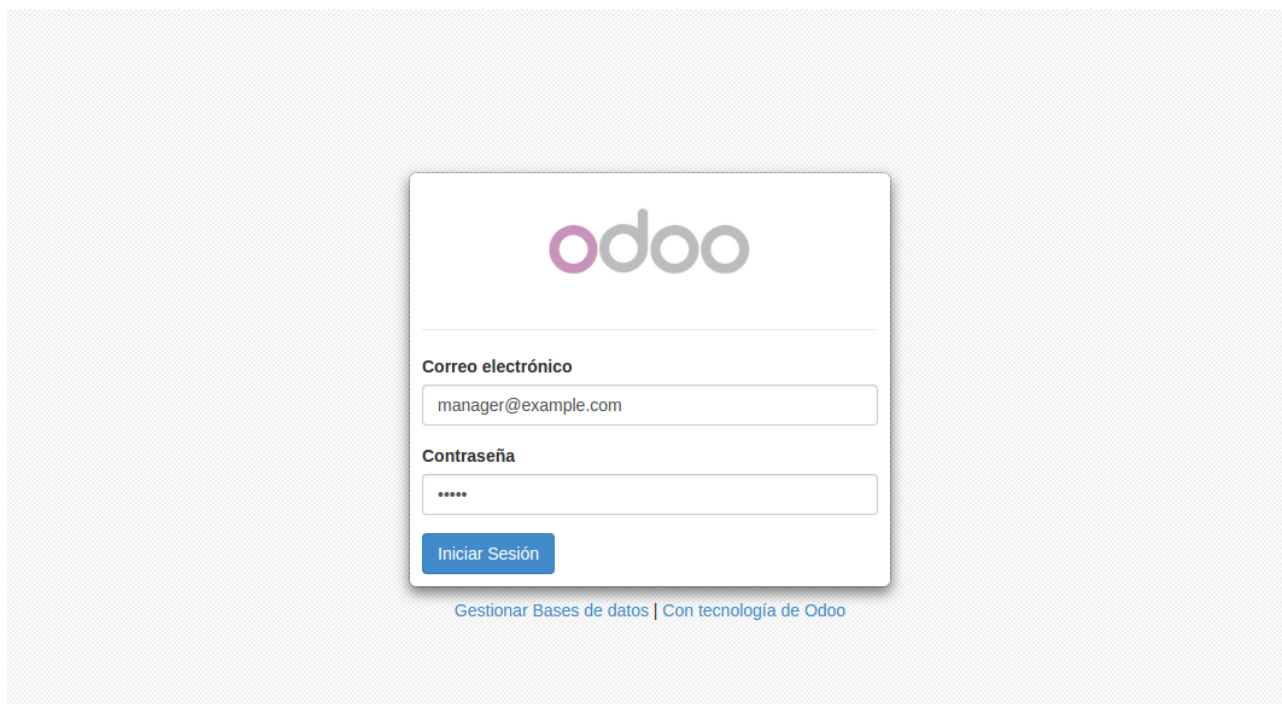


Figura 59: Pantalla Login

### 8.2.2 Preferències d'usuari.

Per a canviar preferències hem d'accedir al menú superior dret, clicar i triar l'opció "Preferencias".

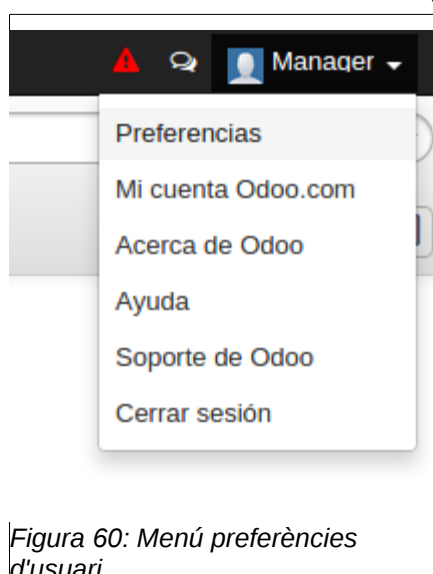
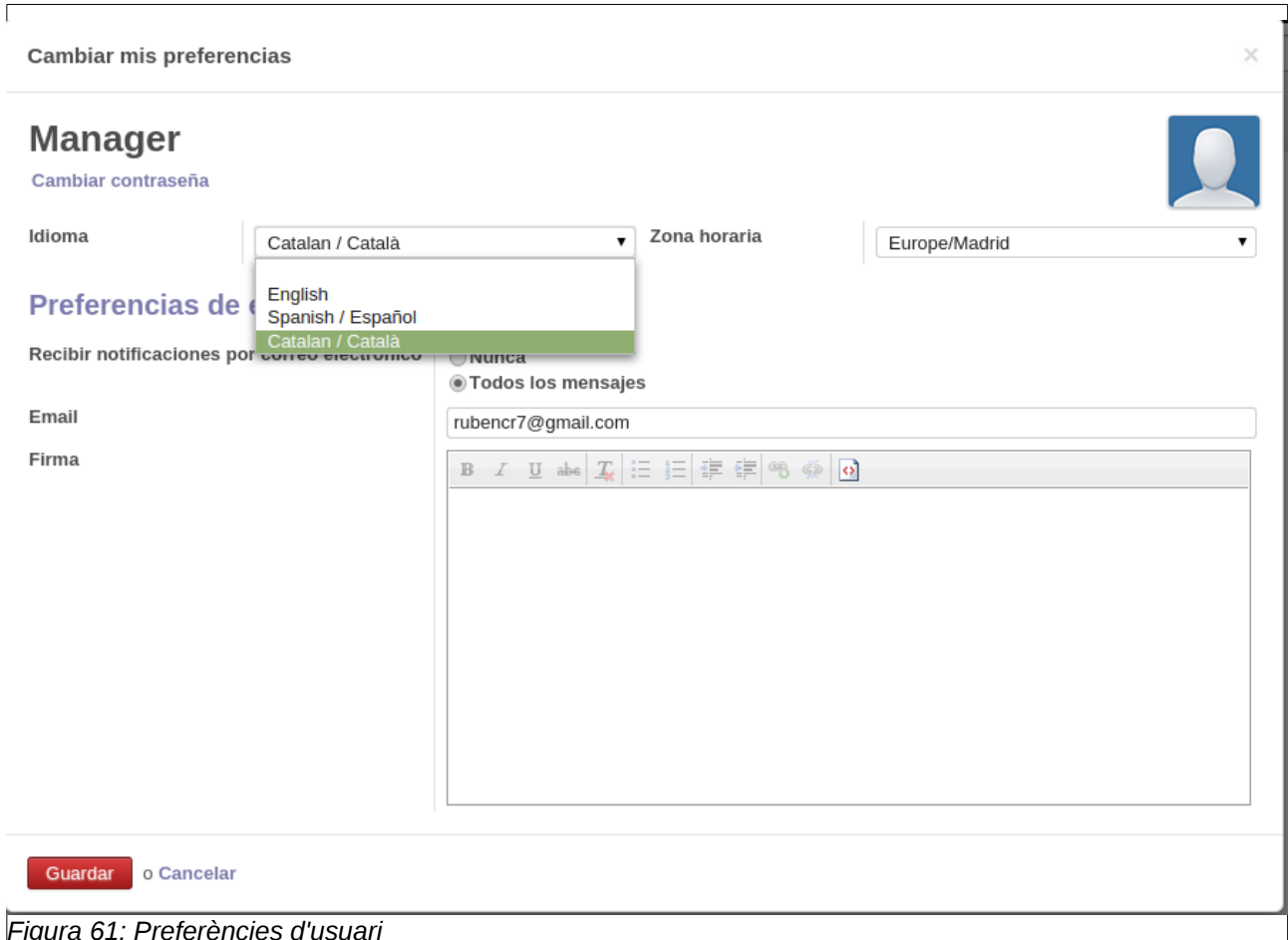


Figura 60: Menú preferències d'usuari

A continuació podem canviar diferents paràmetres com l'idioma, la contrassenya, la zona horària, la foto de perfil, etc:



**Cambiar mis preferencias**

**Manager**  
Cambiar contraseña

Idioma: Catalan / Català    Zona horaria: Europe/Madrid

Preferencias de Idioma: English, Spanish / Español, Catalan / Català

Recibir notificaciones por correo electrónico:  Nunca  Todos los mensajes

Email: rubencr7@gmail.com

Firma

**Guardar** o Cancelar

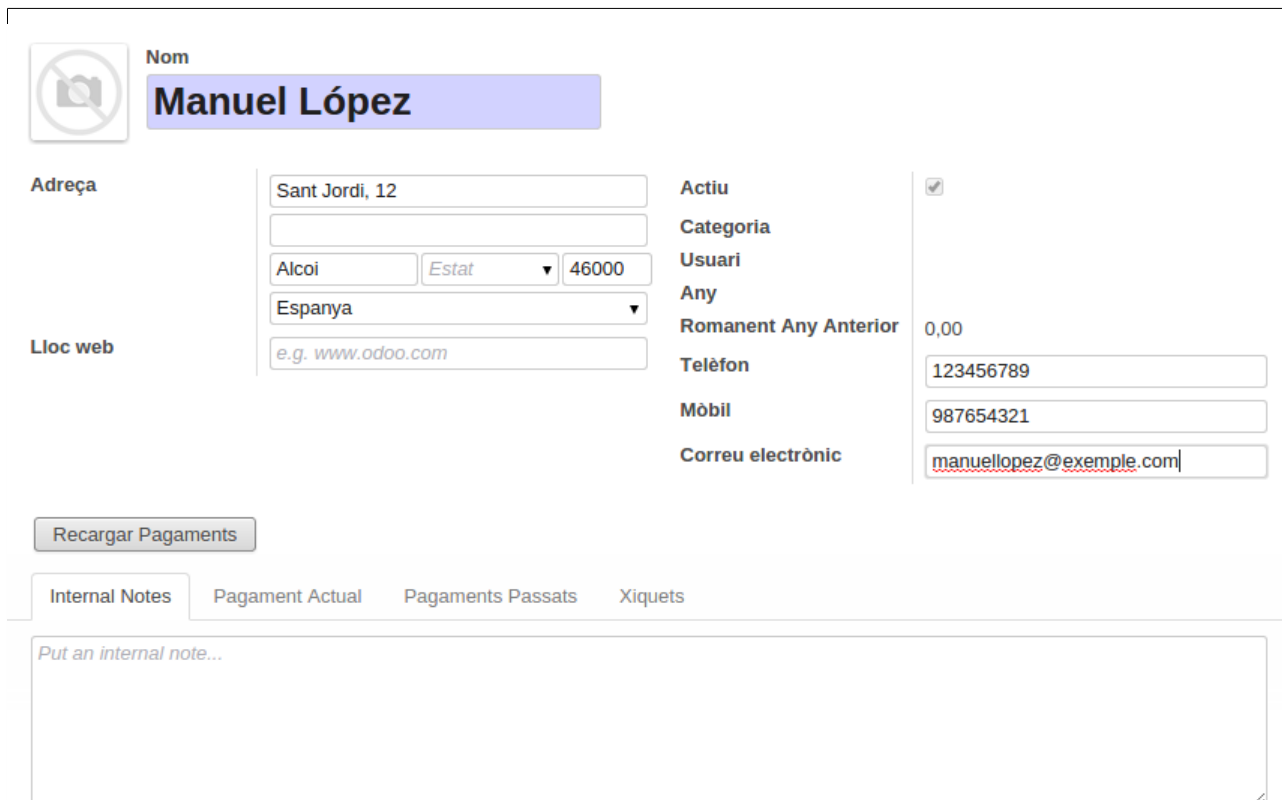
Figura 61: Preferències d'usuari

## 8.2.3 Socis i altres

### 8.2.3.1 Socis

- Crear soci:

Li donarem al boto *crea* i procedirem a introduir les dades:



**Nom**  
Manuel López

**Adreça**  
Sant Jordi, 12

**Lloc web**  
e.g. www.odoo.com

**Alcoi** **Estat** 46000  
Espanya

**Actiu**

**Categoria**

**Usuari**

**Any**

**Romanent Any Anterior** 0,00

**Telèfon** 123456789

**Mòbil** 987654321

**Correu electrònic** manuellopez@exemple.com

Recargar Pagaments

Internal Notes | Pagament Actual | Pagaments Passats | Xiquets

Put an internal note...

Figura 62: Creació de Soci

Una vegada creat el soci ja podrem assignar-li un usuari en cas que siga necessari o bé assignar-li una categoria de fester, des del botó editar podrem editar aquestos camps mitjançant un desplegable:



**Actiu**

**Categoria**

**Usuari**

**Any**

Fester

Figura 63: Assignació de Categoria i Usuari

La casella d'actiu serveix per a activar o desactivar el soci, es a dir, per a fer-lo soci o ex-soci. El botó de *Recarregar pagaments* servirà per a recarregar pagaments quan el soci canvie d'usuari (per a que l'usuari quan entre pugua veure els seus pagaments).

A les pestanyes de baix podrà veure notes internes, pagament actual, xiquets al càrrec, etc.

### 8.2.3.2 Xiquets

Des del menú Xiquets, donar-li al botó crear i introduir dades. Amb els desplegable podrem seleccionar la categoria i el pare del xiquet.



Figura 64: Formulari Xiquets

### 8.2.3.3 Ex-Socis

Aquesta vista té el mateix funcionament que la de socis. En ella podem crear socis igualment, que per defecte seran actius i aniran al menú de socis.

## 8.2.4 Administració

### 8.2.4.1 Anys

Dins del menú d'Anys, donar-li al botó crear:



Figura 65: Formulari Any



Una vegada creat l'Any, que deurà ser únic, ja tindrem a la nostra disposició les diferents funcionalitats implementades mitjançant botons:

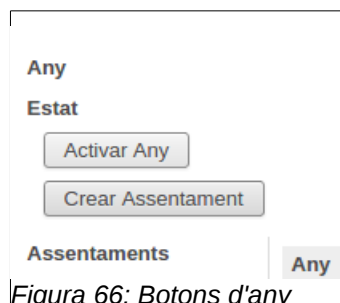
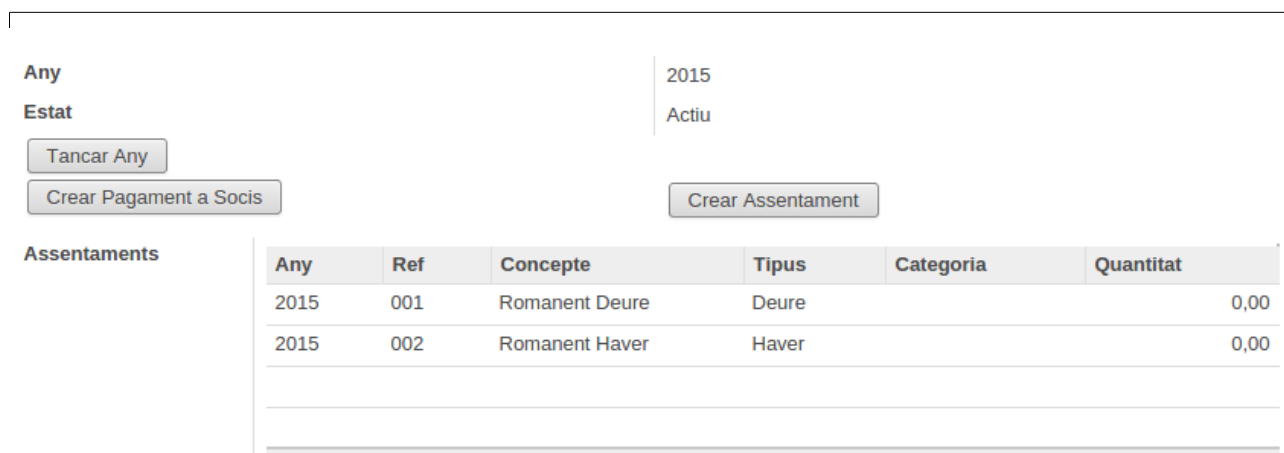


Figura 66: Botons d'any

Podem, pel moment, fer dues accions: Activar l'any o crear assentaments comptables. Activem l'any (en cas d'haver altre any actiu, ens avisa per a que tanquem aquell) i queda el formulari així:



Any 2015  
Estat Actiu  
Tancar Any  
Crear Pagament a Socis  
Crear Assentament

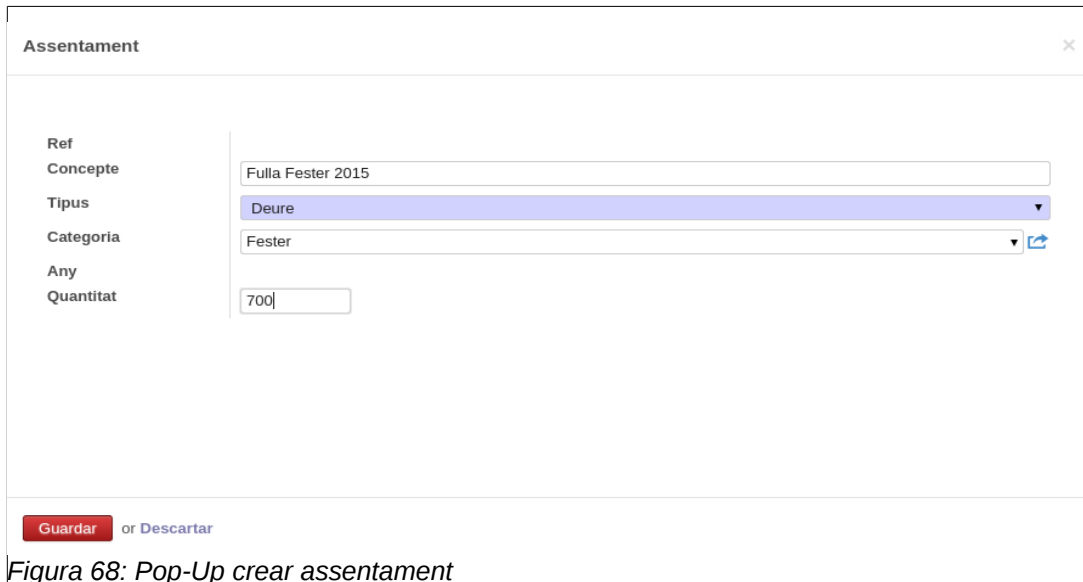
Any	Ref	Concepte	Tipus	Categoria	Quantitat
2015	001	Romanent Deure	Deure		0,00
2015	002	Romanent Haver	Haver		0,00

Figura 67: Vista amb any actiu

Com podem veure, quan s'activa l'any, automàticament es creen dos assentaments comptables, que son *Romanent Deure* i *Haver*. Aquestos assentaments son els que es carregaran quan es cree el pagament en cas que el soci tinga més pagat del compte o dega diners. Amb l'any actiu tenim varies opcions: crear pagament a socis, crear assentament i tancar any.

- **Tancar any:** Tancarà l'any i deixarà d'estar actiu, per tant no es podrà modificar.
- **Crear pagament a socis:** No te cap efecte visual, però carregará el pagament als socis que encara no el tinguen carregat.

- **Crear assentament:** Eixirà un pop-up per a crear un assentament a l'any en qüestió, com mostra la figura:



Assentament

Ref

Concepte Fulla Fester 2015

Tipus Deure

Categoria Fester

Any

Quantitat 700

Guardar or Descartar

Figura 68: Pop-Up crear assentament

Aquest assentament bé pot ser de deure o d'haver (obligat un dels dos, a escollir en un desplegable), tindre categoria o no (en cas de tindre-la, després detectarà que es una fulla, si no la te detectarà un assentament normal) i es pot indicar la quantitat en euros de l'assentament. Se li dona a guardar per a crear-lo. Podem veure l'assentament creat dins l'any 2015 en aquest cas:

Any	Ref	Concepte	Tipus	Categoria	Quantitat
2015	001	Romanent Deure	Deure		0,00
2015	002	Romanent Haver	Haver		0,00
2015	003	Fulla Fester 2015	Deure	Fester	700,00

Figura 69: Assentament creat

Si fem clic sota un assentament podrem carregar-lo als socis mitjançant un botó dins d'un pop-up emergent del assentament, el botó pot ser per a carregar fulla o assentament, depenent del tipus d'assentament:

Obre Assentaments
×

<b>Ref</b>	003
<b>Concepte</b>	Fulla Fester 2015
<b>Tipus</b>	Deure
<b>Categoria</b>	Fester
<b>Any</b>	2015
<b>Quantitat</b>	700,00

Figura 70: Carregar Assentament

En aquest moment que s'ha carregat la fulla als socis, tots els socis de categoria "Fester" tindran l'assentament al seu deure carregat.

- **Tancar any:** Tancarà l'any, de forma que no es podran crear assentaments ni pagaments, l'any passarà a passat.

### 8.2.4.2 Categories

Des del menú Categories, donant-li al botó crear podrem crear distintes categories de tipus senior o infantil (opció senior desmarcada).

<b>Categoria</b>	<input type="text" value="Fester"/>																				
<b>Major</b>	<input checked="" type="checkbox"/>																				
<b>Socis</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">Nom</th> <th style="width: 25%;">Mòbil</th> <th style="width: 25%;">Correu electrònic</th> <th style="width: 25%;">Categoria</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td></tr> </tbody> </table>	Nom	Mòbil	Correu electrònic	Categoria																
Nom	Mòbil	Correu electrònic	Categoria																		

Figura 71: Formulari Categoria

**Categoria**

**Major**

**Xiquets**

Nom	Categoria	Pare

Figura 72: Formulari Categoria

Ens apareix una llista dels socis/xiquets pertanyents a eixa categoria. La llista no és editable des d'aquesta vista.

### 8.2.4.3 Assentament Comptable

Des d'aquesta vista no serà possible crear assentaments, tant sols ho farem des de l'any. El que si que podem fer serà carregar assentaments a socis.

**Ref** 004

**Concepte** Semana 1

**Tipus** Haver

**Categoria**

**Any** 2015

**Quantitat** 0,00

Figura 73: Formulari Assentament Comptable

### 8.2.4.4 Usuaris

Des del menú Usuaris li donem a crear.



Figura 74: Creació Usuari

Tenim l'opció de marcar si es soci o no, per a poder recollir les dades del soci relacionat i després poder enllaçar des de la vista de soci a aquest usuari.

Com a usuari Manager podem canviar la contrassenya dels usuaris des del desplegable superior:



Figura 75: Canviar contrassenya a usuari

### 8.2.4.5 Grups

Des del tree de grups escollim el grup el qual volem editar per a afegir nous usuaris al grup. En aquest cas triem el grup User:

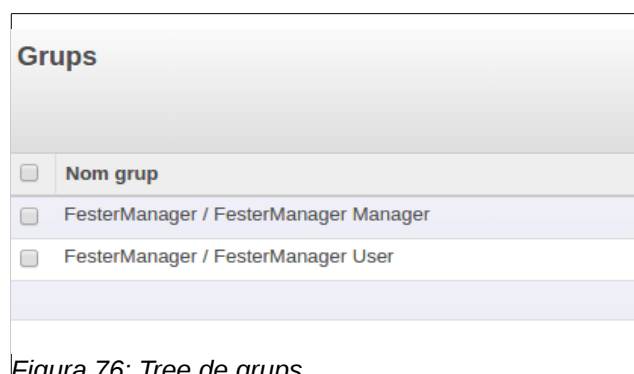


Figura 76: Tree de grups

Una vegada dins del grup li donem a editar, i a l'opció “add an item” per a afegir usuaris al grup:

Socio	Usuari	Latest connection
-------	--------	-------------------

[Add an item](#)

Figura 77: Afegir usuaris a grup

Apareix la següent finestra emergent, per a escollir els usuaris que afegirem al grup:

<input type="checkbox"/>	Socio	Usuari	Latest connection
<input type="checkbox"/>	Administrator	admin	24/08/2015
<input type="checkbox"/>	Manager	manager@exemple.com	24/08/2015
<input checked="" type="checkbox"/>	Manuel López	manuellopez@exemple.com	24/08/2015

or [Cancel](#)

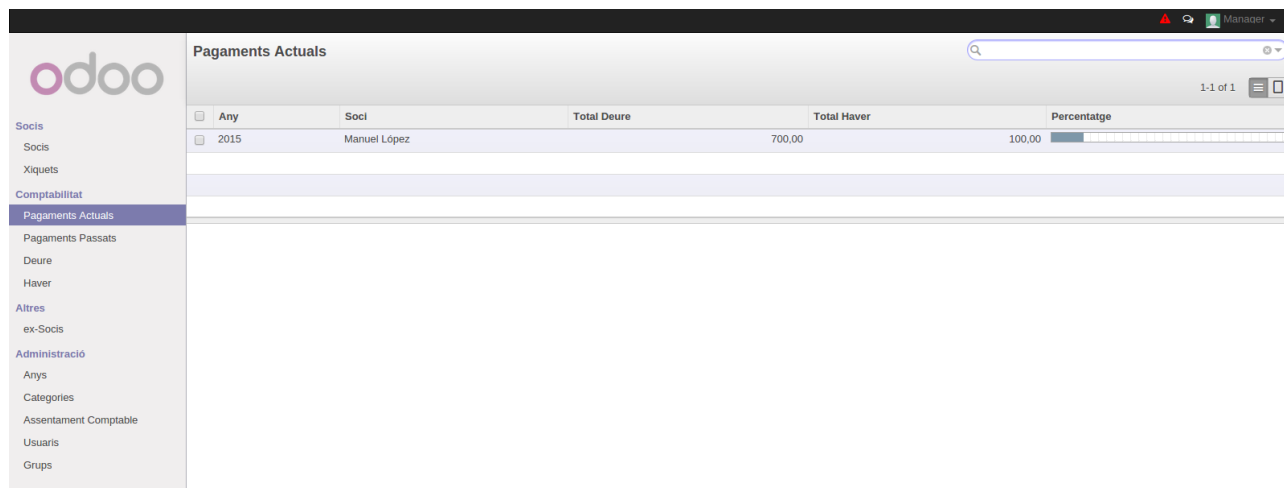
Figura 78: Afegir usuaris a grup

Li donarem al botó select.

## 8.3 Comptabilitat

### 8.3.1 Pagaments actuals

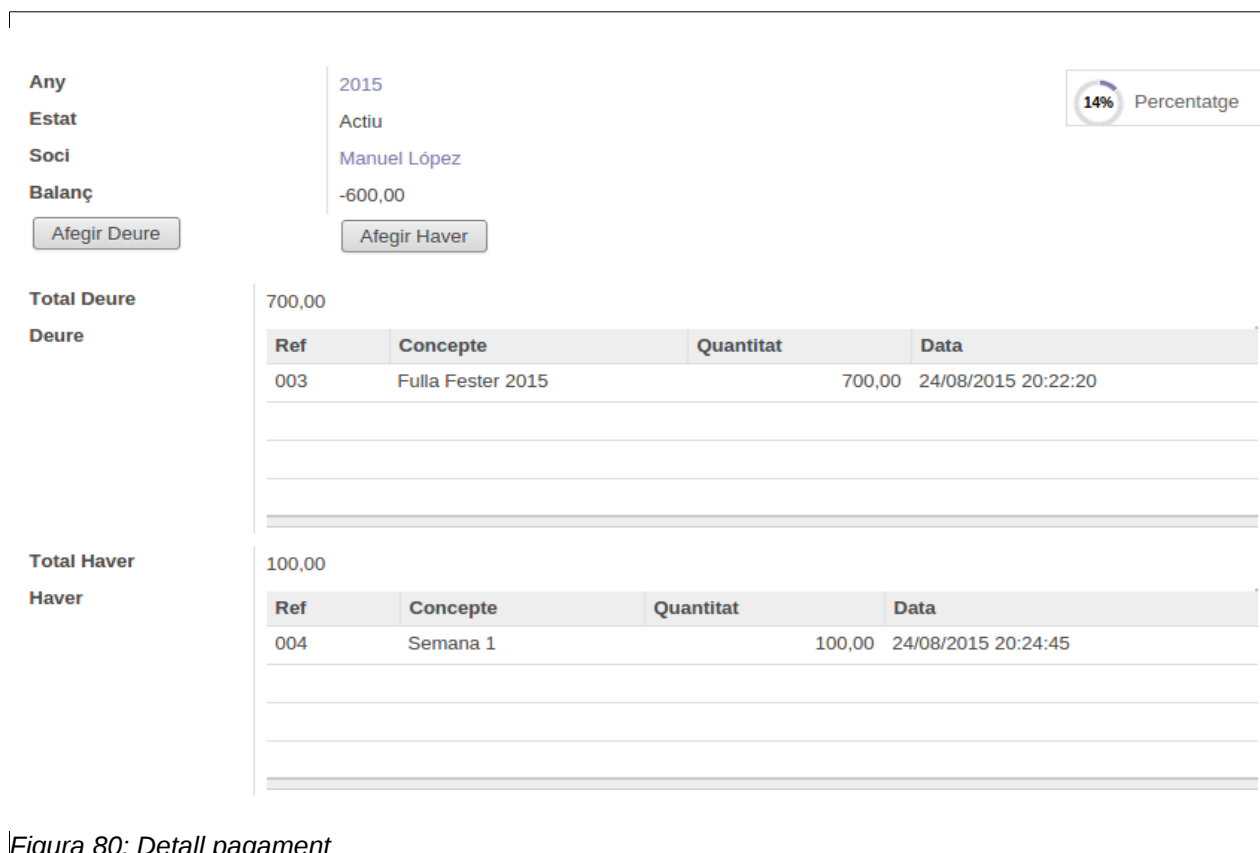
En aquest menú trobarem els pagaments actuals en un tree:



Any	Soci	Total Deure	Total Haver	Percentatge
2015	Manuel López	700,00	100,00	14%

Figura 79: Pagaments actuals

Si cliquem damunt del pagament podem vore'n el detall:



**Any:** 2015  
**Estat:** Actiu  
**Soci:** Manuel López  
**Balanç:** -600,00  
**14%** Percentatge

**Total Deure:** 700,00  
**Total Haver:** 100,00

Ref	Concepte	Quantitat	Data
003	Fulla Fester 2015	700,00	24/08/2015 20:22:20

Ref	Concepte	Quantitat	Data
004	Semana 1	100,00	24/08/2015 20:24:45

Figura 80: Detall pagament

Ací tenim l'opció mitjançant botó d'afegir al deure i al haver:

## Afegir Deure:

**Deure** ×

---

Any	2015
Soci	Manuel López
Ref	003
Concepte	Fulla Fester 2015 <span style="float: right;">▼ ↗</span>
Quantitat	700,00
Data	

---

**Guardar** or **Descartar**

Figura 81: Afegir deure

## Afegir Haver:

**Haver** ×

---

Any	2015
Soci	Manuel López
Ref	004
Concepte	Semana 1 <span style="float: right;">▼ ↗</span>
Quantitat	100
Data	


---

**Guardar** or **Descartar**

Figura 82: Afegir Haver



Podem eliminar assentaments del deure o del haver per si ens hem enganyat, per a que siga possible hi ha que donar-li al botó editar (apareix poal de fem):

Ref	Concepte	Quantitat	Data	
003	Fulla Fester 2015	700,00	24/08/2015 20:22:20	

*Figura 83: Eliminar Assentament de deure o haver*

### 8.3.2 Pagaments passats

Igual que pagament actual pero no podem afegir assentaments nous al deure o haver. Tant sols podem borrar-los.

### 8.3.3 Deure i Haver

Tant sols seran de consulta de dades, i el màxim que hi podrem fer serà eliminar.

## 8.4 Manual d'Usuari

Per a fer login i canviar preferències d'usuari revisar el punt 8.2.1 i 8.2.2 del manual de Manager, ja que es el mateix.

Tindrem tres opcions al menú: *Pagament actual, pagaments passats i el meu perfil.*

- **Pagament actual i passats:** Sols consulta
- **El meu perfil:** Editable:

	<b>Nombre</b> Manuel López		
<b>Dirección</b>	Sant Jordi, 10	<b>Activo</b>	<input checked="" type="checkbox"/>
		<b>Categoría</b>	Fester
	Alcoi Provincia 46000	<b>Usuario</b>	Manuel López
	España	<b>Año</b>	2016
<b>Sitio web</b>	Por ejemplo, www.odoo.com	<b>Remanente Año Anterior</b>	-300,00
		<b>Teléfono</b>	123456789
		<b>Móvil</b>	987654321
		<b>Email</b>	manuellopez@exemple.com

Notas internas Pago Actual Pagos Pasados Niños

Poner una nota interna...

Figura 84: Formulari Soci per a usuari

Es poden editar totes les dades a excepció de *l'usuari, la categoria i actiu*. El que si que podem fer es modificar les preferències de l'usuari fent clic al damunt, en aquest cas damunt de *Manuel López i accedirem a la següent pantalla:*

	<b>Nombre</b> <b>Manuel López</b>
	<b>Dirección de correo electrónico</b> <b>manuellopez@exemple.com</b>
<b>Teléfono</b>	<input type="text" value="123456789"/>
<b>Móvil</b>	<input type="text" value="987654321"/>
<b>Fax</b>	<input type="text" value="Por ejemplo, +32.81.81.37.10"/>

Figura 85: Formulari Usuari per a usuari

Ací podem editar les dades del nostre usuari.

## 9. Futures ampliacions

Encara que l'aplicació compleix sobradament amb el seu propòsit, creguem que es poden fer futures ampliacions, que es deuen de considerar:

- Estudiar les possibilitats d'implementar les altres dues formes d'interactuar amb els pagaments que estan com a opció als requeriments:
  - Remeses mensuals: hi ha usuaris que han autoritzat que se'ls passe un càrrec mensual de determinat import. Com que açò suposa autorització bancària, etc, el que es pot fer es simular-ho i que:
    - Aparega mensualment (en la data determinada per l'administrador) un pagament per a cada usuari que ha habilitat el mode 'remesa'.
    - Que l'administrador pugui 'llançar' manualment la remesa en el moment que ell vulga (apreta el botó carregar remesa) i apareix el pagament a tots els usuaris que han habilitat el mode 'remesa'.
  - Transferència bancària: Per a això, l'aplicació hauria de poder llegir la informació des d'un full d'excel (que generarem des del comptador bancari de la filà).
- Possibilitat de realitzar una aplicació mòbil que treballi en paral·lel amb l'aplicació web per a consulta de dades.
- Estudiar la forma de dotar de més seguretat a l'aplicació, protecció de dades en xarxa.
- Migració a Odoo 9.
- Ampliar la instal·lació, integrant el mòdul de festermanager amb altres que gestionen aspectes com la comptabilitat, relacionant els diferents mòduls per a treballar amb les dades que genera el nostre mòdul.

## 10. Conclusions

Una vegada realitzat el desenvolupament de l'aplicació estem en condicions d'opinar sobre les tecnologies emprades per a portar endavant aquest projecte en el que, a més d'haver après un llenguatge de programació i el funcionament d'un *ERP* complet com és *Odoo*, tenim la satisfacció de veure'l funcionant amb els resultats esperats.

En primer lloc començaré parlant de les meues impressions amb el llenguatge de programació utilitzat en el projecte. En *Python* trobem un llenguatge eficaç per a la programació àgil d'aplicacions. A més de ser apte per a programes de considerable tamany i inclús de gran demanda. Resulta sorprenent la seua velocitat d'execució tot i ser un llenguatge interpretat. La seua curva d'aprenentatge es realment curta en front d'altres llenguatges pareguts, un programador amb una mínima experiència en programació orientada a objectes pot dominar-lo en poc de temps.

Per altra banda està l'agrupació de tecnologies que componen l'entorn del programa *Odoo*. Tenim davant de nosaltres un *ERP* amb unes enormes possibilitats. Trobem una gran carència en les opcions de programació de la interfície però a canvi està a la nostra disposició una integració perfecta amb un llenguatge molt potent que ens permet desenvolupar mòduls en temps relativament curt.

En quant al mòdul desenvolupat. Compleix a la perfecció la seua funció. Ha sigut una grata experiència el poder escoltar i analitzar les necessitats del client, i portar a cap el disseny i la implementació de l'aplicació. Personalment m'ha servit per a ampliar l'experiència amb *Odoo*, una ferramenta molt extesa al món informàtic i en la qual he pogut treballar durant aquestos mesos a les pràctiques d'empresa.

En definitiva, m'ha servit per a millorar en un nou llenguatge de programació desconegut fins ara per a mí i en una eina molt actual en quant a desenvolupament d'aplicacions, que m'ha obert les portes al món laboral. Ha sigut una gran oportunitat per a mí l'elecció i elaboració d'aquest treball de fí de grau.

## 11. Bibliografia

### Libres:

- Daniel Reis, “Odoo Development Essentials”, 2015

### Llocs d'Internet:

- “Python Documentation Index”, <https://www.python.org/doc/>
- “Odoo Technical Documentation”, <https://www.odoo.com/documentation/8.0/>
- “Portal en Español sobre PostgreSQL”, <http://www.postgresql.org.es/>
- “Odoo New API Guideline”, <http://odoo-new-api-guide-line.readthedocs.org/>
- “Wikipedia”, <https://es.wikipedia.org/>
- “NSU, no solo usabilidad: Revista sobre personas, diseño y tecnología”, <http://www.nosolousabilidad.com/manual/>
- “Bitnami Cloud Hosting”, <https://bitnami.com/>
- “Planificador de Proyectos”, <http://www.tomsplanner.es/>

### Software:

- Diagrames: *Giffly Diagrams*
- Editor de Text: *Sublime Text 3*
- Control de Versions: *Gitlab*
- Programa Gestor de base de dades: *PGAdmin 3*

