

# PROJECTE FINAL DE MÀSTER

## SERVIDOR D'ESCRITORI REMOT



### Administració de Sistemes i Xarxes

Autor: Alberto Ponseti Alonso

Consultor: Miguel Martín Mateo.

Consultor Extern: Ricard Iravedra Abellò

Barcelona, Juny de 2012.





## Copyright

---

### Autor

Alberto Ponseti Alonso.

Per a qualsevol comentari o suggeriment contactar amb [albert.ponseti@gmail.com](mailto:albert.ponseti@gmail.com)

### Publicació

Contingut actualitzat el 18 de juny de 2012.

Document creat i basat en OpenOffice.org 3.2.0.

### Llicència



La memòria del projecte "Servidor d'escriptori remot" està sota una [licència de Creative Commons Reconocimiento-CompartirIgual 3.0 Espanya](https://creativecommons.org/licenses/by-sa/3.0/es/).



## Resum

---

El projecte "Servidor d'escriptori remot" té com a propòsit principal realitzar una prova de concepte que permeti la connexió remota a escriptoris de diferents màquines de forma visual, utilitzant programari lliure existent.

La idea principal del projecte va sorgir de la creixent fauna de dispositius informàtics que diàriament tots nosaltres utilitzem. En aquest entorn de seguida sorgeix la necessitat de compartir la informació continguda en aquests dispositius, tant de programes com dades emmagatzemades, principalment de forma remota.

Amb aquesta idea i dins del marc del programari lliure, es va comprovar que existien solucions en quant a connexions remotes a escriptoris tals com FreeNX, XDCMP, RDesktop etc. les quals seràn la base del desenvolupament del projecte.

Per dur a terme aquesta idea i per facilitar el desenvolupament del projecte, s'ha dividit en dues parts.

La primera part està dedicada a obtenir un entorn completament funcional i llest per començar a ser usat per els usuaris. En aquest punt entra en joc el concepte d'appliance (descriu al llarg d'aquest document) que contindrà tot el software requerit, sistema operatiu inclòs, juntament amb el nostre projecte desenvolupat.

La segona part està dedicada a la generació d'un entorn web on es posarà de manifest la prova de concepte del servidor d'escriptori basat en la tecnologia NX representada en el programari lliure FreeNX. Finalitzada aquesta part s'integrarà dins del appliance.

Com a entrega final del projecte es presenta un liveCD amb tot el software generat llest per ser usat en qualsevol màquina.



# Índex de continguts

---

Resum.....	5
1. Introducció.....	9
1.1. Objectius.....	10
1.2. Estructura de la memòria del projecte.....	11
2. Planificació .....	12
2.1. Idea estructurada.....	12
2.1.1 Característiques.....	13
2.1.2. Programari.....	14
2.1.2. Interfícies Web.....	14
2.2. Estudi de viabilitat.....	15
2.2.1. Viabilitat econòmica.....	15
2.2.2. Viabilitat tècnica.....	16
2.2.3. Viabilitat legal.....	16
2.3. Metodologia a seguir.....	16
2.4. Planificació de tasques.....	18
3. Disseny.....	25
3.1. Arquitectura.....	25
3.2. Decisions tecnològiques.....	27
3.3. Persistència de dades.....	30
3.4. Prototip general de la aplicació web.....	32
4. Implementació.....	34

4.2. Programari.....	35
4.3. Mòduls de l'aplicació.....	36
5. Estudi de llicències.....	38
6. Resultats.....	42
6.1. Vista dels resultats.....	42
6.2. Anàlisi dels resultats.....	50
7. Conclusions.....	52
Bibliografia.....	55
Referències electròniques.....	56
Annex I: OpenTrends S.L.....	59
Annex II: Guia ràpida SuseStudio.....	61
AII.1. Com crear una nova appliance.....	62
AII.2. Configuració de l'appliance.....	63



## 1. Introducció

---

En el món tecnològic actual on hi ha tota mena de dispositius informàtics que permeten a la gent estar totalment connectada en qualsevol lloc i hora, neixen noves oportunitats de negoci amb petites i grans aplicacions. De fet, gestionar la gran quantitat de dades que es generen resulta ser un problema greu donat que al final acaben sent dispersades per tots aquests dispositius. Donades aquestes necessitats inicials, seria interessant poder accedir als diferents dispositius mitjançant la tecnologia d'escriptori remot.

En aquest punt, entra OpenTrends, empresa dedicada a la enginyeria de la informació que mitjançant l'aplicació de les últimes tendències en el desenvolupament i la utilització intensiva dels estàndards u solucions de software lliure planteja solucions innovadores per empreses i administracions públiques. (Veure l'annex I).

En el marc de col·laboració entre la UOC i l'empresa externa OpenTrends, es presenta la idea de projecte "Servidor d'escriptori remot" basat en administració de xarxes i sistemes operatius. Per part de la UOC es donarà suport en el plantejament i desenvolupament proveïnt els recursos tècnics com humans. Per altre banda, OpenTrends desenvoluparà la presa de requeriments i suport a la resolució de problemes conceptuals del projecte.

Un cop presentat el rol de cadascuna de les parts participants en el projecte descriurem amb una mica més de cura el plantejament inicial.

Es preten obtenir a partir de software lliure existent un sistema completament funcional i llest per començar a funcionar que ens permeti fer connexions remotes a escriptori. Per tal de facilitar la comprensió, caldrà desenvolupar una petita aplicació web que contingui un gestor d'escriptoris remots.

A partir d'aquestes premisses i les aportades per OpenTrends i tal com es descriu en aquest document, la planificació, el diseny i la implantació així com els resultats obtinguts al llarg de tot el procés.

La resolució d'aquest projecte ens permetrà posar en practica i treballar de forma real amb els conceptes adquirits al llarg del Màster en Programari Lliure, desde analitzar un problema complex fins proposar una solució estructurada a aquest problema. També ens permetrà aprendre a utilitzar els recursos de software lliure disponibles i a integrar-los en el desenvolupament de la nostre solució ja que una de les principals premisses del projecte es la utilització de software lliure tant en el desenvolupament com en la implantació de la solució.

## 1.1. Objectius

Dins del desenvolupament del projecte final es plantejen dos tipus d'objectius, en primer lloc els personals que són els que com a desenvolupador i estudiant de software lliure es vol assolir i en segon lloc els que es volen assolir en aquest projecte.

Els objectius personals que es volen assolir en la realització del projecte son els següents:

- Posar en pràctica els coneixements adquirits al llarg del màster de programari lliure.
- Aprendre a treballar amb les diferents eines de desenvolupament en l'entorn de programari lliure.
- Realitzar una documentació acurada de les diferents parts de projecte, es a dir, des de els requeriments fins a l'implantació de l'aplicació.

Els objectius del projecte que es volen assolir són els següents:

- Creació d'una prova de concepte d'escriptori remot utilitzant programari lliure existent.
- Demostracions de funcionament general del programari implicat utilitzant una interfície web.
- Garantir que no seran necessàries preparacions, instal·lacions ni configuracions del sistema prèvies.
- Posar les bases per poder permetre que l'aplicació pugui ser millorada per la comunitat d'usuaris que puguin estar interessats .
- Familiaritzar-se amb l'ús d'apliances, configurant el software que volem que estigui present, la forma de presentar-se (aspecte visual) i el format de sortida (liveCD, Virtual Machines, etc.)

## 1.2. Estructura de la memòria del projecte

A l'hora de realitzar una bona documentació d'un projecte, o en el nostre cas de la memòria final, cal tenir clar la informació que es vol transmetre i ser a la mesura del possible el més estructurat possible per facilitar la lectura a futurs usuaris o desenvolupadors del projecte. Cal plasmar en el document, les motivacions del projecte, la interacció amb el client amb la presa de requeriments, justificacions de les solucions tecnològiques, metodologia de treball etc. En aquest document i plasmem aquestes idees en els següents apartats:

- **Resum**

Resum global del projecte i descripció del que es veurà al llarg del document. Es una part important ja que es la primera presa de contacte amb la documentació i permet al lector saber de que tracta tot el projecte i si vol o no continuar llegint.

- **Introducció**

Introducció amb les motivacions i objectius. És un apartat on ja s'explica una mica més acuradament de que va el projecte i els diferents participants en ell.

- **Planificació**

Requisits obtinguts per a la correcte funcionalitat de projecte i planificació de la resolució d'aquests.

- **Disseny**

Disseny de l'aplicació a nivell tècnic. Es proposa la solució a les diferents tasques de cara als desenvolupadors del projecte.

- **Implementació**

Resumeix com s'ha implementat la solució.

- **Estudi de llicències**

Estudi de diferent programari per ampliar l'aplicació d'acord amb la seva llicència.

- **Resultats**

Explicació dels resultats obtinguts mostrant èxits i fracassos.

- **Conclusions**

Valoració final del projecte un cop finalitzat.

- **Bibliografia**

Describeu de forma estructurada les fonts d'informació trobades.

- **Annexos**

Informació addicional que es vol introduir com a informació addicional.

## 2. Planificació

---

En aquest apartat es descriu la idea estructurada del projecte mostrant els requeriments. Un cop vista la estructura, estudiarem la viabilitat de dur a terme el projecte per decidir si es viable o no tirar endavant el desenvolupament o si per el contrari, s'ha de descartar per algun motiu concret. Si considerem que el projecte es viable seguirem amb la metodologia que usarem per desenvolupar els requeriments i finalment planificarem el temps de dur a terme les diferents tasques que desencadenaran el temps global per finalitzar el projecte.

### 2.1. Idea estructurada

La idea principal del projecte es la creació d'una prova de concepte d'escriptori remot utilitzant productes de programari lliure existents. Com que un dels principals requisits del projecte es poder fer demostracions del funcionament general del programari implicat sense tenir de fer preparacions, instal·lacions ni configuracions del sistema prèvies, ens adaptarem a un nou concepte de treball, es a dir, usarem un **appliance**.

Els appliances de software, son combinacions pre-configurades d'una aplicació, amb un software intermig i un sistema operatiu integrats en una sola imatge. Aquesta imatge esta feta a mida per executar-se en plataformes estàndards.

OpenTrends proposa per aquest projecte usar la eina de creació d'appliances SUSEstudio. L'appliance tindrà com a software intermig Firefox kiosk i com a sistema operatiu openSUSE 11.4 Minimal X.

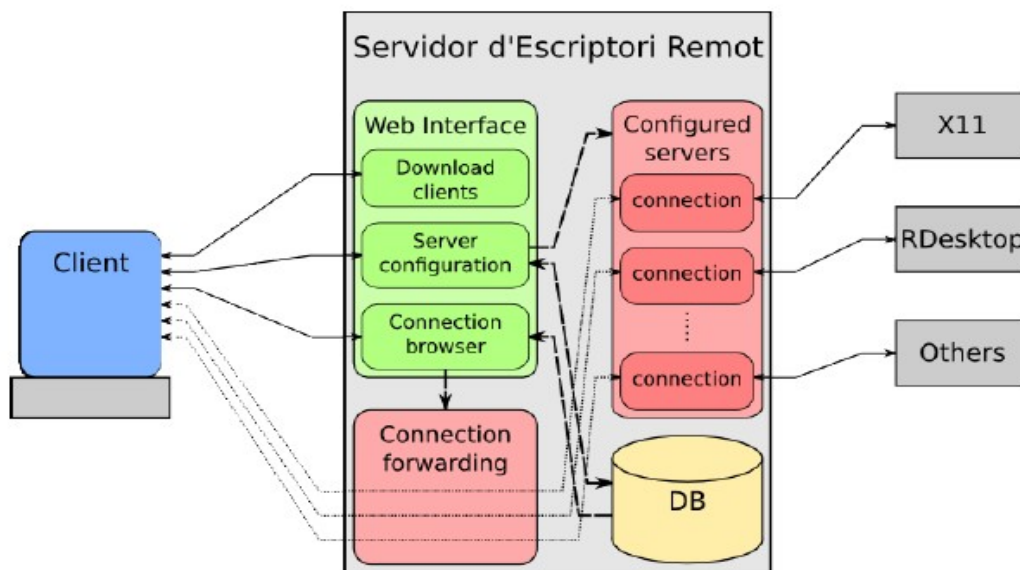
El *appliance* serà capaç de gestionar i fer *forwarding* de connexions a escriptoris remots i cal que suporti com a mínim els protocols següents:

- NX  
El protocol NX realitza connexions remotes X11 molt ràpides que permet als usuaris accedir a escriptoris remots de Linux o Unix inclus sota connexions lentes com les realitzades per modem.
- X11  
El protocol X11 permet la interacció gàfica en xarxa entre un usuari i un o més ordinadors.
- RDP  
El protocol RDP permet la comunicació en la execució d'una aplicació entre un terminal (mostrant la informació processada que rep del servidor) i un servidor Windows. Podem dir que aquest protocol comunica Linux i Windows.

Com es dedueix, dins de l'appliance hi hauran dos sistemes diferenciats un que donara servei als clients i un altre que s'encarregarà de la gestió dels diferents protocols esmentats.

Com podem veure es tracta d'una arquitectura client - servidor un model d'aplicació distribuïda en que les tasques es reparteixen entre els proveïdors de recursos o serveis anomenats servidors i els demandants anomenats clients. Així doncs, un client realitza peticions i el servidor li dona resposta.

La següent figura mostra el esquema conceptual de les funcionalitats generals que el projecte ha d'assolir:



**Figura 1. Esquema conceptual.**

### 2.1.1 Característiques

Com a característiques principals del sistema tindrem les següents les quals ens obligen a determinades desicions de disseny:

- Permetre la connexió mitjançant escriptori remot al propi *appliance* utilitzant el servidor FreeNX
- Habilitat per fer de *proxy* dels serveis d'escriptori remot suportats així com de fer *forwarding* de les seves connexions.
- Habilitat per configurar el sistema via WEB.

- Habilitat per publicar les connexions disponibles via WEB.
- Pàgina de descàrregues dels clients necessaris.
- Gestió d'usuaris de les interfícies WEB.

### **2.1.2. Programari**

Els productes de programari lliure que compondran el Servidor d'escriptori remot, a més a més dels productes inclosos en la base *openSUSE 11.4 Minimal X*, seran com a mínim:

- FreeNX (en la seva última versió estable): Servidor d'una implementació lliure del protocol NX.
- Apache2: Servidor HTTP.
- MySQL: Sistema gestor de bases de dades SQL.
- Firefox: Navegador WEB.
- Clients d'escriptori remot: Clients necessaris per als protocols suportats.

Es poden incloure altres programaris i/o servidors d'escriptori remot per tal de suportar més protocols i/o serveis.

### **2.1.2. Interfícies Web**

El *appliance* tindrà les següents interfícies WEB disponibles. Aquestes han de ser accessibles tant remotament com en local des del propi *appliance*. També caldrà tenir un sistema de gestió d'usuaris per tal de poder regular l'accés a les interfícies:

- Configuració del sistema, ha de permetre configurar els servidors gràfics remots i les seves connexions.
- Publicació de les connexions configurades en forma d'enllaç o d'instruccions de configuració per als clients.

- Pàgina que permet descarregar els diferents clients necessaris per la connexió als escriptoris remots suportats.
- Demostració per tal de guiar als usuaris en la configuració i utilització del productes instal·lats. Ha de servir com a pàgina de benvinguda, menú principal i/o demostració i cal que porti els logos d'openTrends i de la UOC.

En resum, el treball que es realitzarà és la creació d'un *appliance* en format *LiveCD* que inclogui tot el programari necessari instal·lat i pre-configurat. Ha de tenir les interfícies WEB de configuració, de publicació, de descàrrega de clients i de demostració esmentades.

## 2.2. Estudi de viabilitat

A continuació es mostra un petit estudi de viabilitat del projecte, analitzant aspectes econòmics, tècnics i legals.

### 2.2.1. Viabilitat econòmica

Per dur a terme el projecte, no hi ha un pressupost establert però si una durada màxima de tres mesos. Donada la situació econòmica, es requereix el mínim de recursos per finalitzar amb èxit el projecte. Els recursos inicialment proposats son:

- Recursos humans  
Comptarem amb la dedicació durant 3 mesos d'un analista programador, que s'encarregarà d'implementar el projecte. També hi haurà l'intervenció d'un cap de projecte, que s'encarregarà de interactuar amb el client. També planificarà i gestionarà les tasques del projecte.
- Eines de treball  
Només s'usarà programari lliure, per tant el cost en llicències serà nul. També s'usarà el maquinari actual de l'empresa.
- Altres  
No es contemples altres despeses especials, en quant sigui necessari una instal·lació en un espai de hosting s'estudiara el cost. Per ara només es farà entregues en LiveCDs.

**El projecte es perfectament viable econòmicament parlant.**

### 2.2.2. Viabilitat tècnica

El principal problema que podem trobar en l'apartat tècnic, es la desconeixença d'algun programari a utilitzar. Això pot retardar la realització d'alguna de les tasques programades. A pesar de tot, com que comptem amb una ampla experiència en altres tecnologies Apache o MySQL, poden compensar les petites desviacions de temps.

**Podem concloure que el projecte és viable tècnicament.**

### 2.2.3. Viabilitat legal

Tenint en compte que el projecte esta completament desenvolupat en programari lliure, només caldrà tenir en compte les limitacions que poden tenir certes clàusules entre els diferents programaris que hi haurà en l'entrega final.

Com que aquest tema te una complexitat elevada, es dedicarà un cert temps del projecte a estudiar possibles incompatibilitats de cara a l'entrega final, eliminant tot aquell software incompatible.

**Així doncs, el projecte és viable legalment.**

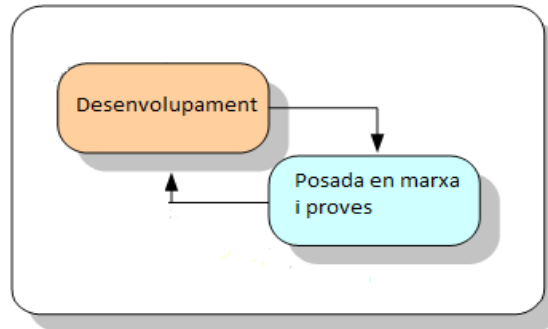
## 2.3. Metodologia a seguir

En el desenvolupament del projecte usarem dos tipus de metodologia de treball, una per crear i actualitzar les dades de l'appliance i un altre per generar la resta del projecte. Per la resta del projecte caldrà aplicar una metodologia de treball standart ja que es tracta d'un projecte de desenvolupament basat en software lliure.

Per crear el *appliance* cal seguir una metodologia que ens ajudi a assolir les fites d'integrar i configurar tot el programari previst en la distribució a mida. Per tant proposem usar una metodologia iterativa per al desenvolupament de l'aplicació ja que aquesta consisteix en anar fent canvis i podrem veure el resultat de forma immediata sense tenir que esperar a tenir altres integracions.

A continuació es mostra un esquema conceptual que ens ajudarà a entendre aquesta metodologia iterativa:





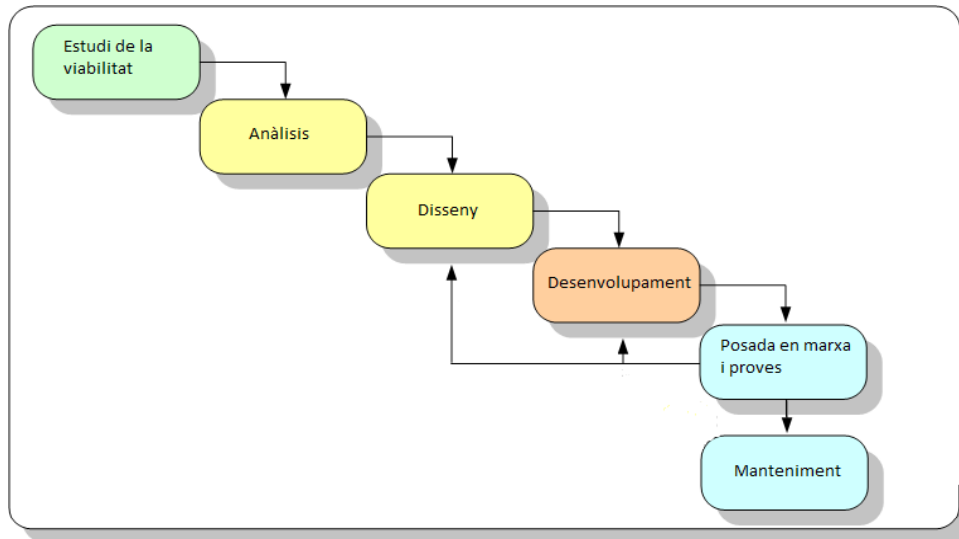
**Figura 2. Metodologia iterativa**

Així doncs dins de la metodologia iterativa seguirem els passos següents per la integració del software dins de l'appliance:

1. Crear un *appliance* en format *LiveCD* amb el SUSEStudio (template *openSUSE 11.4 Minimal X*).
2. Baixar el *LiveCD* i fer-lo funcionar en local amb un hipervisor (usarem VirtualBox).
3. Instal·lar i/o configurar un dels productes al *LiveCD* mentre està funcionant a l'entorn de l'hipervisor.
4. Quan el resultat sigui satisfactori, cal guardar o apuntar els canvis que hi han hagut. Es pot fer una llista de fitxers, un arxíu comprimit (tar) amb els fitxers i les seves rutes absolutes, un dump (en el cas del MySQL) o simplement apuntar què i com ha canviat (per exemple alguna configuració senzilla).
5. Extreure aquesta informació del *LiveCD* de forma que es pugui incloure al *appliance* que tenim al SUSEStudio.
6. Anar al SUSEStudio afegir els canvis i regenerar el *LiveCD* creant una nova versió.
7. **Tornar al punt 2, comprovar que els canvis són correctes i començar el procés de nou amb un altre dels productes que es volen incloure.**

Un cop descrita la metodologia empleada per desenvolupar l'appliance, proposem ara una metodologia per la realització del projecte, que en el nostre cas serà en cascada amb retroalimentació. Seleccionem aquest tipus de metodologia per que la nostre aplicació en un principi es basarà en el protocol NX per dur a terme les connexions a escriptori remot. Però si més endavant es vol incorporar un altre protocol, no farà falta que tornem a fer tot els passos de desenvolupament sino saltar cap a un estat anterior de diseny o desenvolupament que ens permeti introduir el nou protocol.

Així doncs, la figura següent il·lustra aquesta metodologia en cascada que aplicarem al llarg del projecte:



**Figura 3. Cicle de vida del projecte.**

Finalment, per poder tenir un clar concepte de de les diferents tasques a realitzar es recomana dur a terme una comunicació fluida amb els analistes de OpenTrends que s'encarregaran de obtenir feed-back del client final. Fer una planificació semanal, quincenal o mensual de reunions per parlar de les fites assolides i de la posta en comú de la informació acumulada, es una activitat molt recomendada (però no obligatoria).

## 2.4. Planificació de tasques

Per a dur a terme el projecte es presenta a continuació una planificació de tasques on s'especifiquen quatre fases, en les quals es realitzara el projecte. Al finalitzar aquestes fases, es procedira a l'entrega de resultats amb la documentació necessària. Aquesta es la planificació de treball a realitzar:

- **Fase 1:** Preparació del l'entorn del client (7 dies).
  - Crear un *LiveCD* que engegui per defecte el Firefox a pantalla completa, sense pestanyes ni missatges d'inici, amb l'objectiu de familiaritzar-se amb la creació d'*appliances* (*Firefox Kiosk*).

- Modificar l'anterior *LiveCD* de forma que quedin instal·lats i preconfigurats cadascun del programes llistats anteriorment.

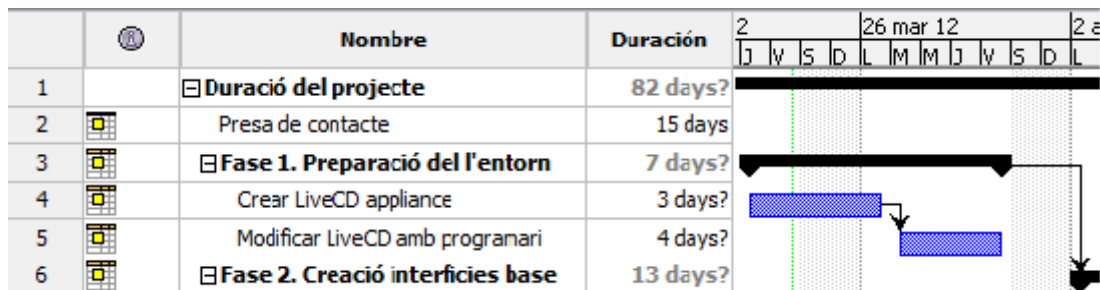


Figura 4. Planificació de la fase 1.

● **Fase 2:** Creació de les interfícies base de l'aplicació (13 dies).

- Fer una interfície WEB que permeti configurar les connexions a servidors gràfics remots (configuració dels serveis).
- Fer una interfície WEB on es publiquin les connexions disponibles.
- Fer una interfície WEB on es permeti la descàrrega dels clients necessaris per la connexió al serveis publicats.

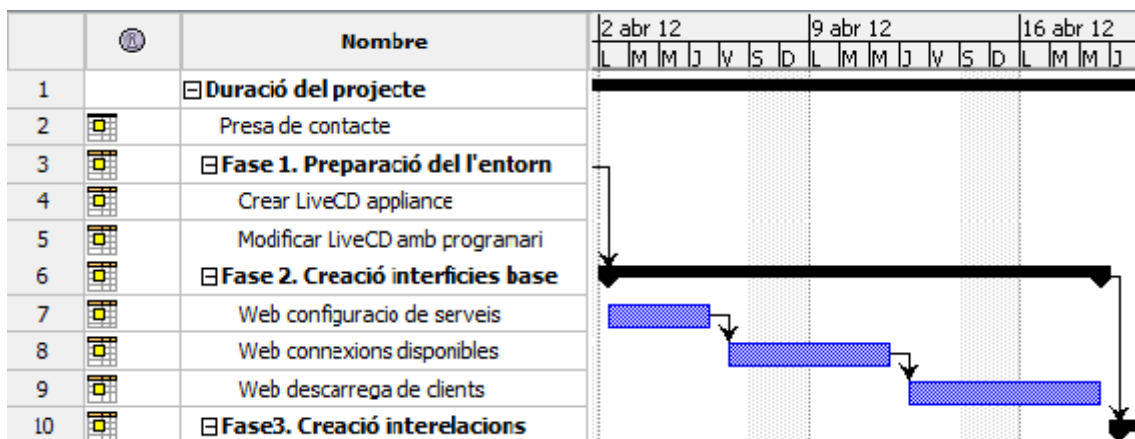


Figura 5. Planificació de la fase 2.

● **Fase 3:** Creació de les interaccions entre pantalles (20 dies).

- Fer una interfície WEB a mode de demostració que enllaci les interfícies anteriors, ha de tenir una explicació del funcionament del *appliance* per tal de guiar als usuaris en la configuració i utilització dels productes, ha de ser la pàgina inicial del Firefox i ha d'incloure els logos de la UOC i OpenTrends.

- Integrar totes les interfícies WEB anteriors al *appliance* incloent una gestió d'usuaris senzilla.

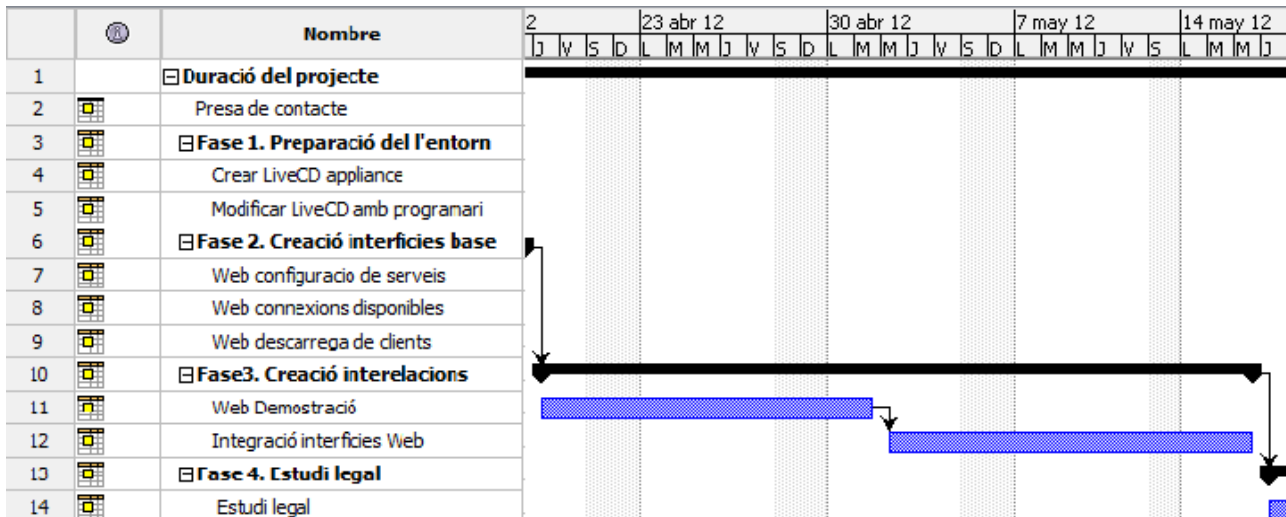


Figura 6. Planificació de la fase 3.

- **Fase 4:** Estudi legal bàsic de llicències (11 dies).

- Fer un estudi legal bàsic sobre les llicències del programari inclòs al *LiveCD* amb l'objectiu de detectar incompatibilitats entre les llicències dels productes instal·lats.

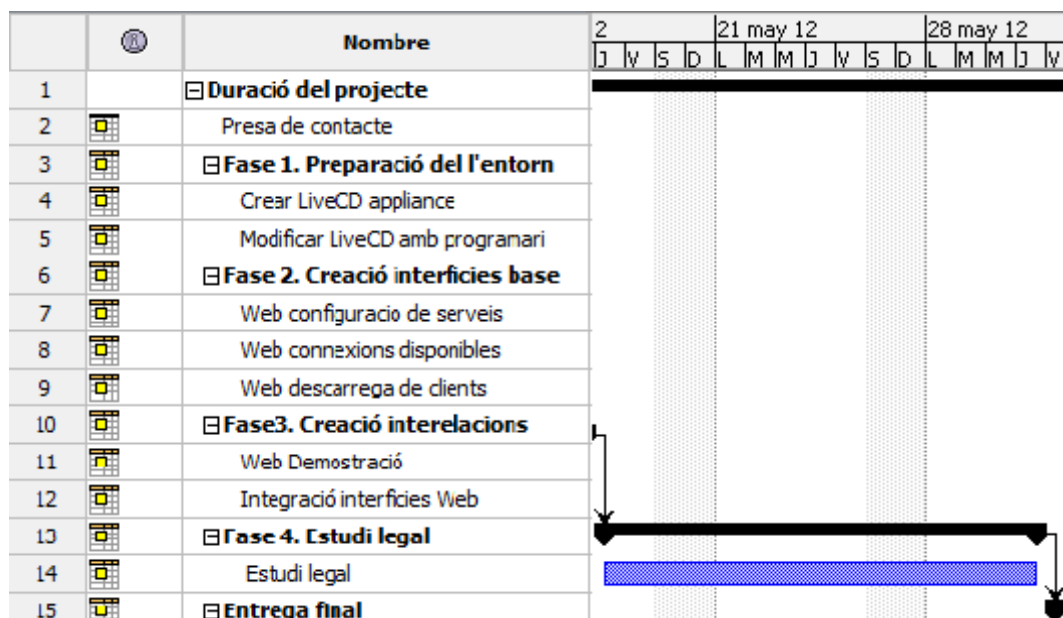


Figura 7. Planificació de la fase 4.

Un cop finalitzada l'ultima fase, es procedirà a fer l'entrega final, que consta de :

- **Entrega final:** Productes i documentació (16 dies).
  - LiveCD amb el Firefox a pantalla completa a l'inici (*Kiosk appliance*).
  - LiveCD amb tot el programari implicat en el Servidor d'escriptori remot instal·lat i pre-configurat. Ha d'incloure totes les interfícies WEB esmentades i ha de poder funcionar de forma autònoma.
  - Documentació del procés d'instal·lació i configuració.

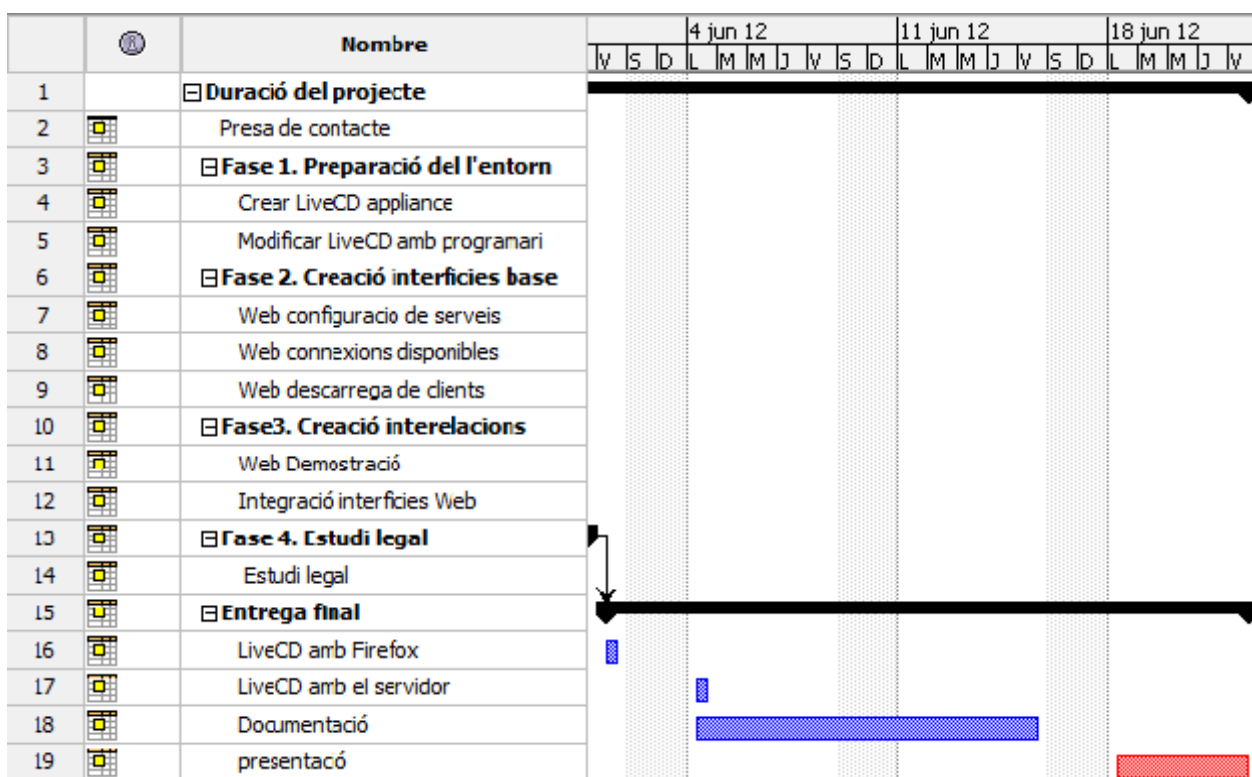


Figura 8. Planificació de la entrega final.

A continuació mostrem el diagrama de Grantt amb la planificació de tasques del projecte:

	📅	Nombre	Duració	Inicio	Terminado
1		☐ Duració del projecte	82 days?	1/03/12 8:00	22/06/12 17:00
2	📅	Presa de contacte	15 days	1/03/12 8:00	21/03/12 17:00
3	📅	☐ Fase 1. Preparació del l'entorn	7 days?	22/03/12 8:00	30/03/12 17:00
4	📅	Crear LiveCD appliance	3 days?	22/03/12 8:00	26/03/12 17:00
5	📅	Modificar LiveCD amb programari	4 days?	27/03/12 8:00	30/03/12 17:00
6	📅	☐ Fase 2. Creació interfícies base	13 days?	2/04/12 8:00	18/04/12 17:00
7	📅	Web configuració de serveis	4 days?	2/04/12 8:00	5/04/12 17:00
8	📅	Web connexions disponibles	4 days?	6/04/12 8:00	11/04/12 17:00
9	📅	Web descarrega de clients	5 days?	12/04/12 8:00	18/04/12 17:00
10	📅	☐ Fase3. Creació interrelacions	20 days?	19/04/12 8:00	16/05/12 17:00
11	📅	Web Demostració	9 days?	19/04/12 8:00	1/05/12 17:00
12	📅	Integració interfícies Web	11 days?	2/05/12 8:00	16/05/12 17:00
13	📅	☐ Fase 4. Estudi legal	11 days?	17/05/12 8:00	31/05/12 17:00
14	📅	Estudi legal	11 days?	17/05/12 8:00	31/05/12 17:00
15	📅	☐ Entrega final	16 days?	1/06/12 8:00	22/06/12 17:00
16	📅	LiveCD amb Firefox	1 day?	1/06/12 8:00	1/06/12 17:00
17	📅	LiveCD amb el servidor	1 day?	2/06/12 8:00	4/06/12 17:00
18	📅	Documentació	10 days?	4/06/12 8:00	15/06/12 17:00
19	📅	presentació	5 days?	18/06/12 8:00	22/06/12 17:00

**Figura 9. Planificació global de tasques.**

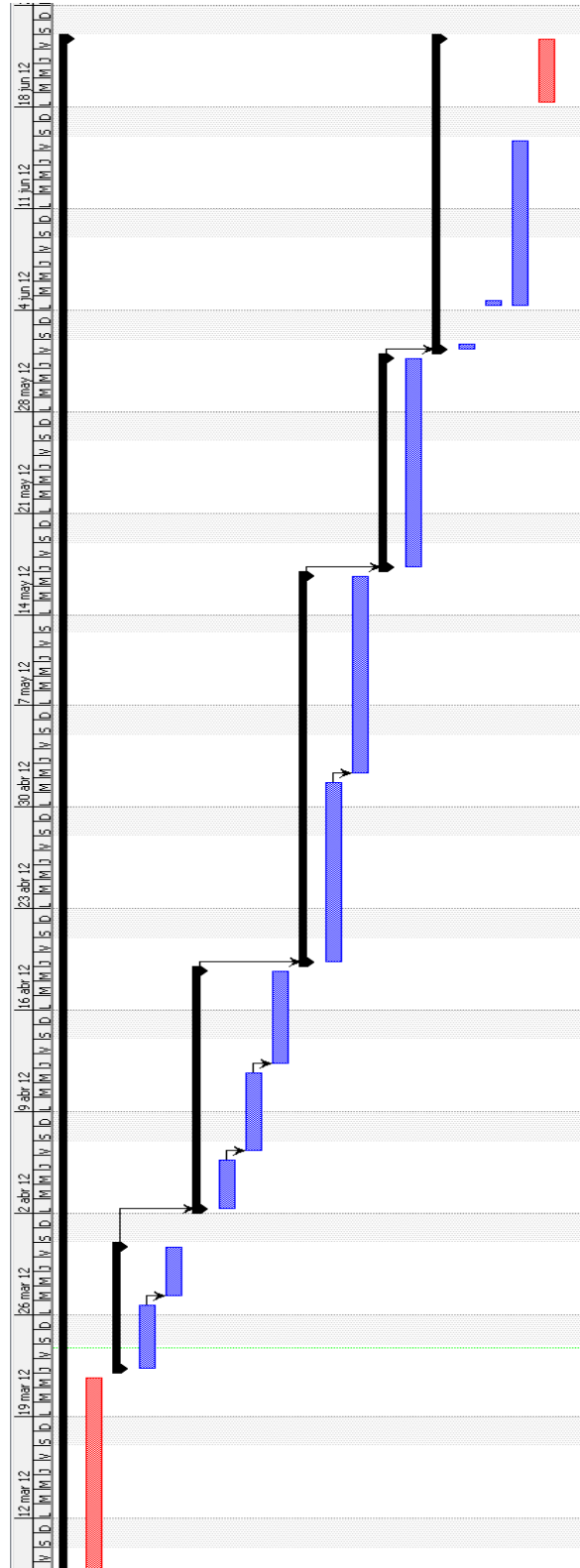


Figura 10. Diagrama de Grand amb el fluxe complet.

Per acabar, mostrem un diagrama de xarxa per mostrar l'estructura global de les tasques planificades:

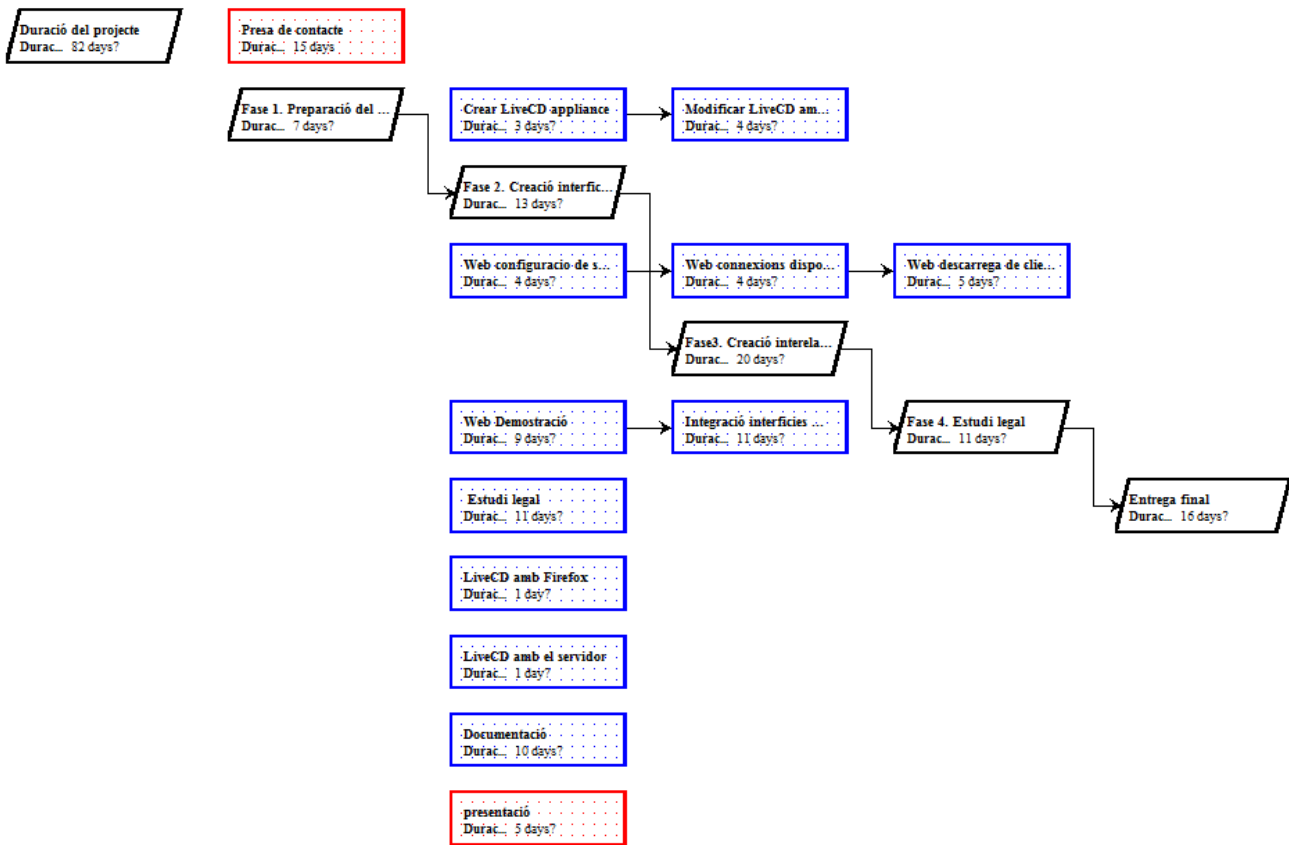


Figura 11. Diagrama de xarxa amb les tasques



### 3. Disseny

---

Donades les característiques del projecte en que el principal objectiu es la creació d'una prova de concepte d'escriptori remot utilitzant programari lliure en el qual s'ha de mostrar el resultat dels diferents connectors remots segons les necessitats dels diferents usuaris, es proposa el següent disseny de la aplicació basada en entorn web.

#### 3.1. Arquitectura

Aquest entorn web es basa en una l'arquitectura global que ve marcada per l'adopció del model client servidor i amb un model relacional de dades. A continuació es detallen les diferents vistes del l'arquitectura global que es basa en:

- **Arquitectura física**, on es reflexa com estan distribuïts físicament els diferents nodes que intervenen en el sistema.
- **Arquitectura lògica**, on es defineixen els components funcionals i la seva relació amb l'interior del sistema.

L'arquitectura física proposada per al nostre projecte es compon dels següents elements:

- **Servidor**

Ordinador remot que proveeix les dades sol·licitades per part dels navegadors dels clients. Per poder proveir les dades el servidor ha de ser capaç de donar servei a:

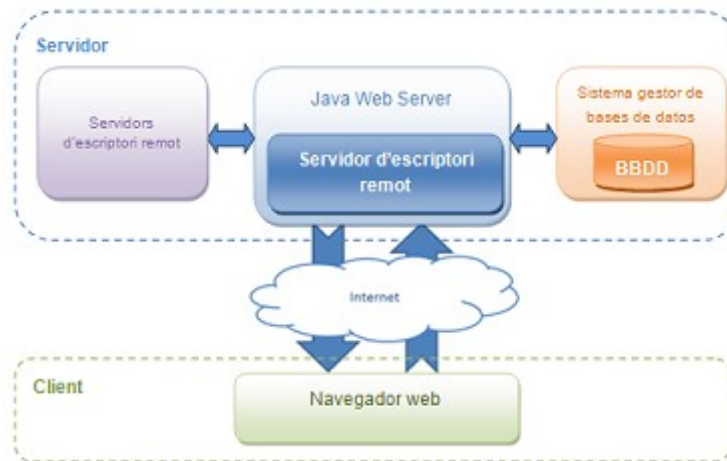
- Aplicacions Java
- Connectors a escriptoris remots
- Bases de dades relacionals

A més ha de estar dotat de sistemes de seguretat i tenir accés a Internet.

- **Client**

Ordinador amb navegador web i capacitat per executar aplicacions web.

En la següent figura recull la disposició de nodes en el sistema mostrant la composició del client i el servidor. Es pot observar que existeix un únic servidor central que presta el serveis de publicació Web, que actua com a servidor d'aplicacions. També esta representat el magatzem de la informació (base de dades) i els connectors a escriptoris remots. Per altre banda es mostra que els usuaris accediran a l'aplicació mitjançant un navegador web.



**Figura 12. Arquitectura física.**

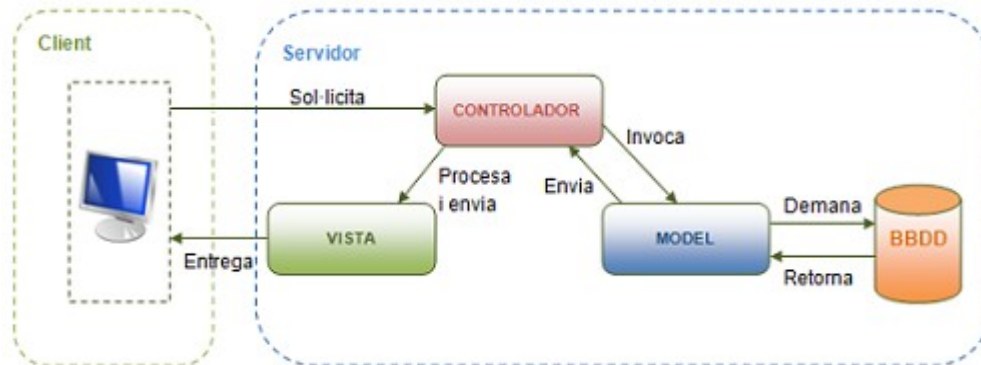
Tant el servidor de base de dades com el servidor web i els connectors a escriptori remot es necessari que es trobin dins de la mateixa xarxa per motius de rendiment i seguretat.

L'arquitectura lògica proposada es basa en el patró de disseny model vista controlador (MVC). Aquest patró es el encarregat de separar la lògica de negoci de la interface de usuari. Es el més utilitzat en la generació d'aplicacions Web ja que facilita la funcionalitat, el manteniment i l'escalabilitat del sistema.

El patró divideix les aplicacions en tres nivells d'abstracció:

- **Model:** Representa la lògica de negoci, s'encarrega d'accedir de forma directa a les dades actuant com a intermediari amb la base de dades.
- **Vista:** Es el encarregat de mostrar la informació al usuari de forma que sigui comprensible per aquests.
- **Controlador:** Es l'intermediari entre la vista i el model. S'encarrega de controlar les interaccions del usuari sol·licitant dades al model i tornant-los cap a la vista per que aquesta els presenti de nou al usuari.

En la figura següent es detalla la composició del MVC:



**Figura 13. Arquitectura lògica.**

El funcionament bàsic del patró MVC es resumeix en :

- L'usuari sol·licita una informació recollida com a petició.
- El controlador captura l'event i invoca les dades del model.
- El model s'encarrega d'interactuar amb la base de dades i recupera la informació sol·licitada. Aquesta es retornada cap al controlador.
- El controlador rep la informació i l'envia cap a la vista.
- La vista processa la informació i la entrega de forma gràfica al usuari.

### 3.2. Decisions tecnològiques

Un cop vista la arquitectura passem a explicar les decisions tecnològiques, justificant el per que de cada decisió presa en cada element de l'aplicació.

- **Llenguatge de desenvolupament**

Entre els diferents llenguatges de desenvolupament (C++, C#, JAVA, etc.) hem decidit utilitzar Java Enterprise Edition per que entre les seves avantatges destaquem la facilitat d'aprendre la seva sintaxis i les seves múltiples llibreries. A més, el seu extens us ens permet trobar una ampla documentació i exemples d'aplicacions.

Gràcies a la tecnologia JAVA, podem utilitzar el tipus de servidor específic que més en convingui, tant en servidors Java HTTP (Tomcat, Jboss, Weblogic, WebSphere, Apache) com a servidors de base de dades.

- **SGBD**

Entre els diferents SGBD (MySQL, Oracle, SQL Server, PostgreSQL, etc.) hem decidit utilitzar MySql. Els motius d'aquesta elecció ha sigut la gran integració del entorn de desenvolupament i la simplicitat de les eines de disseny incorporades. Un altre punt a favor de MySql es que es tracta de software lliure, i per requeriments del projecte descartarem altres softwares que no siguin d'aquest tipus.

- **Presentació**

En la presentació de les dades del projecte s'ha optat per l'estàndard HTML 1.1 i CSS 2.1, que permetrà crear les pàgines JSP que mostrarem a l'usuari. També amb usarem Javascript per a la programació d'interaccions avançades amb el client.

- **Connectors**

Un dels principals components de l'aplicació i el motiu de desenvolupament d'aquests son els connectors a escriptori remot. Usarem el protocol NX per els motius que es descriuen a continuació:

- Permet connexions remotes X11, el que permetrà la interacció gràfica per la xarxa possibilitant l'accés a escriptoris remots.
- Pot realitzar connexions ràpides inclús amb xarxes de baixa velocitat realitzant una compressió directa del protocol X11, es a dir, utilitza un algoritme de compressió més potent que altres tecnologies.
- La informació es enviada per SSH per el que les dades es transmeten xifrades incrementant la seguretat.
- NX usa mecanismes de caché que fa que el número de peticions al servidor disminueixi considerablement.
- El temps que triga en realitzar peticions i respostes es pràcticament nul.

- Usa mecanismes d'adaptació per ajustar-se a les característiques de la xarxa

Un cop vistes les característiques de la tecnologia NX anem a veure els diferents programes que l'utilitzen per les connexions remotes tant en clients com en servidors son els que es mostren a continuació:

- FreeNX: Servidor i client Linux.
- OpenNX: Client (Millora del FreeNX).
- Neatx: Servidor NX desenvolupat per google.
- X2go Servidor de terminal.

En el projecte ens decantem per utilitzar FreeNX ja que es el proposat per OpenTrends i per que ofereix diferents avantatges, entre elles que es fàcilment integrable en aplicacions JAVA. Tot seguit anem a conèixer una mica millor aquest software.

FreeNX es un software que implementa la tecnologia NX, es de codi obert amb llicència GPL i funciona sota plataformes Linux. La base de funcionament es SSH per el que tindrem que tenir instal·lat prèviament el programa ssh.

Com a principal objectiu es planteja que les aplicacions executades a través d'Internet han de tenir el mateix aspecte gràfic que quan s'executen en local, el qual el fa millor que altres eines tals com VNC.

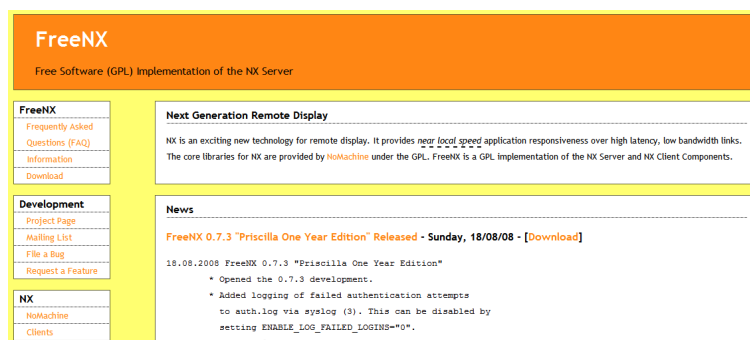


Figura 14. Pàgina principal de FreeNX.

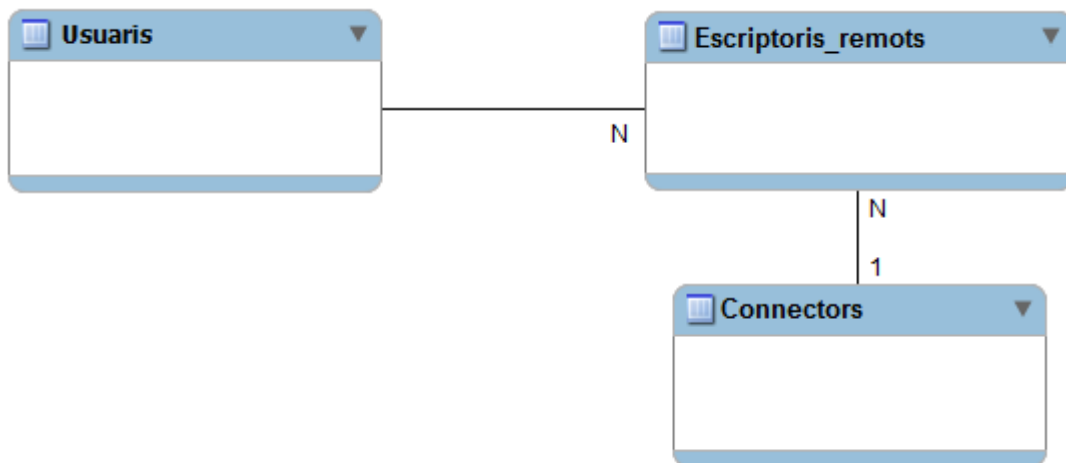
### 3.3. Persistència de dades

Per a al nostre projecte proposem el següent model relacional de base de dades, composta per tres taules principals:

Nom de la taula	Descripció
Usuaris	Conté les dades de registre del usuari.
Escriptoris_remots	Conté les dades dels escriptoris remots introduïts pels usuaris.
Connectors	Conte les dades dels diferents connectors d'escriptori.

**Taula 1. Taules de la base de dades.**

La relació que hi ha entre la taula usuaris es de que cada usuari tindrà definits N escriptoris remots però cada escriptori serà únic per a cada usuari, es a dir es una relació de 1 a N. Per altre banda tenim que cada escriptori remot definit pot tenir un connector, i un connector pot estar definit per N escriptoris remots, es una altre relació 1 a N. A continuació es mostra el diagrama de base de dades.



**Figura 15. Diagrama de base de dades.**

En les següents taules detallem els diferents camps de cada taula:

**Taula Usuaris.**

Columna	Tipus	Nullable	Clau	Descripció
Id	bigInt(10)	No	PK	Identificador d'usuari.
Nom	varchar(50)	No		Nom de l'usuari.
Cognoms	varchar(100)	No		Cognoms de l'usuari.
Nik_name	varchar(30)	No		Nik del usuari per entrar a l'aplicació.
password	varchar(30)	No		Password.
Email	varchar(200)	No		Email de l'usuari.
web	varchar(200)	Si		Web de l'usuari.
Pais	varchar(30)	Si		País de l'usuari.

**Taula 2. Taula usuaris.**

**Taula Escriptoris\_remots**

Columna	Tipus	Nullable	Clau	Descripció
id	bigInt(10)	No	PK	Identificador d'escriptori.
Usuari_id	bigInt(10)	No		Identificador d'usuari.
Nom_escriptori	varchar(200)	No		Nom de l'escriptori.
Login	varchar(30)	No		Login per accedir a l'escriptori.
Password	varchar(30)	No		Password de l'escriptori.
id_connector	bigInt(10)	No		Identificador de connector.
Descripció	varchar(500)	Si		Descripció de l'escriptori.
IP	varchar(30)	No		Ip de l'escriptori.

**Taula 3. Taula escriptoris remots.**

## Taula Connectors

Columna	Tipus	Nullable	Clau	Descripció
id	BigInt(10)	No	PK	Identificador de connector.
Nom_connector	varchar(200)	No		Nom del connector.
Descripció	varchar(500)	Si		Descripció del connector.
Enllaç	varchar(200)	No		Enllaç o ruta del connector.

**Taula 4. Taula connectors.**

### 3.4. Prototip general de la aplicació web

Un cop vist el disseny intern de la aplicació, es proposa el següent prototip de pantalla en el que es basaran les diferents mòduls que s'analitzaran en la implementació. Aquest prototip consta d'un menú lateral compost per:

Menú visible en cas de no estar registrat en l'aplicació:

- Inici  
Mostra informació general del funcionament de l'aplicació.
- Descarrega  
Descarrega dels connectors a escriptoris remots.

Menú visible al estar registrat en el qual apareixeran dues opcions més:

- Configuració  
Es configuren els escriptoris remots.
- Publicacions  
Escriptoris remots disponibles.

A sota d'aquest menú trobem el formulari de login, on l'usuari podrà accedir al seus escriptoris remots. També apareix l'opció per registrar-se.



En la part central es on es mostrarà la informació continguda en els diferents mòduls de l'aplicació. També inclourà el formulari de registre d'usuaris el qual permetrà accedir com a usuari registrat.

En la figura següent es mostra el prototip proposat que serà la base del desenvolupament de l'aplicació web:



**Figura 16. Prototip general de l'aplicació.**

## 4. Implementació

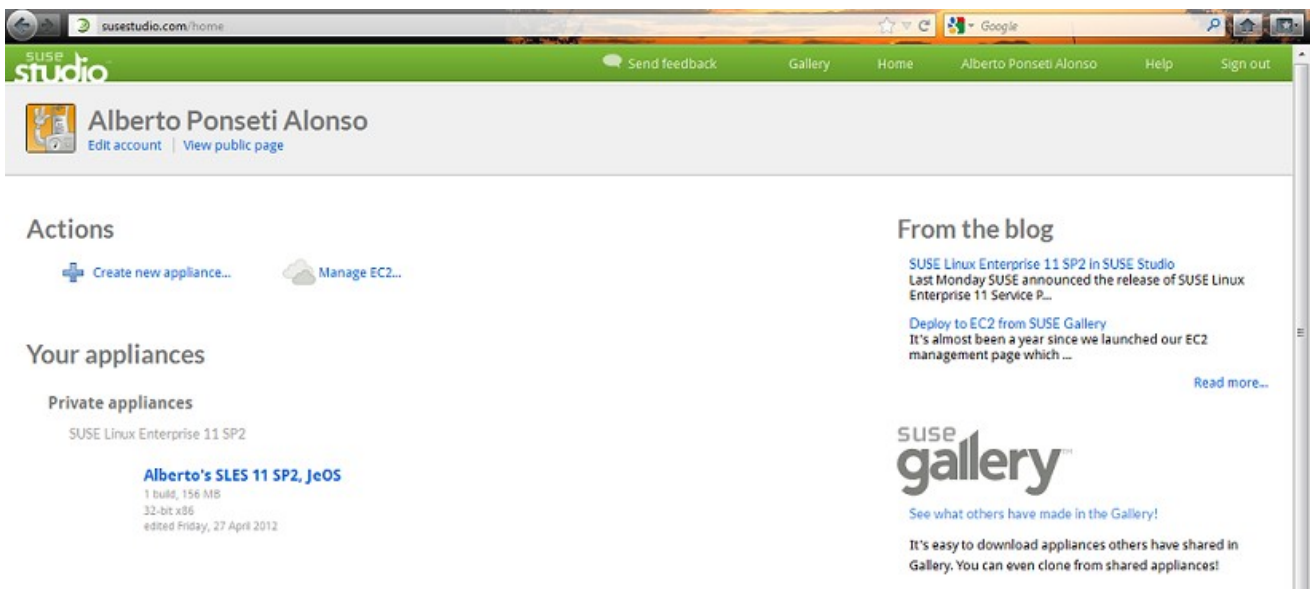
Un cop establert el disseny del projecte passem a la seva implementació. Com a punt de partida, establim l'entorn on s'executara el programari (appliance) seguidament descriurem el programari involucrat i finalment els mòduls implementats.

### 4.1. Appliance

Per facilitar les instal·lacions de cara a l'entrega, usarem un appliance que descrivim a continuació.

Per posar-nos en situació, direm que un appliance es una pila d'aplicacions que conté un sistema operatiu, software d'aplicació i les dependències necessàries amb la configuració i fitxers de dades per al correcte funcionament. Tot esta preinstal·lat, preintegrat i llest per a funcionar.

Així dons, el sistema estarà basat en un appliance de tipus Firefox kiosk, el qual desplegarà el navegador al iniciar-se. Per crear-lo cal utilitzar la eina de creació d'appliances SUSEStudio ([www.susestudio.com](http://www.susestudio.com)), que de forma fàcil i guiada en permet crear una imatge iso amb el programari desitjat. En l'apartat Annex II es parla de com crear un appliance i la forma de configurar-lo.



**Figura 17. Creació de appliance.**

La base que usarem sera un openSUSE 11.4 Minimal X que és un template d'appliance que hi ha al SUSEStudio. Aquest appliance, anirà incloent tot el programari necessari i la aplicació desenvolupada.

## 4.2. Programari

Un cop creat i configurat l'appliance afegirem dins d'aquest el següent programari usant la metodologia iterativa indicada en l'apartat de disseny. Aquest programari està format per:

- **Programari de servidor**

Inclou tot el programari que permet que s'executin aplicacions web, des de el servidor fins la base de dades. Instal·larem doncs:

- Apache2: Servidor HTTP.
  - Es troba a <http://httpd.apache.org/download.cgi>
  - Ha de contenir l'aplicació web de gestió i demostració dels escriptoris remots.
    - L'aplicació es concentrara en un únic fitxer ".war".
- MySQL: Sistema gestor de bases de dades SQL.
  - Es troba <http://dev.mysql.com/downloads/>
- Clients d'escriptori remot: Clients necessaris per als protocols suportats.
  - FreeNX (en la seva última versió 0.7.3): Servidor d'una implementació lliure del protocol NX.
    - Es troba a <http://freenx.berlios.de/download.php>
  - Altres connectors (X11, Rdesktop, etc.)

- **Programari en el client**

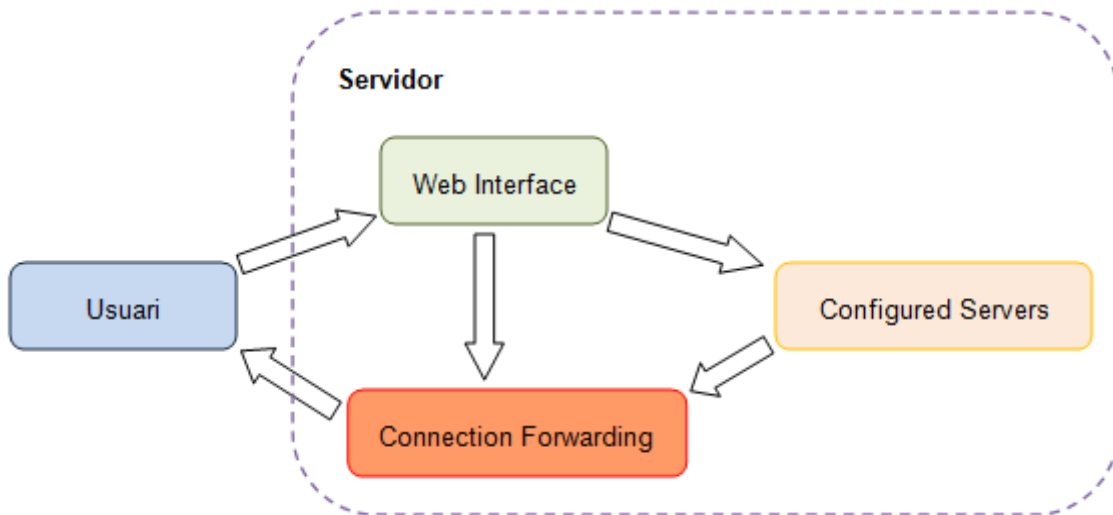
Com a programari de client usarem el navegador que ens permetrà accedir a la aplicació web desenvolupada.

- Firefox: Navegador WEB.
  - Es troba a <http://www.mozilla.org/>

El Mozilla Firefox és un navegador web lliure desenvolupat a partir del projecte Mozilla. L'objectiu del Firefox és desenvolupar un Mozilla més petit, lleuger i ràpid mitjançant l'extracció i redisseny del component de navegador de la suite de programari Mozilla. De la mateixa manera que Mozilla, Firefox és multiplataforma i utilitza el llenguatge d'interfície XUL.

### 4.3. Mòduls de l'aplicació

Dins del nucli de la nostre aplicació de servidor d'escriptori remot, es compon de tres mòduls principals, com mostrem en la següent figura.



**Figura 18. Aspecte de la implementació.**

- **Web interface**

La web interface conté les diferents pantalles de l'aplicació (amb estil proposat en el prototip de la fase de disseny) i dona servei i funcionalitats als usuaris. Aquestes funcionalitats son:

- Baixar software de clients

Es mostren els diferents clients disponibles amb una explicació de les seves característiques amb un enllaç que descarrega el software.

- Configuració dels servidors

Formulari on es posen les dades de configuració de l'escriptori remot. Aquestes dades només seran accessibles per a l'usuari registrat que les ha introduït, amb les quals les podrà editar i eliminar.

- Connexió amb els servidors remots.

Pantalla on apareixeran els escriptoris remots configurats anteriorment pel usuari. Aquest pot elegir el llistat mostrat i accedir directament cap al seu escriptori.

A part de les funcionalitats aquestes interfícies han d'incloure per petició expressa els logos de la Universitat Oberta de Catalunya i Opentrends tal i com es mostra en la maqueta descrita en el disseny.



**Figura 19. Logos que ha d'incloure a l'aplicació.**

- **Connection forwarding**

Per tal d'enllaçar les interfícies web amb els servidors configurats, s'inclou aquest mòdul de connexió encarregat de fer l'enllaç entre aquests dos mòduls. Per fer l'enllaç usarem la tecnologia NX, que permet realitzar connexions remotes molt ràpides que permet als usuaris accedir a escriptoris remots inclús amb xarxes lentes.

Com hem descrit abans, usarem una implementació lliure d'aquest protocol anomenada FreeNX.

- **Configured servers**

Inclou els diferents connectors a servidors remots com FreeNX, X11, Rdesktop, etc. S'encarrega de donar el servei necessari per poder accedir remotament. Aquests connectors han estat prèviament configurats. En un principi només es configurarà el connector FreeNX per realitzar la prova de concepte. En posteriors iteracions s'intentarà incloure un altre protocol amb un altre software.

## 5. Estudi de llicències

En aquest apartat es dedicarà a fer un petit estudi de les llicències que de forma implícita conviuran en el projecte final i que poden causar incompatibilitats entre elles i per tant hem de estar previnguts per no tenir ensurts en temes legals, els quals ens poden significar connotacions penals i econòmiques.

Hi han molts tipus de llicències, però descriurem les que poden afectar al nostre projecte, aquestes son:

- **GNU GPL**

Es de tipus copyleft, i es la més recomenada per a paquets de software. Tots els elements desenvolupats han de mantenir la mateixa llicència

- **GNU LGPL**

Amb característiques GPL però menys restrictiu amb l'incorporació de programari amb altres tipus de llicència.

- **BSD simplificada**

Es basa en la BSD original, eliminant la clausula de privacitat. Es permisiva sense copyleft però compatible ammb GNU GPL.

- **MIT**

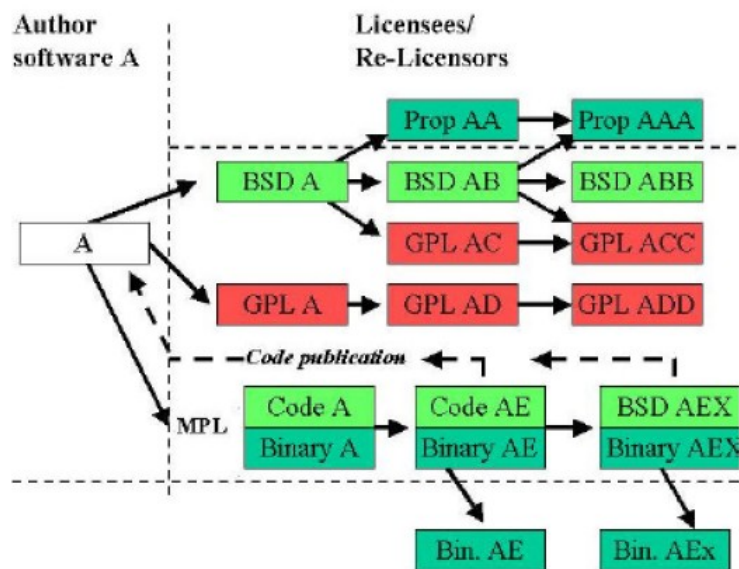
Permet usar el software lliurement, copiar, publicar, distribuir, sub-licenciar sempre que s'inclougi la nota de copyright en totes les distribucions.

Un dels principals problemes de compatibilitat sorgeix del software que permet fer les connexions a escriptori remot pel qual hem d'anar amb compte a l'hora de compartir aquests software dins de l'aplicació. El programari que usarem en un primer moment es FreeNX, però podem usar d'altres que descrivim a continuació amb el tipus de llicència associada:

Programari	Llicència
RealVNC	GPL / propietaria comercial
XDCMP	MIT License
Microsoft Terminal Server (RDP)	MS-EULA
Citrix ICA	Propietaria
Freenx	GPL

**Taula 5. Tipus de llicència.**

Com podem veure hi ha una varietat de programari associada a una varietat de llicències pel que cada elecció pot comportar un seguit de restriccions cap a altres llicències. Com podem veure en la següent figura on es representen tres de les llicències lliures més usades, mostra com la elecció del tipus de llicència pot afectar en la hora de llicenciar o rellicenciar versions del programari original.



**Figura 20. Esquema de elecció de llicències.**

Observem que les restriccions es van complicant a mesura que es van introduint modificacions en els termes de les llicències. Observem que passa amb cadascuna d'aquestes que ens permetrà veure una mica més clar que està passant:

**BSD:** una aplicació llicenciada amb BSD permet que altres versions puguin tenir altres tipus de llicències, tant propietàries, com BSD o GPL.

**GPL:** aquesta llicència aplica la necessitat del *copyleft*, fent que les noves versions de la aplicació siguin sempre lliures i llicenciades sota GPL.

**MPL:** aplica llicències dobles al codi font i als executables, obligant a tornar a l'autor les fonts modificades i permetent llicenciar els binaris com a propietaris.

Com s'esperava el tema de les llicències es complex i per tant com a solució més senzilla per al nostre projecte serà utilitzar sempre que es pugui llicències GPL.

Sabent que la nostra aplicació implementarà el protocol NX trobem programari compatible com el que es mostra en la següent taula i que seran el principals candidats a implementar-se després de l'ús de FreeNX:

<b>Programari</b>	<b>Llicència</b>
OpenNX	GPL
Neatx	GPL
X2go	GPL

**Taula 6. Programari compatible amb tecnologia NX**

- **Conclusions**

Podem dir com a conclusió que vistes les tendències actuals en que cada cop hi ha més participació en el món del programari lliure, usar llicències que permetin la divulgació de informació i coneixement va mes enllà dels temes estrictament del desenvolupament de software. Així doncs acabem amb aquestes tres premisses:

- El software es coneixement i ha de poder-se estendre sense impediments
- Utilitzar programari amb llicències propietaries comporta la prohibició de la cooperació i la creació de comunitats de desenvolupadors. Per tant es preferible apartar-nos.
- Utilitzar programari lliure amb llicències tals com GPL es una forma de contribuir a la societat aportant coneixement.

- **Altres llicències**

També podem trobar programari que usi altres sistemes de llicències per això resumeixo algunes de les significatives en el ús de programari lliure indicant els principals aspectes que la identifiquen:

Llicència: **Eiffel Forum (EFL)**

Compatible GPL: Si (v2)

Certificada OSI: Si



Llicència: **IBM Public License**

Compatible GPL: No

Certificada OSI: Sí

Llicència: **Intel Open Software**

Compatible GPL: Si

Certificada OSI: Si

Llicència: **Perl**

Compatible GPL: Si

Certificada OSI:

Lliure segons DFSG: Si

Comentaris: Llicència dual AL/GPL

Llicència: **Python**

Compatible GPL: Si

Certificada OSI: Si

Libre según DFSG: Si

Llicència: **Zope Public License (ZPL)**

Compatible GPL: Si (v2)

Certificada OSI: Si

Lliure segons DFSG: Si

Llicència: **W3C Software**

Compatible GPL: Si

Certificada OSI: Si

Lliure segons DFSG: Si

Llicència: **OpenLDAP**

Compatible GPL: Sí (v2.7)

Certificada OSI: No

## 6. Resultats

---

A continuació es descriuen els resultats obtinguts en la finalització del desenvolupament del projecte, en el qual intervien diferents tecnologies i metodologies per generar una prova de concepte d'un servidor d'escriptori remot.

En primer terme analitzarem el programari generat i el seu funcionament, per donar una visió global del projecte i de les funcionalitats implementades.

Seguidament analitzarem el resultat obtingut, comparant els objectius inicials amb el resultat final, així com la planificació proposada amb el temps real que s'ha empleat en la realització i finalització les cadascuna de les tasques.

### 6.1. Vista dels resultats

Una de les primeres tasques que es va realitzar, va ser generar un appliance on s'executaria les diferents parts del projecte. Aquest appliance, un cop finalitzat, es gravarà en un CD-ROM demostració del qual no farà falta cap programari ni instal·lacions addicionals.

Així doncs, el appliance generat esta basat en el sistema operatiu SuSe i es de tipus Firefox kiosk, el qual desplegarà el navegador al iniciar-se. Com a programari indispensable, s'inclou:

- MySQL: Sistema de base de dades.
- Apache: Servidor d'aplicacions on corra el programari web.
- Connectors de Servidors remots: Com a exemple per defecte s'usa el programari FreeNX.

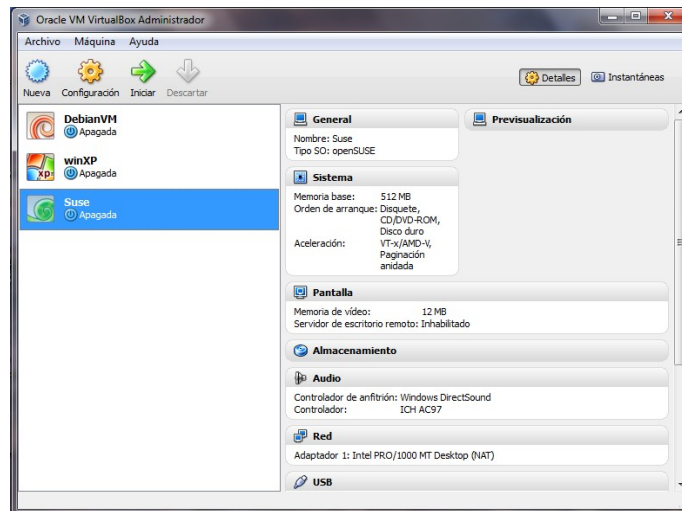
Per dur a terme les proves, l'appliance s'executa sobre una maquina virtual creada amb Oracle Virtual Box.

Aprofitant aquest programari creem altres maquines virtuals on crearem els nostres escriptoris de prova. Aquests son els que s'accedirà mitjançant la nostra aplicació.

El contingut de virtual Box es compon per els següents sistemes operatius que ens serviran per realitzar les proves de connexió remota:

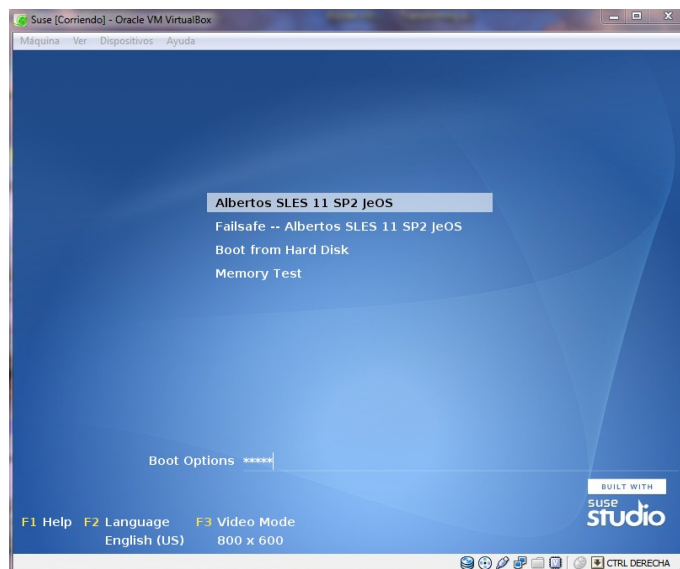
- Debian
- Windows XP
- SuSe

En la següent figura es mostra aquesta configuració:



**Figura 21. Màquines virtuals creades en Virtual Box.**

Recordem que el appliance es trobarà encapsulada en una iso auto-bootable que simularem en un CD-ROM virtual dins del mateix Virtual Box. Un cop introduïda la iso iniciem la maquina virtual.



**Figura 22. Inici de l'appliance basat en SuSe Studio.**

Un cop iniciada l'appliance s'obrirà el navegador FireFox, en que en la seva pagina d'inici tenim per defecte la nostre aplicació web.

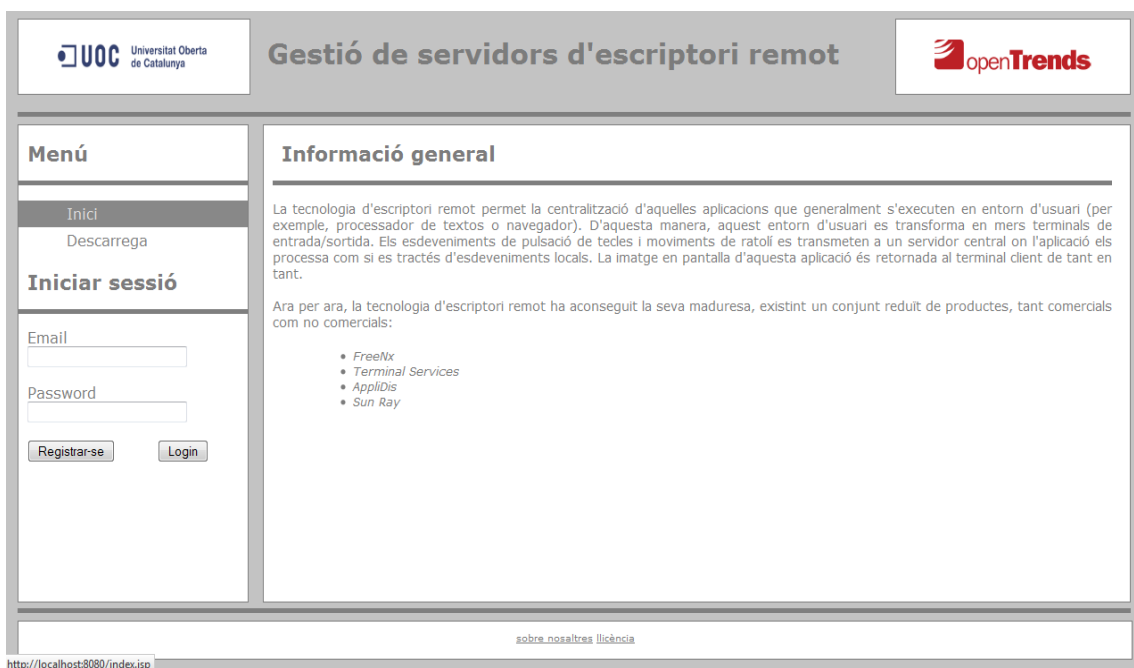
El resultat de l'implementació de la web, el veiem a continuació amb la següent guia de funcionament:

- **Pàgina d'inici:** Es la pàgina inicial, la qual s'obre automàticament al inicialitzar l'appliance SuSe. Consta d'un menú de navegació i ens permet iniciar una sessió amb usuari i password. També hi ha informació general sobre el funcionament dels escriptoris remots amb els diferents connectors.

El menú consta de les opcions:

- Inici: Mostra la pantalla principal
- Descarrega: Va cap a la pàgina de descarregues.

Aquestes opcions son les que es mostren per defecte a usuaris no registrats. En la figura següent es mostra aquesta pantalla:



**Figura 23. Inici de l'aplicació web.**

- **Pàgina registre d'usuari:** Dins de la pàgina principal, existeix en la part inferior del menú principal la opció d'iniciar sessió. Abans de poder iniciar sessió cal que l'usuari estigui registrat, per tant caldrà que accedeixi al formulari de registre mitjançant el botó de "registrar-se". Per el moment només caldrà que introdueixi el nom, cognoms, l'email i el password.

The screenshot shows the 'Gestió de servidors d'escriptori remot' interface. At the top left is the UOC logo and name. At the top right is the 'openTrends' logo. The main content area is divided into two columns. The left column, titled 'Menú', contains links for 'Inici' and 'Descarrega', and a section for 'Iniciar sessió' with 'Email' and 'Password' input fields, and 'Registrar-se' and 'Login' buttons. The right column, titled 'Registre d'usuari', contains a form with 'Nom', 'Cognoms', 'Email', and 'Password' input fields, and a 'Guardar' button. A small link 'sobre nosaltres llicència' is visible at the bottom of the page.

**Figura 24. Registre d'usuari.**

Si tot ha anat correctament l'aplicació ens mostra el missatge de que ens hem registrat correctament, i podrem tornar a l'inici de l'aplicació.

This screenshot shows the same 'Gestió de servidors d'escriptori remot' interface as Figure 24, but with a successful registration message. The 'Registre d'usuari' section now displays the text 'Usuari registrat correctament' and a single 'Inici' button. The rest of the interface, including the 'Menú' and 'Iniciar sessió' sections, remains the same. The 'sobre nosaltres llicència' link is also present at the bottom.

**Figura 25. Registre d'usuari finalitzat correctament.**

Un cop registrats, introduïm el nostre email i password en l'apartat d'iniciar sessió. Aquesta acció ens portara l'aparició de noves opcions en el menú de navegació.

Així dons el menú desbloquejara aquestes dues opcions:

- Configuració: Mostra el formulari de configuració d'escriptoris remots.
- Publicacions: Llista els escriptoris configurats per a l'usuari registrat.

Com a informació complementaria, el sistema indica l'usuari registrat mostrant el seu nom i cognoms, com podem veure en la següent figura:



**Figura 26. Inici de sessió.**

- **Configuració del sistema:** Permet configurar els servidors gràfics remots i les seves connexions, on les dades del formulari es guarden dins de la base de dades. Per el moment l'introducció d'escriptoris es il·limitada.



Figura 27. Configuració d'escriptori remot.

- **Publicació de les connexions:** S'encarrega de mostrar els diferents escriptoris remots configurats en la pantalla de publicacions, juntament amb la possibilitat d'accedir de forma automàtica als diferents escriptoris configurats per l'usuari.



Figura 28. Llistat d'escriptoris remots publicats

- **Pàgina de descarregues:** Lloc que conté els diferents clients necessaris per la connexió als escriptoris remots suportats i on l'usuari els pot obtenir mitjançant descarrega directa o mitjançant la pàgina web dels creadors dels connectors.








**Figura 29. Descarrega del software .**

Un cop vist el resultat de la implementació de les diferents pantalles ens centrem en la funcionalitat principal de l'aplicació. Es tracta de connectar amb l'escriptori remot, usant el connector predefinit. Per això tornem a la pantalla de publicacions on apareixen els diferents escriptoris definits, on seleccionem el que desitgem mitjançant la bombolla de selecció i fem click en el botó de "Connectar".

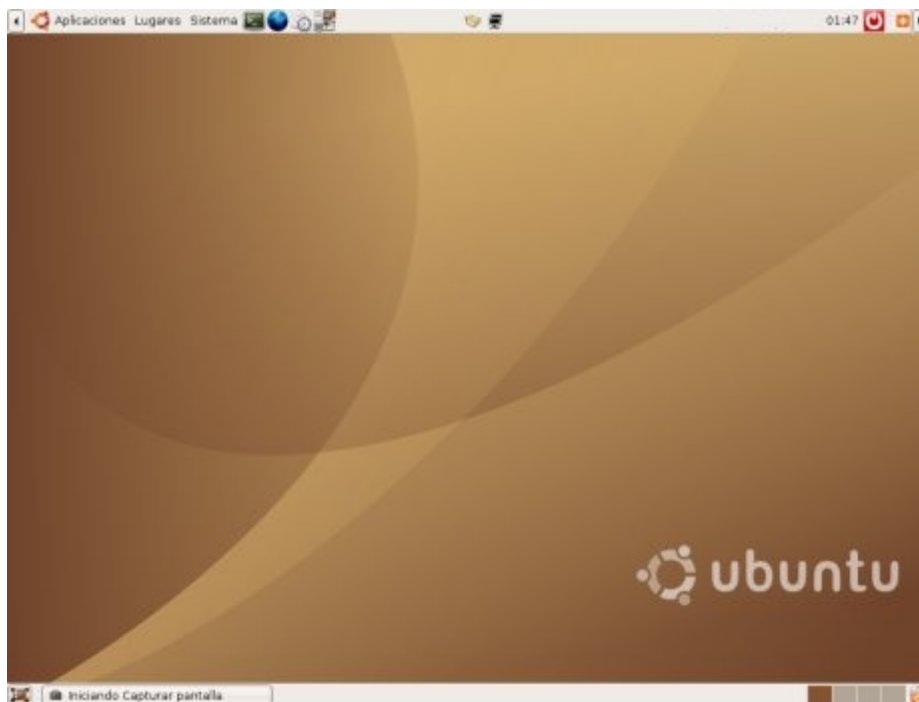


**Llistat d'escriptoris remots**

Escriptori remot	IP	Connector	Seleccionar
 <a href="#">Escriptori PC Windows XP</a>	192.168.1.171	FreeNX	<input checked="" type="radio"/>
 <a href="#">Escriptori PC Linux SuSe</a>	192.168.1.172	FreeNX	<input type="radio"/>
 <a href="#">Escriptori PC Linux Debian</a>	192.168.1.173	FreeNX	<input type="radio"/>
 <a href="#">Escriptori PC Linux Ubuntu</a>	192.168.1.174	FreeNX	<input type="radio"/>
 <a href="#">Escriptori PC Linux</a>	192.168.1.175	FreeNX	<input type="radio"/>

**Figura 30. Connectar amb el escriptori remot.**

Si la connexió s'efectua correctament ha d'aparèixer una nova finestra amb el escriptori remot seleccionat.



**Figura 31. Finestra amb el escriptori seleccionat.**

## 6.2. Anàlisi dels resultats

Després de mostrar els resultats de manera gràfica, en aquest apartat es vol analitzar des d'un punt de vista diferent, es dir, en termes d'objectius, planificació, implementació i ajust als resultats previstos inicialment.

En primer lloc analitzem els objectius que hem assolit:

- La familiarització de l'ús d'apliances en la implementació de projectes.
- La creació d'una la prova de concepte de connexió d'escriptoris remots mitjançant programari lliure.
- Generació de la web demostració que implementa de forma visual la prova de concepte.

En aquest punt destacar que en conjunt s'han assolit els objectius inicialment proposats.

A continuació analitzarem la planificació proposada en el disseny del projecte, que es resumeixen en la següent taula:

Fase	Tasca	Dies planificats	Dies reals	Desviació	Motius
	Presa de contacte	15	15	0	
Fase 1	Crear liveCD appliance	3	3	0	
Fase 1	Modificar LiveCD amb programari	4	4	0	
Fase 2	Web – Configuració de serveis	4	5	1	
Fase 2	Web – connexions disponibles	4	5		
Fase 2	Web – descarrega de clients	5	5	1	
Fase 3	Crear Web demostració	9	12	3	
Fase 3	Integració interfícies web	11	14	3	
Fase 4	Estudi legal	11	11	0	
Entrega	Generar LiveCD	2	2	0	
Entrega	Documentació	10	12	2	
Entrega	Presentació	5	5	0	

**Taula 7. Tasques planificades i desviacions.**

El resultat d'aquesta planificació no ha estat la esperada ni la desitjada en el moment de la definició. Al llarg del desenvolupament de les tasques han sorgit una serie de retards que han afectat al correcte lliurament de la documentació prevista i de la generació del codi en el temps estimat.

	<b>Planificació projecte</b>	<b>Projecte real</b>	<b>Desviació</b>
<b>Dies totals</b>	83	93	10

**Taula8. Total desviació del projecte.**

Vistos els problemes sorgits, hem detectat que hi ha diferents defectes en la planificació localitzats en:

- La definició del temps en dies.

Al proposar el temps en dies no es pot fer un seguiment exacte en qüestió d'hores reals dedicades. Es pot dir que la irregularitat de les hores dedicades ha afectat en el desenvolupament final de l'aplicació. Per això, la planificació en hores ens permet ser més regular a l'hora de dedicar temps i a intercalar tasques secundaries no previstes inicialment. També es pot estimar la duració en dies de forma més flexible dilatant o escurçant la tasca a realitzar.

- La no inclusió del temps de documentar en la finalització de cadascuna de les fases.

Al finalitzar cada fase no es va incloure un temps de dedicació a la documentació demanada (En en nostre cas seguiment de les PACs), pel que ha desencadenat una serie de retards en les entregues i finalització en el temps estimat de les diferents fases.

Com a resultat final de la planificació podem dir que no ha estat del tot satisfactòria ja que ha acumulat un retard d'unes dues setmanes.

Com a resultat de la implementació no es destaquen problemes importants en l'aplicació de les tecnologies proposades ni en el desenvolupament de les diferents funcionalitats de projecte desenvolupat.

## 7. Conclusions

---

El projecte que hem realitzat ha contribuït de manera molt important en posar en practica els diferents conceptes adquirits en el camp del programari lliure. Els resultats obtinguts ens plantegen una serie de reflexions que ens reforcen a l'hora de fer una bona implementació d'un projecte.

Aquestes reflexions i des de el punt de vista dels resultats obtinguts vistos en el anterior apartat ens porten a realitzar els comentaris que descrivim a continuació.

Un dels principals coneixements adquirits gràcies a aquest projecte es el de l'utilització i configuració d'appliance, eina útil per poder generar entorns software empaquetats per poder distribuir de forma ràpida. També és útil a l'hora de plantejar entorns de treball o de proves sense tenir que gastar massa temps en preparar un entorn físic real.

Per altre banda esta la creació de la interfície web, on hem desenvolupat les tècniques de maquetació i la generació d'applets JAVA per la interacció entre els diferents mòduls de l'aplicatiu integrant les funcionalitats descrites.

Una de les parts més importants del projecte ha estat la integració del protocol NX amb el programari FreeNX que ens ha permès realitzar connexions a escriptoris remots de forma ràpida. No obstant val a dir que la utilització d'aquest software no ha estat del tot com s'esperava ja que hi han hagut problemes en la configuració d'aquest i la integració amb l'aplicació. Però aquests problemes ens han permès adquirir més coneixements dels que en un principi podíem haver obtingut en el cas que tot hagués funcionat correctament.

Pel que fa als objectius del projecte i des de el meu punt de vista crec que s'han assolit en la seva majoria deixant potser pendants la connexió d'escriptoris remots amb altres protocols que hem nombrat en diferents apartats però que ens han quedat en el tinter.

Una de les futures accions per acabar de arrodonir el projecte seria preparar aquests protocols amb programari adequat i fer-lo funcionar correctament. En la elecció del programari, hem de tenir cura amb els temes de llicència com hem vist en un apartat de la memòria i cal que seguim les recomanacions que em exposat, ja que es un tema que pot afectar en possibles accions futures.

Al llarg del desenvolupament del projecte, des de les primeres fases de presa de contacte fins a la finalització del projecte i de la conseqüent memòria del projecte, hem anat adquirint una experiència que ens permetrà afrontar nous reptes en un futur proper en el món laboral. Hem vist nous softwares i protocols de connexió entre ordinadors, hem plantejat solucions al diferents problemes trobats, hem planificat un projecte amb encerts i errades, potser amb més errades que encerts però que ens han ensenyat que no sempre les coses surten com es planifiquen, hem dissenyat l'estructura del projecte i finalment hem desenvolupat aquest disseny. En aquest punt es pot dir que ha sigut una tasca gratificant des de el meu punt de vista personal.

Segurament es podrien millorar moltes de les tasques realitzades però l'essència del projecte continuaria intacte a pesar de que ara potser en aquest moment de finalització del projecte les plantejaria d'una altre manera.

En definitiva esperem que sigui un punt de partida per a futures implementacions o font d'inspiració per a nous projectes sempre que sigui mantenint la filosofia del desenvolupament del programari lliure.



## Bibliografia

---

Mohammed J. Kabir. "La biblia del servidor Apache 2".  
Ed. Anaya Multimedia

Steve McConnell. "Desarrollo y gestión de proyectos informáticos"  
Ed. McGrawHill

Adoración de Miguel – Paloma Martínez. "Diseño de base de datos. Problemas resueltos"  
Ed. Alfamega – Ra-Ma.

Chris Negus. "Suse Linux Toolbox: 1000+ Commands"  
Ed. John Wiley and sons Ltd. 2007.

## Referències electròniques

---

SuseStudio project site

<http://www.susestudio.com> (Juny 2012)

Creative commons licence

<http://creativecommons.org> (Juny 2012)

Manual d'escriptori remot amb FreeNX

[http://www.wikilearning.com/monografia/escritorio\\_remoto\\_con\\_freenx/6865-1](http://www.wikilearning.com/monografia/escritorio_remoto_con_freenx/6865-1) (Juny 2012)

Manual Linux SUSE

<http://redes-linux.com/manuales/SuSE/SuSE-Linux-Userguide-8.1.0.3.pdf> (Juny 2012)

Descripció de la tecnologia NX

<http://es.scribd.com/doc/79831549/Clientes-Ligeros-y-Servidores-NX> (Juny 2012)

Pàgina d'inici de NoMachine

<http://www.nomachine.com> (Juny 2012)

Pàgina d'inici de FreeNX

<http://freenx.berlios.de> (Juny 2012)

Potocol de compressió NX

<http://www.nomachine.com/documentacion/NX-ProtocolCompression.php> (Juny 2012)

Repositoris de FreeNX en Ubuntu

<http://free.linux.hp.com/~brett/seveas/freenx> (Juny 2012)

Servei de terminal remot amb NX

[http://www.linux-magazine.es/issue/35/021-024\\_NXLM35.crop.pdf](http://www.linux-magazine.es/issue/35/021-024_NXLM35.crop.pdf) (Juny 2012)

Instal·lació de FreeNX en OpenSuse

<http://www.tormentadebits.com/2012/03/freenx-en-linux-opensuse.html> (Juny 2012)



Instal·lació NX server

<http://es.kioskea.net/faq/2595-instalar-nx-server> (Juny 2012)

Llicència GNU

<http://www.gnu.org/licenses/licenses.es.html> (Juny 2012)

Anàlisi de llicències en software lliure

[http://www.miriamruiz.es/slides/doc\\_licencias\\_de\\_software\\_libre.pdf](http://www.miriamruiz.es/slides/doc_licencias_de_software_libre.pdf) (Juny 2012)

Llicències de software lliure

[http://en.wikipedia.org/wiki/Free\\_software\\_licence](http://en.wikipedia.org/wiki/Free_software_licence) (Juny 2012)

Alternatives al les llicències lliures i open source (Juny 2012)

<http://derecho-internet.org/node/325>



## Annex I: OpenTrends S.L.

---

Anem a conèixer una mica millor l'empresa on es realitzaran les practiques del Projecte Final de Màster de Programari Lliure, es tracta de "OpenTrends Solucions y Sistemas S.L.". Es una empresa d'enginyeria de software, que té com objectiu proporcionar solucions i sistemes d'informació per a les empreses, mitjançant l'aplicació de les últimes tecnologies i tendències en el desenvolupament d'aplicacions "Open Source".

Dins d'aquest marc "OpenTrends" treballa i dona servei en les següents tecnologies amb el programari lliure associat a les solucions donades.

### Gestor de continguts

- Portals (Liferay)
- Gestor de continguts (OpenCms)
- Gestor documental (Alfresco)

### Projectes empresarials

- ERP (openBravo)
- eCommerce (OfBIZ / Magento)
- CRM (SugarCRM)
- BI (Pentaho / Jasper)

### Administració electrònica

- Formularis Intel·ligents (openPDF)
- BMP (OpenTram)
- Seguretat (PKI, firma electrònica...)

### Infraestructura

- Distribucions Linux
- Comunicacions Unificades (openComs)
- Cloud Computing

### Arquitectura i disseny avançat en Java

- Disseny d'arquitectures JavaEE
- Frameworks de desenvolupament
- Qualitat de codi

Alguns exemples de projectes que han estat implantats i actualment en funcionament son:

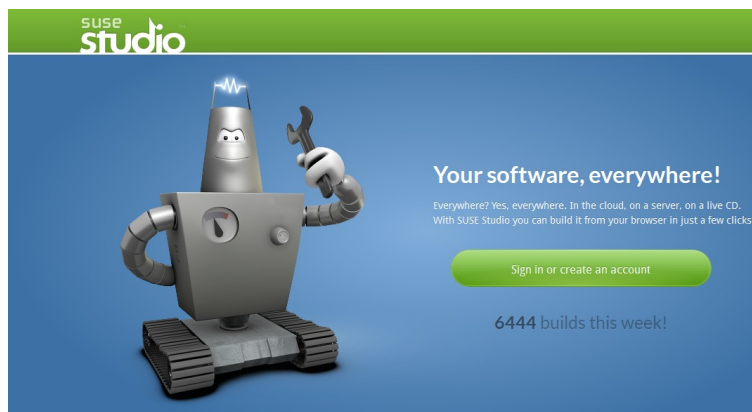
- Gencat.mobi: Mou-te  
Es tracta d'un buscador de rutes en Internet promogut per la Generalitat de Catalunya amb la voluntat d'anar integrant tota l'oferta de transport públic que opera dins el territori català.

- RUMI (Registre Unificat de Maltractament Infantil)  
S'implementa un simulador que forma part del projecte.
- "Gestió de Beques per les Vacances d'Estiu 2011"  
Projecte per l'ajuntament de Barcelona per al temps de la campanya d'inscripcions per a beques per vacances. Fa us d'un framework corporatiu per a javaEE i openFrameIMI.  
Mes info: <http://bcn.cat/vacances>
- Portal mòbil de TMB  
Portal mòbil de TMB adaptat a dispositius com iPhone, Pocket PC, Blackberry y Android.

## Annex II: Guia ràpida SuseStudio

En aquest annex descriurem de forma ràpida el procés de creació d'un appliance amb SuseStudio (<http://www.susestudio.com>), que ofereix un entorn web on de forma visual es va configurant el appliance amb les característiques desitjades.

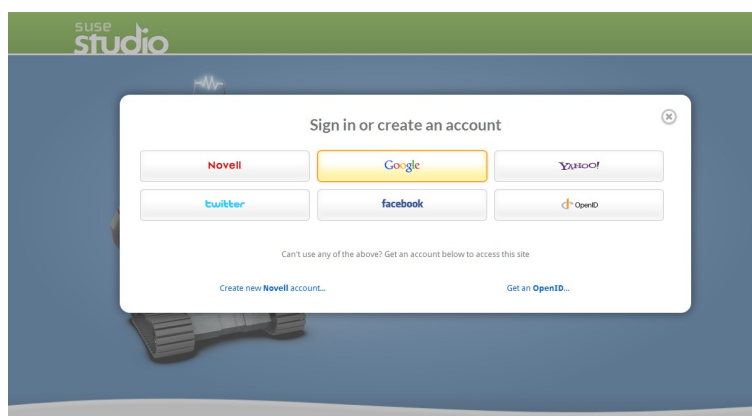
Primer de tot entrem a la pàgina web de SuseStudio i el primer que ens demana es entrar o crear un compte.



**Figura AII.1. Web inicial SuseStudio.**

Per poder accedir dins de la web SuseStudio hem de tenir creat un compte en un dels següents espais web, es cas contrari haurem de crear-lo.

- Novell
- Google
- Yahoo
- twitter
- Facebook
- OpenID

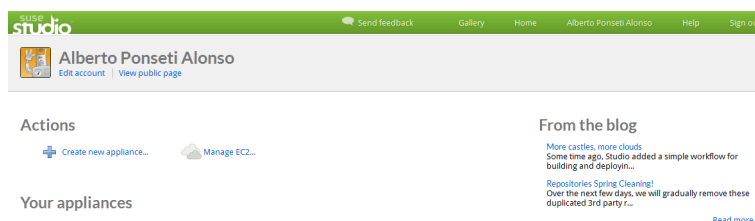


**Figura AII.2. Possibilitats de comptes acceptades.**

Un cop logats dintre d'un compte, podem accedir a dins de l'aplicació web on ja podrem crear la nostre appliance.

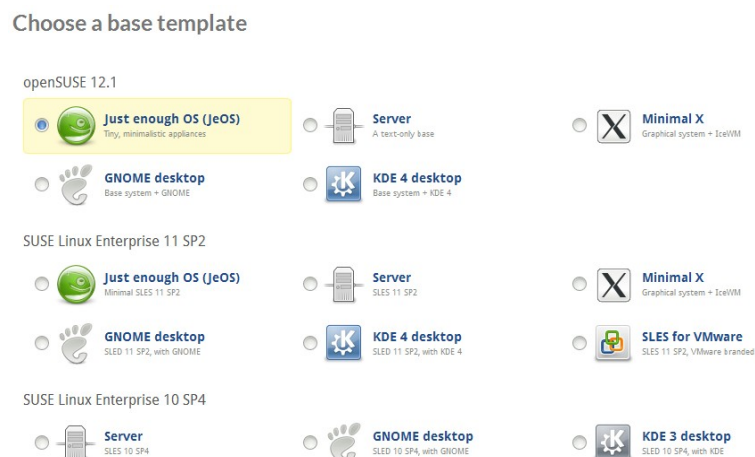
## AII.1. Com crear una nova appliance

Una vegada entrem com a usuari en la part superior apareix el nostre nom i una mica més avall dins del apartat "Actions" hi ha el botó "Create new appliance".



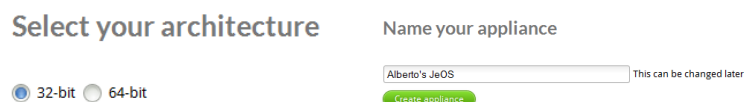
**Figura AII.3. Crear nova appliance.**

Seleccionada aquesta opció ens apareixeran una sèrie de sistemes operatius preconfigurats amb característiques diferents que permeten adaptar a cada necessitat. Per exemple, en el nostre cas hem usat la configuració Suse 11sp2 Minimal X.



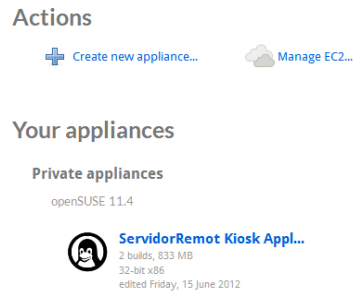
**Figura AII.4. Selecció de tipus de sistema operatiu Suse.**

Seguim baixant i ens demana el tipus d'arquitectura, que pot ser de 32 o 64 bits. Per exemple, per ser més accessibles podem usar 32 bits. Finalment posem nom a l'appliance i ja podem passar a configurar el sistema per dins.



**Figura AII.5. Tipus d'arquitectura i nom de l'appliance.**

Ara que tenim creada l'appliance en el menú principal apareixerà en les nostres appliances la nova creada.



**Figura AII.6. Appliances creades.**

## AII.2. Configuració de l'appliance

Com hem vist en l'apartat anterior on hem creat la nostre appliance tenim a la vista un enllaç que ens porta cap a la pàgina de configuració on apareixen una serie de noves opcions.

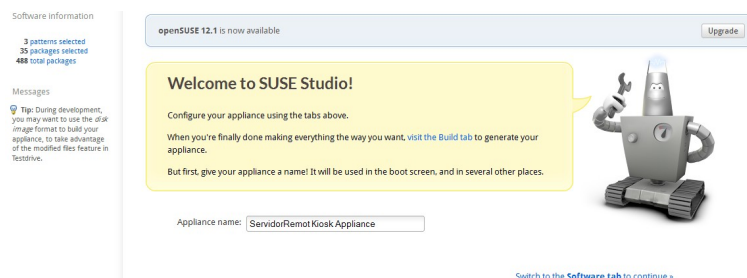


**Figura AII.7. Tabs amb les opcions de configuració.**

Anem a conèixer cadascuna d'aquestes opcions:

- Start

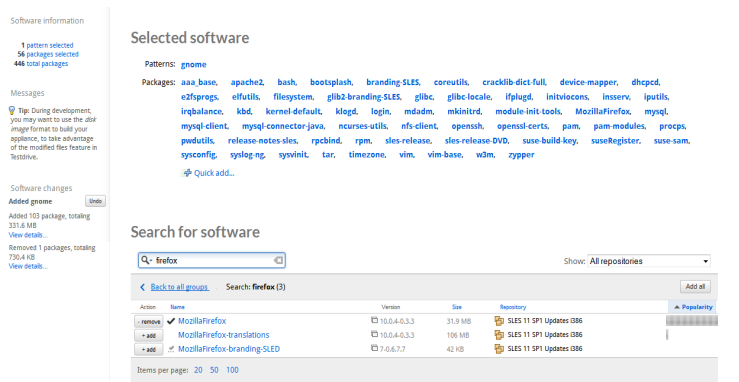
Permet configurar el nom de l'appliance, no té cap altre funcionalitat.



**Figura AII.8. Menú Start.**

- Software

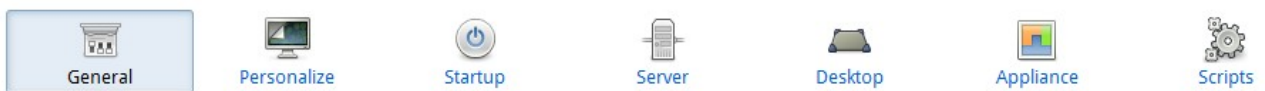
En aquesta pestanya seleccionarem tot el software que volem incloure dins del appliance, des de servidors a gestors de base de dades passant per navegadors o entorns grafics d'escriptori. Per ferir aquest software només cal fer la cerca i seleccionar dins de la cerca els paquets desitjats.



**Figura AII.9. Menú de software.**

- Configuration

Es el menú més extens i que ens permet refinar més la configuració de l'appliance. Dins d'aquesta pestanya ens apareixen noves opcions, les quals apareixen en la següent figura.

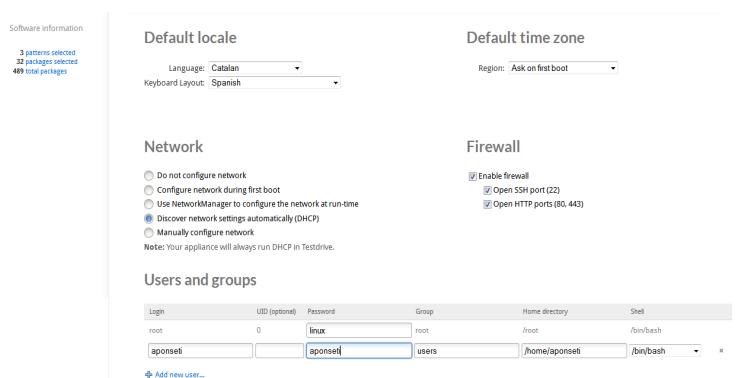


**Figura AII.10. Tabs amb les opcions de configuració.**

Aquestes opcions es descriuen com:

- General

Es el menú que conté les dades de xarxa i seguretat, a més de configurar els usuaris.

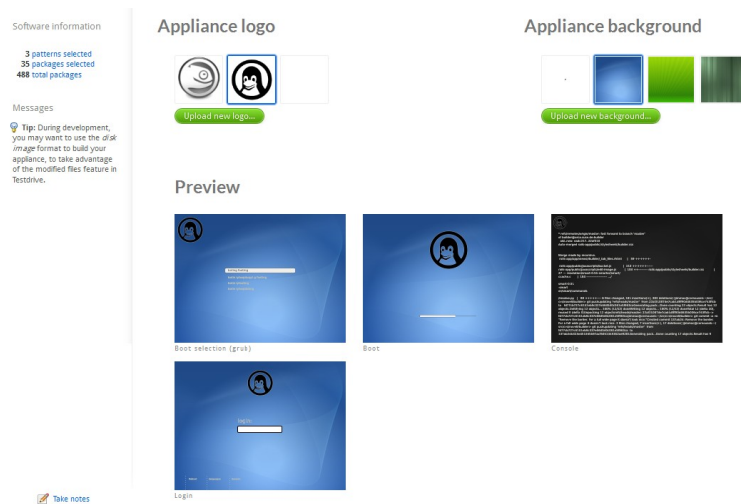


**Figura AII.11. Opcions de la configuració general.**



- Personalize

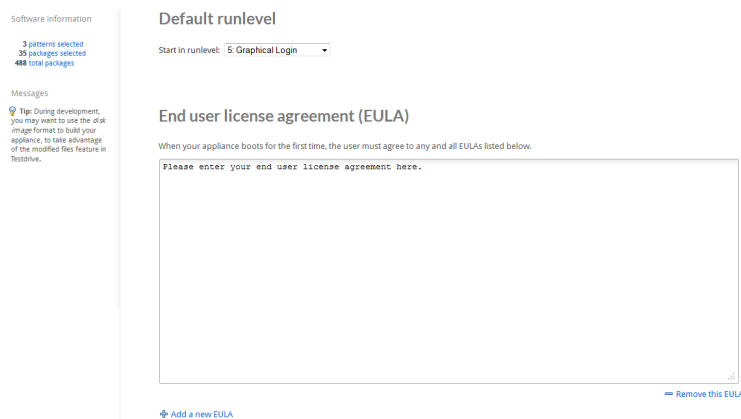
Amb aquest menú podem personalitzar l'entorn gràfic que es desplegarà al executar el appliance. Podem afegir logotips i fons de pantalla, i veure en directe la previsualització dels canvis realitzats.



**Figura AII.12. Opcions de personalització de l'entorn.**

- Startup

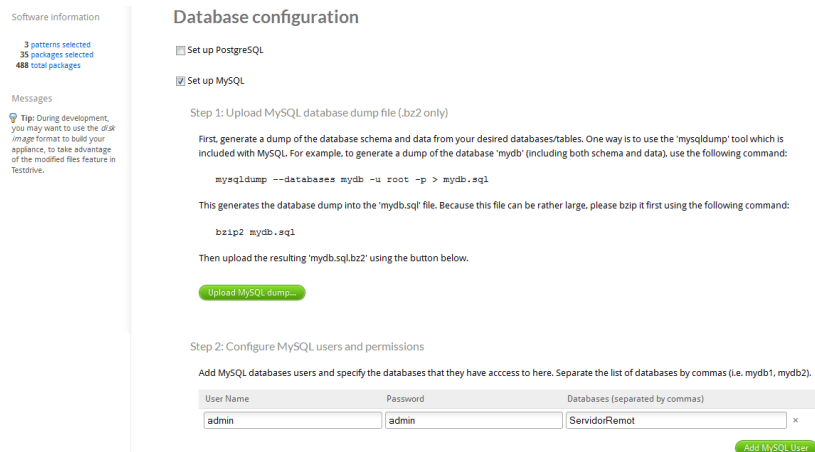
Aquí configurarem el inici de la sessió amb un missatge inicial, que podem deixar per defecte o podem indicar les dades que desitgem.



**Figura AII.13. Configura un missatge inicial de termes de llicència.**

- Server

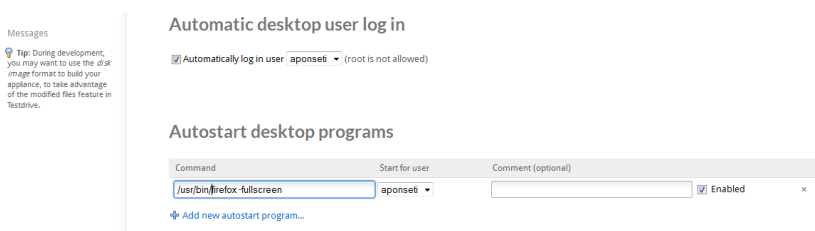
Permet configurar el programari de servidor que donarà servei a les diferents aplicacions. Es pot configurar programari de base de dades com MySQL o PostgreSQL indicant els usuaris per a les diferents bases de dades contingudes.



**Figura AII.14. Configura opcions de servidor.**

- Desktop

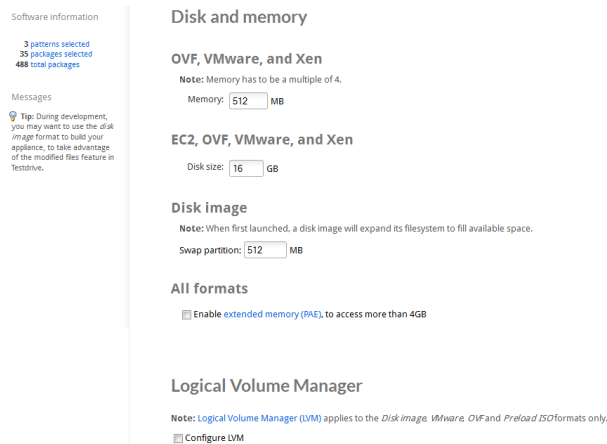
Aquest apartat es un dels destacats a l'hora de configurar l'appliance ja que s'encarrega d'inicialitzar l'escriptori. El primer que permet fer es iniciar la sessió amb un usuari concret predefinit, que serà l'usuari per defecte. Un cop tenim l'usuari podem fer comandes per que s'executin al arrancada del escriptori, podem fer que arrenquin programes, execució de scripts, etc. Per exemple podem fer que s'obri un navegador Firefox amb una pàgina web predeterminada



**Figura AII.15. Configura les comandes inicials d'escriptori.**

- o Appliance

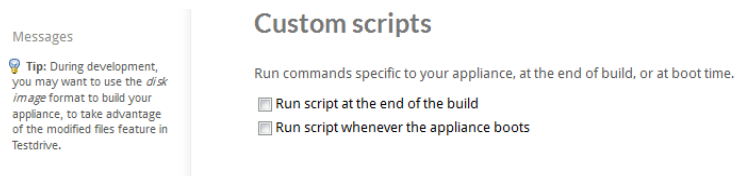
Dins del menú Appliance trobem opcions de configuració més de tipus "físic" tals com definició de disk o espai de memòria. Això és útil de cara a treballar en màquines virtuals o amb LiveCDs.



**Figura AII.16. Configura opcions de servidor.**

- o Scripts

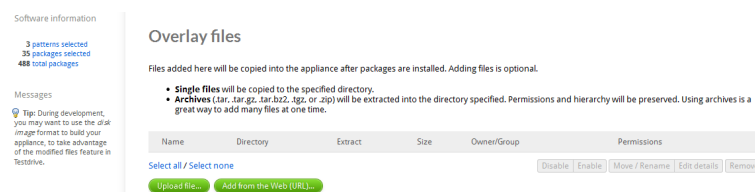
Aquest menú permet incloure scripts dins de l'appliance.



**Figura AII.17. Configura els scripts.**

- Files

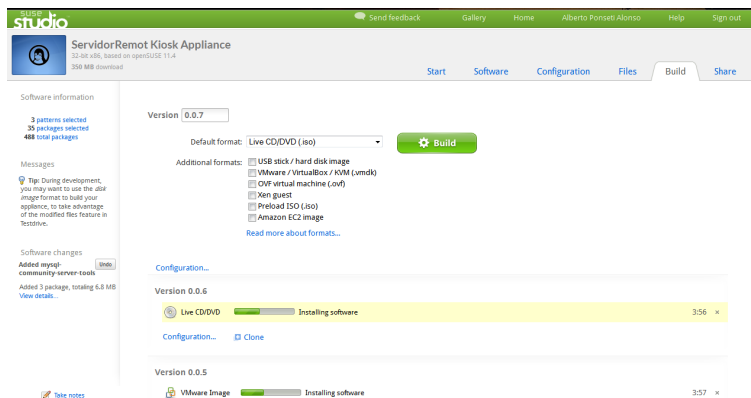
En aquesta opció podem introduir fitxers dins dels directoris de l'appliance, com per exemple fitxers war en el servidor Apache.



**Figura AII.18. Pantalla per afegir contingut a l'appliance.**

- Build

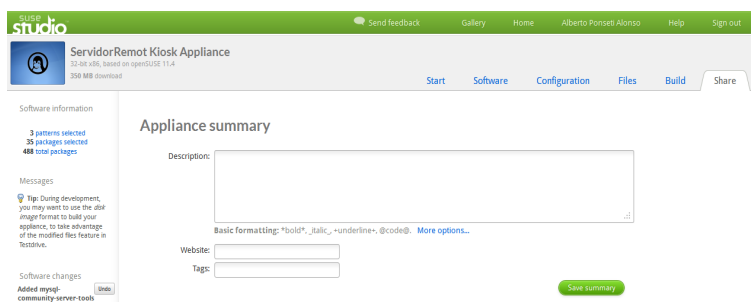
Quan arribem en aquest punt podem dir que ja tenim la nostre appliance creada i configurada prer començar a treballar. Per tant només farà falta empaquetar-la en el format de treball. Podem seleccionar entre varis fitxers de sortida tals com per a LiveCD, Maquines virtuals, Dispositius USB, etc.



**Figura AII.19. Pantalla de configuració del fitxer de sortida.**

- Share

Per acabar, podem fer que la nostre appliance estigui disponible des de la mateixa web de SuseStudio incloent dades tals com la descripció, el webside i els tags pels quals volem que s'identifiqui.



**Figura AII.20. Pantalla per compartir l'appliance.**

Donem per acabada aquest petit annex de creació i configuració d'un appliance, que espero sigui un bon punt de partida per als no iniciats en aquest tipus de software.

