

# MIGRACIÓ DE SISTEMES TIC D'UNA MULTINACIONAL A SISTEMES DE CODI OBERT

Memòria – Treball final de grau

David López Berja

*Davidlb82@uoc.edu*

Tutor: Joaquín López Sánchez-Montañés

## Resum.

Trencar amb la utopia de que a les empreses es necessari la utilització de sistemes de pagament perquè son millors i mes estables. Es pretén fer un recorregut que demostrí que l'estabilitat de tot sistema radica en tenir el control sobre seus serveis, eliminant la dependència de tercers. Utilitzant sempre que sigui possible eines de codi obert com sistemes operatius Linux i programari lliure, s'explica el camí com s'està realitzant la migració dels servidors, serveis i programari a una gran multinacional per tal de millorar la productivitat, la eficiència i obtenir una reducció dels costos.

En aquest document recollim les explicacions i justificacions dels sistemes i serveis que s'han implementat a tots els nivells de la companyia, sistemes i serveis destinats a millor els recursos i la infraestructura informàtica, els sistemes i programari destinats als usuaris. Recull també la formació necessària per l'èxit del projecte així com un petit anàlisi del seu impacte econòmic.

**Paraules clau:** GPL, GNU, Productivitat, Escalabilitat, Robustesa.

## Abstract.

Breaking with the utopia than in the companies is necessary use paid systems because are better and more stables. Our intention is to take a tour that demonstrate that the stability in all systems lies in to have the control over his services, removing third party dependences. Using always that be possible open source tools like Linux operating systems and free software, we explain the way how to we are doing the migration of servers, services and software in a big company for improve the productivity, the efficiency and obtaining a cost reduction.

In this document we collect the explanations and justifications of systems and services that we implemented in all levels of the company, systems and services destinate to improve IT resources and infrastructure, the systems and software destinate to the users. It collects too the necessary training for the project successful and a little resume of the economic analysis.

**Keywords:** GPL, GNU, Productivity, Scalability, Hardiness.

<b>INTRODUCCIÓ</b> .....	<b>3</b>
<b>1. ELS SERVIDORS I ELS SEUS SERVEIS</b> .....	<b>4</b>
1.1. LA VIRTUALITZACIÓ.....	4
1.2. TRIA DEL SISTEMA OPERATIU.....	9
1.3. EL SISTEMA GESTOR DE BASE DE DADES.....	11
1.4. DEFINICIÓ DELS SERVEIS.....	17
1.5. DISTRIBUCIÓ DELS SERVEIS.....	32
1.6. SISTEMA DE COPIES DE SEGURETAT.....	34
1.7. PROJECTES DE FUTUR.....	35
<b>2. LES MAQUINES DE CONTROL DE PRODUCCIÓ</b> .....	<b>37</b>
2.1. ELS SERVEIS DE PREMSAT.....	38
2.2. ELS SERVEIS DE SINTERITZAT.....	39
<b>3. LES EINES PER L'EQUIP DE DESENVOLUPADORS</b> .....	<b>41</b>
3.1. EINES DE DESENVOLUPAMENT.....	41
3.2. GESTIÓ DE LLIBRERIES I VERSIONAT D'APLICACIONS.....	43
3.3. ELS ESCENARIS DE DESENVOLUPAMENT.....	45
<b>4. LES EINES PELS USUARIS</b> .....	<b>45</b>
4.1. EL SISTEMA OPERATIU.....	45
4.2. PROGRAMARI ESPECÍFIC.....	46
<b>5. FORMACIÓ</b> .....	<b>48</b>
5.1. GESTIÓ, MANTENIMENT I INSTAL·LACIÓ DE SERVIDORS.....	48
5.2. BASE DE DADES MYSQL.....	49
5.3. ESCENARIS I AUTOMATITZACIÓ.....	50
5.4. PROTOCOLS DE COMUNICACIÓ.....	51
5.5. CREACIÓ D'APLICACIONS WEB.....	51
5.6. CREACIÓ D'INFORMES.....	52
5.7. PROGRAMARI BASIC D'USUARI.....	52
5.8. EINES D'OFIMÀTICA, LIBREOFFICE.....	53
5.9. PROGRAMARI AVANÇAT D'USUARI.....	55
<b>6. IMPACTE ECONÒMIC DEL PROJECTE</b> .....	<b>56</b>
6.1. LES EINES DE SERVIDOR.....	56
6.2. LES EINES D' USUARIS.....	59
6.3. RESUM.....	59
<b>7. DEBILITATS, AMENACES, FORTALESES I OPORTUNITATS (DAFO)</b> .....	<b>60</b>
<b>8. CONCLUSIONS</b> .....	<b>61</b>
<b>9. BIBLIOGRAFIA</b> .....	<b>62</b>

## Introducció.

Empresa familiar dedicada a la pulvimetal·lúrgia donant servei principalment als components del sector de l'automoció. Es converteix en una multinacional amb 7 fabricques, distribuïdes 4 a territori nacional, 3 a Catalunya i una a Aragó, una fabrica als Estats Units, un altre fabrica a Wuhu city a la Xina i finalment una a Hongria.

La [pulvimetal·lúrgia](#) es una tecnologia molt complexa però que ha tingut molt pocs canvis, el que provoca que el canvis tecnològics no afectin al nucli de negoci. El departament TIC d'una empresa multinacional on el seu negoci principal esta molt lluny de les TIC ha de poder garantir que tota la infraestructura i serveis d'informàtica adquireixin un nivell d'estabilitat i escalabilitat que permeti afrontar els canvis tecnològics de tal manera que afecti el menys possible als serveis de producció.

Per les característiques de la empresa el pressupost destinat a IT es molt mes reduït que per altres departaments que treballen directament per la producció i això fa que els serveis de les TIC tinguin un ventall de sistemes que inclou sistemes amb mes de 30 anys d'antiguitat. Amb aquest panorama i tenint en compte la manera d'actuar de les multinacionals com Microsoft que força als seus clients a seguir unes directrius d'actualitzacions, on els ocasiona problemes d'incompatibilitat amb sistemes antics, això ens obligà a prendre una decisió per garantir el millor nivell de servei i menor impacte sobre el sistemes que treballen per producció.

En aquest context a l'any 2004, quan l'empresa nomes tenia 4 fabricques, les tres catalanes i un taller als Estats Units, vam començar un projecte de migració dels serveis TIC a sistemes que ens garanteixin un major control del cicle de vida dels nostres propis desenvolupaments. En aquell moment ja estava previst el creixement de l'empresa amb la construcció d'una nova fabrica, la qual cosa ens obligà a fer un plantejament que ens garantir l'escalabilitat. La data clau va ser durant la setmana santa de 2004 on vàrem migrar el servidor de la seu principal del grup d'un sistema operatiu [Novell](#) a una distribució Linux. A partir d'aquí s'inicia una travessia complicada on s'ha requerit de molt d'ingeni per anant solucionant tots els reptes tècnics que ha representat aquesta decisió.

Aquest projecte fa un recull d'aquesta travessia que el departament de TIC portem liderant i implementant durant els darrers 16 anys i que arriba al seu punt mes dur a l'actualitat on estem en mig del procés de migració de les estacions de treball dels usuaris on el repte principal es trencar la barrera de la resistència al canvi.

# 1. Els servidors i els seus serveis.

Dins la infraestructura de les TIC els servidors son els pilars que determinarà i definirà la resta de serveis que s'utilitzaran, el fet de triar el camí del software lliure d'alguna manera condiciona els serveis que es podran implementar, o si mes no els tipus de tecnologies que s'adaptarien millor als nostres servidors i amb les quals obtindríem un millor rendiment. Es per això que la definició dels servidors es un del punts mes crítics dins del projecte de migració i cal ser molt específic perquè aquesta definició serveixi com a estàndard de definició dins de la companyia. A tots els centres de la casa ens hem de trobar una estructura molt similar que segueixi els procediments definits.

Dins d'aquest apartat ens trobarem l'estudi i la valoració realitzada tenint en compte les premisses del projecte referents a l'estabilitat i l'escalabilitat dels sistemes de producció.

## 1.1. [La virtualització.](#)

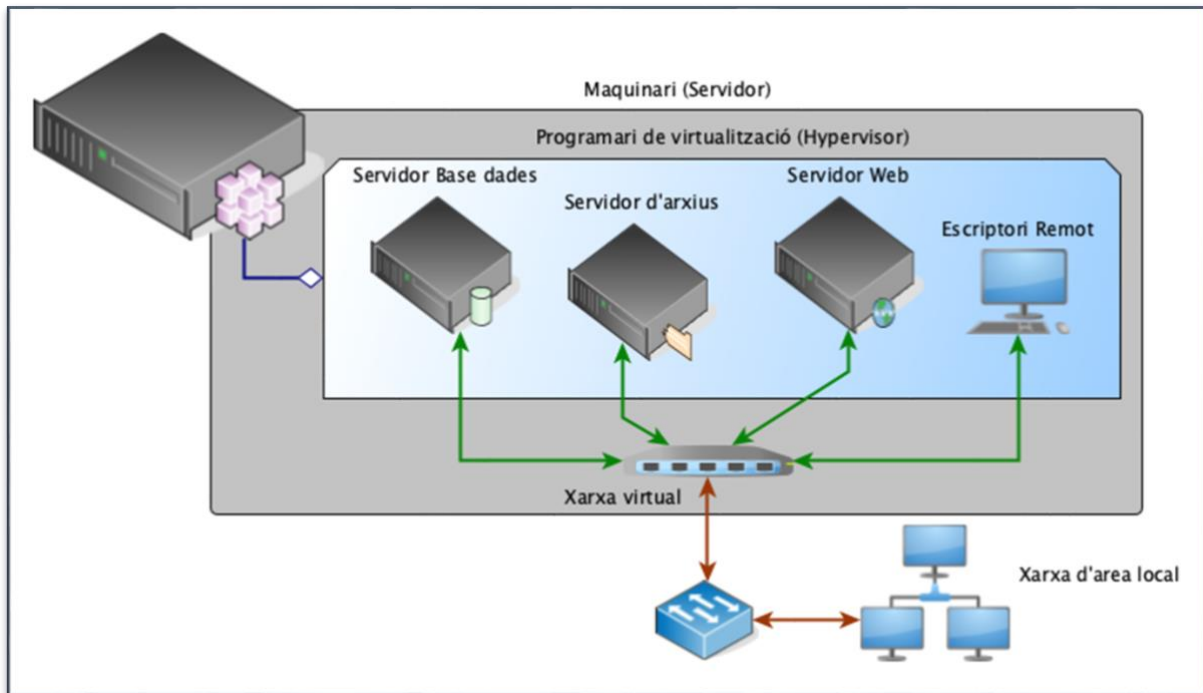
La infraestructura TIC d'una sola fabrica requereix tenir mes d'un servidor per poder-hi donar servei a totes les necessitats, tenint en compte això, dins del projecte la virtualització es una tècnica que cal explicar detalladament, entendre quina aportació te dins el nostre sistema quan a l'escalabilitat, seguretat i robustesa que ens ofereix. No valorar si aquest tipus de tecnologia es interessant dins de l'abast d'un projecte d'aquestes característiques seria un error.

### 1.1.1. Que es la virtualització i com funciona.

La virtualització es una tecnologia que permet mitjançant un programari específic, anomenat *hypervisor*, la creació de versions virtuals de recursos tecnològics com plataformes de maquinari, sistemes operatius, dispositius d'emmagatzematge o qualsevol altre dispositiu de xarxa.

Aquesta tècnica ens ajuda a extreure un altíssim rendiment a un equip informàtic, un únic servidor físic, conegut com amfitrió (*host*), ens permet tenir mes d'un servei virtual, conegut com convidat (*guest*).

Per la virtualització es molt important tenir en compte les característiques del maquinari amfitrió, com son el processador, la memòria o l'espai d'emmagatzematge, ates que aquestes característiques son les que determinen la quantitat de serveis convidats podem allotjar a dins del nostre amfitrió.



Cada servei té els seus propis requeriments quan a processador, memòria o disc necessari, però existeix una sèrie de regles que cal tenir en compte a l'hora de començar la creació de serveis virtuals:

- No podem assignar tots els recursos de l'amfitrió perquè el propi sistema operatiu necessita recursos per poder-hi funcionar.
- Cada fil del processador equival a un processador virtual, això vol dir que per exemple un processador amb 10 nuclis i 20 fils equivaldria a tenir 20 processadors virtuals per assignar-ho als nostres serveis virtuals.
- No reservar els recursos assignats a un servei, fixar en exclusivitat, ja que normalment un servei no utilitza el 100% dels recursos assignats i això permet que l'amfitrió pugui distribuir els recursos si fos necessari entre els serveis.
- Existeix una tècnica arriscada relacionada amb l'anterior, que consisteix en assignar un 120-130% de la memòria perquè habitualment un servei mai utilitza el 100% de la memòria assignada, normalment estan per sota del 75% i puntualment podem tenir pics del 100, d'aquesta manera es l'amfitrió que treu més rendiment de la memòria que té. Personalment no recomano aquesta tècnica i mai assignaria més d'un 100% de la memòria de l'amfitrió tenint com a referència que el 100% seria restant al menys 1Gb per garantir el funcionament de l'amfitrió.

## 1.1.2. Avaluació dels possibles sistemes.

La principal característica que ha de tenir aquest tipus de programari es que ha de garantir un nivell de servei que millori o iguali la seguretat que ofereix el propi maquinari. Dit amb altres paraules, no podem deixar en mas d'un programari inestable o en el que no tenim confiança la nostra infraestructura de servidors. En aquest punt es també molt important tenir en compte que el nostre pressupost es limitat per tant ens calia trobar un programari que ens proporcionés un nivell de garantia excel·lent amb un preu mínim. Al mercat existeixen diferents solucions, nosaltres només vàrem centrar-nos en aquelles que millor acceptació tenien al mercat i que a l'hora podien arribar a complir amb els nostres requeriments, per això vam establir els criteris que s'enumeren a continuació:

1. Programari estable, ha de ser un projecte de programari de garanties, hem de garantir la vida del programari així com la seva estabilitat.
2. Escalabilitat, el programari ha de permetre una fàcil i ràpida escalabilitat tant horitzontal com vertical.
3. Facilitat de manteniment, ha de ser un programari que sigui fàcil d'instal·lar i mantenir, l'ideal es que no requereixi grans coneixements per fer un manteniment ràpid.
4. Exportació e importació de serveis, poder-hi exportar he importar serveis virtuals entre diferents servidors es vital per garantir un bon nivell de serveis. A més facilita la migració i millora de servidors.
5. Facilitat per la gestió remota, els nostres serveis IT centrals estan a Barcelona i ens cal poder-hi realitzar manteniment dels nostres servidors en fabriques remotes, es importantíssim que el programari faciliti aquesta tasca.
6. Compatibilitat amb el maquinari, un dels principals problemes de tot programari es la compatibilitat amb els diferents maquinaris, es per això que a l'hora de triar un programari determinat cal avaluar si amb el nostre maquinari ofereix un alt grau de compatibilitat.
7. Optimització dels recursos, els recursos de maquinari han d'estar destinats als serveis per tant es important que aquest tipus de programari no consumeixi molts recursos de la maquina per funcionar.
8. Emmagatzematge remot, dona un alt grau d'escalabilitat i garanties tenir tot el emmagatzematge en remot, això implica que el maquinari amb el programari de virtualització no tindria emmagatzemats els dades en local, d'aquesta manera un canvi de maquinari resulta molt senzill i a més afegeix un alt grau d'escalabilitat.

9. Cost reduït, es important mantenir el pressupost del departament als nivells destinats per tant la solució no pot ser a un preu prohibitiu.

Tenint en compte aquesta llista de requeriments, trobem al mercat un gran ventall de programari que compleix amb les nostres expectatives alguns les compleixen totes altres només algunes, ens centrem només en aquells que ens resulten més interessants. A continuació tenim un petit resum del programari analitzat:

- **Microsoft Hyper-V Server:** aquest va quedar descartat automàticament, perquè Microsoft no ens genera prou confiança com per deixar res a les seves mans. A més quan a preu està fora de requeriments.
- **Citrix Xen Server:** Es un programari basat en codi obert i té una versió de pagament i un altre versió lliure. Les seves característiques quan a rendiment i seguretat són molt bones. Citrix és una empresa que ofereix garanties però que no són experts en virtualització.
- **Xen Project:** Es un projecte de codi obert recolzat pel projecte The Linux Foundation. Es tracta d'un programari molt similar al de Citrix, de fet Citrix es basa en aquest projecte, però el seu rendiment no és el més òptim.
- **VMware vSphere:** Probablement el programari més avançat i amb millor rendiment del mercat, VMware són líders en la virtualització i la seva compatibilitat amb els diferents maquinaris i el seu rendiment estan molt per sobre de la seva competència. VMware ofereix moltes utilitats de pagament però tenen una versió totalment gratuïta del seu programari coneguda com VMware vSphere ESXi.
- **Proxmox:** Programari molt similar a VMware vSphere en quan a característiques i rendiment. És un programari basat en Debian i de codi obert. Les seves principals avantatges és que a més de permetre servidors virtuals també dona suport als contenidors. Un contenidor és un mini servei que comparteix la base del sistema operatiu del amfitrió i això fa que sigui molt més lleuger.

### 1.1.3. Avantatges.

Existeixen una llista d'avantatges genèrics de la virtualització però en aquest punt enumerarem i explicarem quins avantatges aporta la virtualització dins el nostre projecte:

- **Optimització del maquinari:** La virtualització ens permet extreure un major rendiment del maquinari, un servei sense virtualitzar consumeix entre un 5% i un 20% dels



recursos del maquinari això implica que la resta fins el 100% es podria aprofitar i optimitzar el recursos de maquinari amb aquesta tècnica.

- **Estalvi econòmic en maquinari:** La pròpia optimització comentada anteriorment fa que es redueixi el nostre nombre de servidors i això es provoca un estalvi econòmic quan a despesa en maquinari i despesa energètica.
- **Reducció d'espai ocupat:** Agrupar serveis redueix el nombre de servidors i reduir l'espai ocupat al centre de processament de dades (CPD) es un punt molt important, les fabriques mes antigues quan es van dissenyar no es va plantejar cap lloc per als CPD, de fet no existia la informàtica en molts casos la qual cosa ha provocat que els CPD tinguin un espai reduït.
- **Millora de la migració de maquinari:** No tenir lligat els serveis a un maquinari concret ens ajuda a poder actualitzar el maquinari de manera molt rapida ja que podem migrar el serveis a un maquinari nou sense problemes i de forma molt rapida.
- **Seguretat:** Augment del nivell de seguretat en cas de fallada de maquinari ja que tenint un sistema de copies de seguretat dels serveis es molt ràpid aixecar els serveis en un altre maquinari.
- **Escalabilitat i flexibilitat:** Permet incorporar nous serveis de forma molt rapida, fàcil i barata, així com aporta flexibilitat a l'hora de moure serveis a maquinari mes potent i fer una distribució mes eficient dels recursos.
- **Reutilització del maquinari:** Els serveis mes crítics han de tenir un maquinari de garanties i això ens obliga a no allargar la vida del maquinari mes de 5 anys. Un cop finalitzat aquest cicle el maquinari s'aprofita per incloure serveis menys crítics com escriptoris remots per usuaris, servidors de proves, etc.

#### 1.1.4. Inconvenients.

Al igual que amb els avantatges els inconvenients s'han de centrar en el nostre projecte, i la llista de inconvenients es molt reduïda.

- **Increment de dependència del maquinari:** Al tenir molts serveis al mateix maquinari increment el risc de que una fallada de maquinari afecti a mes d'un servei.
- **Increment del cost del maquinari:** La teoria ens deia que això seria un desavantatge però en la realitat en aquest projecte el maquinari que estem utilitzant no ha incrementat el seu cost, però si obtenim un millor rendiment.

### 1.1.5. Conclusió.

La virtualització proporciona molt més avantatges que inconvenients, a més els inconvenients són fàcilment mitigables amb un bon sistema de còpia de seguretat dels serveis crítics. És per això que finalment vam decidir utilitzar la virtualització dins del nostre projecte.

En el nostre cas hem triat la opció de VMware vSphere ESXI perquè ens proporciona una facilitat en la instal·lació i manteniment que no ens obliga a destinar molts recursos. El rendiment obtingut és espectacular a més un altre dels punts que va marcar la diferència és que el catàleg de VMware és molt ampli i tenim molts de serveis que sense haver-hi de canviar res podem afegir a la nostra infraestructura.

Finalment aquesta tècnica ens permet tenir 40 servidors físics que tenen allotjats un total de 162 serveis virtuals. El rati d'utilització de maquinari és espectacular situant en 4.05 serveis virtuals a cada servidor físic.

## 1.2. Tria del sistema operatiu.

A l'hora de triar el sistema operatiu per la nostra organització el primer que ens cal analitzar és si els serveis que tenim són dependents del sistema operatiu així com quins serveis necessitem oferir. En aquesta línia és important realitzar un bon anàlisi per tal de triar el sistema operatiu dels nostres servidors. Els serveis més crítics de la nostra organització dependran d'aquest sistema per tant requereix raonament amb bona informació.

### 1.2.1. Anàlisi de requeriments.

El primer pas és realitzar una llista dels requeriments actuals de la nostra organització quan a serveis crítics es refereix, per tal de veure quin o quins sistemes operatius encaixarien millor en la nostra infraestructura.

- **Correu electrònic:** El servei de correu és imprescindible per a tota organització, és la manera de comunicar-nos tant internament com externament.
- **Compartició d'arxius:** Els nostres usuaris necessiten compartir documentació, dins del seu departament, amb altres departaments e inclús entre fabricues diferents.
- **Gestió d'usuaris:** A una multinacional és necessari o al menys molt important tenir una gestió dels usuaris de tota l'organització.
- **Gestió de privilegis:** Per una compartició eficient, eficaç i segura es requereix poder-hi gestionar els permisos la informació.

- **Noms de domini:** Tenir un servidor de noms de domini intern facilita molt les tasques de manteniment d'equips i de serveis, ja que utilitzant el nom de domini per arribar als serveis en comptes de l'adreça IP directament dona més flexibilitat, escalabilitat i balanceig als nostres serveis.
- **Cues d'impressió:** En el nostre cas tenim serveis encara que treballen amb [MS-DOS](#) i l'única via per imprimir es utilitzant una cua d'impressió.
- **Pàgines web:** Les noves aplicacions estan realitzades en format web per tant es requereix d'un servei que doni accés a aquest tipus de contingut.
- **Adreces IP dinàmiques:** Tenim molts usuaris mòbils per tant hem de tenir un servei que ofereixi una adreça [IP](#) concreta a un usuari a la seva fabrica i una adreça dinàmica quan visita altres fabricues.

### 1.2.2. Sistemes operatius candidats.

Un cop tenim la llista dels requeriments més crítics que ha de cobrir el nou sistema operatiu podem veure que qualsevol distribució de Linux pot oferir-nos aquests serveis amb un nivell de garanties i robustesa excepcionals. De totes les distribucions al nostre projecte només acceptarem aquelles que ofereixen un grau de garanties, es per això que es descarten distribucions com [Ubuntu](#) perquè les seves actualitzacions són amb paquets que ofereixen menys garanties que altres distribucions. Es per això que al nostre projecte només tenim en compte les quatre grans distribucions, que a més on base de la resta de distribucions del mercat.

- **[OpenSuse](#):** Es una distribució basada en Suse Linux Enterprise que ofereix moltes característiques d'aquesta versió comercial. El principal inconvenient es que es tracta d'una versió en desenvolupament continu sense actualitzacions, fet que no es gens segur per a servidors de producció.
- **[Gentoo](#):** Es tracta de la que possiblement es la versió de Linux més robusta però també la més difícil de mantenir i instal·lar. Gentoo es una distribució basada en paquets fonts que es compilen cada cop per la maquina concreta on s'instal·len. Requereix de molt de coneixements i la resolució de problemes es molt complicada.
- **[CentOS](#):** Distribució basada en Red Hat Enterprise server mantinguda per la comunitat i per tant garanteix suport a llarg termini on cada versió garanteix un suport als seus paquets de 4 anys.

- **Debian:** Es tracta d'una distribució que es basa en l'estabilitat del sistema operatiu, per la qual cosa utilitza paquets molt garantits, ofereix un suport a molt llarg termini i dona una estabilitat al sistema operatiu que ofereix tranquil·litat als administradors. Es una de les distribucions més antigues de Linux.

### 1.2.3. Conclusió.

Debian ens ofereix tot allò que necessitem amb els nivells de robustesa i simplicitat que requerim, tot això acompanyat per un suport a molt llarg termini que per als nostres serveis es el millor, per tant per a tots els nostres servidors Linux vam establir el criteri de que sempre intentaríem utilitzar com a sistema operatiu Debian o bé una distribució basada en Debian en cas de tenir la necessitat de un programari més actualitzat.

La principal problemàtica de les distribucions de Linux es habitualment la localització dels arxius de configuració o llibreries son diferents depenent de la distribució i això no ajuda al manteniment. Es per això, que es important de cara als administradors si tots els servidors son el més semblants possible per tant tria una distribució o una base ajuda a millorar la facilitat per a realitzar els manteniments dels servidors i els seus serveis degut a que resulta més senzill. A més, utilitzar una distribució o base de sistema ajuda als administradors a millorar els coneixements del sistema operatiu i el seu comportament.

## 1.3. El sistema gestor de base de dades.

La base de dades es un dels pilars més importants de tot sistema informàtic, per tant trobar una solució segura, flexible, escalable i que a més ofereixi el millor rendiment possible es una tasca molt important per a tot departament de tecnologies de la informació.

Quan es fa la posta en marxa d'un nou sistema, completament nou, els requeriments quan a base de dades poden ser flexibles i això dona més marge de moviment, no es el cas del nostre projecte, on en de tenir molt en compte el sistema actual. Continuant amb el mateix criteri d'impulsar les solucions [GPL](#), hem d'analitzar el nostre sistema perquè la migració a de permetre mantenir el sistema treballant sense cap impacte sobre els sistemes de producció derivats del canvi de base de dades.

### 1.3.1. Consideracions prèvies.

Es parteix de [DataFlex](#), un sistema de programació que integra un entorn de desenvolupament propi amb bases de dades pròpies, a l'inici del projecte la versió de DataFlex es la 9.1, una versió visual anomenada Visual DataFlex que ja disposa d'una eina de

desenvolupament per Windows. Inicialment tenim un 90% dels programes fets utilitzant aquest llenguatge de programació i totes les bases de dades estan en format DataFlex. El sistema inclou aplicacions MS-DOS i aplicacions Windows 16/32 bits, un requeriment obligatori es mantenir tota la història de dades, això fa que la base de dades de futur a de permetre transferir tota la informació antiga cap a la nova base de dades.

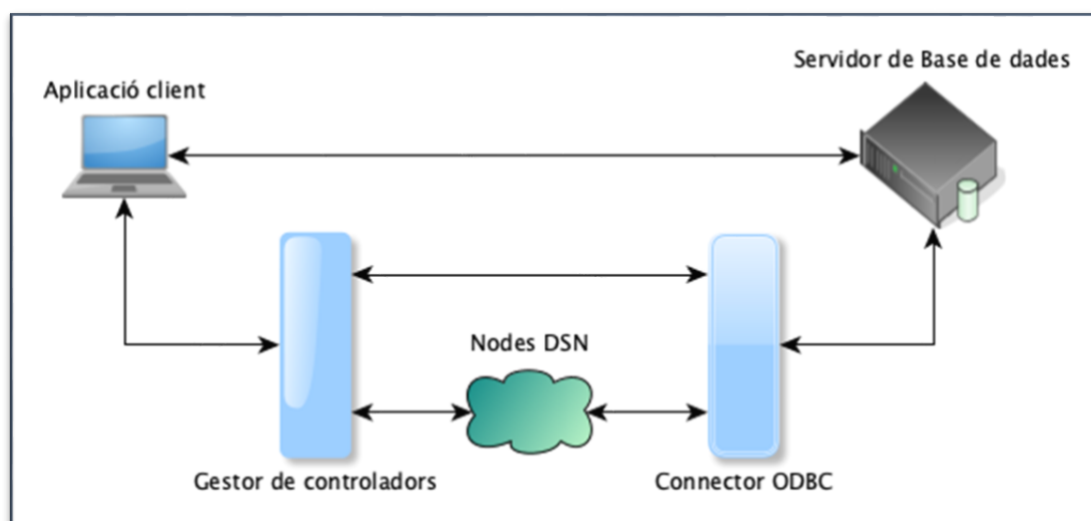
Es un requeriment obligatori el realitzar una migració progressiva mantenint tots els serveis, programes i dades sense afectar a la producció, el que implica que hauran de conviure dades en format Dataflex amb dades SQL amb el servidor que triem.

La totalitat dels informes estan realitzats en [Crystal Reports](#) versió 11, una eina de disseny de informes capaç de connectar amb base de dades DataFlex, aquests informes han de continuar treballant amb la nova base de dades mentre es substitueix per una altra opció que mes endavant veurem.

Finalment un dels requeriments mes importants per part de la direcció es que totes les fabricques han de ser autònomes quan a servidors i bases de dades es refereix, el que implica que la producció no es pot parar pel fet de no tenir connexió exterior i per tant es imprescindible continuar recollint i servint dades en cas d'una fallada de comunicacions externes (Internet).

### 1.3.2. Definició de la solució.

DataFlex disposa d'un connector de [ODBC](#) per Windows que ofereix la possibilitat d'accedir a les dades DataFlex utilitzant eines de desenvolupament compatibles amb ODBC, però en canvi no pot utilitzar ODBC per accedir a Base de dades diferents de DataFlex. Aquesta possibilitat en brinda un pont per poder-hi treballar amb aplicacions on es pot combinar dades DataFlex amb SQL.



Utilitzar ODBC ens permet generar aplicacions de traspàs de dades, com fer també aplicacions per mantenir dades a un servidor SQL i dades DataFlex simultàniament, aquesta darrera funcionalitat es perillosa però necessària, moltes vegades hem de tenir les dades duplicades en els dos llocs temporalment degut a que les aplicacions MS-DOS de DataFlex només són compatibles amb el seu format i es necessari mantenir les dades en els dos sistemes mentre tant no tenim actualitzades les aplicacions. Aquesta solució ha de ser temporal per dos motius principals, el primer es que lligar les aplicacions a ODBC es lligar-se a Microsoft i això no està dins de l'àmbit del nostre projecte que el que pretén es tot el contrari, i el segon es perquè mantenir dos bases de dades amb les mateixes dades però tecnologies diferents es molt perillós i difícil.

Mitjançant un component per Visual DataFlex (versió de DataFlex per Windows), [Mertech Flex2SQL](#) es possible connectar de forma nativa l'entorn de programació DataFlex amb base de dades SQL entre les quals es troba compatibilitat amb: [Microsoft SQL](#), [PostgreSQL](#), [MySQL](#) i [Oracle](#). Mertech presenta una limitació quan a la connexió de base de dades, per tant el fet de tenir una versió de DataFlex concreta ens obliga a mantenir la versió de Mertech que ens limita amb quines versions concretes dels servidors de base de dades podem treballar. Un altre requeriment del connector de Mertech es que una connexió només pot accedir a una base de dades SQL (schema data).

Amb les dades recollides i tenint en compte l'essència principal d'un projecte GPL ens trobem amb dues opcions, PostgreSQL o MySQL i nosaltres ens vam decantar per utilitzar MySQL, en aquest cas les comparacions són molts similars però en el moment de triar base de dades MySQL era la més utilitzada i amb més suport i fonts d'informació al nostre abast, per això va fer decantar la balança cap a MySQL. Utilitzant un paquet de Linux [lampp](#) que inclou MySQL 5.1 de forma empaquetada que resulta molt senzill d'instal·lar i distribuir, a més de que no es dependent del sistema operatiu. Aquesta solució ens ajuda a garantir que durant la migració de totes les aplicacions el funcionament del sistema no es veurà afectat, aportant a més, la possibilitat de poder treballar en paral·lel amb les dades antigues juntament amb les dades noves.

Finalment hem de tenir un servidor de base de dades a cada fabrica per tal de complir amb els requeriments de direcció. Aquest servidor només pot tenir una base de dades amb totes les taules per tal de complir amb els requeriments de Mertech i DataFlex. Evidentment en una multinacional existeixen moltes dades que ha de ser comunes per tant hem de tenir algunes

dades centralitzades i això es un dificultat que cal resoldre complint amb els requeriments esmentats.

Tota la solució presenta tres dificultats principals que van resultar complicades de resoldre:

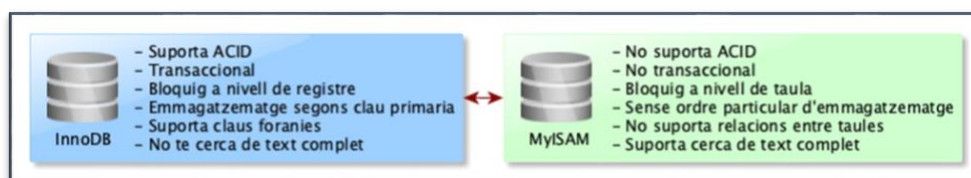
1. Nomes podem accedir a una base de dades (schema) mentre tinguem aplicacions DataFlex.
2. Hem de complir amb els requeriments de direcció quan a la independència de les fabricues.
3. Hem de tenir dades centralitzades sense oblidar la primera dificultat i complint amb els requeriments de direcció.

Per resoldre les anteriors problemàtiques vam crear una base de dades anomenada “*produccio*” on tenim totes les taules que formen la base de dades de cada fabrica, d'aquesta manera aconseguim complir amb els dos primers requeriments.

Per les dades centralitzades vam crear un servidor de dades central amb una base de dades diferent anomenada “*general*”, aquí tenim totes les taules amb les dades comunes per tot el grup. Aquestes dades es repliquen sobre totes les bases de dades de producció.

L'últim punt a tractar es el motor de base de dades, considerem com a vàlids per al nostre projecte els dos motors mes comuns per a bases de dades MySQL: [InnoDB](#) i [MyISAM](#).

InnoDB es un motor de base de dades mes potent amb mes opcions que permet per exemple tenir integritat referencial, suporta atomicitat, consistència, aïllament i durabilitat (ACID). Però el nostre projecte te uns requeriments que per poder-ho solucionar es necessari prescindir de la potencia d'InnoDB i utilitzar MyISAM, un motor molt mes simple que com a punt mes fort es que dona un millor rendiment. En el nostre cas el projecte inclou una base de dades que ha d'estar en mes d'una localització, per això es millor MyISAM. Per tal de poder-hi implementar la solució que explicarem al següent punt necessitem mantenir la integritat de les dades mitjançant el programari.



En conclusió al nostre projecte el motor que millor s'adapta es MyISAM, a mes te un sistema de replicació que molt simple que ens interessa i ens ajudarà molt durant la travessia de la transició que es presenta.

### 1.3.3. [Funcionament del sistema.](#)

A continuació es descriu el funcionament del sistema aplicat al nostre projecte sense arribar a profunditzar en detalls massa tècnics, les parts on actua algun desenvolupament propi inclouen accés al codi del programa.

#### La base de dades general.

A la seu central de l'empresa tenim un servidor de base de dades que inclou totes les dades comunes per a totes les fabricques del grup. En aquesta base de dades s'inclou informació com els usuaris, les empreses, els proveïdors, les maquines, etc.

Per aconseguir una correcta diferenciació i funcionament d'aquesta base de dades es obligatori establir unes regles per totes les taules:

- El nom sempre començarà amb “*general\_*” per exemple: *general\_mestre\_centres*.
- Han de tenir una clau primària auto incremental, per exemple *id\_centres*.
- Es crea una taula “*sincros\_generals*” per portar un control dels canvis sobre les taules generals.
- Totes les taules tindran [tres disparadors \(triggers\)](#), després de cada inserció, actualització o esborrat d'una dada insereixen un registre amb la informació referent al canvi que s'ha de replicar a la taula *sincros\_generals*.
- Totes tindran tres camps de control:
  - a. *Id\_usuaris\_stamp*: per controlar l'usuari que fa la darrera modificació.
  - b. *Marcadetemps*: per tenir la data i hora de la darrera modificació.
  - c. *Recversion*: Un camp especial per controlar la versió del registre, es tracta d'un valor incremental mitjançant un disparador.

#### La base de dades de producció.

Cada fabrica te la seva pròpia base de dades de producció amb totes les taules amb les dades exclusives de la fabrica. Un requeriment per mantenir compatibilitat es tenir totes les taules a la mateix base de dades però establir un sistema de nomenclatura per les taules de cara a un futur poder-hi separar-ho en diferents bases de dades si fos necessari.

Totes les taules de producció començaran el seu nom amb l'àmbit de les dades, per exemple les taules de matriceria seran “*matriceria\_nomdelataula*”. Això facilita l'agrupació de les taules per àmbit de sistema, matriceria, premsat, forns, barreges, etc.



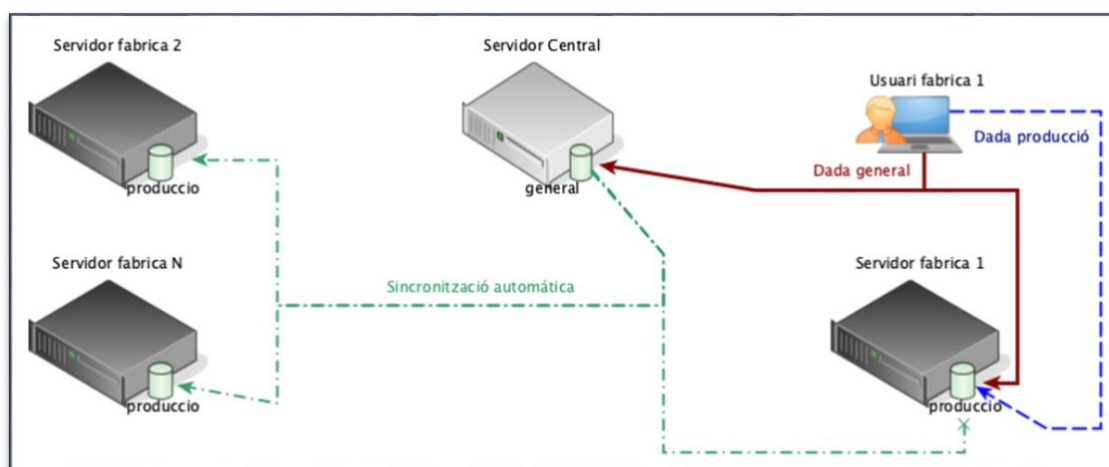
Segons el requeriment totes les fabriques ha de ser autònomes però han de poder consultar les dades del grup, es per això que totes les taules de la base de dades generals estan duplicades a totes les bases de dades de producció de totes les plantes amb unes petites diferències i requeriments quan a funcionament:

- Son taules de només lectura, no es modifiquen directament a la base de dades de producció.
- No tenen cap valor auto incremental ni cap valor per defecte com marques de temps, etc.
- No tenen cap disparador.

### Interacció de les aplicacions amb les base de dades.

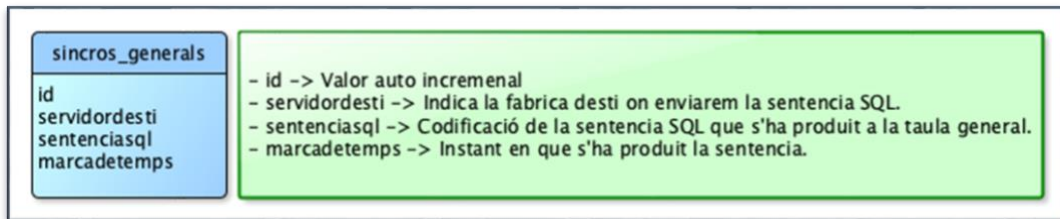
Totes les aplicacions treballen amb la base de dades de producció i tenen al seu abast la consulta de dades comunes gràcies a la duplicitat de les taules generals. Les modificacions sobre les dades de producció son directes, a diferència de les taules generals que tenim a la base de dades de producció.

Les modificacions sobre les dades generals es pot efectuar des de qualsevol fabrica sempre que existeixi connexió amb el servidor central. Un cop realitzada la modificació amb rutines de doble gravació que explicarem a l'apartat de les eines de desenvolupament, es realitza també la mateixa operació a la taula general corresponent dins la base de dades de producció, d'aquesta manera els canvis realitzats a un fabrica estan al moment a la mateixa fabrica, a la resta de fabriques s'utilitza una automatització que enviarà els canvis a les taules generals de la base de dades de producció de la resta de fabriques. La imatge següent presenta un esquema de funcionament de la solució implementada:



### Sincronització de dades de general a producció.

Totes les modificacions que es produeixen a les taules generals provoquen que els disparadors insereixin un registre a la taula sincros\_general.



Per tal de processar les dades d'aquesta taula utilitzem un desenvolupament propi fet en Java que utilitza tres classes i un arxiu de configuració:

- [StructTaula.java](#): Analitza i retorna l'estructura. El constructor de la classe requereix la connexió de base de dades i la taula a analitzar.
- [SyncClass.java](#): Es la classe encarregada de tota la feina de sincronització, dins d'aquesta classe es troben totes les funcions que s'encarreguen d'analitzar la sentència codificada, transformar-ho en una sentència SQL i enviar-la al servidor destí.
- [Syncrolocals.java](#): Classe principal del programa, encarregada de cridar a la classe de sincronització.
- [Config.ini](#): Arxiu de configuració on s'indica les dades del servidor general i totes les fabricques on cal processar les sentències.

El compilat Java syncrolocals.jar s'executa cada 5 minuts en el servidor Linux de la base de dades general utilitzant el programador de tasques del sistema, el servei [cron](#). Un requeriment del programa es que cal fer una crida per cada fabrica que volem sincronitzar, ja que a mes de la localització de l'arxiu de configuració es necessari indicar la planta.

```
# Disparadors base de dades general a base de dades de produccio de les fabricques
0,5,10,15,20,25,30,35,40,45,50,55 * * * * java -jar /path/sincrolocals/syncrolocals.jar /path/sincrolocals/config.ini fabrica1 #de General a fabrica1
0,5,10,15,20,25,30,35,40,45,50,55 * * * * java -jar /path/sincrolocals/syncrolocals.jar /path/sincrolocals/config.ini fabrica2 #de General a fabrica2
0,5,10,15,20,25,30,35,40,45,50,55 * * * * java -jar /path/sincrolocals/syncrolocals.jar /path/sincrolocals/config.ini fabricaN #de General a fabricaN
```

Les crides anteriors miren l'arxiu de configuració config.ini i busquen la configuració de la fabrica corresponent, el programa Java fa una selecció de dades de la taula sincros\_general condicionat pel camp servidordesti = fabrica1, d'aquesta manera s'envien totes les sentències a la fabrica1 i s'esborren de la taula de sincronització.

## 1.4. Definició dels serveis.

El llistat de serveis que es poden tenir configurats a una fabrica podria arribar a ser-hi interminable, en aquest projecte només s'inclou els anàlisis en profunditat d'aquells serveis que considerem crítics i essencials pel funcionament de les nostres fabricues.

La consideració de servei crític en el nostre cas fa referència a aquells serveis que afecten directament a la producció, es a dir tots els serveis que en cas de fallida impacten directament sobre el procés productiu de la fabrica, i per tant impedeixen que una part o tot el sistema de producció pugui funcionar amb normalitat.

### 1.4.1. [Noms de domini.](#)

Tot dispositiu connectat a una xarxa requereix d'una adreça IP per tal tenir connectivitat i comunicació amb la resta de dispositius, habitualment utilitzem servidors de noms de domini per convertir els noms de les pàgines web en el seu equivalent d'adreça IP. El servei utilitzat per aquesta tasca es el servei conegut com [BIND](#), un servei que es pot instal·lar i gestionar molt fàcilment.

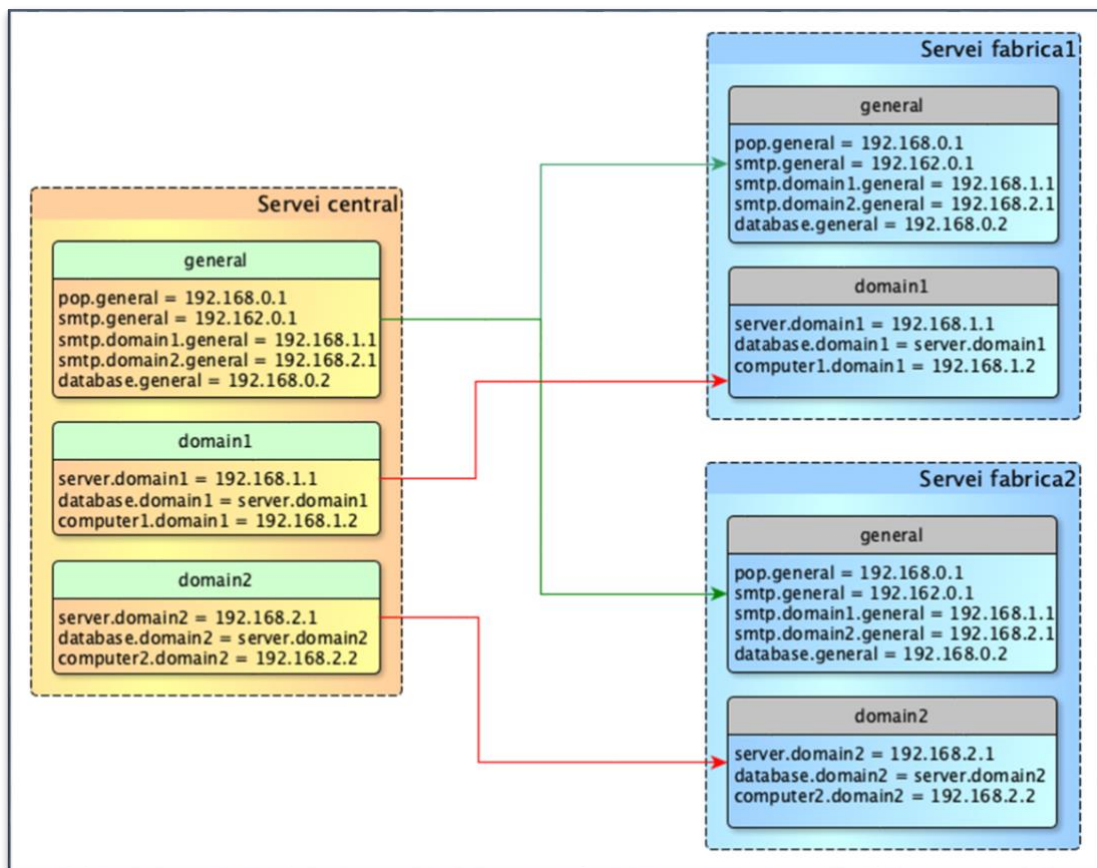
El nostre projecte inclou aquest servei perquè ens proporciona una sèrie de avantatges respecte a utilitzar directament l'adreça IP.

- **Flexibilitat:** El fet de tenir un nom assignat a un servei ens proporciona la flexibilitat de que si aquest servei canvia d'adreça només actualitzant el servidor de noms tots els clients que connecten al servei queden actualitzats, altrament hauríem de configurar tots els clients amb la nova adreça.
- **Creació de dominis interns:** Essent un servei intern podem utilitzar noms de domini propis per definir diferents dominis en funció de la fabrica, o la localització.
- **Balanceig mitjançant DNS ([round robin](#)):** Utilitzem aquesta tècnica per que tots els serveis que tenim publicats a l'exterior tenen una adreça d'accés extern però en el nostre DNS intern aquesta adreça es diferent. En el cas del servei d'enviament de correu el servidor es diferent si l'usuari esta fora de l'empresa, ho mirarem mes endavant.

El servidor de noms es un servei que es pot configurar de forma local a cada fabrica, de fet, per complir requeriments de direcció ha de ser així, hem de procurar mantenir sempre l'autonomia de les fabricues. Però aquest servei es basa en la configuració de zones i les zones poden ser mestres o esclaves, així es com podem aconseguir que cada fabrica tingui el seu propi servei i a l'hora tenir un servei centralitzat, configurant les fabricues com a esclaves del servei central de noms.

Al servidor central de noms de domini tenim totes les zones pròpies creades com a zones mestres, per tant tota la gestió està centralitzada sobre aquest servidor. A les fàbriques tenim un servidor que té les zones necessàries per la fàbrica configurades com esclaves de la zona mestra al servidor central.

A la següent imatge podem veure un esquema de la seva configuració interna i el seu funcionament tal i com hem explicat anteriorment:



Una altra configuració especial que tenim és amb els dominis DNS que tenim publicats a l'exterior, existeixen una sèrie de serveis que requereixen connectivitat des de fora de l'empresa com ara per exemple el correu corporatiu. Al nostre servidor de noms central tenim també publicades aquestes adreces però en comptes de fer referència a adreces externes tenim assignades les adreces internes, d'aquesta manera cap ordinador de l'organització ha de sortir a internet per accedir a un servei propi. Un altra funcionalitat és la modificació del servidor de correu de sortida ([SMTP](#)) on tenim una adreça diferent per cada centre, explicarem això més detalladament a l'apartat del correu.

#### 1.4.2. [Replicació de base de dades.](#)

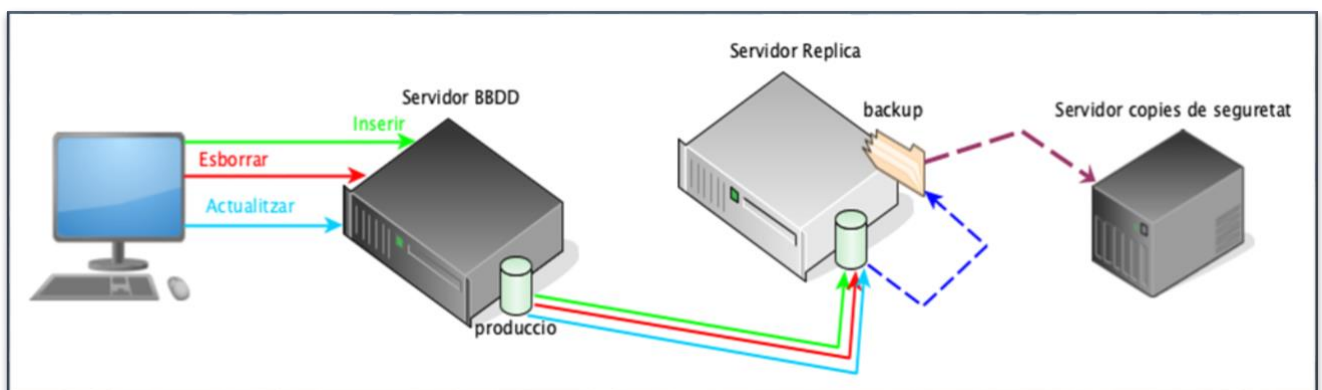
Tenir còpia de seguretat de la base de dades és obligatori per qualsevol organització, hi ha molts sistemes per realitzar una còpia de seguretat de la base de dades de producció però no es recomana utilitzar cap directament sobre aquesta base de dades. Una base de dades que està treballant 24 hores durant els 7 dies de la setmana i els 365 dies de l'any no pot tenir moments de denegació de servei diaris per realitzar una còpia de seguretat, és aquí on entra en joc el sistema de replicació.

El motor de base de dades triat, MyISAM, permet configurar un altre servidor de base de dades com a esclau del servidor mestre. Per configurar aquest servei es necessita modificar l'arxiu de [configuració de MySQL al mestre](#) per assignar les bases de dades (schema) que es replicaran, també cal [inicialitzar el servidor esclau](#) indicant el seu mestre.

El servidor esclau el que fa és que totes les transaccions que es produeixen sobre el mestre seran replicades al servidor esclau, per tant tenim exactament un mirall de la base de dades en un servidor diferent, i és aquest servidor el que utilitzarem per fer les còpies de seguretat de forma segura.

Per realitzar una còpia de seguretat del servidor de base de dades és important garantir que no tenim dades al [cache](#) de les taules per això al servidor replica de la base de dades utilitzem un guió ([script](#)) configurat un cop al dia al programador de tasques del sistema. Bàsicament el seu funcionament és aturar la base de dades de replica garantint que es buida el cache, llavors fa una sincronització de totes les taules a un altre directori que posteriorment s'enviarà a sistema de còpies de seguretat.

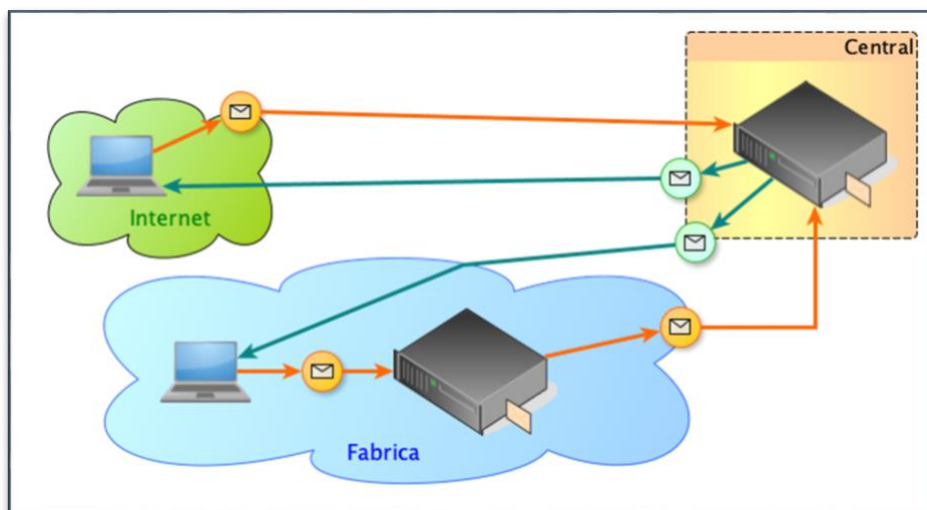
El següent esquema mostra el funcionament de la solució de replicació aplicada al nostre projecte:



### 1.4.3. [Correu electrònic.](#)

El servei de correu electrònic es un dels serveis mes crítics i a l'hora mes complicats de mantenir, es per això que cada cop mes les companyies opten per utilitzar proveïdors externs per aquest tipus de servei. Un servei com aquest es tan ampli i tan complex que entrar a detallar-ho suposa un projecte per si mateix, per tant farem una explicació des de un prisma menys tècnic i profund.

La solució aplicada al correu consisteix en tenir un servidor de correu centralitzat amb accés des de l'exterior on tots els usuaris de la companyia quan estan fora reben i envien correu utilitzant aquest servidor, gracies a les assignacions de noms de domini que tenim a l'exterior. En canvi els usuaris que es troben dins la intranet fem servir el nostre servidor de noms per aconseguir que quan enviïn correu utilitzin el correu local de la seva fabrica, d'aquesta manera l'enviament es molt mes ràpid i no esta condicionat a la velocitat de transferència contra el servidor central. Pel que fa a la recepció tots els usuaris accedeixen únicament a un servidor el servidor de correu central.

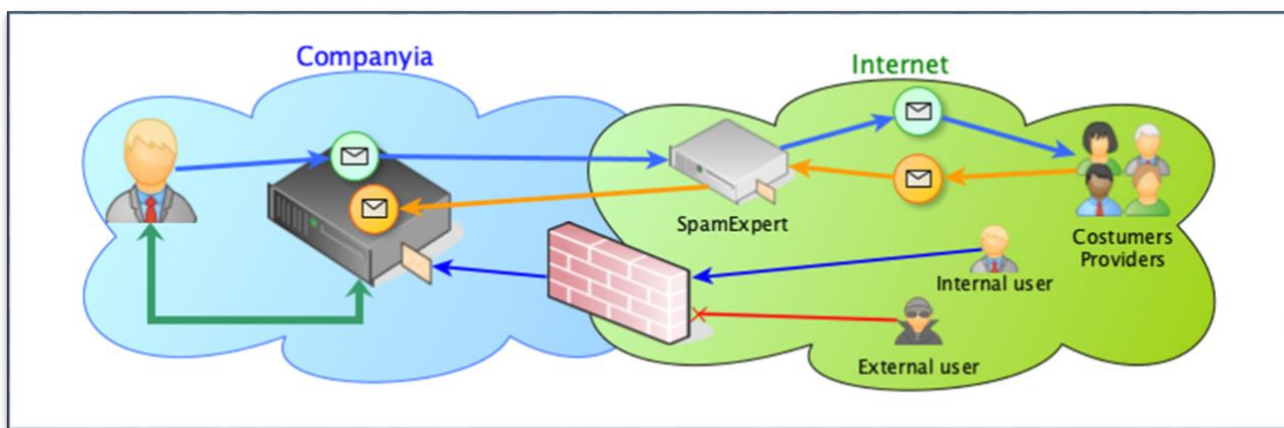


### El servidor de correu corporatiu.

Situat a la seu central de la companyia, es tracta d'un servidor Debian on mitjançant la instal·lació i configuració d'un servei de correu de codi obert anomenat [Postfix](#), aquest servei utilitza el gestor de base de dades MySQL per gestionar les configuracions i comptes de correu el que fa que encara sigui mes interessant. El servidor Postfix es nomes l'encarregat de gestionar el servei de correu d'entrada i de sortida, [IMAP](#) o [POP3](#) per l'entrada i SMTP per la sortida. Per ampliar l'experiència de l'usuari tenim instal·lat un servei de correu web de codi obert anomenat [modoboa](#), que es l'encarregat de proporcionar una interfície web perquè els usuaris puguin veure i enviar correus així com modificar configuracions com per exemple respostes automàtiques.

Tot servidor de correu corporatiu es que disposi d'accés des de l'exterior a de poder-hi rebre el correu dels nostres clients i proveïdors, es per això que requereix una configuració específica a l'encaminador de xarxa de la seu principal, que redirigeix una adreça IP pública cap als ports de correu del nostre servidor.

A nivell de seguretat, es el nostre tallafocs l'encarregat d'evitar que aquest servidor es pugui accedir de forma externa, el tallafocs evita que cap altre servei o port estigui accessible des de fora de la companyia, només podem accedir als ports de correu mitjançant autenticació per usuari i contrasenya o al servidor web per accedir al correu web. Finalment la gestió del correu brossa es un tema molt complicat i en aquest punt es millor subcontractar uns experts, ells s'encarreguen d'analitzar tot el correu d'entrada i de sortida per garantir que el nostre servidor de correu ni envia ni rep correu brossa.



### Els servidors d'enviament de correu a les fabricques.

A cada fabrica tenim un servei de correu molt senzill instal·lat al nostre servidor, aquest servei s'encarrega únicament de gestionar una cua d'enviament de correu, gràcies a la configuració pròpia del servidor de noms una modificació a l'adreça del correu de sortida fa que els usuaris d'una determinada fabrica enviïn els seus correus a aquest servidor dins la pròpia fabrica. El servidor te instal·lat el servei [EXIM](#), un servei de correu molt simple que tenen totes les distribucions de Debian.

El servei esta destinat a fer que els clients de correu dels usuaris a l'hora de fer els enviaments de correu sigui molt més ràpid i no depenguin de la velocitat de la xarxa fins al servidor central. En seus internacionals l'enviament de cada correu podria suposar un minut d'espera, per això vam decidir agilitzar aquesta tasca amb aquest servei.

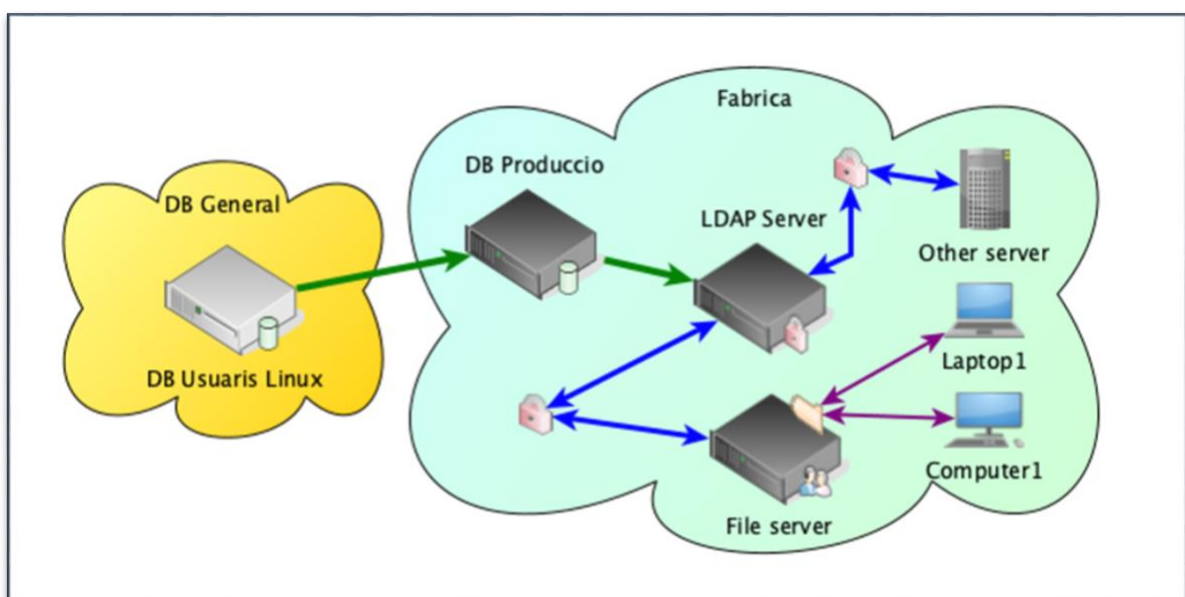
#### 1.4.4. [Gestió d'usuaris.](#)

La part més important de tota organització es la gestió dels usuaris i grups, existeixen dues maneres d'organitzar aquesta informació, de forma local a la fabrica o de manera centralitzada per a tota l'organització. Tota organització te una gran quantitat de perfils d'usuaris diferents i en general quan es tracta d'una multinacional existeixen perfils només vàlids per un àmbit local de fabrica i però també existeixen perfils que han d'existir a totes les fabricues.

Tots els servidors Linux disposen del seu propi e intern gestor d'usuaris, utilitzar aquest implica que tots els usuaris que volen accedir a un servidor s'han de crear al servidor en qüestió, aquest metodologia es molt improductiva i molt ineficient.

Es per això que vam incorporar al nostre projecte un servei de codi obert anomenat [OpenLDAP](#), una implementació que permet l'accés a un servei de directori ordenat i distribuït. Aquest servei habitualment s'utilitza per gestionar la informació dels accessos d'usuari i juntament amb la modificació de la configuració dels servidors es pot utilitzar per gestionar l'autenticació dels usuaris i dels grups. OpenLDAP també es considera una base de dades tot i que el seu sistema d'emmagatzematge es molt diferent.

La gestió de les dades de LDAP es una mica complicada i poc flexible a més cal garantir l'autonomia de les fabricues, per això es necessari tenir un servidor LDAP a cada fabrica. Centralitzar aquest servidor pot arribar a provocar molts problemes en cas de fallada de comunicacions degut a que el cache te una caducitat i un risc molt gran es la pèrdua de la base de dades d'autenticació, el que provocaria la denegació d'accés als servidors i la conseqüent fallada dels serveis.

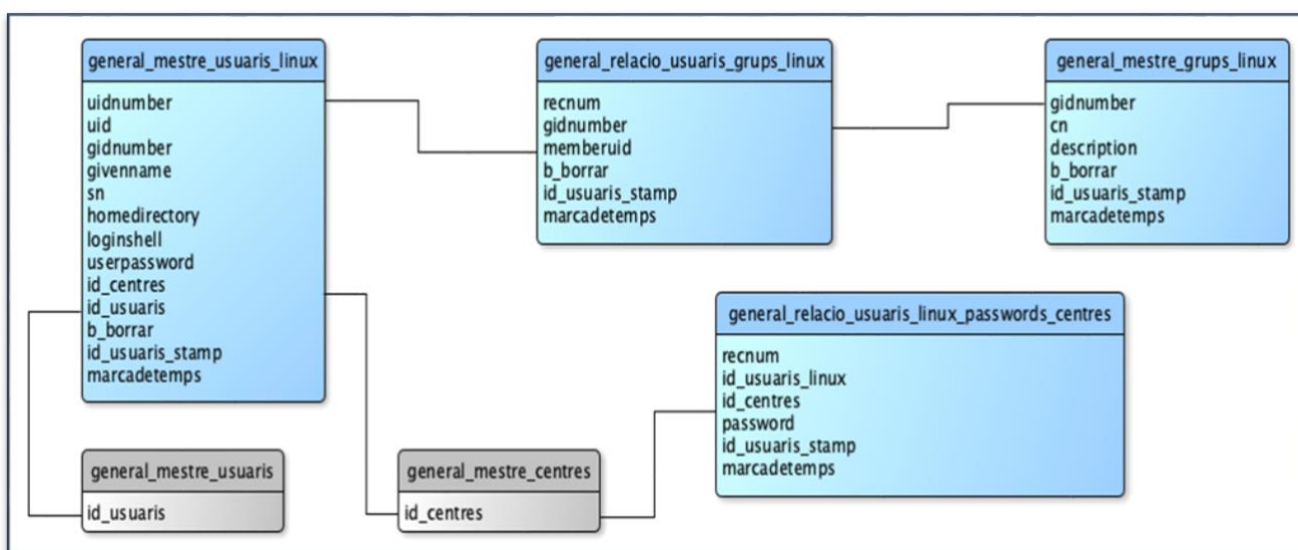


### Base de dades d'usuaris.



Aprofitant el nostre sistema de bases de dades generals vam dissenyar unes taules a la base de dades on tenim tota la informació d'usuaris i grups, aquesta informació es replicada a cada fabrica. D'aquesta manera aconseguim un servidor de LDAP local a cada fabrica amb informació centralitzada.

La particularitat de les dades d'aquestes taules es que no es poden esborrar dades directament i per això totes les taules tenen un camp de control que s'activa quan es vol esborrar una dada important i al cap d'una setmana es netejant les dades, així assegurem que tots els servidors estan actualitzats.



Com es pot veure a l'esquema anterior, els usuaris Linux poden estar relacionats amb usuaris de l'empresa o no, ja que molts usuaris d'ordinador no corresponen a cap usuari (empleat). Tots els usuaris de Linux tenen un centre de treball assignat, el seu centre per defecte.

Els grups es relacionen amb els usuaris mitjançant una taula de relació ja que un usuari pot pertanyer a molts grups i a un grup poden pertanyer molts usuaris, la millor solució aquí es una taula de relació.

Finalment la taula de que relaciona contrasenyes, els usuaris tenen una contrasenya i aquesta es la seva per defecte, nosaltres necessitem que en diferents centres un mateix usuari pugui tenir contrasenyes diferents. Per això la taula de relació amb les contrasenyes, d'aquesta manera podem garantir diferents contrasenyes pel mateix usuari i a l'hora controlem si un usuari esta donat d'alta a un determinat centre, no tots els usuaris poden interactuar amb els diferents centres.

### Creació i actualització de les dades LDAP.

El nostre projecte té una instal·lació i configuració molt particular, on al servidor LDAP de cada fabrica tenim una tasca programada encarregada d'executar un projecte de desenvolupament propi realitzat en [python](#) on traslладem les dades del nostre MySQL local de la fabrica, utilitzant les taules generals, a la base de dades pròpia LDAP. Aquest programari propi es compon de tres arxius:

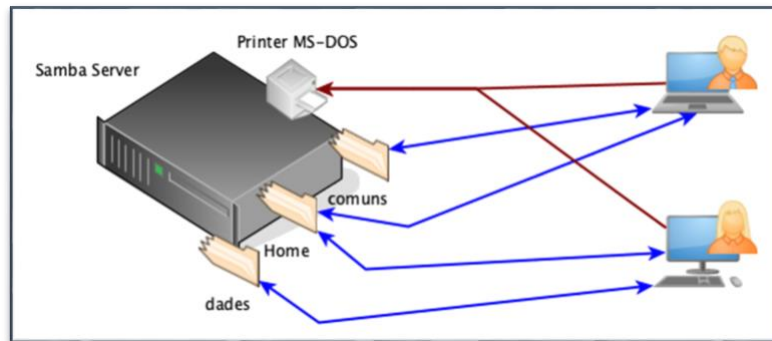
- [Ldap\\_sync.py](#): Encarregat de connectar a les dades de les taules d'usuaris i grups i seleccionar segons la marca de temps tots aquells registres que la seva marca sigui superior a la darrera actualització, tenim aquesta dada a l'arxiu ini. Un cop te les dades seleccionades transforma i insereix o actualitza les dades sobre la base de dades de LDAP.
- [Ldap\\_sync.ini](#): Arxiu de configuració on tenim les dades de connexió amb el servidor i les marques de temps de les darreres actualitzacions d'usuaris, grups o relacions usuaris i grup.
- [Ldap\\_sync.sh](#): Guió encarregat d'executar el programa python i fer l'actualització de les dades de cache del servidor.

Tots els servidors d'autenticació (LDAP) de cada fabrica tenen instal·lat aquest programa i configurat perquè un cop cada hora s'executi, així es com mantenim un servei d'autenticació centralitzat i local a l'hora.

#### 1.4.5. [Compartició de recursos.](#)

Un dels serveis més utilitzats a tota xarxa d'ordinadors es la compartició de directoris, tot i que cada vegada més la tecnologia s'està actualitzant i aquest tipus de servei està quedant molt obsolet. El nostre projecte requereix obligatòriament l'ús de directoris compartits, ja que com hem vist al començament les bases de dades antigues estan basades en arxius els quals s'accedeixen mitjançant els recursos compartits. Per la creació d'aquest recursos utilitzem un servei de codi obert anomenat [samba](#) que consisteix a una implementació del protocol de xarxa SMB i fa possible accedir als recursos compartits utilitzant Linux, Windows o Mac OS.

Als nostres sistemes encara tenim programari crític desenvolupat per MS-DOS, aquest sistema per poder-hi imprimir requereix de la connexió a un recurs compartit, una impressora de xarxa, el servidor samba també es l'encarregat d'aquesta tasca.



La definició de recursos samba permet assignar uns drets determinats sobre cada definició de recurs, per fer això utilitza els usuaris i grups del sistema operatiu que en el nostre projecte es basen en un servidor LDAP.

L'únic inconvenient es que samba requereix la creació de l'usuari a samba, això es fa assignant una contrasenya a l'usuari amb una comanda específica (`smbpasswd -a`), aquesta identificació assigna un usuari a un servidor samba. El servei LDAP es capaç de registrar aquesta informació a la seva base de dades, la problemàtica es que la contrasenya de samba només funciona per un determinat servidor i això ens desmuntaria tota la flexibilitat del nostre muntatge. Actualment els servidors de samba no estan connectats a LDAP i cal crear els usuaris a cada servidor.

### **Millors en fase de desenvolupament.**

La problemàtica de la creació dels usuaris samba a cada servidor es pot resoldre mitjançant una automatització similar a la utilitzada per la creació de les dades del servidor LDAP.

Aquesta millora està encara en fase d'estudi, concretament en la definició de requeriments i les problemàtiques a resoldre.

Utilitzant la taula de relació d'usuaris amb la contrasenya i la fabrica (`general_relacio_usuaris_linux_password_centres`) es viable utilitzar un programa python amb els següents requeriment:

- Determinar la fabrica on s'està executant per recuperar el seu identificador. Això es fàcil perquè a cada base de dades de producció tenim una taula de paràmetres de la fabrica, un dels paràmetres indica aquest valor.
- Seleccionar les dades de la taula de contrasenyes filtrades per la fabrica i la marca de temps superior a la darrera actualització.
- Fer un recorregut per les dades executant la comanda de creació de contrasenya samba per l'usuari seleccionat.

Quan a les problemàtiques a resoldre prèviament a fer la implementació del codi cal resoldre les següents qüestions:

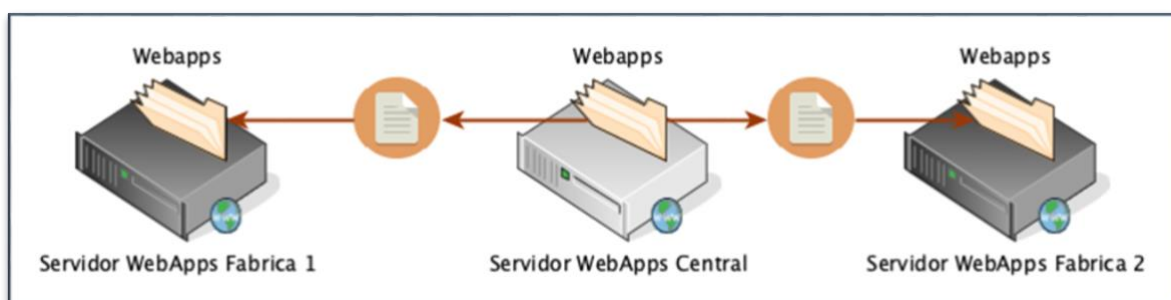
1. La taula de contrasenyes ha d'estar plena i aquesta millora es molt nova encara falten moltes dades.
2. Tractament dels usuaris que no tenen contrasenya, implica altres modificacions ja que hauríem d'assumir que si el camp de la contrasenya es nul·la indicaria que l'usuari no te contrasenya. Actualment la taula no permet el valor nul per aquest camp.

#### 1.4.6. [Transferència d'arxius entre servidors.](#)

Tota companyia amb mes d'una seu i mes d'un servidor per cada seu segurament resulta imprescindible mantenir una sèrie d'informació sincronitzada entre servidors o inclús entre diferents fabriques.

##### Transferència o sincronització unidireccional.

La [sincronització](#) es una tècnica que compara un origen amb un destí i envia totes les diferències de l'origen cap el destí, per defecte la sincronització es una tècnica unidireccional. El servei de sincronització esta disponible a tots els servidors Linux utilitzant un servei anomenat [rsync](#). Els nostres servidors han de tenir moltes dades sincronitzades degut a que son dades corporatives o directoris idèntics a totes les fabriques. Per exemple el directori d'aplicacions web, on el directori mestre hem definit que esta a la seu central de la companyia i es l'origen de les dades, aquesta carpeta es sincronitza periòdicament a tots els servidors web de cadascuna de les fabriques.



Una millora d'aquest sistema es una implementació pròpia realitzada en python que per un determinat directori fa un recorregut dins el primer nivell de directoris i si troba uns arxius de configuració determinats actua en funció d'aquesta configuració i en cas contrari fa sincronització de tot el directori. Aquesta implementació esta destinada a controlar quins arxius i a quines fabriques es sincronitza un determinat directori, el exemple clar es el directori

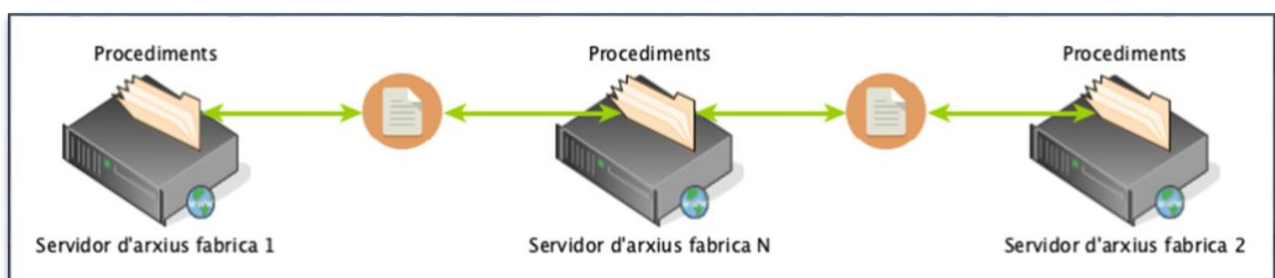
d'aplicacions web on a vegades una determinada aplicació només està disponible o actualitzada en determinades fàbriques però no en totes.

Aquesta implementació consta de tres arxius:

- [Syncapps.py](#): Programa encarregat de fer el recorregut de l'arbre de directoris i realitzar una sincronització en funció de si troba o no configuracions específiques per el determinat directori. En cas de no tenir arxius de configuració sincronitza segons les dades per defecte incloses al codi del programa, tots els arxius a totes les fàbriques.
- [Syncapps.conf](#): Arxiu que indica la configuració específica per un determinat directori, aquí es pot desactivar el sincronisme o especificar a quina o quines fàbriques es farà la sincronització.
- [Syncapps.inc](#): És l'arxiu d'exclusions del programa rsync, serveix per indicar quines coses volem sincronitzar i quines no.

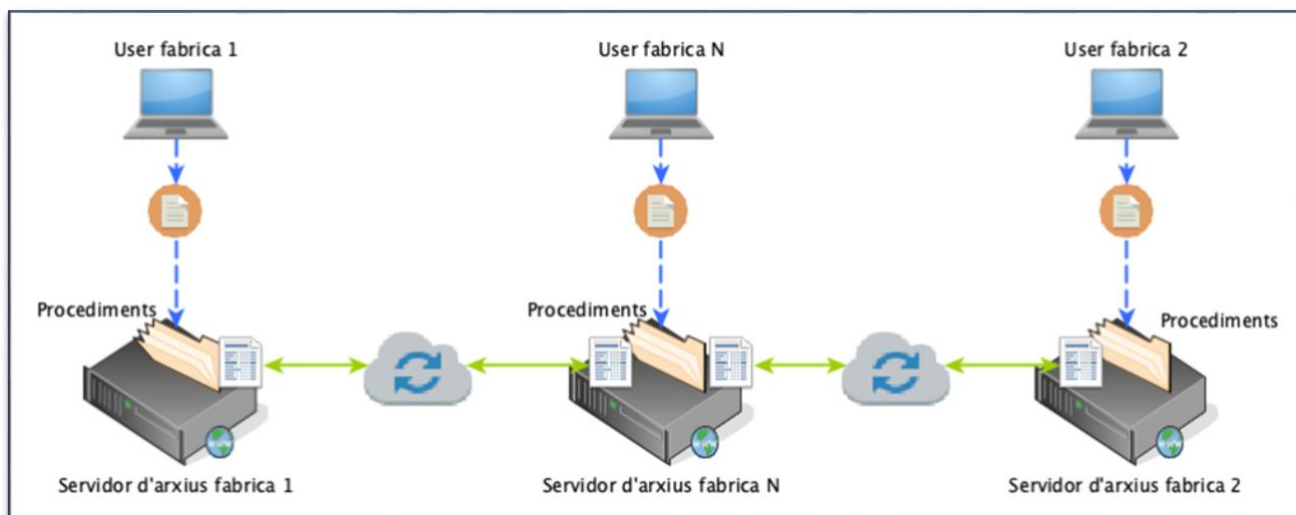
### Transferència o sincronització bidireccional.

Realitzar la sincronització bidireccional és una tasca una mica més complicada ja que requereix d'un programari específic que faci una comparació entre els dos recursos i enviar la informació més actual a cadascun dels recursos aconseguint que les modificacions de cada recurs s'envii a l'altre, habitualment aquest tipus de programari necessita d'un arxiu on porta un índex de modificació i registres de canvi a cadascun dels recursos, d'aquesta manera es capaç de detectar les modificacions i els conflictes que puguin tenir. En aquest tipus de sincronització no existeix un mestre i un esclau, un origen i un destí, tots dos recursos són igual i tenen la mateixa prioritat, aquesta és la major problemàtica.



El nostre projecte requereix tenir recursos que es poden afegir documents des de qualsevol fàbrica i ha d'actualitzar-se a totes les fàbriques del grup, documentació com ara plànols, procediments, etc. Per la realització d'aquesta tasca utilitzem un programari de codi obert i multi plataforma anomenat [Unison](#).

Es recomanable assignar un centre encarregat d'executar la sincronització i en el nostre cas es tracta del servidor de la seu principal, aquest es l'encarregat d'executar l'unison per cada recurs definit i comparar amb cadascuna de les fabricques. El funcionament d'aquest programari es molt senzill, nomes requereix definir un arxiu amb el [perfil de sincronització](#), cal crear un arxiu per cada recurs i servidor.



#### 1.4.7. [Serveis web.](#)

En el nostre camí de migració d'aplicacions d'escriptori cap a sistemes multi plataforma i multi dispositiu es fa imprescindible disposar dels serveis mes flexibles i escalables, aquests son els serveis web, serveis encarregats d'allotjar les pagines i aplicacions web. Dins d'aquest apartat trobem tres serveis que al nostre projecte han adquirit un estat de servei crític i imprescindible, un servidor de pagines web, un servidor d'aplicacions web i un servidor d'intercanvi d'arxius. Totes les fabricques de la companyia tenen instal·lats i configurats els tres servidors, a tots els nivells del projecte cal respectar el requeriment de direcció referent a l'autonomia de les fabricques.

##### **Servidor de pagines web.**

El servidor web de codi obert mes extens arrel del mon es el servidor [apache](#), un servei molt senzill d'instal·lar i configurar, amb molts mòduls o extensions que afegeixen un gran valor a aquest servei.

Les nostres aplicacions web estan basades en [PHP](#) i [JavaScript](#) a mes utilitzem també programes en python que gracies als mòduls d'apache i a la [nostra configuració](#) podem executar-ho tot directament des de qualsevol navegador.

En resum, els nostres servidors de pàgines web utilitzen apache configurat per treballar amb llibreries PHP (actualment treballem amb versió 5.5) i amb programes python, amb això es cobreix pràcticament el 65% de les nostres necessitats web.

### **Servidor d'aplicacions web.**

Moltes vegades no es prou amb una pàgina web per fer funcionar una aplicació sinó que necessitem una programació de més baix nivell però basada en web, això ho realitzem mitjançant aplicacions Java basades en els conceptes dels [Java servlets](#) i [JavaServer pages](#).

Inicialment vam començar utilitzant un servidor de codi obert molt potent anomenat [GlassFish](#), un projecte que va caure en mans d'[Oracle](#) i va estar durant molts anys sense continuïtat, el que ens motivà a migrar a un servei menys potent però més estable com es [Apache Tomcat](#), actualment estem en un procés de migració ja que requereix modificació del codi de les aplicacions per adaptar-les a Tomcat.

El servei d'aplicacions web es l'encarregat de la generació d'informes mitjançant els servlets de Java, així com aplicacions per dispositius mòbils basades en els JavaServer pages. Aquest servei suposa un 30% del nostre servei web, per tant es un servei molt crític, sobre tot per la part de la generació d'informes, ja que s'utilitza per la impressió de totes les fulles de control de la producció.

### **Servidor d'intercanvi d'arxius**

Dins de l'apartat de serveis web ens trobem amb la necessitat de tenir un servidor que ens permeti realitzar l'intercanvi d'arxius de forma eficient, ràpida i segura, tant a nivell local de fabrica com entre les diferents fabricues. Aquest servei suposa només un 5% de les nostres necessitat web per no per això es menys crític que el altres dos.

Probablement un dels protocols més flexibles, segurs i eficients per realitzar aquesta tasca es el protocol [FTP](#) (File Transfer Protocol), un protocol creat específicament per aquest tipus de servei. Els servidors Debian incorporen als seus repositoris oficials un servidor FTP molt senzill i que cobreix els nostres requeriments, es tracta del servidor [ProFTPD](#).

Els serveis FTP utilitzen la pròpia autenticació del sistema operatiu per la validació d'usuari, en aquest cas interactua amb el nostre servei de validació d'usuaris basat en LDAP. A totes les fabricues disposem d'un servidor FTP actiu al servidor d'arxius, d'aquesta manera podem accedir mitjançant FTP o samba segons ens interressi a determinada documentació. L'avantatge principal que en proporciona aquest servei es que es pot accedir des qualsevol tipus d'aplicació de forma fàcil, ràpida i independent de la plataforma a documentació.

### 1.4.8. [Monitorització de la infraestructura informàtica.](#)

Finalitzant amb la definició dels serveis crítics parlarem precisament de com monitoritzar tots aquests serveis que hem definit abans per anticipar-nos a possible problemes o denegacions de serveis.

La bona informació i a temps estalvia molts de problemes, per això es important poder monitoritzar no només els serveis més crítics sinó també la infraestructura informàtica com ara els commutadors de xarxa, els servidors, encaminadors, punts d'accés sense fils i un llarg etcètera que formen tot els nostres sistemes i serveis.

Després de realitzar investigacions i proves amb un sistema de codi obert anomenat [Nagios](#) vam veure que encara necessitàvem algun sistema més robust, senzill i flexible. Es en aquest punt on vam trobar un altre sistema de codi obert que a més disposa d'una versió empresarial anomenat [PandoraFMS](#), un servei que aporta monitorització remota dels protocols de comunicacions més coneguts i estesos ([WMI](#), [SNMP](#), [TCP](#), [UDP](#), [ICMP](#), [HTTP](#), ...) a més de monitorització local mitjançant l'agent de pandora que es pot instal·lar als servidors i equips per tal d'arribar a profunditzar encara més el nivell de monitorització, controlant per exemple l'ocupació del disc, la carrega del sistema operatiu o l'execució d'un determinat servei.

La manca d'informació es un problema però l'excés d'informació també pot ser un greu problema, un servidor de monitorització com aquest disposa de tantes opcions que fa necessari racionalitzar la informació realment crítica que cal mostrar o avisar quan succeeix algun problema. La tendència dels administradors i els tècnics de sistemes es de monitoritzar en excés, el que provoca massa soroll al sistema i dificulta la interpretació dels problemes més crítics, per això la primera tasca que vam realitzar es una definició del nivell de monitorització que volíem al nostre sistema i vam arribar al consens de monitoritzar només els següents serveis:

- Encaminadors, commutadors, i punts d'accés sense fils de la fabrica.
- Servidors crítics: base de dades, servidor web, servidor d'aplicacions, correu i FTP.
- Pantalles d'informació de producció.
- Serveis crítics de producció.
- Sistema de còpies de seguretat.
- Serveis d'Internet i d'interconnexió entre fabricues.



Les alertes del sistema de monitorització es gestionen mitjançant correu electrònic o be un canal de [Telegram](#), en el nostre cas tenim [canals de Telegram](#) per cada fabrica on rebem en temps real les possibles fallades de servei.

En total al nostre sistema tenim un servidor de monitorització per cada fabrica encarregat de vigilar els serveis definits a mes de un servidor adicional que monitoritza tots els serveis centrals.



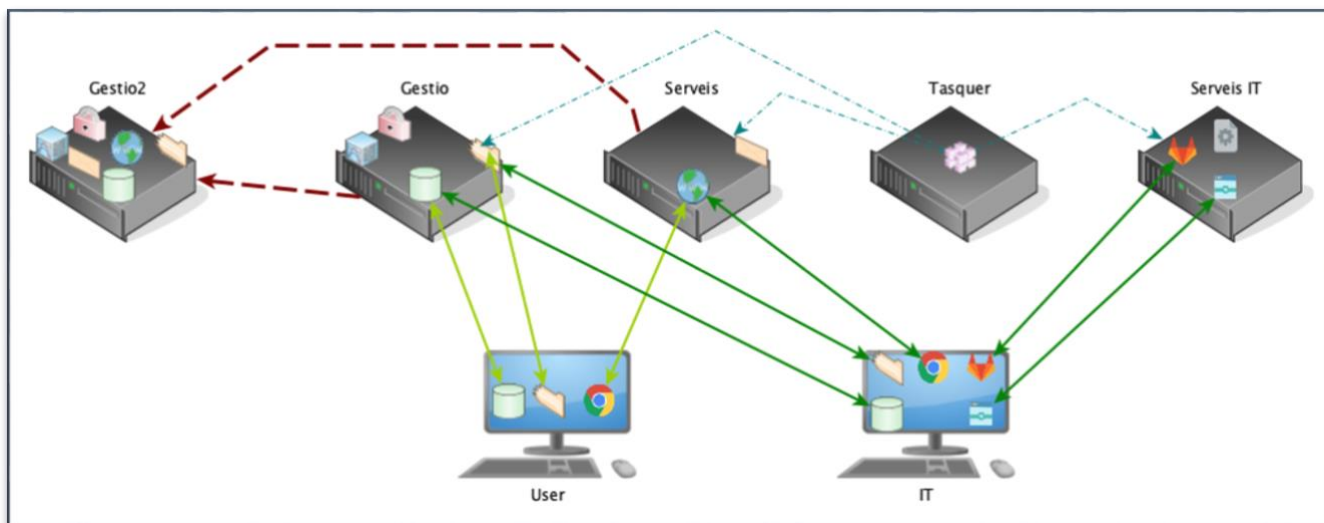
## 1.5. [Distribució dels serveis.](#)

Els serveis necessaris per mantenir un correcte funcionament de tot el sistema informàtic son molts i es necessari fer un anàlisi profund per determinar la quantitat de servidors que tindrem per poder-hi oferir tots els serveis. Quan parlem de distribució no només hem de pensar en el balanceig de la carrega dels servidors sinó també en la cripticitats dels serveis i l'escalabilitat dels sistema. En el nostre projecte tenim serveis dependents de versions determinades com ara la base de dades que poden impedir el progrés d'altres serveis com els web, per exemple, es per això que vàrem realitzar un anàlisi dels serveis necessaris a cada fabrica, no només els crítics, per tal de definir com els distribuïrém.

- **Servidor principal (gestio):** Encarregat de donar servei de base de dades, tant MySQL com DataFlex, per tant encarregat també del servei de compartició. Al ser-hi el servidor

principal de la fabrica inclou els servei de noms de domini, d'assignació d'adreces IP i el servidor d'autenticació LDAP.

- **Servidor web (serveis):** Els serveis web son el nostre futur quan a aplicacions per tant convé tenir-los tots agrupats en un servidor que puguem implementar les millores futures dels serveis. Aquest equip ofereix el servei de pagines i aplicacions web, apache i Tomcat, a mes del servidor FTP i el servidor d'enviament de correu de la planta.
- **Servidor replica (gestio2):** No nomes fa la replica de MySQL sinó que es un mirall exacte del servidor principal amb els serveis web, d'aquesta manera en cas de fallada dels dos anteriors nomes cal canviar l'adreça IP i reduïm el temps de denegació de servei metres reparam el servidor caigut.
- **Servidor de tasques:** Es el servidor destinat a realitzar totes les tasques de sincronització i manteniment de dades.
- **Servidor informàtica:** Destinat a allotjar els serveis exclusius del departament d'informàtica, com ara servidor de versió i el servidor de llibrerries.



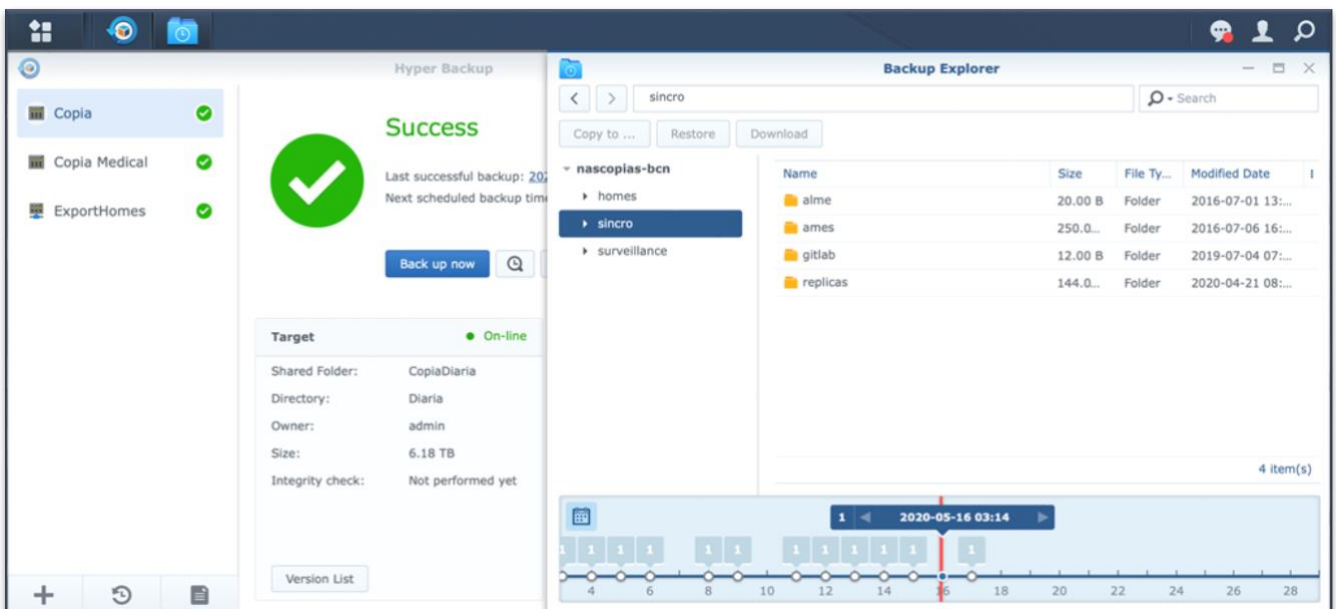
Aquesta distribució ens permet no nomes balancejar la carrega dels servidors sinó també incrementar la seguretat davant una fallada de serveis, amb el maquinari actual i utilitzant servidor virtuals es fàcil millorar les prestacions dels servidors i fins i tot mantenir un sistema de còpies de seguretat tot el servidor que en cas de fallada seria bastant ràpid restablir els serveis. Tot i així convé fer aquesta distribució perquè a vegades una tasca de manteniment es capaç d'ocupar el 100% de processador i provocar la denegació de serveis del servidor.

## 1.6. Sistema de còpies de seguretat.

Inicialment les còpies de seguretat les fèiem amb un servidor Linux utilitzant únicament rsync i eines de compressió, però això a mes de ineficient va arribar un punt en que era impossible degut a que la compressió de les dades per la còpia diària superava les 24 hores, un altre punt negatiu era la recuperació de dades de la còpia, que es feia molt complicat. En el moment en que ja no érem capaços de realitzar una còpia diària dins vam començar a investigar i provar un sistema d'emmagatzematge de xarxa (NAS) de la companyia [Synology](#).

Els dispositius NAS de Synology incorporen una distribució pròpia de Linux que inclou tot un sistema operatiu amb manteniment basat en web anomenat DSM. Aquest sistema te un repositori d'aplicacions les quals disposa d'aplicacions destinades a còpia de seguretat de dades, còpia de seguretat de maquines virtual i un llarg etcètera de serveis com servidor web, servidors de base de dades o ERP. La introducció d'aquest dispositiu als nostres sistemes ens ha aportat una millora important no només en el camp de les còpies de seguretat sinó per altres serveis.

La configuració del sistema de còpies de seguretat realitza un sincronisme diari de les dades al servidor de còpies i aquest mitjançant una aplicació específica de realitza una còpia de seguretat incremental, intel·ligent i espaiada que manté 76 versions de les dades, l'equivalent a 1 any.



Gràcies a aquest serveis hem aconseguit guanyar en eficiència ates que necessitem menys capacitat de disc per tenir més versions de còpies i en proporciona més serveis.

## 1.7. Projectes de futur.

En el món de la informàtica s'avança molt ràpidament i això fa que sempre es presentin oportunitats per millorar l'eficiència de serveis existents o fins i tot la necessitat de nous serveis que fins ara no es consideraven d'interès. Actualment hi ha cinc projectes de millora dels nostres serveis, tres han començat a principis d'any i els altres dos derivats de la crisi del Covid19.

### 1.7.1. Sistema de videoconferència propi.

El sistema actual està basat en dispositius físics, un sistema de videoconferència per sales de reunions amb un maquinari destinat i poc flexible que obliga a fer reunions utilitzant les sales i sense possibilitat de connectar cap equip que no sigui els propis del sistema. Un altre problema d'aquest sistema és que no suporta més de 5 sales connectades simultàniament.

Cada cop és més necessari poder realitzar reunions ràpides i prou flexibles com perquè qualsevol persona amb qualsevol dispositiu i estigui on estigui, a una sala, a casa seva o a un hotel es pugui incorporar a la reunió sense problemes.

Abans de començar amb la cerca de possibles sistemes cal definir els requeriments que ha de complir:

- Idealment hauria de ser de codi obert i amb el mínim cost possible.
- Sistema multi plataforma i multi dispositiu.
- Poder-hi realitzar compartició de pantalla.
- Suport per al menys 10 usuaris connectats simultàniament.

Amb aquests requeriments que no són els primers per poder-hi començar a valorar serveis ens han sortit dos interessants que actualment els tenim instal·lats i acabant les configuracions per poder començar a realitzar proves en breu.

- [OpenMeetings](#): Un sistema de codi obert desenvolupat per Apache que disposa de capacitat de donar servei de formació en línia, vídeo conferència i disposa d'una funcionalitat de pissarra virtual. Es tracta d'un programari multi plataforma però no és multi dispositiu, només està disponible per ordinador.
- [Jitsi](#): Es tracta d'una plataforma de codi obert que disposa de servei web i dona la possibilitat d'instal·lar el nostre propi servidor Jitsi, amb lo que tenim més control sobre la seguretat de les sales. Jitsi compleix amb tots els requeriments fins ara definits.

## 1.7.2. Sistema de formació en línia.

Fins ara tota la formació impartida per l'empresa es feia a les aules, mai fins la situació viscuda pel Covid19 s'havia plantejat la possibilitat de realitzar formació en línia. Aquest projecte esta just començat, encara no s'han definit de forma clara els requeriments del sistema però si que hauria de complir alguns mínims:

- Sistema de codi obert.
- Multi plataforma, preferentment basat en web.
- Creació de cursos amb classes per videoconferència, connexió amb els sistemes de comentats anteriorment (Jitsi o OpenMeetings preferentment).
- Creació de cursos asíncrons, on l'alumen pugui seguir el curs amb l'ajut d'un fòrum del curs.
- Possibilitat de creació de formularis d'avaluació i lliurament d'exercicis.

En aquest moment hem començat a mirar un servei de codi obert bastant utilitzat com es el [moodle](#) i tenim realitzada la instal·lació d'un servidor amb una configuració bàsica per començar les proves del servei.

## 1.7.3. Contenedors de serveis i aplicacions.

Els nostres sistemes son molt dependents encara de determinades versions i això ens obliga a limitar la versió del sistema operatiu que podem instal·lar, la qual cosa ens esta limitant altres millores per culpa de les versions o hem de crear mes servidors del compte. La solució a aquest problema es senzilla i molt antiga, es tractar de la creació de contenidors amb els serveis, d'aquesta manera es desvincula la versió del sistema operatiu i podem mantenir un contenidor preparat i igual per totes les fabriques que a mes manté la versió que necessitem de cada servei sense.

Hi ha diferents gestors de contenidors per nosaltres estem avaluant la utilització de [Docker](#), que es un sistema de virtualització a nivell de sistema operatiu, permet fer exactament el que nosaltres necessitem, crear un contenidor amb un determinat servei i les seves llibreries i que sigui independent del sistema operatiu. L'únic inconvenient que te aquest sistema es que no es un sistema de codi obert, si be si que disposa d'una versió de la comunitat que es gratuït.

Aquest projecte s'ha endarrerit l'inici de les proves i encara no tenim definits quins serveis hem de crear el seu contenidor tot i que el primer candidat es la nostra base de dades.

### 1.7.4. Gestió d'adreces IP.

Un projecte començat i aturat de moment es el de la gestió de les adreces IP, actualment totes les fabricques disposen d'un rang d'adreces lliures per les visites del personal d'altres fabricques però això te la problemàtica de que si algú connecta un dispositiu a la nostra xarxa sempre disposarà d'una adreça i per seguretat això es un problema. Idealment tots els dispositius mòbils de l'empresa haurien de tenir una assignació de l'adreça IP segons la seva adreça física ([MAC](#)).

El projecte consta d'unes taules a la base de dades general que gestionen les adreces físiques els equips i relacionen aquestes dades amb una determinada fabrica i adreça IP. Cada dispositiu tindrà per cada fabrica que pot visitar una adreça concreta.

La part en desenvolupament del projecte es un programa que a partir de les dades de la base de dades a cada fabrica sigui capaç de generar l'arxiu de configuració de servidor d'adreces IP.

### 1.7.5. Gestió d'esdeveniments i registres.

Un dels projectes mes necessaris per tota infraestructura informàtica juntament amb la monitorització, comentada anteriorment i que tenim resolta, es el anàlisis dels registres i esdeveniments de diferents dispositius i serveis.

En fase de proves inicials tenim a una fabrica un servidor [Elasticsearch](#) un servidor que basa moltes funcionalitats en codi obert però que no es tracta d'un programari de codi obert, te versió gratuïta. El servidor Elastic s'encarrega d'inserir a la seva base de dades la informació dels registres d'esdeveniments de dispositius com ara servidor, encaminadors, commutador de xarxa i treballar aquesta informació per donar interpretació dels esdeveniments.

## 2. [Les maquines de control de producció.](#)

En els següents dos apartats parlarem de com hem resolt la problemàtica d'utilització de programari propietari mitjançant la implementació de programari propi que realitza les funcions mes eficientment i integrades al nostre sistema. Inclouran explicacions a un nivell teòric de cada projecte sense profunditzar en la solució de codi font dels projectes per motius de confidencialitat i propietat intel·lectual del programari. He triat aquests dos projectes perquè son els que donen un major valor afegit a la feina de tot el departament d'informatica.

La pulvimetal·lúrgia es una tecnologia la qual mitjançant una barreja de pols base ferro i barrejada amb altres materials es posa en un motllo, es premsa per obtenir la forma de la peça i després es sinteritza per donar-hi duresa. Aquest es el procés mes basic, barrejar, premsar i sinteritzar. Per aquest procés basic es necessari disposar d'una bona informació, controlant el procés basic estalviem problemes amb els complexes.

Es per això que els majors reptes de desenvolupament passen per tenir una informació i controls acurats en aquest punts de la producció, el nostre sistema inclou moltes millores basades en la utilització de sistemes de codi obert però nomes resumirem aquelles que han suposat la substitució de programari propietari de molt alt cost i basat en Windows.

## 2.1. Els serveis de premsat.

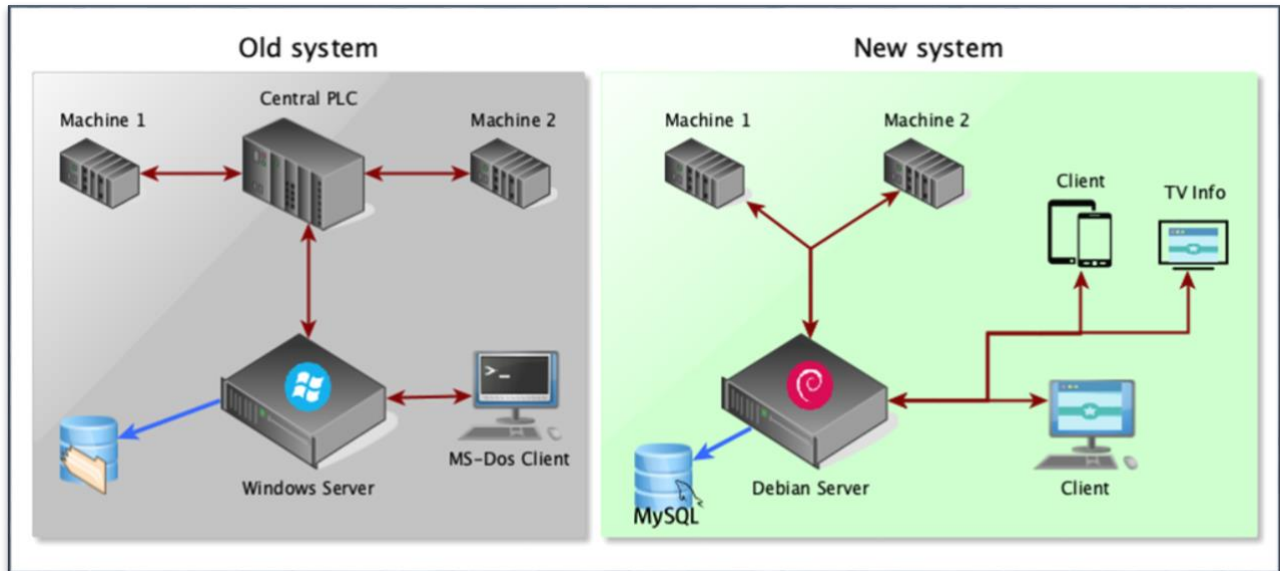
La quantitat de peces premsades, els temps de treball de les maquines, les aturades i averies son informació que resulta imprescindible per poder-hi portar un bon control de la producció. Totes les maquines tenen uns autòmats programables (PLC) que registren informació dels diferents sensors de la maquina i es pot interactuar en funció dels esdeveniments de cada sensor.

El sistema estava compost per un servidor Windows amb una aplicació que mirava els esdeveniments de les maquines i els introduïa en una base de dades propietària. A mes del servidor tenia una aplicació client desenvolupada en software propietari i MS-DOS. Totes les maquines es connectaven a un autòmat central que feia de pont entre el servidor Windows i les maquines, aquesta solució implicava cablejar cada maquina cap a l'autòmat central.

Dins el projecte de migració a eines de codi obert resultava vital i necessari millorar aquest sistema amb una implementació mes integrada. Un nou sistema que substitueix el servidor Windows per un servidor Debian i la implementació d'un dimoni de Linux a la banda servidor, implementat en python molt mes estable, segur e integrat que a mes treballa directament amb les nostres bases de dades MySQL. Aquesta implementació requereix de la creació de classes per poder-hi connectar he interactuar amb els autòmats de cada maquina i eliminar l'autòmat central que feia de pont amb les maquines i aquesta implementació ha requerit una feina d'investigació dels diferents proveïdors dels autòmats per crear els protocols de comunicacions de cadascun d'ells.

La part del client consisteix en tot un desenvolupament web que interactua amb cadascuna de les maquines controlant el seu estat, la producció que esta fent i els esdeveniments produïts.

Finalment una millora afegida a aquest sistema ha sigut la creació d'una visualització web amb la situació de tot un grup de màquines. Aquesta implementació es mostra a la zona de producció en unes pantalles perquè els operaris poden veure ràpidament si alguna màquina de la seva zona té un problema, a més de la situació mensual de la producció per aquell grup de màquines.



## 2.2. Els serveis de sinteritzat.

Al sinteritzat hi ha uns forns llargs de cinta per on passen les peses i mitjançant unes regulacions de temperatura i gasos s'aconsegueixen les atmosferes necessàries per aplicar la duresa desitjada a cada peça.

L'antic sistema consistia en un sistema propietari anomenat [Wizcon](#) basat en programari [SCADA](#) que permetia la creació d'interfícies d'usuari amb els paràmetres dels forns, temperatura, gasos i alarmes. Aquest programari portava inclosa tota la implementació de protocols de comunicació necessaris per connectar amb els diferents dispositius que regulen els forns.

Les dues problemàtiques principals d'aquest sistema es que requereix una llicència d'utilització de programari per cada forn a més de llicència per cada desenvolupador i d'altra banda no es podia integrar amb les nostres bases de dades, per tant les regulacions s'havien de fer a mà i la consulta de dades només es podia fer mitjançant el programari específic.

Per tal d'evitar la dependència d'un programari propietari, molt car i poc integrat així com per millorar la qualitat de la informació i la integració amb el nostre sistema de producció s'inicia

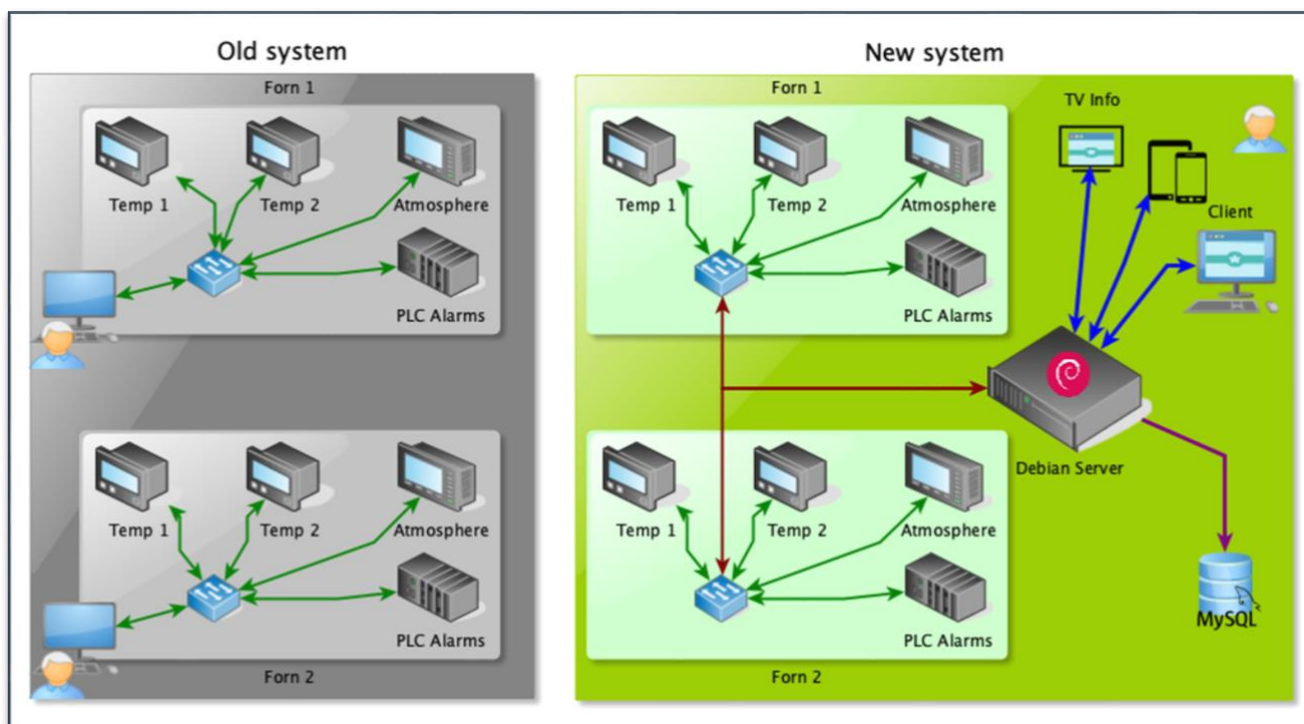


un projecte de migració de tot el sistema de control dels forns a un desenvolupament propi utilitzant eines de codi obert.

El nou projecte implica la creació de tot una sèrie de llibreries de comunicacions realitzades en Java per aconseguir la comunicació amb els diferents dispositius que regulen els forns, llibreries pròpies per comunicació amb els sensors de temperatura, gasos i per l'autòmat del forn.

El projecte d'integració es basa en traslladar els petits ordinadors solitaris que controlen de forma aïllada cada forn en un únic servidor Linux per la fabrica encarregat de connectar amb cadascun dels forns i dels seus dispositius mitjançant un dimoni fet en python. El dimoni utilitza el programari específic Java de control dels forns i les llibreries de comunicacions. El servidor s'encarrega no només de controlar els forns sinó d'enregistrar les dades en el nostre servidor de base de dades MySQL, els nostres clients ens obliguen a tenir un històric de dades de al menys 15 anys.

La integració de tot el sistema proporciona la possibilitat d'automatitzar els valors que s'han d'aplicar a cada forn en funció de la peça que s'introduirà. A més per la banda del client es crea un projecte web que permet la consulta i modificació de les propietats dels forns. A banda de l'estalvi econòmic que suposa aquest projecte val la pena mencionar la millora de productivitat i qualitat que suposa la integració amb tot el sistema de producció.



## 3. Les eines per l'equip de desenvolupadors.

Quan es marca una direcció a seguir en temes de desenvolupament es important la implicació de tot l'equip de desenvolupadors, aportant idees i prenent les decisions que ens ajudaran a construir un camí comú. Per tal de poder consolidar un equip es important que tots els seus membre es sentin integrats, es per això que vam decidir tots junts una les regles i normes pel desenvolupament de les futures aplicacions.

Una de les principals decisions que marca la dinàmica a seguir quan al desenvolupament es la prioritació de treballar en entorns multi plataforma evitant sempre que sigui possible la creació d'aplicacions d'escriptori client-servidor. Aquestes dues premisses que semblen molt simples son força complicades i ens motivà a crear una documentació amb els estàndards de desenvolupament, una guia inicial pels desenvolupadors.

### 3.1. Eines de desenvolupament.

La part del programari destinada a ser executada per part dels servidors es un apartat molt important i en aquest punt es on hem de seguir la mateixa línia del projecte, els nostres servidors son distribucions de Debian per tant les eines destinades a programar a la banda dels servidors han de ser compatibles amb els nostres sistemes operatius.

Un altre punt molt important a l'hora de fer desenvolupaments d'aplicacions amb bases de dades SQL es tenir molt en compte el temps de resposta de les sentencies SQL. El llenguatge de programació SQL es molt potent, disposa de moltes funcions i podem arribar a obtenir dades molt treballades nomes amb l'ús de sentencies SQL, això es un greu error, cada sentencia SQL que supera un determinat temps processant dades impacta negativament sobre el rendiment del servidors SQL.

Els servidors SQL donen prioritat a les sentencies de selecció de dades per davant de la resta, per tant, hem hagut d'establir una política a l'hora de realitzar sentencies evitant fer treballar al servidor de base de dades i processant les dades per programa desviant així la carrega contra el servidor d'aplicacions o l'equip de l'usuari final.

La reutilització del codi es vital per tot equip de desenvolupadors per tant un altre de les nostres prioritats es intentar en la mesura del possible generar classes el mes estàndards possibles per tal de reutilitzar aquest codi, un exemple clar seria una classe per connectar amb la base de dades amb les funcions bàsiques que normalment s'utilitzen.

Pel que fa a les eines utilitzades per realitzar desenvolupaments destinats a ser-hi executat en els nostres servidors es defineixen tot una sèrie de programari i components.

### Llenguatges de programació.

Cada llenguatge de programació te les seves avantatges i funcionalitats respecte als altres i existeixen molts llenguatges multi plataforma, en el nostre cas volem prioritzar les aplicacions tipus web perquè ens donen millor rendiment i flexibilitat i això condiciona una mica els llenguatges que utilitzarem. Definir quins llenguatges volem utilitzar i aconseguir destresa en la seva utilització ens ajuda a aconseguir millorar la productivitat a en de realitzar projectes nous, i es per això que els llenguatges que utilitzarem seran:

1. **PHP i JavaScript**: Utilitzats per les aplicacions web, son el llenguatges encarregats de implementar les funcionalitats de les aplicacions.
2. **HTML**: Per la realització de l'entorn visual de les aplicacions web, per aquest llenguatge utilitzem **CSS** per enriquir-ho i per mantenir un homogeneïtat de les pàgines.
3. **Python**: Enfocat al tractament de les dades i utilitats de traspàs d'informació bàsicament. Tot i que a mida que ens anem familiaritzant veiem un potencial molt gran per aquest llenguatge dins la nostra organització.
4. **QT**: Utilitzat al desenvolupament de l'entorn visual de les aplicacions d'escriptori basades en Python.
5. **Java**: Aplicacions que requereixen arribar a un nivell inferiors a la **capa d'aplicació** i per tant requereix interactuar amb el maquinari. També utilitzat per aplicacions mes complexes que serien molt complicades o impossibles de realitzar amb els anteriors. I finalment per la implementació dels Servlets.
6. **JavaFX**: Que s'utilitza per la creació de les aplicacions Java d'escriptori.
7. **JasperReports**: Tot i que no es pot considerar un llenguatge de programació ja que el llenguatge es Java, ho posem dins aquest apartat perquè es el llenguatge utilitzat per la creació d'informes.

### Entorns de desenvolupament.

Per tal de homogeneïtzar les eines establim els entorns de desenvolupament perquè tots els membres de l'equip utilitzin les mateixes eines.

- [NetBeans](#), entorn multi plataforma i de codi obert que incorpora compatibilitat per diferents llenguatges de programació, en el nostre cas s'utilitza per tot el desenvolupament d'aplicacions web (HTML, CSS, PHP i JavaScript).
- [PyCharm](#): Utilitzem la versió community amb parts de codi obert per les implementacions de aplicacions en llenguatge Python.
- [Qt Creator](#): Un entorn de desenvolupament gràfic per la creació de tota la part visual de les aplicacions desenvolupades mitjançant Python. No es un programari de codi obert però disposa de versió gratuïta.
- [Eclipse](#): Un entorn de desenvolupament multi llenguatge i multi plataforma basat en llicència de codi obert que utilitzem per realitzar els desenvolupaments d'aplicacions basades en Java i on integrem l'entorn de desenvolupament gràfic de Java, JavaFX en la seva versió de codi obert, [Openjfx](#).
- [Jaspersoft Studio](#): Entorn de desenvolupament basat en Eclipse, i per tant en codi obert, per la creació dels informes de JasperReports.

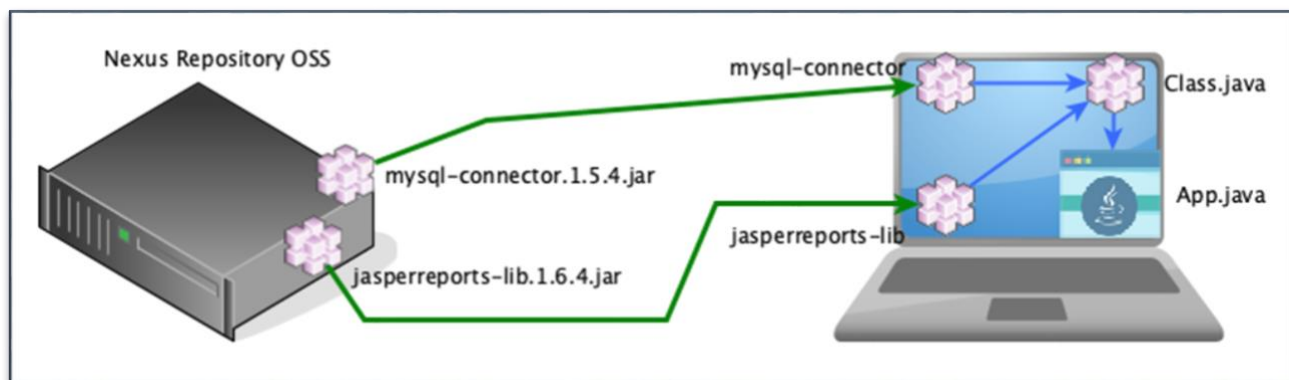
### 3.2. [Gestió de llibreries i versionat d'aplicacions.](#)

Un dels principals mal de caps a l'hora de realitzar desenvolupaments son les llibreries i el versionat de les aplicacions, cadascun d'ells te un impacte diferent i per tant les solucions aplicades també son diferents.

#### **Gestió de les llibreries.**

A l'hora d'implementar projectes nous sempre es un problema les diferents versions de les llibreries que utilitzarem, el fet d'utilitzar una versió determinada pot tenir un impacte imprevisible tant a nivell d'execució com de compatibilitat a nivell de servidors com a nivell dels clients finals.

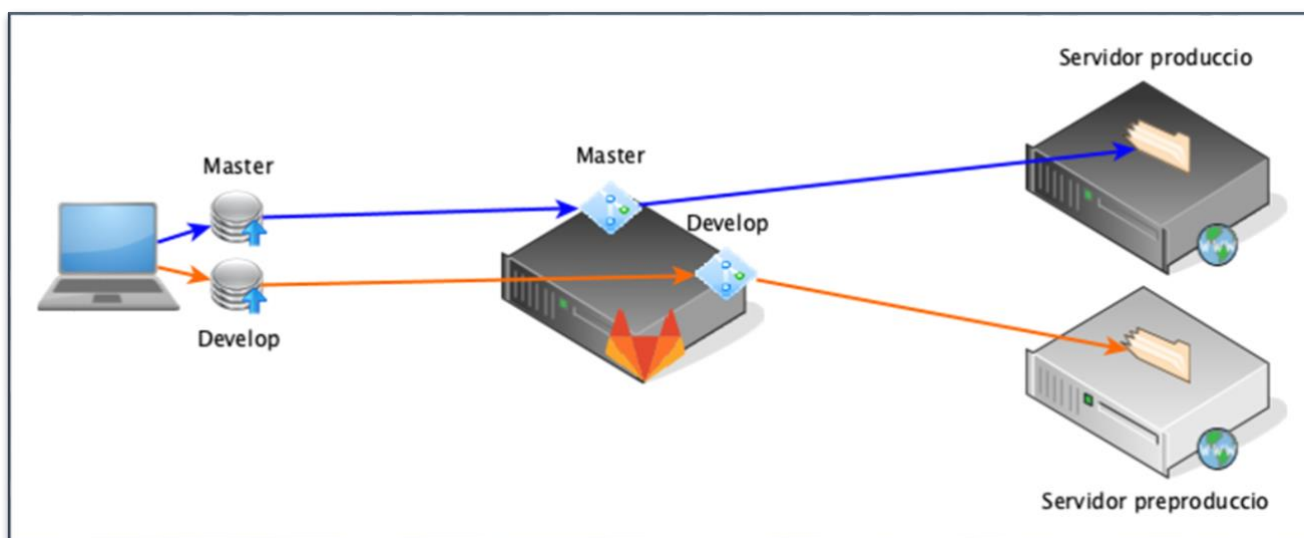
Es per això que dins el nostre sistema hem incorporat un servidor de repositoris capaç de gestionar les versions de les llibreries que tenim validades entre altres funcionalitats. Incorporem al nostre projecte un servidor Linux amb d'instal·lació del servei [Nexus Repository OSS](#), un gestor de repositoris gratuït compatible amb repositoris Java, Python o Docker entre d'altres. Aquest servidor es l'encarregat de garantir que tot l'equip de desenvolupadors utilitzem les mateixes llibreries.



### Versionat d'aplicacions.

Els projectes de desenvolupament habitualment incorporen mes d'un desenvolupador a l'equip de desenvolupament per tant habitualment mes d'un membre de l'equip treballa modificant el codi font i incorporant noves funcionalitats. La manera de gestionar els canvis i els possibles conflictes de forma eficient, així com portant un bon control, es utilitzant un servidor de versions, d'aquesta manera es poden establir fluxos de treball per diferents membres i centralitzar les versions del codi font sobre aquest servidor.

Per raons evidents de confidencialitat i per tal de protegir les nostres implementacions hem incorporat a la nostra infraestructura un servidor propi de versionat utilitzant un servidor Linux i la instal·lació local del servidor de versions de codi obert [GitLab](#). Aquest servei en permet a mes de tenir el versionat controlar el cicle de d'integració continua del nostre programari.



### 3.3. [Els escenaris de desenvolupament.](#)

Finalment un punt bastant important es que l'equip de desenvolupament disposi d'un escenari de proves durant el desenvolupament el mes semblant als servidors de producció possibles. [Vagrant](#) es un programari de codi obert molt potent que permet la creació d'escenaris basant-se en la virtualització, el que fa que requereixi un programari específic de virtualització, que en el nostre cas es [VirtualBox](#). Utilitzant Vagrant i una eina per l'automatització de d'instal·lació, [ansible](#), l'equip de desenvolupadors disposa al [nostre repositoris git d'un escenari](#) preparat per utilitzar idèntic als servidors de producció, el que facilita el desenvolupament d'aplicacions i el control d'errors.

## 4. Les eines pels usuaris.

Històricament cada vegada que Microsoft ens obligà a actualitzar el sistema operatiu en patit moltíssim per adaptar les nostres aplicacions. La nostra relació amb Microsoft va començar a deteriorar-se a partir de Windows 8 degut a la immensa quantitat de problemes que ens va provocar adaptar-nos, però el punt crític i que va suposar trencar amb Microsoft com a sistema operatiu per usuaris va se amb la sortida de Windows 10 on hem tingut molts mes problemes de compatibilitat i a cada actualització ha anat a pitjor fent que no funcionin moltes de les nostres aplicacions.

Els nostres sistemes inclouen moltes aplicacions on el seu cicle de vida es molt llarg i es per això que no ens podem permetre que ningú no ens obligui a actualitzar les aplicacions per seguir la seva corrent. Es evident que trencar definitivament es impossible degut a que existeix molt de programari de tercers que ens obliga a tenir Microsoft però la nostra aspiració es arribar a tenir un parc d'equips amb del 75% de sistemes operatius basats en Linux.

### 4.1. El sistema operatiu.

Un cop decidit que Windows no serà el nostre sistema operatiu pel futur ens hem de plantejar quin serà aquest sistema operatiu. A nivell de servidors ja vam realitzar un estudi on es va decidir utilitzar Debian, aquest sistema operatiu es també vàlid per a utilitzar-se com a sistema d'escriptori però la principal problemàtica que presenta es la compatibilitat amb el maquinari nou i que les versions de programes i llibreries validades son massa antigues per a un sistema operatiu d'usuari.

Si Debian es vàlid per servidor però esta massa antiquat, podem triar un sistema basat en Debian que estigui mes actualitzat. Existeixen moltes distribucions de Linux que estan basades en Debian però hem de prioritzar que es tracti d'un projecte consolidat i estable, no estem sortint de Microsoft per empitjorar, per això ens vam fixar en Ubuntu, una de les distribucions mes estables i consolidades que existeixen i a mes basat en Debian.

Ubuntu te una versió que es la considerada com mes estable distingida per les sigles LTS (Long Term Support) que significa que es una versió que inclou un suport d'actualitzacions de 5 anys, considerem que aquesta es la versió que ens aporta garanties i seguretat a mes de que ens permet mantenir les versions del programari mes controlades.

L'entorn d'escriptori es un punt crític per als usuaris, es per això que dins de les versions basades en Ubuntu l'usuari pot triar el seu entorn on es senti mes còmode. El mes important es que l'usuari accepti aquest canvi. Les versions valides son:

- Ubuntu LTS: La nostra distribució per defecte, i que incorpora l'escriptori [GNOME](#).
- Linux Mint: Una distribució basada en Ubuntu LTS que porta per defecte l'escriptori [cinnamon](#), un escriptori molt lleuger que es molt semblant a Windows 7. Aquesta distribució es molt recomanada per equips molt antics, ja que funcionen realment molt be i fluids.
- Kubuntu LTS: La única diferencia amb Ubuntu es que la K indica que porta per defecte l'escriptori [KDE](#), un escriptori molt mes pesat que els anteriors però molt semblant a Windows 10.

## 4.2. Programari específic.

La migració d'un sistema a un sistema operatiu de codi obert implica que moltes aplicacions de tercers ja no estan disponibles, llavors cal buscar alternatives robustes per poder-hi adaptar-nos al nou sistema. En aquest punt es descriuen les necessitats dels usuaris i amb quin programari han sigut resoltes.

**Navegador web:** Habitualment els navegadors web son un mal de caps per la compatibilitat amb determinades pagines com ara les governamentals. Les distribucions triades incorporen per defecte [Mozilla Firefox](#) un navegador de codi obert i multi plataforma amb un grau de compatibilitats equivalent a [Microsoft Edge](#) o [Internet Explorer](#).

**Client de correu:** Una de les eines mes utilitzades a les organitzacions es un bon client de correu electrònic, que si pot incorporar alguna funcionalitat extra com ara gestió de contactes

o calendari ens ajuden amb la nostra feina diària. Aquestes necessitats queden totalment resoltes amb [Mozilla Thunderbird](#), programari de codi obert i multi plataforma el que ens ajuda a poder-hi mantenir una mateixa solució de client de correu per a tota l'organització.

**Paquet ofimàtic:** La creació de documents de text, fulles de càlcul o presentacions son les necessitats mes imprescindibles per a tota (Comunitat, 2020) organització, per resoldre aquestes necessitats utilitzem un paquet de codi obert molt complet i compatible amb [Microsoft Office](#) anomenat [LibreOffice](#).

**Edició d'imatges:** Determinats usuaris necessiten realitzar tractaments a les fotografies o imatges, per aquesta tasca utilitzem el programari de codi obert [Gimp](#).

**Edició de vídeos:** Usuaris mes avançats necessiten poder-hi realitzar muntatges i edició de vídeo, per aquesta tasca tenim tres programes depenent del nivell de l'usuari i el tipus de vídeo:

1. [OpenShot](#): Programari de codi obert molt senzill per realitzar muntatges simples, l'alternativa ideal per al usuaris que coneixen [Windows Movie Maker](#) o [iMovie](#).
2. [Shotcut](#): Un editor també de codi obert una mica mes complet que l'anterior però també es menys intuïtiu.
3. [Lightworks](#): Programari multi plataforma alternatiu a [Pinnacle Studio](#), [Vegas Pro](#) o [Adobe Premier](#). No es programari de codi obert però ofereix una versió gratuïta molt completa.

**Reproducció multimèdia:** No sembla que aquest tipus de programari sigui d'us professional però poder reproduir vídeos de proveïdor i clients es una tasca mes bastant habitual, en aquest cas utilitzem un programari de codi obert compatible amb molts formats, [VLC media player](#).

**Creació d'esquemes i plànols:** Sobre tot els usuaris de manteniment necessiten programari per realitzar esquemes i plànols en dos dimensions, una alternativa a [Microsoft Visio](#), tot i que no es tan complet utilitzem un programari de codi obert anomenat [Dia](#) que cobreix per complet les nostres necessitats en aquest sentit.

**Creació de diagrames:** La creació de diagrames de flux, UML o diagrames de decisió son una necessitat habitual i per resoldre-la utilitzem un programari de codi obert anomenat [yEd](#).

**Maquines virtuals:** Moltes de les nostres aplicacions encara necessiten Microsoft Windows per poder-hi treballar i treballar amb Windows 10 es impossible per raons de compatibilitat per això utilitzem una maquina virtual de Windows 7 amb el programari de codi obert [Virtualbox](#). Per la llicència de Windows, fem una [baixada de la versió OEM de Windows 10 Pro](#).



**Capturador de pantalla:** Enriquir documents, correus o cursos amb captures de pantalla es una necessitat que hem de cobrir, per resoldre-la utilitzem un programari de codi obert que trobem al repositoris d'Ubuntu anomenat [Flameshot](#).

## 5. Formació.

A un projecte d'aquestes característiques es fa totalment necessari incloure un programa formatiu tant per al personal del departament d'informàtica com pels usuaris. La formació es un dels pilars mes importants per la companyia, amb un programa formatiu que inclou reconeixements i compensacions, per tant no resulta gens complicat preparar i realitzar formacions perquè això forma part de l'ADN de l'empresa.

En aquest apartat repassarem les formacions que s'han considerat mes importants, cadascuna d'aquestes requereix una preparació prèvia de documentació, presentacions i practiques, nosaltres les resumirem explicant a quines persones va dirigides cada formació i perquè, a mes dels seus objectius.

### 5.1. Gestió, manteniment i instal·lació de servidors.

Quan el negoci principal de la companyia no es la informàtica els membres del departament d'informàtica solen ser molt mes versàtils i polivalents, la figura de l'administrador de sistemes existeix però no clarament definida i es important que en absència d'un administrador els tècnics de sistemes puguin realitzar tasques de manteniment de servidors, identificació i resolució d'incidències.

Aquesta formació dona per suposat que els alumnes tenen uns coneixements mínims de sistemes operatius Linux.

#### Objectius:

- Conèixer com es realitza la instal·lació d'un servidor: Quines opcions hem d'instal·lar i quines particions hem de crear i perquè s'han de crear.
- Instal·lar i configurar tots els serveis crítics amb l'objectiu de conèixer quins o on estan els arxius de configuració de cada servei.
- Identificar les incidències analitzant el comportament del servidor i els seu arxius de registre.
- Solucionar les incidències identificades prèviament.

- Aprendre a crear usuaris, grups i recursos compartits.
- Realitzar els manteniments del servidor. Identificar l'ocupació dels discos, actualització de paquets, creació de tasques programades, etcètera.

### **Metodologia.**

Els alumnes realitzaran la instal·lació completa d'un servidor en entorn local de proves utilitzant virtualbox. Un cop instal·lat el servidor farem simulacions d'incidències i han de ser capaços de identificar la incidència i resoldre-la.

## **5.2. Base de dades MySQL.**

Enfocat principalment a l'equip de informàtica i a usuaris molt avançats que tenen la confiança del departament d'informàtica, aquests usuaris son capaços de realitzar els seus propis informes utilitzant directament les bases de dades i això allibera de feina al departament. Tot i que es defineix com una única formació tenim dos graus de profunditat del curs, en funció de si els alumnes son personal d'informàtica o no.

### **Objectius generals.**

- Conèixer les bases de dades de la companyia.
- Aprendre el llenguatge de consultes, SQL.
- Optimització de consultes.
- Creació de vistes.

### **Objectius per informàtics.**

- Crear de taules, disparadors, funcions i procediments d'emmagatzematge.
- Definició dels índex de les taules i entendre com funcionen els índex de cerca i els de control de registre únic.
- Identificació i resolució d'incidències amb les consultes, identificar consultes que generin bloqueig i usuaris actius.
- Anàlisis i millora de les consultes lentes (slow queries).

### **Metodologia.**

Explicarem a nivell teòric el llenguatge SQL, les funcions i com s'estructuren les sentències per obtenir els millors resultats.

Els alumnes treballaran amb una la base de dades de probes utilitzant el programari de codi obert [DBeaver](#) per la realització de consultes, la creació de les taules, els disparadors i les vistes.

Es presenten varies sentències SQL ineficients i el alumnes han d'aconseguir millorar les sentències redissenyant per complet la consulta.

### 5.3. [Escenaris i automatització.](#)

El desenvolupament d'aplicacions que s'executaran al servidor tenen la principal problemàtica en que els sistemes operatius del servidor i dels desenvolupadors son diferents, amb llibreries diferents.

Els desenvolupadors han de tenir i conèixer eines per poder-hi crear escenaris de probes de forma rapida i utilitzant les mateixes configuracions i llibreries que els servidor de destí on s'executaran les aplicacions.

#### **Objectius.**

- Conèixer la eina Vagrant i les seves comandes bàsiques.
- Conèixer Ansible per l'automatització.
- Aprendre a crear escenaris de desenvolupament amb Vagrant.
- Millorar la creació d'escenaris utilitzant de Ansible.
- Compartir dades entre l'escenari i la maquina de desenvolupament.
- Compartir els escenaris de desenvolupament.

#### **Metodologia.**

Utilitzarem els ordinadors dels desenvolupadors per la creació d'un escenari adaptat i configurat per fer les probes de desenvolupament d'aplicacions web amb bases de dades MySQL. Els alumnes han de realitzar la instal·lació tant del servidor com dels seus serveis utilitzant Vagrant utilitzant pel seu aprovisionament ansible.

Es realitzaran tres practiques:

1. Creació de l'escenari i configuració del Vagrantfile.
2. Instal·lació dels paquets utilitzant ansible.
3. Compartició dels directoris de dades amb l'escenari i configuració dels paquets per utilitzar aquestes dades. Directori de base de dades i pagines web.

## 5.4. Protocols de comunicació.

La creació d'aplicacions destinades a comunicar-se amb els diferents dispositius i perifèrics requereix conèixer molt bé els diferents protocols de comunicacions. Aquesta formació pretén que els desenvolupadors més experts puguin compartir els seus coneixements amb la resta de companys.

La formació es centrarà en mostrar com s'han resolt les problemàtiques internes per comunicar les aplicacions amb dispositius com PLC, balances, lectors de codi de barres, entre d'altres.

### Objectius.

- Aprendre els diferents protocols de comunicacions que utilitzem.
- Programació de llibreries de comunicacions.
- Interactuar entre les aplicacions i els dispositius.

### Metodologia.

Treballarem en la creació d'una llibreria simple per comunicació RS232 amb una balança i la posterior utilització en una aplicació per agafar el pes d'una peça.

## 5.5. Creació d'aplicacions web.

El canvi de direcció en el desenvolupament d'aplicacions requereix que els desenvolupadors recordin els llenguatges de programació web. Aquesta formació refrescarà la memòria i ens ajudarà a establir els criteris comuns a l'hora de la creació d'aplicacions web.

### Objectius.

- Creació de formularis web utilitzant HTML i CSS.
- Desenvolupament d'aplicacions web utilitzant PHP i JavaScript.
- Conèixer les llibreries de programació: [jQuery](#) i [Bootstrap](#) amb la plantilla [AdminLTE](#).
- Estructuració dels projectes d'aplicacions web.
- Creació de projectes amb multi idioma.

### Metodologia.

Busquem un projecte real i realitzem el desenvolupament d'aquesta aplicació en format curs formatiu. Els alumnes han de ser capaços de desenvolupar una part de l'aplicació i entre tots al final ajuntar-ho i veure el resultat.

## 5.6. Creació d'informes.

Dirigida principalment als desenvolupadors es dirigeix també als usuaris mes avançats de la nostre companyia. Els requisits d'aquesta formació es assistir al curs de base de dades. La combinació dels dos cursets ajudaran als alumnes a explorar les dades del nostre sistema mitjançant la creació d'informes amb JasperReports.

### Objectius.

- Conèixer l'entorn de creació d'informes Jaspersoft Studio.
- Crear informes simples amb dades del sistema.
- Crear informes amb subinformes.
- Entendre com treballar amb paràmetres als informes.
- Treballar amb Informes amb en multi idioma.
- Creació d'informes complexes amb paràmetres, subinformes i formules complexes.

### Metodologia.

Els alumnes instal·laran i configuraran l'entorn de programació de informes, es realitza una presentació de tot l'entorn i es fa comparació amb l'anterior eina Crystal Reports per tal de mostrar als alumnes les diferències.

Es trien una sèrie d'informes fets en Crystal Reports i es migraran a Jasper entre tots els alumnes. Al final els alumnes han de presentar dos informes, un completament nou fet amb Jasper i un migrat d'un informe existent fet en Crystal.

## 5.7. [Programari basic d'usuari.](#)

Dirigit principalment als usuaris que migren de sistema operatiu a Linux, la formació es divideix en dues parts, la part inicial enfocada a acostumar l'usuari al treball amb el nou sistema operatiu mostren les diferències amb Microsoft Windows i l'altra que mostrarà les eines bàsiques que s'utilitzaran amb aquest nou entorn.

### Objectius.

- Aprendre a instal·lar del sistema operatiu Linux.
- Ensenyar els requeriments i les conseqüències de la migració.
- Conèixer les diferències entre Windows i Ubuntu.
- Aprendre a treballar amb el nou sistema d'arxius.

- Mostrar els diferents entorns d'escriptori.
- Personalització i configuració del sistema operatiu.
- Instal·lació d'aplicacions.
- Comandes bàsiques de terminal.
- Aprendre a treballar amb el programari basic com el client de correu o les captures de pantalla.
- Ensenyar als alumnes a treballar amb les maquines virtuals de Windows per accedir al programari antic i intercanviar informació entre la maquina física i la virtual.

### Metodologia.

Els alumnes realitzaran una instal·lació completa del sistema operatiu amb l'objectiu de que puguin adoptar-ho a nivell particular, tractar de convèncer als alumnes de les millores que ofereix treballar amb Linux respecte Windows.

Es practicarà amb les tasques diàries dels usuaris, com ara consultar documents de la xarxa i locals, enviar i rebre els correus, adjunts i organització de la informació dins el directori Home.

## 5.8. [Eines d'ofimàtica, LibreOffice.](#)

Les eines ofimàtiques son un dels programaris mes utilitzat dins de la companyia es per això que aquesta formació s'ha de dirigir a tots els usuaris de l'empresa. Les eines ofimàtiques estan en constant evolució es per això que al llarg dels anys s'ha anat modificant i millorant amb les noves funcionalitats que ofereix el programari.

Aquest curset es necessari dividir-ho en quatre cursos per el seu grau d'importància i la necessitat de separar-ho en funció dels diferents programes.

### 5.8.1. Curs d'introducció al LibreOffice.

#### Objectius.

- Descarrega i instal·lació de l'aplicació.
- Aprendre a treballar amb el nou l'entorn.
- Consell i les bones practiques per la creació de documents, reducció d'imatges, tipus de lletra.
- Com posar el noms dels arxius i claus de seguretat per l'accés o l'edició.
- Ortografia, gramàtica i les configuracions regionals.

## **Metodologia.**

Ensenyarem als alumnes com instal·lar i configurar el paquet per intentar que adoptin aquest com a paquet ofimàtic a nivell personal.

Es fa una presentació de tot el programari mostrant les diferències amb altres programaris, es centra la formació en que els alumnes puguin aprendre com esta distribuïts els menús i le barres d'eines al nou entorn.

S'imparteixen també consell per tal de millorar a l'hora de la creació de qualsevol document aplicant seguretat i millorant la informació que s'insereix.

### **5.8.2. Curs del document de text.**

#### **Objectius.**

- Aprendre a treballar amb els formats utilitzant els estils.
- Creació de plantilles de documents.
- Treballar amb el editor de formules.
- Crear documents amb índex de contingut i referències.
- Treballar amb documents tècnics amb diagrames de flux.

#### **Metodologia.**

Per la realització d'aquesta formació existeix el requisit que els alumnes hagin assistit al curs d'introducció.

Utilitzant el tipus de documentació que habitualment es genera dins de la companyia els alumnes aprendran a utilitzar el programari practicant amb el tipus de documents que habitualment generen.

### **5.8.3. Curs de la fulla de càlcul.**

#### **Objectius.**

- Entendre a tractar el full de càlcul com si fos una petita base de dades.
- Aprendre a treballar amb els estils i formats per millorar la productivitat.
- Treballar amb formats condicionals.
- Aprendre a utilitzar les formules i els gràfics.
- Importació i exportació de dades amb el sistema de producció.

- Millorar els documents amb llistes de contingut i protecció de manipulació.
- Crear taules dinàmiques.

### **Metodologia.**

Per la realització d'aquesta formació existeix el requisit que els alumnes hagin assistit al curs d'introducció.

El curs consta de 5 practiques en un format progressiu, les practiques s'utilitzen pel seguiment de l'avaluació continua, cada practica es una continuació de l'anterior fins arribar a la practica final on tenim la suma de tot els conceptes que s'han après.

Habitualment les fulles de càlcul estan creades de forma molt ineficient i en aquesta formació es pretén que els usuaris millorin la seva productivitat fent fulles mes potents però mes senzilles.

## **5.8.4. Curs de presentacions.**

### **Objectius.**

- Aprendre a treballar amb els patrons de diapositives.
- Treballar amb gràfics i diagrames.
- Entendre com s'ha de treballar amb arxius multimèdia, imatges, vídeo i àudio.
- Treball amb transicions i animacions.
- Configurar les opcions de la presentació.
- Treballar amb les notes de presentació.

### **Metodologia.**

Per la realització d'aquesta formació existeix el requisit que els alumnes hagin assistit al curs d'introducció.

Els alumnes treballaran en la creació d'una presentació establint una sèrie de directrius perquè la presentació tingui un format corporatiu. La practica final consisteix en que els alumnes han de fer-hi una presentació de tema lliure utilitzant tots els mecanismes apresos.

## **5.9. Programari avançat d'usuari.**

Existeix un grup dels nostres usuaris que necessiten utilitzar eines mes avançades per realitzar les tasques de preparació de vídeos o catàlegs pels clients. Altres tasques son la



creació d'esquemes i diagrames de flux, per això necessiten conèixer el programari específic com ara els editors de imatge o de vídeo, la creació de diagrames o treballar amb esquemes.

### **Objectius.**

- Crear i editar vídeo utilitzant l'eina OpenShot.
- Treballar amb l'edició d'imatges utilitzant Gimp.
- Creació de diagrames mitjançant yEd.
- Creació d'esquemes i plànols utilitzant Dia.

### **Metodologia.**

El curs es distribueix en quatre practiques separades per programari, es tracta de fer un vídeo presentació, un catàleg de mostres amb fotografies de peces editades i millorades, creació d'un diagrama de procés i finalment un plànol de distribució de maquines d'una secció.

Els alumnes aprendran amb la practica, utilitzant el programa i veien les similituds i diferencies amb altres programes utilitzats.

## **6. Impacte econòmic del projecte.**

Per mesurar l'impacte econòmic d'un projecte tan gran com aquest seria necessari tenir dades concretes de hores dedicades a la programació i posta en marxa de cadascuna de les solucions implementades, també tenir el cost per hora de cadascun dels membres que intervenen en cada part, dades que son complicades de mesurar i d'obtenir per confidencialitat.

Un altre punt molt important i també molt difícil de mesurar es la millora de productivitat que aporta cadascuna de les solucions implementades.

Es per això que aquest impacte econòmic nomes tindrà en compte els conceptes que es puguin comptabilitzar de forma inequívoca, únicament dades tangibles.

### **6.1. Les eines de servidor.**

A la banda dels servidors l'impacte econòmic es pot distribuir en tres conceptes, maquinari, programari i energia, aquest últim al tractar-se de dades poc tangibles no profunditzarem.

#### **6.1.1. Maquinari.**

La virtualització de servidors ens proporciona no nomes un estalvi en servidors sinó també en commutadors de xarxa i sistemes d'alimentació ininterrompuda (SAI).

Actualment tenim un total de 40 servidors físics que tenen un total de 162 serveis virtuals, d'aquests serveis virtuals 47 son escriptoris remots i els tractarem diferent. Per tant tenim un total de 115 servidors virtuals que si fossin maquines físiques suposaria tenir 75 maquines mes de les actuals.

Cada servidor te dues fonts d'alimentació, normalment de 550w i dues connexions a xarxa connectades a diferents commutadors per seguretat. Això suposa un increment de les necessitats de SAI i commutadors. Per tant l'increment en 75 maquines incrementa en 82.5kw i suposa un increment en 8 commutadors de xarxa:

- 550w per 2 fonts i per 75 maquines fan 82.500w o el que es el mateix 82.5 Kw. Per calcular la necessitat farem una extrapolació del sistema actual, per 40 equips, 44Kw tenim un total de 16Kva. Per tant el nostre increment suposa necessitar un total de 30 Kva. Si actualment tenim un total de 8 equips de 2 Kva aquest increment suposaria tenir un total de 15 nous equips, amb un cost aproximant de 1100€ per equip.
- 75 punts de xarxa dividit en commutadors de 24 ports son 3,125 commutadors es a dir 4 però necessitem doblar aquest valor per seguretat, per tant 8 commutadors, cada commutador per als CPD ens costen al voltant de 400€. Actualment es pot donar servei amb 4 commutadors als 40 servidors.
- El maquinari dels servidors es obligatori disposar de doble font d'alimentació i controladora RAID. Nosaltres treballem amb servidors DELL i un servidor senzill amb aquestes característiques te un cost de uns 1800€ aproximadament.
- La renovació dels diferents maquinaris s'estableix que de mitjana es cada 5 anys.

Concepte	Real	Estimat	Estalvi	Estalvi anual
Servidors	72.000€	207.000€	135.000€	27.000€
Commutadors	1.600€	4.800€	3.200€	640€
SAI	8800€	25.300€	16.500€	3.300€
<b>Total</b>	<b>82.400€</b>	<b>237.100€</b>	<b>154.700€</b>	<b>30.940€</b>

### 6.1.2. Programari.

A la banda dels servidors cal comptabilitzar l'estalvi en llicències de sistema operatiu i les llicències del servidor de base de dades.

- Considerem llicències de Windows 7 per servidors simples i Windows server per als servidors més complexes. Actualment ja utilitzem molts Windows 7 per màquines que realitzen alguna tasca que requereix programari Windows.
- Pel que fa als sistemes operatius, d'aquest 115 servidors només 65 són servidors Debian, i dels 65 només 15 requeririen sistema operatiu servidor la resta poden ser sistemes operatius d'usuaris. En principi les llicències serien OEM per tant es renovarien amb les màquines, cada 5 anys.
- Els servidors d'arxius a més requereixen llicència per cada usuari que es connecta, tenim un total de 1200 usuaris al nostre sistema dels quals uns 200 es connecten a varis servidors, hem de contar tenir llicències per uns 2000 usuaris en total.
- La base de dades també és un servei que cal valorar, els sistemes de pagament a més requereixen a més d'una llicència per cada servidor, nosaltres tenim 7 fabricques, es necessari tenir llicència per usuari nominatiu, això vol dir llicència per cada dispositiu que es connecta a la base de dades, el nostre parc d'ordinadors es compon per 1165 ordinador d'usuari i 162 servidors per tant un total de 1327 connexions.
- El servidor de correu és un servei que cal valorar l'impacte econòmic de tenir-ho com a servei extern. Actualment tenim un total de 654 bústies de correu, habitualment els serveis de correu facturen per bústies, usuaris. Per fer l'estipulació hem agafat [Microsoft Exchange](#) el servei més econòmic que es factura mensualment a 3,40€ per usuari al mes.

Concepte	Real	Estimat	Estalvi	Estalvi anual
Llicències servidor	14.550€	35.550€	21.000€	4.200€
Usuaris servidor	0€	70.600€	70.600€	14.120€
Llicència base dades	0€	6.384€	6.384€	1.277€
Usuaris base dades	0€	282.651€	282.651€	56.530€
Correu corporatiu	0€	134.070€	134.070€	26.814€
<b>Total</b>	<b>14.550€</b>	<b>529.255€</b>	<b>514.705€</b>	<b>102.941€</b>

## 6.2. Les eines d' usuaris.

Per mesurar aquestes dades només tindrem en compte les llicències de sistema operatiu i les llicències del paquet d'ofimàtica que són el que realment entren dins de la migració, la resta de programari no es tindrà en compte tot i que pot arribar a suposar un estalvi considerable degut a que sempre hem utilitzat aquest programari lliure.

- En total comptabilitzats tenim a tota l'organització 1165 ordinadors dels quals només uns 295 poden treballar sense utilitzar Windows com a sistema operatiu, en el futur aquesta xifra anirà creixent però un encara tenim aplicacions dependents de Windows al nostre sistema de producció.
- De la totalitat d'ordinadors uns 678 necessiten utilitzar programari ofimàtic i d'ells 183 ja tenen llicència de Microsoft Office per raons de compatibilitat amb programari de tercers. Per tant ens deixa un total de 495 usuaris que valorarem la solució ofimàtica de pagament.
- El programari d'usuari es fa una valoració a 8 anys, que és el temps de vida de mitja estimada d'un ordinador a la nostra companyia.

Concepte	Real	Estimat	Estalvi	Estalvi anual
Sistema operatiu	130.500€	174.750€	44.250€	5.531€
Paquet ofimàtic	45.750€	169.500€	123.750€	15.469€
<b>Total</b>	<b>176.250€</b>	<b>344.250€</b>	<b>168.000€</b>	<b>21.000€</b>

## 6.3. Resum.

Comptabilitzant l'estalvi anual de cadascun dels apartats valorats aquest projecte suposa un estalvi anual a la companyia de 154.881€ i esperem que en el futur les dades d'estalvi anual encara siguin més bones a mida que anem migrant més dels nostres sistemes i aplicacions a programari lliure o aplicacions a mida basades en desenvolupaments de codi obert.

Recordem que aquest estalvi es només tenint en compte les dades que podem comptabilitzar i per valorar els preus s'han utilitzats preus de mercat del mes de Març de 2020.

## 7. Debilitats, Amenaces, Fortaleses i Oportunitats (DAFO).

En l'àmbit de la gestió de projectes es bo fer un exercici DAFO per identificar punts a millorar i punts on s'ha d'estar mes alerta.

### Debilitats:

Encara el projecte no te un recolzament clar i oficial per part de la direcció, per això moltes vegades resulta mes difícil implementar noves solucions.

El poder de decisió del departament d'informàtica encara esta per sota del de direcció de fabrica.

El cap del departament d'informàtica informa directament a direcció general però no te consideració de CIO ni forma part de la direcció de la companyia. No assisteix a les reunions de comitè executiu on moltes vegades es parla de solucions informàtiques.

### Amenaces:

La resistència al canvi dels usuaris juntament amb el recolzament poc clarificat de la direcció pot suposar tornar enrere en decisions del departament.

Hem triat Ubuntu com a sistema operatiu, aquest sistema pertany canonical i entre ells i Microsoft s'ha establert una relació que pot afectar a la nostra decisió.

El programari de la comunitat pot acabar discontinuat, ja ens ha passat i hem de tenir cura de trobar alternatives abans de quedar obsolets.

### Fortaleses:

L'equip d'informàtica estem tots en la mateixa línia del programari lliure i formem un equip ben consolidat i format.

La direcció general te molta confiança en tots els membres del departament.

A la llarga les nostres decisions s'han anat consolidant com a bones solucions i hem encertat bastant anticipant-nos moltes vegades als problemes que sorgien.

### Oportunitats:

Les noves tècniques per la creació de contenidors de serveis pot suposar un estalvi en quan al numero de servidors.

La crisi actual del COVID-19 ens ha obert noves línies d'investigació per la millora del sistema de videoconferència i de formació en línia.

## 8. Conclusions.

La fort aposta per la utilització de programari lliure ha proporcionat molta independència i robustesa al nostre sistema a mes d'un estalvi importantíssim per la companyia, els problemes amb els serveis derivats per aplicacions de tercers s'ha reduït considerablement i ara per ara tenim un sistema prou fiable com per garantir uns cicles de vida molt llargs per als nostres serveis.

Introduir la virtualització ha suposat un salt qualitatiu per als nostres servidors, gracies a aquesta tècnica hem aconseguit optimitzar-los moltíssim i gracies a aquesta memòria de projecte ens hem adonat que no estem aprofitant tota la potencia que ens ofereix aquesta eina i hem començat una revisió per optimitzar encara mes els servidors. La virtualització de serveis ens aporta molta seguretat amb l'ajuda dels sistemes de còpies dels serveis i això ens garanteix que els temps de recuperació de serveis siguin molt menors a 4 hores. El cost de serveis que garanteixen menys de 4 hores per la resolució d'un problema es considerable.

Treballar per tenir sistemes propis de control de producció requereix d'un equip gran i molt preparat, aquest camí obliga a tenir mes recursos però el benefici es molt gran, els sistemes propis de control de producció s'integren amb la resta de sistemes perfectament i amb les necessitats de la companyia en cada moment, el que proporciona una gran font d'informació de qualitat. Aquest fet, ajuda a impactar directament sobre els nostres productes millorant la seva qualitat, així com també el processos de producció, fet que fa que es produeixi una millora de la productivitat.

La introducció del programari lliure per als usuaris es una tasca complicada inicialment però cada vegada hi tenim mes usuaris que valoren positivament el canvi, entre ells molts dels directius i això dona recolzament al departament per continuar millorant en aquesta línia.

La formació ha de ser una punt de partida per la millora de tots els sistemes, es per això que el departament d'informàtica considera la formació imprescindible fins a tal punt que tenim la nostre pròpia cartera de cursos planificada anualment on tractem de formar-nos en noves tecnologies, llenguatges de programació, tècniques de treball i un llarg etcètera, la formació en el negoci de les TIC ha de ser una part de la feina diària.

Com a conclusió final, a tots els nivells els sistemes informàtics han de ser senzill i pràctics, no han de complicar la vida ni als usuaris ni als membres del departament. Les coses senzilles son les que funcionen i tenen millor acceptació, i aquesta ha de ser la línia a seguir en el futur, la simplicitat del sistema.

## 9. Bibliografia.

- Fernández, J. C. (19 / Març / 2017). *Conoce estos 5 software de virtualización, ideal para virtualizar varios sistemas operativos en un servidor.* Recollit de RZ redes zone: <https://www.redeszone.net/2017/03/19/conoce-estos-5-software-de-virtualizacion-ideal-para-virtualizar-varios-sistemas-operativos-en-un-servidor/>
- Hasan, M. (sense data). *Best Linux Server Distro: Top 10 Compared and Our Recommendation.* Recollit de Ubuntupit: <https://www.ubuntupit.com/best-linux-server-distro-top-10-compared-recommendation/>
- Mertech. (sense data). *Migrate from DataFlex to SQL Databases.* Recollit de Mertech: <https://www.mertech.com/migrate-dataflex-to-sql>
- Comunitat. (27 / Abril / 2020). *Microsoft SQL Server.* Recollit de Wikipedia: [https://en.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](https://en.wikipedia.org/wiki/Microsoft_SQL_Server)
- Comunitat. (2020, Març 24). *DataFlex.* Retrieved from Wikipedia: <https://en.wikipedia.org/wiki/DataFlex>
- Comunitat. (25 / Abril / 2020). *GNU General Public License.* Recollit de Wikipedia: [https://en.wikipedia.org/wiki/GNU\\_General\\_Public\\_License](https://en.wikipedia.org/wiki/GNU_General_Public_License)
- Comunitat. (19 / Març / 2020). *Hypervisor.* Recollit de Wikipedia: <https://en.wikipedia.org/wiki/Hypervisor>
- Comunitat. (27 / Abril / 2020). *Novell.* Recollit de Wikipedia: <https://en.wikipedia.org/wiki/Novell>
- Comunitat. (2020, Abril 25). *Open Database Connectivity.* Retrieved from Wikipedia: [https://en.wikipedia.org/wiki/Open\\_Database\\_Connectivity](https://en.wikipedia.org/wiki/Open_Database_Connectivity)
- Comunitat. (3 / Març / 2020). *Powder metallurgy.* Recollit de Wikipedia: [https://en.wikipedia.org/wiki/Powder\\_metallurgy](https://en.wikipedia.org/wiki/Powder_metallurgy)
- Comunitat. (28 / Abril / 2020). *PostgreSQL.* Recollit de Wikipedia: <https://en.wikipedia.org/wiki/PostgreSQL>
- Comunitat. (29 / Abril / 2020). *MySQL.* Recollit de Wikipedia: <https://en.wikipedia.org/wiki/MySQL>
- Comunitat. (26 / Abril / 2020). *Oracle Database.* Recollit de Wikipedia: [https://en.wikipedia.org/wiki/Oracle\\_Database](https://en.wikipedia.org/wiki/Oracle_Database)
- Comunitat. (27 / Abril / 2020). *XAMPP.* Recollit de Wikipedia: <https://en.wikipedia.org/wiki/XAMPP>
- Oracle Corporation. (sense data). *Connector/ODBC architecture.* Recollit de MySQL: <https://dev.mysql.com/doc/connector-odbc/en/connector-odbc-architecture.html>
- Comunitat. (24 / Abril / 2020). *InnoDB.* Recollit de Wikipedia: <https://en.wikipedia.org/wiki/InnoDB>
- Comunitat. (8 / Juliol / 2019). *MyISAM.* Recollit de Wikipedia: <https://en.wikipedia.org/wiki/MyISAM>
- N, K. (11 / Octubre / 2019). *What is the difference between MyISAM and InnoDB?* Recollit de Knowhost: <https://www.knownhost.com/wiki/developmental/mysql-myisam-innodb>
- Comunitat. (21 / Abril / 2020). *Database trigger.* Recollit de Wikipedia: [https://en.wikipedia.org/wiki/Database\\_trigger](https://en.wikipedia.org/wiki/Database_trigger)
- Comunitat. (2 / Març / 2020). *Round-robin DNS.* Recollit de Wikipedia: [https://en.wikipedia.org/wiki/Round-robin\\_DNS](https://en.wikipedia.org/wiki/Round-robin_DNS)
- Comunitat. (11 / Maig / 2020). *Simple Mail Transport Protocol.* Recollit de Wikipedia: [https://en.wikipedia.org/wiki/Simple\\_Mail\\_Transfer\\_Protocol](https://en.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol)
- Comunitat. (3 / Maig / 2020). *Cache (Computing).* Recollit de Wikipedia: [https://en.wikipedia.org/wiki/Cache\\_\(computing\)](https://en.wikipedia.org/wiki/Cache_(computing))
- Comunitat. (6 / Maig / 2020). *BIND.* Recollit de Wikipedia: <https://en.wikipedia.org/wiki/BIND>
- Both, D. (7 / Abril / 2017). *Build your own DNS server on Linux.* Recollit de Opensource: <https://opensource.com/article/17/4/build-your-own-name-server>
- Comunitat. (11 / Maig / 2020). *Postfix (software).* Recollit de Wikipedia: [https://en.wikipedia.org/wiki/Postfix\\_\(software\)](https://en.wikipedia.org/wiki/Postfix_(software))
- Comunitat. (24 / Abril / 2020). *Internet Message Access Protocol.* Recollit de Wikipedia: [https://en.wikipedia.org/wiki/Internet\\_Message\\_Access\\_Protocol](https://en.wikipedia.org/wiki/Internet_Message_Access_Protocol)
- Comunitat. (9 / Maig / 2020). *Post Office Protocol.* Recollit de Wikipedia: [https://en.wikipedia.org/wiki/Post\\_Office\\_Protocol](https://en.wikipedia.org/wiki/Post_Office_Protocol)

- Comunitat. (3 / Maig / 2020). *EXIM*. Recollit de Wikipedia: <https://en.wikipedia.org/wiki/Exim>
- Comunitat. (4 / Maig / 2020). *OpenLDAP*. Recollit de Wikipedia: <https://en.wikipedia.org/wiki/OpenLDAP>
- Comunitat. (11 / Maig / 2020). *Cron*. Recollit de Wikipedia: <https://en.wikipedia.org/wiki/Cron>
- Comunitat. (14 / Maig / 2020). *Python (programming language)*. Recollit de Wikipedia: [https://en.wikipedia.org/wiki/Python\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))
- Comunitat. (8 / Maig / 2020). *Samba (software)*. Recollit de Wikipedia: [https://en.wikipedia.org/wiki/Samba\\_\(software\)](https://en.wikipedia.org/wiki/Samba_(software))
- Comunitat. (17 / Abril / 2020). *Synchronization (computer science)*. Recollit de Wikipedia: [https://en.wikipedia.org/wiki/Synchronization\\_\(computer\\_science\)](https://en.wikipedia.org/wiki/Synchronization_(computer_science))
- Pierce, B. C. (2012). *Unison File Synchronizer*. Recollit de cis.upenn.edu: <https://www.cis.upenn.edu/~bcpierce/unison/download/releases/stable/unison-manual.html>
- Comunitat. (25 / Abril / 2020). *Apache HTTP Server*. Recollit de Wikipedia: [https://en.wikipedia.org/wiki/Apache\\_HTTP\\_Server](https://en.wikipedia.org/wiki/Apache_HTTP_Server)
- Comunitat. (15 / Maig / 2020). *PHP*. Recollit de Wikipedia: <https://en.wikipedia.org/wiki/PHP>
- Comunitat. (14 / Maig / 2020). *JavaScript*. Recollit de Wikipedia: <https://en.wikipedia.org/wiki/JavaScript>
- Comunitat. (22 / Abril / 2020). *JavaServer Pages*. Recollit de Wikipedia: [https://en.wikipedia.org/wiki/JavaServer\\_Pages](https://en.wikipedia.org/wiki/JavaServer_Pages)
- Comunitat. (1 / Maig / 2020). *GlassFish*. Recollit de Wikipedia: <https://en.wikipedia.org/wiki/GlassFish>
- Comunitat. (12 / Maig / 2020). *Apache Tomcat*. Recollit de Wikipedia: [https://en.wikipedia.org/wiki/Apache\\_Tomcat](https://en.wikipedia.org/wiki/Apache_Tomcat)
- Comunitat. (6 / Maig / 2020). *ProFTPD*. Recollit de Wikipedia: <https://en.wikipedia.org/wiki/ProFTPD>
- Comunitat. (15 / Abril / 2020). *Nagios*. Recollit de Wikipedia: <https://en.wikipedia.org/wiki/Nagios>
- Comunitat. (15 / Abril / 2020). *Pandora FMS*. Recollit de Wikipedia: [https://en.wikipedia.org/wiki/Pandora\\_FMS](https://en.wikipedia.org/wiki/Pandora_FMS)
- Pandora FMS Team. (2020). *Documentation*. Recollit de Pandorafms: <https://pandorafms.org/docs/monitoring-documentation/>
- Comunitat. (8 / Maig / 2020). *Network-attached storage*. Recollit de Wikipedia: [https://en.wikipedia.org/wiki/Network-attached\\_storage](https://en.wikipedia.org/wiki/Network-attached_storage)
- Comunitat. (14 / Maig / 2020). *Jitsi*. Recollit de Wikipedia: <https://en.wikipedia.org/wiki/Jitsi>
- Comunitat. (9 / Maig / 2020). *Apache OpenMeetings*. Recollit de Wikipedia: [https://en.wikipedia.org/wiki/Apache\\_OpenMeetings](https://en.wikipedia.org/wiki/Apache_OpenMeetings)
- Comunitat. (13 / Abril / 2020). *Moodle*. Recollit de Wikipedia: <https://en.wikipedia.org/wiki/Moodle>
- Comunitat. (15 / Maig / 2020). *Docker*. Recollit de Wikipedia: [https://en.wikipedia.org/wiki/Docker\\_\(software\)](https://en.wikipedia.org/wiki/Docker_(software))
- Comunitat. (3 / Maig / 2020). *MAC address*. Recollit de Wikipedia: [https://en.wikipedia.org/wiki/MAC\\_address](https://en.wikipedia.org/wiki/MAC_address)
- Comunitat. (13 / Maig / 2020). *Elasticsearch*. Recollit de Wikipedia: <https://en.wikipedia.org/wiki/Elasticsearch>
- Comunitat. (9 / Maig / 2020). *SCADA*. Recollit de Wikipedia: <https://en.wikipedia.org/wiki/SCADA>
- Comunitat. (20 / Març / 2020). *ACID*. Recollit de Wikipedia: <https://en.wikipedia.org/wiki/ACID>
- Comunitat. (14 / Març / 2020). *Java servlet*. Recollit de Wikipedia: [https://en.wikipedia.org/wiki/Java\\_servlet](https://en.wikipedia.org/wiki/Java_servlet)
- Comunitat. (15 / Maig / 2020). *File Transfer Protocol*. Recollit de Wikipedia: [https://en.wikipedia.org/wiki/File\\_Transfer\\_Protocol](https://en.wikipedia.org/wiki/File_Transfer_Protocol)
- Wikipedia. (17 / Maig / 2020). *Qt (software)*. Recollit de Wikipedia: [https://en.wikipedia.org/wiki/Qt\\_\(software\)](https://en.wikipedia.org/wiki/Qt_(software))
- Comunitat. (28 / Abril / 2020). *HTML*. Recollit de Wikipedia: <https://en.wikipedia.org/wiki/HTML>
- Comunitat. (18 / Maig / 2020). *Java (software platform)*. Recollit de Wikipedia: [https://en.wikipedia.org/wiki/Java\\_\(software\\_platform\)](https://en.wikipedia.org/wiki/Java_(software_platform))
- Comunitat. (19 / Maig / 2020). *CSS*. Recollit de Wikipedia: [https://en.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](https://en.wikipedia.org/wiki/Cascading_Style_Sheets)
- Comunitat. (11 / Maig / 2020). *JavaFX*. Recollit de Wikipedia: <https://en.wikipedia.org/wiki/JavaFX>
- Comunitat. (21 / Abril / 2020). *PyCharm*. Recollit de Wikipedia: <https://en.wikipedia.org/wiki/PyCharm>
- Comunitat. (23 / Abril / 2020). *Qt Creator*. Recollit de Wikipedia: [https://en.wikipedia.org/wiki/Qt\\_Creator](https://en.wikipedia.org/wiki/Qt_Creator)



- Comunitat. (14 / Maig / 2020). *Eclipse (software)*. Recollit de Wikipedia: [https://en.wikipedia.org/wiki/Eclipse\\_\(software\)](https://en.wikipedia.org/wiki/Eclipse_(software))
- Openjfx. (2020). *Getting Started with JavaFX*. Recollit de Openjfx: <https://openjfx.io/openjfx-docs/>
- Comunitat. (1 / Febrer / 2020). *JasperReports*. Recollit de Wikipedia: [https://en.wikipedia.org/wiki/JasperReports#IDE\\_integration](https://en.wikipedia.org/wiki/JasperReports#IDE_integration)
- Comunitat. (4 / Maig / 2020). *GitLab*. Recollit de Wikipedia: <https://en.wikipedia.org/wiki/GitLab>
- Comunitat. (11 / Maig / 2020). *Vagrant (software)*. Recollit de Wikipedia: [https://en.wikipedia.org/wiki/Vagrant\\_\(software\)](https://en.wikipedia.org/wiki/Vagrant_(software))
- Comunitat. (16 / Maig / 2020). *VirtualBox*. Recollit de Wikipedia: <https://en.wikipedia.org/wiki/VirtualBox>
- Comunitat. (11 / Maig / 2020). *Ansible (software)*. Recollit de Wikipedia: [https://en.wikipedia.org/wiki/Ansible\\_\(software\)](https://en.wikipedia.org/wiki/Ansible_(software))
- Comunitat. (1 / Maig / 2020). *GNOME*. Recollit de Wikipedia: <https://en.wikipedia.org/wiki/GNOME>
- Comunitat. (19 / Abril / 2020). *Cinnamon (desktop environment)*. Recollit de Wikipedia: [https://en.wikipedia.org/wiki/Cinnamon\\_\(desktop\\_environment\)](https://en.wikipedia.org/wiki/Cinnamon_(desktop_environment))
- Comunitat. (17 / Maig / 2020). *KDE*. Recollit de Wikipedia: <https://en.wikipedia.org/wiki/KDE>
- Comunitat. (16 / Maig / 2020). *Firefox*. Recollit de Wikipedia: <https://en.wikipedia.org/wiki/Firefox>
- Comunitat. (17 / Maig / 2020). *Mozilla Thunderbird*. Recollit de Wikipedia: [https://en.wikipedia.org/wiki/Mozilla\\_Thunderbird](https://en.wikipedia.org/wiki/Mozilla_Thunderbird)
- Comunitat. (18 / Maig / 2020). *LibreOffice*. Recollit de Wikipedia: <https://en.wikipedia.org/wiki/LibreOffice>
- Comunitat. (7 / Maig / 2020). *Gimp*. Recollit de Wikipedia: <https://en.wikipedia.org/wiki/GIMP>
- Comunitat. (17 / Maig / 2020). *OpenShot*. Recollit de Wikipedia: <https://en.wikipedia.org/wiki/OpenShot>
- Comunitat. (13 / Maig / 2020). *Shotcut*. Recollit de Wikipedia: <https://en.wikipedia.org/wiki/Shotcut>
- Comunitat. (22 / Abril / 2020). *Lightworks*. Recollit de Wikipedia: <https://en.wikipedia.org/wiki/Lightworks>
- Comunitat. (14 / Maig / 2020). *VLC media player*. Recollit de Wikipedia: [https://en.wikipedia.org/wiki/VLC\\_media\\_player](https://en.wikipedia.org/wiki/VLC_media_player)
- Comunitat. (23 / Decembre / 2019). *Dia (software)*. Recollit de Wikipedia: [https://en.wikipedia.org/wiki/Dia\\_\(software\)](https://en.wikipedia.org/wiki/Dia_(software))
- Comunitat. (13 / Octubre / 2019). *yEd*. Recollit de Wikipedia: <https://en.wikipedia.org/wiki/YEd>
- Comunitat. (29 / Abril / 2020). *DBeaver*. Recollit de Wikipedia: <https://en.wikipedia.org/wiki/DBeaver>
- Comunitat. (6 / Maig / 2020). *jQuery*. Recollit de Wikipedia: <https://en.wikipedia.org/wiki/JQuery>
- Comunitat. (15 / Maig / 2020). *Bootstrap (front-end framework)*. Recollit de Wikipedia: [https://en.wikipedia.org/wiki/Bootstrap\\_\(front-end\\_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework))
- Microsoft. (2020). *Comparar los planes de Microsoft Exchange online*. Recollit de Microsoft: <https://www.microsoft.com/es-es/microsoft-365/exchange/compare-microsoft-exchange-online-plans>