

TEMA 6 - INDEX

1 Sistemes empresarials de gestió d'informació

1.1 Objectius bàsics

- 1.1.1 Presa de decisions
- 1.1.2 Facilitar les tasques

1.2 Programari empresarial

- 1.2.1 Aplicacions genèriques
- 1.2.2 Programari a mida
- 1.2.3 Programari estàndard
- 1.2.4 ERP
- 1.2.5 Implantació d'un ERP

1.3 Tendències

2 Serveis web

2.1 La web programable

2.2 Els serveis web

- 2.2.1 Característiques dels serveis web
- 2.2.2 Tipus de serveis web

2.3 Big Web services

- 2.3.1 Funcionament
- 2.3.2 SOAP
- 2.3.3 WSDL
- 2.3.4 UDDI
- 2.3.5 Provar serveis web

2.4 Architectures basades en REST

- 2.4.1 Els recursos
- 2.4.2 Representacions
- 2.4.3 HTTP

2.5 Desenvolupament de serveis web

- 2.5.1 Desenvolupament de serveis basats en SOAP
- 2.5.2 Desenvolupament de serveis basats en REST
- 2.5.3 Entorns de desenvolupament

3 Informàtica en núvol

3.1 Característiques

- 3.1.1 Ubiquïtat
- 3.1.2 Infraestructura informàtica
- 3.1.3 Adaptació a les necessitats
- 3.1.4 Reducció de costos

3.2 Classificació

- 3.2.1 Segons la propietat
- 3.2.2 Segons els nivells de servei ofert

3.3 Resistències a l'adopció de la informàtica en núvol

- 3.3.1 Privacitat i seguretat de les dades
- 3.3.2 Captivitat en el proveïdor
- 3.3.3 Aspectes Legals

1. Sistemes empresarials de gestió d'informació

Els sistemes informàtics han canviat la manera de treballar de les empreses en molts aspectes però sobretot en el camp de la informàtica de gestió¹. Però la informatització de les empreses no acaba en la informàtica de gestió, ja que actualment es fan servir sistemes informàtics per controlar pràcticament qualsevol dels aspectes del seu funcionament, des de la facturació fins al detall més mínim de la relació amb els clients; i tot amb l'objectiu d'aconseguir que les empreses siguin més competitives i eficients.

1.1 Objectius bàsics

Si bé inicialment els programes informàtics només **es feien servir per agilitzar les tasques de l'empresa**, actualment tenen un paper fonamental en moltes empreses.

Essent molt generalistes podem dir que els sistemes informàtics s'han implantat amb **dos objectius bàsics**:

- **Facilitar les tasques als treballadors.**
- **Facilitar les preses de decisions dels dirigents.**

És des d'aquest prisma que sovint es fan les valoracions sobre l'èxit o el fracàs de la implantació d'un sistema informàtic en una empresa.

1.1.1 Presa de decisions

Un dels objectius comuns en gairebé totes les organitzacions és obtenir i mantenir un avantatge competitiu, que a la vegada proporciona un avantatge a llarg termini.

Això planteja un seguit de qüestions, com ara:

- S'ha d'invertir?
- S'ha de produir més?
- S'ha de contractar més personal?
- ...

Malgrat que totes les empreses són diferents, **tenen en comú que aquest avantatge competitiu** sovint es basa en la **presa de decisions en el moment adequat, i normalment en un curt espai de temps**, ja que descobrir tard les necessitats del mercat pot fer que algú altre se n'aprofiti i aconsegueixi posicionar-se i sigui molt difícil competir-hi.

El fet que les decisions que es prenguin puguin **afectar directament el futur** de l'empresa i que s'hagin de prendre en el moment adequat fan dels sistemes informàtics un recurs valuósíssim per a qualsevol organització. Les decisions ja no s'han de prendre per intuïcions ni de manera aleatòria, sinó **basades en les dades que es poden obtenir dels sistemes informàtics**.

Podem dividir les dades en dos grans grups:

- **Dades internes**: generades **per l'empresa mateixa**, com ara nombre de vendes, preus dels proveïdors i capacitat de producció de l'empresa.
- **Dades de l'entorn**: la **situació del mercat**, els competidors, la legislació...

¹La informàtica de gestió és una de les branques més importants de la informàtica ja que és l'encarregada de donar suport als sistemes d'informació de les empreses.

Una de les funcions determinants de les empreses és extreure informació a partir de les dades que pugui recollir però s'hi ha d'afegir un factor important: aquesta informació ha d'estar disponible en el moment oportú.

El problema més gran rau en el fet que la societat de la informació està fent que **tothom rebi una gran quantitat de dades** al dia i a pesar que **moltes d'aquestes dades són irrelevantes** per als objectius de l'empresa n'hi ha moltes que són o podrien ser importants en la presa de decisions i per tant s'han de poder detectar.

Això fa que les empreses es vegin **obligades a tractar una gran quantitat d'informació**, i la gestió d'aquesta informació és una **pràctica costosa** tant en l'ús de recursos com en persones.

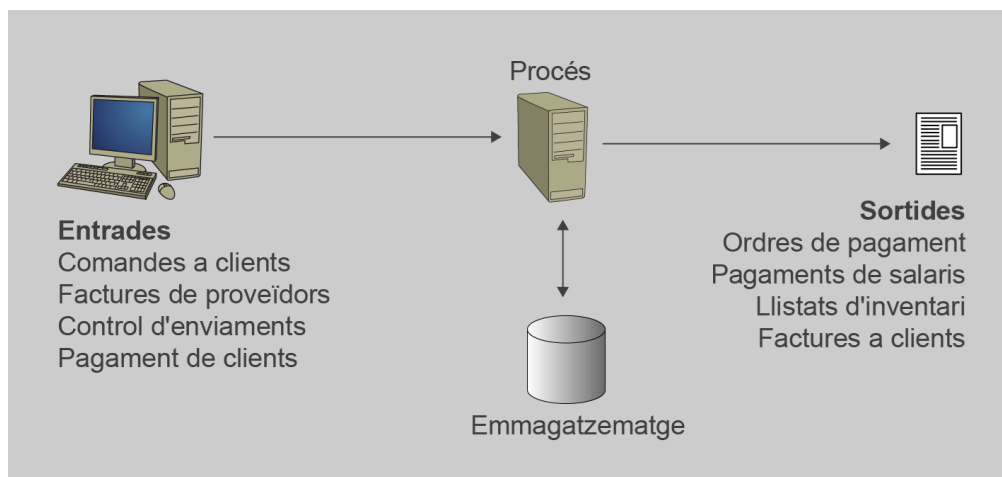
Per aquest motiu els sistemes informàtics són bàsics per proporcionar els **mecanismes per destriar les dades que serveixen de les que no serveixen**, per **organitzar i classificar** les dades amb vista a obtenir la informació necessària en el moment adequat.

1.1.2 Facilitar les tasques

El fet que la informatització ha facilitat les tasques que havien de dur a terme els treballadors es veu molt clarament en el que ha passat en el procés de transaccions de les empreses. El procés de transaccions comprèn tota una sèrie de tasques rutinàries² com control d'estocs, elaboració d'albarans, elaboració de factures, o comptabilitat que s'han de fer en una empresa i que abans requerien molts recursos per desenvolupar-se, com mostra la figura 1.1.

Aquest procés es caracteritza pel fet de tenir una gran quantitat de dades d'entrada a les quals s'aplica un procés, que no sol ser gaire complex, i que genera una gran quantitat de sortides.

Figura 1.1. Procés de transaccions d'una empresa



Abans de l'aparició de la informàtica, aquests processos, tot i que no solen ser complexos, **consumien una gran quantitat de recursos**: s'hi havia de destinar molt de personal, calia lloc per emmagatzemar les dades...

²El fet que les activitats del procés de transaccions siguin rutinàries no vol dir que no siguin importants ja que el mal funcionament d'alguna d'aquestes tasques pot portar greus problemes a les empreses.

La informatització va agilitzar notablement aquests processos, alliberant molts dels recursos que es dedicaven al procés de transaccions per destinar-los a altres tasques. L'èxit va ser tan gran que actualment el procés de transaccions informatitzat està implementat en qualsevol empresa.

Els sistemes informàtics han contribuït i estan contribuint a fer les tasques més ràpides i menys propenses a errors i per tant permeten a les empreses reduir costos i oferir més bon servei.

1.2 Programari empresarial

Hi ha una gran i molt variada quantitat d'aplicacions que es poden fer servir en les empreses. Els programes que es fan servir en les empreses van des de programes d'ofimàtica fins a programes totalment especialitzats per a una empresa d'un sector concret.

Per programari empresarial s'entén qualsevol tipus de programari que ajudi a mesurar o millorar la seva activitat.

És molt difícil fer una classificació del programari empresarial però una de possible és basar-se en quina és la idea base per crear-lo. Des d'aquest punt de vista podem dividir el programari que es fa servir per gestionar una empresa en:

- Aplicacions genèriques.
- Programari a mida.
- Programari estàndard.
- ERP (programari de gestió integrada o enterprise resource planning).

En general les empreses grans solen tenir un control més gran sobre tots els aspectes del negoci, mentre que les més petites tendeixen a tenir un programari amb funcions similars però amb un cost i funcions inferiors.

1.2.1 Aplicacions genèriques

En les empreses també es fan servir tota una sèrie d'aplicacions que no estan desenvolupades específicament per a la gestió de l'empresa sinó que són de caràcter més general, com aplicacions ofimàtiques, navegadors web o programes de correu.

Aquestes aplicacions, tot i que no estan desenvolupades específicament per a la gestió de l'empresa, a vegades hi tenen un paper molt important; els navegadors web, per exemple, són vitals per a l'accés a alguns serveis web; o pel que fa a l'exportació de dades, se solen importar en algun paquet ofimàtic per fer estadístiques o anàlisi de resultats que no estan definits en el programa de gestió principal.

Fins i tot es donen casos d'empreses molt petites que poden fer servir programes genèrics com per exemple aplicacions ofimàtiques per portar un control simplificat de la seva activitat empresarial.

1.2.2 Programari a mida

En els principis de l'adopció de la informàtica per part de les empreses el més corrent era que aquestes empreses definissin els seus sistemes informàtics fent un programa que s'adaptés perfectament al seu funcionament o a les seves necessitats.

El **programari a mida** consisteix a **desenvolupar un nou programa informàtic que permeti informatitzar l'activitat de l'empresa.**

La idea és **contractar informàtics per desenvolupar un programa** que permeti informatitzar les tasques de la gestió de l'empresa. Com que es tracta de **programes específics per a l'organització**, aquests programes s'adapten perfectament a les seves necessitats.

Però aquests programes també presenten una **sèrie de problemes**:

- S'ha d'**analitzar amb detall el funcionament de l'empresa** per comprendre perfectament què es vol fer.
- La implantació és un **procés llarg i propens a errors**, ja que el programa **no s'ha provat abans**.
- S'adapten tan bé a l'empresa que **qualsevol canvi en el funcionament normalment provoca que també s'hagin de fer canvis en el programari**. Això fa que aquest tipus de programari sigui molt **car de mantenir**.

1.2.3 Programari estàndard

El **programari estàndard** de gestió d'empreses és un programari desenvolupat **específicament per poder funcionar en la majoria d'organitzacions.**

El programari estàndard està pensat **específicament per fer tasques concretes en una empresa genèrica global** o d'un **sector concret**. La idea que hi ha al darrere d'aquestes aplicacions és intentar **abaratar els costos i minimitzar els errors** que es produeixen en el desenvolupament d'aplicacions a mida oferint un paquet que faci les funcions **tan genèricament com sigui possible** perquè es pugui usar en **diferents empreses**.

Tot i que les empreses són totes diferents, la realitat és que hi ha una part molt important de tasques que és comuna a totes³. I gràcies a això es poden crear paquets genèrics que es poden reutilitzar en diferents empreses.

Aquest sistema permet:

- **Abaratar molt els costos** perquè el programari ja està **desenvolupat i provat**.
- **Reduir el temps d'implantació** del programari. Generalment n'hi ha prou amb entrar les dades i ja es pot començar a treballar.

Existeixen molts programes d'aquest tipus que normalment es concentren en les necessitats de les **petites i mitjanes empreses**, oferint en **un sol paquet control** d'estocs, punt de venda, comptabilitat i facturació.

Els inconvenients d'aquests programes estan en l'adaptació:

- Totes les funcions estan pensades per funcionar "en qualsevol lloc" i per tant **no solen adaptar-se perfectament a les especificitats de cap empresa**.
- Poden provocar que sigui **l'empresa la que s'hagi d'adaptar al funcionament del programa** i no al revés.

³Tot i la semblança entre les tasques de les empreses, les petites especificitats que les fan diferents són les que les defineixen realment.

A més, a vegades els programes que s'adaptarien més bé als diferents departaments d'una empresa no són del mateix desenvolupador. Per exemple, es pot donar el cas que el programa de comptabilitat que va més bé sigui el de la companyia A, mentre que el de gestió de magatzem sigui el de la companyia B.

Tenir programari de diferents empreses generalment provoca tota una sèrie de problemes però normalment el més important és “com ho fem perquè els programes es comuniquin les dades entre si sense problemes?”

Fins i tot quan s'aconsegueix aquesta comunicació hi sol haver tota una sèrie de dades d'un programa que no es passaran mai a l'altre (sia perquè no les fa servir o perquè no li calen). I això genera el que anomenem illes informàtiques.

Es produeix una illa informàtica quan en una empresa hi ha diferents departaments que tenen les seves pròpies dades i només en comparteixen una part amb la resta de l'empresa.

A més s'ha de tenir en compte que els programes no són perfectes, i regularment s'hi solen detectar problemes, evolucionen, i en surten noves versions. Pot passar que una actualització d'algun dels programes faci que es generin problemes en l'intercanvi de dades amb els altres programes, que a partir d'una determinada funció es facin tasques de manera diferent...

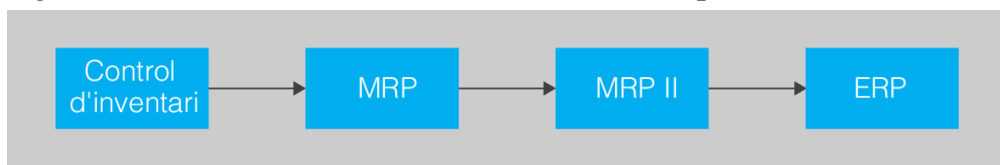
Malgrat aquests inconvenients, el programari estàndard és una opció molt vàlida que s'està fent servir en la gestió d'una gran quantitat d'empreses petites i mitjanes.

1.2.4 ERP

En el moment en què va fer falta unir les diferents aplicacions dels diferents departaments d'una empresa es va veure que tenir un sistema d'emmagatzematge únic milloraria l'eficiència de l'organització.

Els programaris de gestió integrada o enterprise resource planning (ERP) van sorgir a partir dels sistemes industrials de planificació de recursos de fabricació o material requeriment planning (MRP i MRP II) amb l'objectiu d'integrar tots els sistemes d'informació de les empreses en un de sol (figura 1.2).

Figura 1.2. Evolució dels sistemes informàtics empresarials



El que fan els ERP és expandir el concepte d'MRP II a totes les àrees de negoci, des de finances i control de projectes fins a recursos humans.

Els ERP són sistemes integrals per a la gestió de l'empresa. Es tracta d'un programa integrat que permet avaluar i gestionar el negoci.

Són sistemes de gestió d'informació que automatitzen moltes de les tasques d'operació i producció dels negocis d'una empresa.

Els sistemes modulars estan formats per diferents components o mòduls que ofereixen solucions a determinats processos de negoci i que es poden integrar lliurement per formar l'aplicació final. El funcionament modular permet a qualsevol empresa poder adaptar el programa als canvis i a les necessitats de cada moment, integrant-hi o eliminant-ne mòduls segons les noves necessitats.

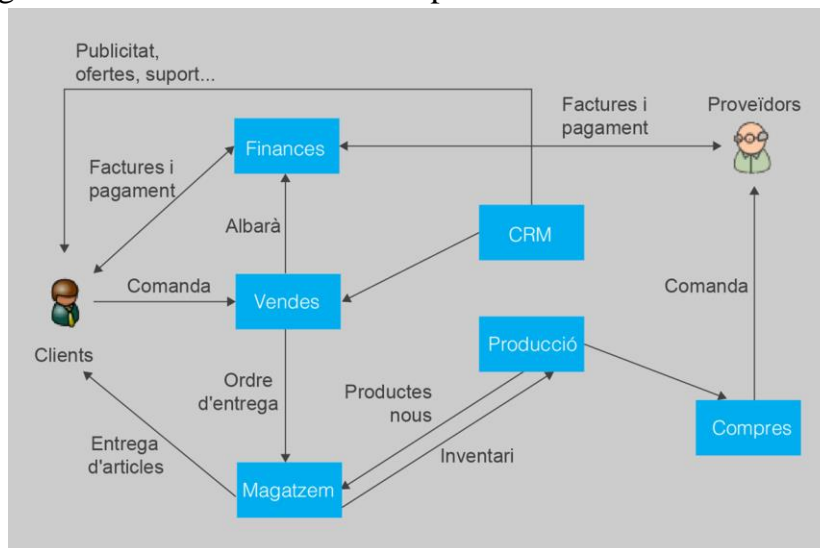
A més, el fet de funcionar de manera modular permet que els ERP es puguin adaptar a un conjunt diferent d'empreses i sectors ja que cada empresa només cal que integri els mòduls que realment necessita.

Molts dels ERP, mitjançant algun tipus d'eines de desenvolupament, permeten adaptar mòduls concrets a les necessitats particulars de l'empresa.

La gran **diferència** entre els ERP i les aplicacions tradicionals està en el fet que els **ERP intenten abastar tots els processos de l'empresa i no solament una part**, com feien les aplicacions tradicionals. Això fa que es puguin integrar totes les àrees i procediments de l'empresa en una sola aplicació i permet obtenir-ne una visió global d'una manera més senzilla.

La integració de tots els aspectes del negoci en una mateixa aplicació fa que se **simplifiquin les comunicacions entre els processos**. Si partim d'un sistema tradicional els diferents departaments han d'establir múltiples comunicacions amb els altres departaments per poder fer la seva feina (figura 1.3).

Figura 1.3. Comunicacions entre processos amb sistemes tradicionals



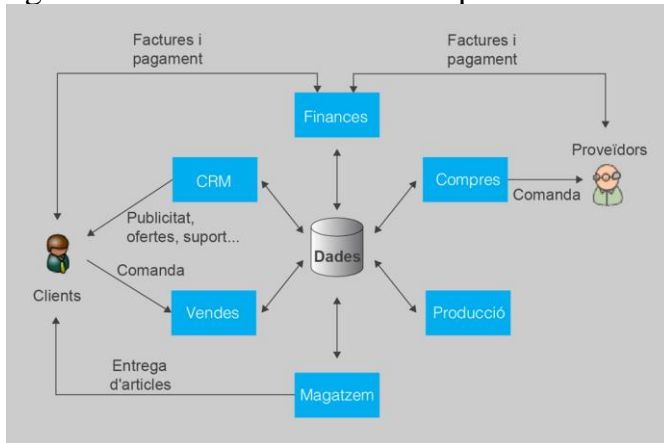
En canvi, amb un ERP, els mòduls no cal que es comuniquin amb els altres mòduls; simplement ho han de fer amb la font de dades comuna (figura 1.4).

Els ERP són conscients de l'ús d'aplicacions de propòsit general en les empreses i solen estar pensats per interactuar amb paquets d'ofimàtica o navegadors.

Tampoc no han quedat aliens a l'explosió d'èxit que ha estat Internet:

- Pràcticament tots permeten que s'hi pugui treballar des de navegadors web (i eliminen així la necessitat de clients específics).
- S'integren perfectament amb els programes de correu electrònic i comencen a oferir integració amb xarxes socials o correu electrònic.

Figura 1.4. Comunicacions entre processos amb un ERP

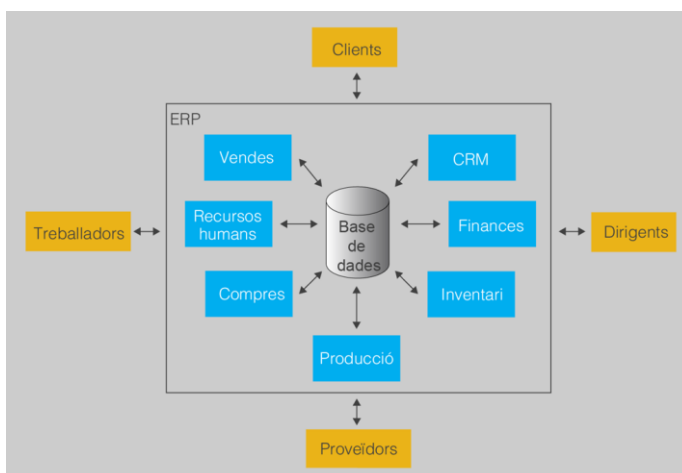


Característiques dels ERP

Simplificant podem dir que un ERP està format per dos components bàsics (figura 1.5):

- Una base de dades centralitzada.
- Un grup de mòduls o aplicacions.

Figura 1.5. Esquema bàsic d'un ERP



Base de dades centralitzada

Els ERP funcionen amb una base de dades centralitzada en què interactuen tots els programes de manera que les dades només s'emmagatzemen una sola vegada i mai no hi ha dades disgregades.

Aquest repositori central de dades garanteix que, com que la informació és en un sol punt, en qualsevol moment es pugui obtenir una imatge de l'estat en què es troba l'empresa. Això elimina la possibilitat de creació d'“illes informàtiques”.

També es minimitza l'intercanvi d'informació entre departaments i la possibilitat que algú treballi amb dades desactualitzades ja que la informació és la mateixa per a tothom.

Mòduls

Els mòduls són un conjunt d'aplicacions que es poden integrar per formar el sistema de programari. És bastant habitual que cadascun d'aquests mòduls coincideixi amb les unitats funcionals de l'empresa (compres, vendes, inventari, finances...).

Per tant, un dels grans avantatges dels ERP respecte als altres sistemes és que amb un sol programa s'integra tota la gestió de l'empresa. Això porta avantatges associats:

- El temps d'aprenentatge es redueix ja que no cal aprendre com funcionen diferents entorns, perquè és el mateix en tots els programes.
- El sistema es pot adaptar a les necessitats de cada moment de l'empresa, afegint nous mòduls per fer noves tasques o canviant la manera en què funcionen els mòduls instal·lats.

Tots els mòduls comparteixen la informació que van generant amb els altres mòduls de manera que en qualsevol punt es pot obtenir informació fiable i veraç.

Inconvenients

No tot són avantatges en els ERP; també tenen alguns inconvenients, com ara:

- La implantació⁴ sol ser complexa ja que cal que hi intervingui i s'hi impliqui tota la empresa. Hi ha nombrosos exemples d'implantacions d'ERP que han fallat perquè no s'ha implicat tota la empresa en la implantació o per resistències al canvi d'alguns departaments.
- És bastant habitual que la implantació de l'ERP comporte que s'hagi de canviar la manera de funcionar d'algun procés de l'empresa per adaptar-lo a la manera de fer-ho de l'ERP, o bé que s'hagi de reprogramar algun mòdul per adaptar-se al funcionament l'empresa. Tot això fa que les fases de desenvolupament i prova dels canvis que s'han dut a terme siguin més importants, i que per tant el temps d'implantació encara sigui llarg.

Però hi ha altres factors que poden fer que el temps d'implantació sigui més llarg, com ara:

- Problemes en la integració de les dades existents en el nou sistema.
- Resistència d'alguns departaments al canvi.
- Problemes d'aprenentatge de les noves eines per part dels treballadors.

Un altre problema que s'ha de tenir en compte és que es passa a dependre d'un sol proveïdor de programari, i això converteix l'empresa en dependent del proveïdor i del seus costos de llicències i de manteniment, que sovint són importants.

Per tant, els ERP ofereixen molts avantatges però no són recomanables per a empreses que canvien sovint o que siguin molt descentralitzades.

Tipus d'ERP

Mentre inicialment els ERP estaven pensats per funcionar en qualsevol empresa, independentment del sector en què treballaven, actualment es concentren més a adaptar-se per cobrir les necessitats de les petites i mitjanes empreses. D'aquesta manera han anat apareixent ERP destinats específicament a determinats sectors empresarials.

Es podria fer una classificació dels ERP des d'un punt de vista de les empreses a què van dirigits en:

- ERP horitzontals: aplicacions pensades per poder funcionar en qualsevol empresa.

⁴Normalment el temps d'implantació d'un ERP en una empresa supera l'any.

- ERP **verticals**: aplicacions pensades per treballar en una **activitat concreta**.

ERP verticals

Un **ERP vertical** és una solució amb totes les funcionalitats estàndard com control de magatzem, compres, vendes o finances, però a més **s'hi afegeixen les necessitats concretes d'un sector determinat i s'hi adapten**.

Un dels motius pels quals els ERP horitzontals tenen un temps d'implantació tan gran és perquè **resolen molt bé tot el que és generalista** però no solen resoldre gaire bé aquelles parts que són més específiques d'un sector industrial en concret. És en aquest punt que els ERP verticals prenen força ja que, com que són adaptats per desenvolupadors amb coneixements sobre l'entorn i el negoci, el temps d'implantació es redueix considerablement.

En el cas dels **ERP horitzontals**, és corrent que el **cost d'adaptació del programa sigui fins i tot superior al cost de les llicències**, ja que calen **moltes hores d'adaptació per aconseguir la integració amb l'empresa**. En canvi, en el cas dels verticals, l'adaptació requereix pocs canvis i per tant el resultat final també té un cost inferior.

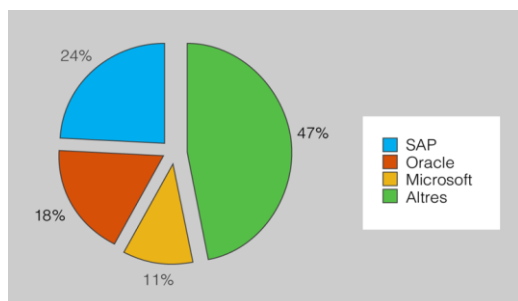
Tot i això, els **ERP verticals tenen més tendència que els horitzontals** a fer que sigui l'empresa la que s'hagi d'adaptar a l'ERP i no al revés; això acostuma a ser degut al fet que fa les coses d'una manera diferent de com es fan en l'empresa, i portar a terme un canvi en l'àmbit de desenvolupament eliminaria l'avantatge de tenir un ERP vertical.

Principals programes d'ERP

No hi ha cap ERP perfecte i tots tenen punts forts i punts febles, i per tant un dels aspectes fonamentals perquè la implantació d'un ERP tingui èxit en una empresa és triar-lo adequadament.

Hi ha diversos productes ERP disponibles en el mercat, entre els quals destaquen els oferts per les tres grans empreses desenvolupadores d'ERP: SAP AG, Oracle i Microsoft. Però hi ha moltes més solucions, com les que integren les empreses Sage (molt populars pels productes Contaplus o Facturaplus), Epicor, IFS o Ross (figura 1.6).

Figura 1.6. Distribució dels ERP en el mercat segons Panorama Consulting Group.



Els ERP més populars són: (Como pregunte esto le mato)

- SAP R/3.
- Oracle e-Business Suite.
- Oracle JD Edwards.
- Oracle Peoplesoft.
- Microsoft Dynamics.

- Epicor ERP.
- Sage.
- Infor ERP.
- Lawson M3 ERP.

Codi obert

També en el món del programari lliure s'hi pot trobar programari ERP, els fabricants del qual sovint aconsegueixen els ingressos per mitjà de l'oferta de serveis als clients.

Entre els programes **ERP de codi obert** destaquen:

- OpenBravo.
- OpenERP.
- Compiere.
- ADempiere.
- Opentaps.

El codi obert ofereix unes característiques extres que sempre s'han de tenir en compte a l'hora de valorar si cal implantar aquestes solucions o no:

- L'avantatge més clar és el **preu**. Els ERP de codi obert es poden aconseguir sense cap mena de cost i per tant es redueixen els costos d'implementació i les llicències d'usuari.
- **No crea cap dependència d'un proveïdor informàtic** ja que el programari passa a ser **controlat per l'empresa** i es pot fer servir per crear un ERP propi amb molts menys recursos dels necessaris en un sistema de pagament.
- Com que **disposa del codi les possibilitats d'adaptació del programa a les necessitats de l'empresa són màximes**. Les solucions de pagament generalment permeten **personalitzar els mòduls** però sempre dins d'uns límits. En el cas dels sistemes de codi obert aquests límits no existeixen.

Però el programari lliure no és exempt de problemes, la qual cosa fa que moltes empreses no s'atreveixin a fer-los servir:

- **No sempre hi ha prou documentació del funcionament dels programes**. Això pot incrementar el temps d'aprenentatge.
- Les **actualitzacions** del programari solen ser **imprevisibles** i no sempre es documenten els canvis que es fan.
- **Els projectes a vegades desapareixen**. La desaparició d'un projecte normalment acaba obligant a canviar de producte.
- A pesar que gairebé tots ofereixen el producte lliure però amb una versió de pagament on s'inclou suport, en alguns casos el suport no sol ser tan important com el que s'ofereix en el cas del programari propietari.

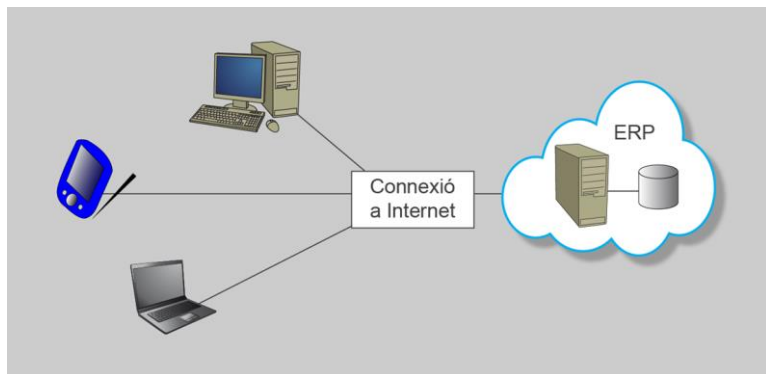
Els ERP de codi obert són una **alternativa totalment vàlida** i que s'ha de valorar a l'hora de triar quin és l'ERP que s'adapta més a les necessitats de l'empresa, però sempre tenint en compte les fortaleeses i les debilitats que té.

Informàtica en núvol

La majoria d'aquests ERP s'estan adaptant o ja s'han adaptat per poder funcionar en el que es coneix com a informàtica en núvol o més concretament com a SaaS (software as a service).

La idea és no haver de fer cap mena d'instal·lació de programari en els servidors de l'empresa sinó accedir al programa per Internet, normalment amb un navegador web. Això permet que les tasques d'instal·lació i manteniment s'externalitzin, ja que és el proveïdor el que se n'encarregarà (figura 1.7).

Figura 1.7. ERP en núvol



Aquesta característica permet:

- Fer servir el programa sense haver d'instal·lar cap programari, ja que aquest programari està instal·lat en els servidors del venedor.
- Que no sigui necessari per a l'empresa fer cap mena de despesa en infraestructura de maquinari (no cal comprar un servidor, implementar mecanismes de còpia de seguretat...) ni en personal informàtic qualificat.

N'hi ha prou amb un ordinador i una connexió a Internet per poder treballar amb l'ERP. Generalment el cost d'aquest tipus de serveis es paga via subscripció, la qual cosa implica un preu molt inferior al que es demana per comprar el paquet de programari i el lloguer dels tècnics responsables de la instal·lació.

Funcionar d'aquesta manera garanteix que el programari sempre està actualitzat ja que l'empresa desenvolupadora s'encarrega de fer-ho en els seus propis servidors. No cal que l'empresa prengui riscos ni tingui despeses extres fent actualitzacions en els seus servidors. Però sempre s'ha de tenir en compte que treballar d'aquesta manera implica uns riscos:

- Requereix que hi hagi una connexió permanent a Internet i per tant fa més dependent l'empresa de la companyia de telefonia.
- Una fallada en els servidors de l'empresa allotjadora pot provocar que s'aturi completament el funcionament de l'empresa ja que no es pot accedir al programa que ho gestiona tot.
- Com que les dades estan emmagatzemades en un lloc remot, en cas de problemes no és possible recuperar-les.

A més, l'ús de serveis en núvol pot comportar problemes legals ja que les dades deixen d'estar emmagatzemades en l'empresa per passar a estar-ho en els servidors del proveïdor, i a vegades aquests proveïdors no són en el mateix país en què treballa l'empresa.

Per tant es pot **infringir la llei de Protecció de Dades de Caracter Personal** (LOPD) a causa de les limitacions imposades en l'ubicació d'emmagatzematge ja que entre les dades amb les que treballa l'empresa n'hi sol haver de caràcter personal.

La LOPD defineix entre altres coses **en quins casos es pot permetre a un tercer accedir a les dades d'una organització** o quines condicions s'han de complir per moure les dades d'un país a un altre.

Això obliga, abans de contractar un servei en núvol, a **assegurar-se que l'empresa proveïdora del servei garanteixi que les dades que seran emmagatzemades en els seus servidors** es desaran sense infringir la llei.

1.2.5 Implantació d'un ERP

Els ERP són **programes complexos d'implantar** i no és rar que la implantació en una empresa pugui esdevenir un fracàs. Des d'un punt de vista empresarial un fracàs pot passar perquè **calgui molt més temps o diners del que s'havia previst** per implantar una determinada solució, perquè aquesta solució no funcioni com ho hauria de fer, que no es faci servir perquè és massa complicada o perquè no hi hagi confiança en les dades obtingudes.

En el mercat hi ha **centenars d'ERP amb característiques i preus diferents**. És bàsic que abans de triar un ERP es tingui clar com és l'empresa i quines són les seves necessitats concretes. La tria adequada de l'ERP és un dels factors determinants perquè tingui èxit la implantació. Els ERP són programes molt específics, i **que un ERP tingui èxit en una empresa no és garantia que tingui èxit en una altra**, encara que les dues siguin semblants.

Els ERP no són un tipus de programari que es compra i s'instal·la i comença a funcionar sinó que **requereix que sigui adaptat per usar-lo en l'empresa**.

Sempre cal un temps de parametrització i modificació dels ERP abans de poder fer-los servir. No n'hi sol haver prou amb instal·lar el programari, sinó que **cal una fase per fer totes les adaptacions** necessàries per poder engegar el sistema.

És rar que no s'hagi de modificar res en un ERP. Normalment sempre hi ha algun aspecte del funcionament que va més bé si es canvia, o que s'ha de canviar per força. Per exemple, les llistes de resultats normalment solen ser molt genèriques perquè es confia que es modificaran en la fase d'implantació.

Amb la implantació d'un ERP generalment el que **es busca és optimitzar els processos de l'empresa** i en aquest sentit és fàcil que també s'hagi de canviar algun dels processos actuals de l'empresa o bé hagi de desaparèixer. Per aquest motiu **un dels factors clau en l'èxit de la implantació és que s'hi impliquin totes les àrees del negoci**. La **resistència dels usuaris acostumats a treballar d'una manera** a canviar-la és un altre factor determinant perquè no hi hagi èxit en la implantació.

El temps d'adaptació varia segons diferents aspectes però generalment hi influeix sobretot quin ERP s'ha triat, quins són els mòduls que s'han d'implantar i la mida de l'empresa on s'implanta l'ERP. L'allargament del temps d'implantació sovint és un altre dels factors pels quals fallen les implantacions de l'ERP.

Un altre factor que **sovint provoca molts problemes** en la implantació d'un ERP és el **traspàs de dades**. Les empreses ja solen tenir molts programes i molt diversos per portar la gestió i

generalment es requereix que es puguin connectar amb l'ERP i això a vegades és una tasca molt complexa.

Un factor que s'ha de tenir en compte és que no cal que l'ERP sigui implementat tot de cop ja que la naturalesa modular que té fa que es pugui implementar mòdul a mòdul, i generalment seguir aquesta estratègia sol tenir més garanties d'èxit.

Per fer l'adaptació normalment s'ha de recórrer a una empresa integradora que ajuda en la implementació de l'ERP i ajuda a l'empresa en els aspectes del negoci relacionats amb el canvi: maquinari, programari o procés de canvi. La tria d'una bona empresa integradora també és un factor determinant en l'èxit o la fallada de la implantació. La seva experiència i capacitat d'entendre l'empresa i d'adaptar adequadament l'eina és un altre factor determinant.

1.3 Tendències

Els ERP, com qualsevol tipus de programari, estan evolucionant i oferint noves possibilitats que fins ara no oferien.

És difícil dir com seran els ERP dels propers anys però sembla que les tendències van cap a oferir interfícies d'usuari altament personalitzables que es puguin usar en diferents dispositius, i permetin integrar eines d'ofimàtica i missatgeria electrònica i tasques col·laboratives.

La majoria dels ERP ja estan muntats basant-se en arquitectures orientades a serveis (SOA), que permeten que el programari pugui interactuar per fer tasques automàtiques sense que calgui intervenció humana. Una part important de l'arquitectura SOA, els serveis web (web services), facilita les interaccions entre els que ofereixen un servei i els que el consumeixen, de manera que permeten fer servir les funcions globals de negoci des de programes autònoms.

La tendència cap als serveis en núvol sembla que ja és imparable; en aquest cas, es basa a permetre que les empreses puguin fer servir el seu ERP des de qualsevol lloc amb una simple connexió a Internet. Les tecnologies en núvol permeten que els usuaris es despreocupin de les actualitzacions i que puguin fer servir llicències de mòduls flexibles (pagar segons les necessitats del moment).

També hi ha una tendència que cada vegada es facin servir nous mòduls que s'aprofitin de l'increment de la interoperativitat que ofereixen les arquitectures orientades a serveis (SOA) per oferir mòduls de simulació d'escenaris (pensats per preveure “què passaria si...” però amb la possibilitat d'obtenir automàticament dades externes a l'empresa) o per fer tasques que no s'estaven duent a terme com poden ser la mineria de dades⁵ (data mining).

⁵Mineria de dades: Es tracta d'una tecnologia pensada per descobrir informació desconeguda a partir de les dades que ja té l'empresa fent servir tècniques diverses relacionades amb la intel·ligència artificial.

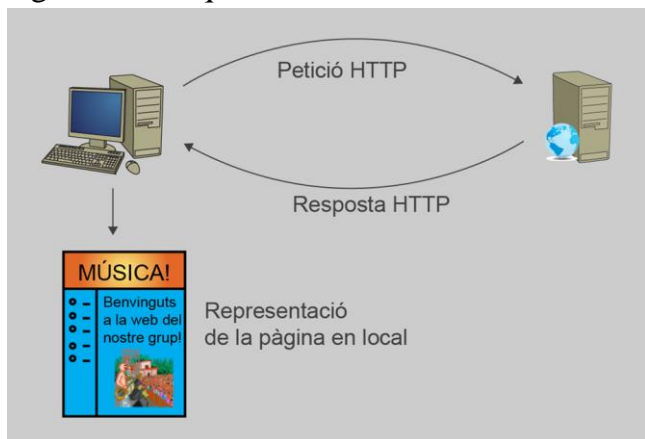
2. Serveis web

Si Internet ha tingut l'èxit que ha tingut ha estat gràcies a la creació de la WorldWideWeb. Però quan es va crear la web durant els anys vuitanta, la idea de manera com havien de ser les pàgines era molt diferent de l'actual. En el moment en què es van crear aquestes pàgines, estaven pensades només per mostrar informació estàtica. Per aquest motiu es va implementar un sistema senzill basat en dos protocols simples: HTTP i HTML.

El llenguatge de marques HTML és l'encarregat d'especificar de quina manera s'han de representar les pàgines.

HTTP és un protocol simple basat en ordres simples (GET, POST, PUT, HEAD, DELETE) que permeten recuperar un recurs mitjançant una simple petició. El protocol pot ser tan simple perquè el resultat és representat en local pel navegador.

Un client des d'un navegador demana un arxiu i el servidor simplement l'envia (figura 2.1).
Figura 2.1. Esquema de funcionament de la web



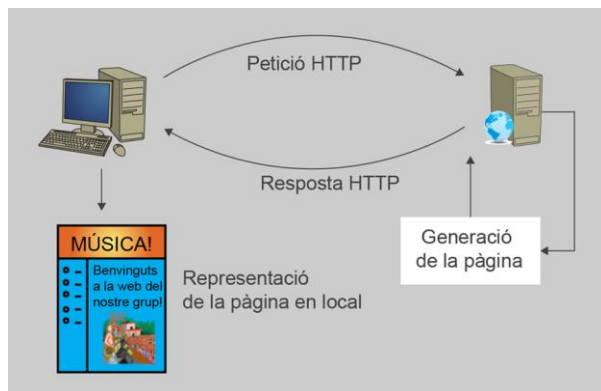
Un aspecte clau d'aquestes pàgines és que un cop representada la pàgina és l'usuari el que se l'ha de mirar i interpretar el contingut.

L'èxit d'Internet va començar a fer que la gent cada vegada demanés més de les pàgines web, que van començar a permetre la interacció amb els usuaris mitjançant el que s'anomenen pàgines web dinàmiques. La idea és que, en comptes de limitar-se a enviar un fitxer des del servidor web, es generi la pàgina segons la informació enviada per l'usuari.

Com que la tasca es fa en el servidor això no afecta gens cap dels protocols bàsics.

El fet que la pàgina es generi per encàrrec no canvia gens el fet que en realitat es continuï enviant amb el protocol HTTP un arxiu HTML (figura 2.2).

Figura 2.2. Web amb pàgines dinàmiques



Si ens ho mirem bé veurem que en realitat el que estan fent les pàgines web dinàmiques és oferir un servei. Es pot entrar en un cercador

qualsevol o en una botiga d'Internet i escriure-hi una paraula i això generarà una llista de resultats. El web fa el servei de cercar i presentar els resultats i evita que sigui l'usuari qui ho hagi de fer.

Així, doncs, el web és un lloc ple de dades (com preus, articles, horaris de trens o imatges) i de serveis (com cercadors, jocs o botigues) que han estat causa de l'èxit d'Internet.

El problema del web actual està en el fet que la majoria de les pàgines que conté comparteixen una idea comuna: estan pensades perquè sigui una persona la que s'encarregui d'interpretar-ne el contingut.

Per a un programa la informació de les webs és realment complicada d'interpretar ja que els que han creat les pàgines les han creades pensant de quina manera es poden representar les dades, i no que aquestes dades s'hagin de processar.

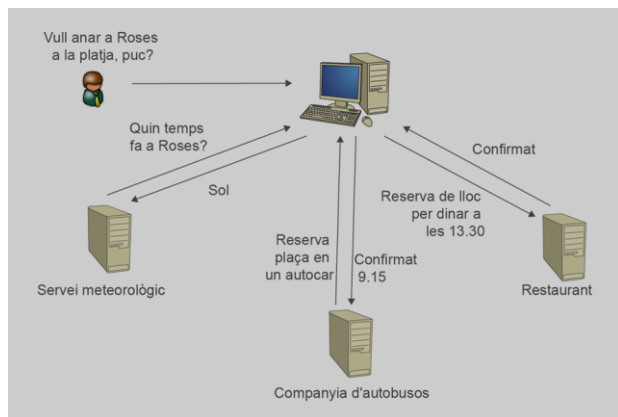
2.1 La web programable

El que es coneix com la web programable és el mateix que la web normal però pensada per ser interpretada per programes.

En comptes d'estar basada en HTML està basada en documents de marques que puguin ser processats fàcilment per determinats programes (XML, JSON...) i, per tant, que serveixin perquè els programes puguin fer tasques automatitzades a partir de la informació que han recollit.

La idea és aconseguir que un programa d'ordinador, a partir d'una informació inicial -per exemple que es vol anar a "la platja a Roses"-, s'encarregui automàticament de comprovar quin dia farà, de contractar el bitllet d'autobús i reservar una taula per dinar en un restaurant. I tot sense que calgui intervenció de l'usuari (figura 2.3).

Figura 2.3. Contractar un viatge a Roses amb servei web



La idea és crear una web diferent de l'actual, que algú ha anomenat la **web semàntica**, que sigui **comprensible per a les màquines i per als humans** de manera que els programes hi puguin fer **tasques automatitzades amb el mínim d'intervenció humana**.

Els serveis web ens permeten crear programes amb components distribuïts que no tinguin cap problema per ser executats en qualsevol tipus de plataformes o de sistemes operatius. Això ho aconsegueixen basant-se fonamentalment en protocols estàndard i oberts.

2.2 Els serveis web

Microsoft va definir el terme **servei web** (Web services) l'any 2000 per descriure un **conjunt d'estàndards que permetien als ordinadors comunicar-se per una xarxa**.

El nom prové d'aquí:

- Web va sorgir perquè, tot i que tal com estan especificats aquests serveis no tenen cap problema per funcionar amb altres protocols, la comunicació es fa sobretot mitjançant el protocol HTTP.
- La idea que hi ha al darrere de servei és que es pugui aconseguir que **el sistema remot faci una tasca de manera que no calgui que aquesta tasca es faci en el sistema local**.

Un servei web és un programari que **ofereix una interfície per fer una tasca, que pot ser usat per altres sistemes independentment de quina plataforma i llenguatge faci servir**.

Fent servir un llenguatge més proper als programadors també es podria definir un servei web com un mètode disponible remotament mitjançant una xarxa, mitjançant un protocol particular que normalment és l'HTTP.

Com que es vol que el sistema sigui multiplataforma **cal que tots els sistemes entenguin i puguin interpretar correctament les dades que s'hi transmeten**. Per aquest motiu el més corrent és que es faci servir un llenguatge de marques (XML, JSON...) per passar les dades d'un extrem a l'altre de la comunicació.

2.2.1 Característiques dels serveis web

Hi ha unes característiques que distingeixen els serveis web dels altres sistemes distribuïts:

- Fan servir una **infraestructura oberta**.
- Són **interoperables**.
- Fan servir un **disseny modular**.

O sigui que es desenvolupen **fent servir estàndards públics que siguin independents del fabricant** (com per exemple amb els protocols HTTP o XML) de manera que s'afavoreix que s'hi pugui accedir des de pràcticament **tots els sistemes i plataformes**.

La **interoperativitat** garanteix que **no importi en quin llenguatge de programació estiguin desenvolupats** (C, C#, Java, Python, Perl...) ni tampoc des de quina **plataforma** es treballi (Windows, Linux...) ja que si poden interpretar les dades (i com que estan en un estàndard públic ho faran) poden operar entre si sense problemes.

El **disseny modular** és un altre aspecte clau, ja que **permet que els serveis web es pugin integrar amb altres serveis per formar-ne de més grans i complexos**, de manera que permet crear serveis complexos a partir de components més senzills i a més sense que sigui important en quines plataformes s'estiguin executant cadascun dels components.

Un programa **pot ser descomposat en tasques més senzilles**, algunes de les quals també les fan altres programes. Si es parteix d'un disseny modular⁶ es pot **aprofitar el codi que executen aquests serveis per no haver de reprogramar-los una vegada i una altra**.

Els serveis que s'ofereixen en els serveis web són diferents dels tradicionals a Internet, que principalment es basaven en el contingut, ja que en aquests serveis el que es genera són dades que poden ser usades per uns determinats programes per dur a terme tasques.

2.2.2 Tipus de serveis web

Segons el W3C hi ha dos grans tipus de serveis web: “Els que són compatibles amb REST, que tenen com a objectiu principal **manipular representacions XML** de recursos fent servir un conjunt uniforme d'operacions sense estat; i els **arbitraris**, en què el servei **exposa un conjunt arbitrari d'operacions**.” De manera que tenim dos grans grups de serveis web:

- **Big Web services**: serveis que **exposen operacions** perquè les facin servir els servidors. Estan basats en el **protocol SOAP**.
- **Web API**: serveis basats en l'arquitectura REST -que basa el seu funcionament **en els recursos**-.

2.3 Big Web services

Els coneguts com a big Web services són serveis web que **envien i reben la informació empaquetada en un format XML** anomenat SOAP. Es poden definir a partir de tres funcions, tot i que no totes són estrictament necessàries per treballar:

- Una manera de **definir i transmetre els missatges**.
- Un llenguatge de definició de la manera com es poden fer servir els serveis.
- Un sistema de descobriment de serveis.

2.3.1 Funcionament

Els big Web services **es basen molt en XML**, ja que fer servir aquest llenguatge en sistemes distribuïts aporta alguns avantatges amb els quals no es comptava anteriorment:

- Es poden **comunicar dos sistemes totalment diferents ja que XML garanteix la portabilitat**. Com que les dades que s'envien són XML cap sistema no té problemes per entendre'n la informació.

⁶La modularitat dels serveis web és un dels factors bàsics a l'hora de definir el que s'ha anomenat arquitectura basada en serveis (SOA).

- En definir el que es vol fer i les respostes en SOAP no importa quin és el protocol de transport que es faci servir sinó la informació continguda en les dades XML.
- Es poden definir els tipus de dades fent servir XSD (XML Schemas Definition) per definir les dades que s'han de passar entre sistemes. Això permet que es puguin fer comprovacions de tipus, que els programes sàpiguin quins tipus de dades han d'enviar o han de rebre...
- Es poden expandir les possibilitats del protocol fent servir els espais de noms. Com qualsevol altre document XML es poden expandir les capacitats d'aquests protocols per afegir-hi funcionalitats mitjançant els espais de noms.

Format dels missatges

Per definir el format dels missatges que es transmeten es fa servir el protocol SOAP, que està basat en XML.

Per transportar aquests missatges se sol fer servir HTTP (protocol de transferència d'hipertext o hypertext transfer protocol) tot i que es poden fer servir altres protocols de transport com els de correu electrònic, FTP (protocol de transferència de fitxers o file transfer protocol) o BEEP (blocks extensible exchange protocol).

Definició del servei

Per poder fer servir un servei web que no s'ha fet servir mai i que fins i tot pot haver estat descobert automàticament és molt important disposar d'una manera de conèixer quins serveis ofereix i com s'ha de fer per usar-los.

La manera que tenen aquests serveis web de definir com s'ha de fer per poder usar el servei i quin servei s'ofereixen és mitjançant un altre protocol anomenat WSDL (Web services description language), que també està basat en XML.

Descobriments de serveis

Si es pretén que els programes funcionin sols cal que tinguin alguna manera de poder descobrir serveis que no coneixien anteriorment de manera automàtica.

Es va definir UDDI (descripció, descoberta i integració universals o universal description, discovery and integration) per proporcionar la capacitat de permetre als programes descobrir serveis web que s'adeqüin a les necessitats que tinguin.

Altres protocols

A part dels que s'anomenen protocols bàsics, els big Web services disposen d'un grup més important de protocols per afegir noves funcions a SOAP.

Aquests protocols, el nom dels quals normalment comença per les lletres WS-, solen ser definits per tres organitzacions d'estàndard:

- El World Wide Web Consortium (W3C).
- L'Organization for the Advancement of Structured Information Standards (OASIS).
- La Web Services Interoperability Organization (WS-I).

Aquestes organitzacions han definit més de cinquanta estàndards diferents per permetre definir qualsevol aspecte que afecti els serveis web com seguretat, gestió de serveis, gestió d'errors, o confiança entre els comunicants.

Alguns dels estàndards serveis web:

- WS-Discovery. Defineix un protocol per descobrir serveis en la xarxa local.
- WS-Addressing. Defineix un mecanisme per transportar informació d'adreces.
- WS-Security. És una extensió per afegir seguretat a SOAP.
- WS-Resource. És una família d'especificacions que permeten manipular estructures de dades remotes.
- WS-Reliability. Permet assegurar l'enviament de missatges crítics.
- WS-Policy. Permet als serveis web definir les seves característiques de seguretat, qualitat de servei...
- WS-Trust. Afegeix extensions de seguretat per gestionar testimonis d'autenticació (Security-Token) i relacions entre els participants en un intercanvi segur.

Interoperativitat

Una de les crítiques que es fan a aquest tipus de serveis web és que el fet d'haverhi tants estàndards definits per diferents organitzacions crea una dispersió que fa difícil poder oferir productes que garanteixin la tan volguda interoperativitat.

Per posar ordre al desgavell, un grup d'empreses el 2002 van crear una organització d'estàndards anomenada WS-I⁷. Es tracta d'una organització que no té com a objectiu definir noves versions dels protocols, sinó que se centra a definir de quina manera es pot aconseguir desenvolupar serveis web interoperables. Per fer-ho va definint el que anomena perfils (profiles), que són una sèrie de normes i estàndards que s'han de fer servir per assegurar que el servei és compatible amb els altres.

Un dels perfils que defineixen i que és més usat és el WSI-BP (WSI Basic Profile) que només fa servir els protocols més elementals: SOAP, WSDL, WS-Addressing, UDDI. Amb el WSI-BP es limiten les possibilitats a l'hora de crear serveis web però s'aconsegueix que aquells que el compleixin siguin interoperables amb total seguretat. Moltes de les plataformes de generació de serveis web només creen Web services compatibles amb WSI-BP.

2.3.2 SOAP

SOAP va ser el primer estàndard de serveis web que es va desenvolupar i actualment és una recomanació del W3C tot i que originalment va ser desenvolupat per un grup d'empreses. És un acrònim de simple object access protocol (protocol d'accés d'objecte simple), però actualment aquest nom ha perdut sentit ja que ha quedat demostrat que SOAP no és simple, i a més no té gaire a veure amb l'accés a objectes.

Han anat sorgint diferents versions de SOAP al llarg dels anys i actualment se'n fan servir dues, les versions 1.1 i 1.2, l'especificació de les quals es pot trobar a la pàgina del W3C (www.w3.org/TR/soap).

És un entorn de missatges que permet la comunicació entre programes que corren en diferents servidors amb independència del llenguatge i del sistema operatiu en què funcionin. Un dels avantatges que aporta SOAP és que fa molt més simple la comunicació entre sistemes que estiguin en xarxes diferents del que es podia fer abans amb altres sistemes com CORBA o DCOM.

⁷WS-I ara és membre d'OASIS tot i que no ha canviat el seu objectiu fonamental.

Aconsegueix la independència del llenguatge i de la plataforma basant els missatges en XML i s'aprofita que l'èxit de la web ha fet que tots els sistemes tinguin implementat el protocol HTTP (tot i que pot funcionar en altres protocols) per fer-lo servir per l'intercanvi de missatges.

A més de definir el format dels missatges també defineix quines són les regles que han de regir aquest intercanvi, què passa en cas d'error i com s'ha de lligar amb els altres protocols. Amb els serveis web d'aquest tipus s'exposen tota una sèrie de funcions mitjançant les quals es poden fer programes que poden fer que el servei faci tasques determinades i retorni resultats amb els quals es pugui treballar.

Per tant, qualsevol missatge que s'envii cap a un servei web va dins d'un missatge SOAP en el qual s'especifica quin és el servei que es vol fer servir i quines són les dades que s'hi envien.

Missatges SOAP

SOAP és un entorn de missatges basat en XML i per tant necessita definir tot allò que fa referència a la manera com s'han de gestionar els elements següents:

- Tipus de missatges.
- Format dels missatges.
- Protocols de transport.

Tipus de missatges

L'estàndard SOAP defineix dos tipus de patrons de missatges:

- El més habitual, que està basat en petició i resposta, que és l'habitual en el protocol HTTP.
- I un altre tipus de només resposta, que l'estàndard anomena SOAP Response. La idea és que el servei respongui quan rep un missatge no SOAP amb una resposta SOAP.

Però els dos patrons també es poden combinar per aconseguir altres patrons, com definir missatges només d'entrada, o fins i tot establir una conversa.

Composició dels missatges

Un missatge SOAP és bàsicament un document XML i per tant ha de complir totes les normes de creació de documents XML. Per tant, ha de tenir un element arrel anomenat Envelope que serveix per identificar els missatges SOAP i la versió que es fa servir. La versió s'identifica a partir de l'espai de noms especificat.

Dins de l'element arrel hi pot haver un element <Header> opcional i un <Body>, que és el que representa el contingut del missatge.

L'estàndard no defineix cap funció per a l'element <Header> però generalment se sol fer servir per donar informació o definir el contingut del missatge.

Com que SOAP és XML, diem que hi ha un format de missatges que es pot expandir mitjançant espais de noms.

Protocol de transport

Els missatges poden viatjar en qualsevol protocol però el més usat és **HTTP** i per tant és el que explicarem aquí.

Per fer una petició cal enviar un missatge amb el mètode POST al servei web en què s'identifiqui on és el servei, i en el tipus hi ha d'haver application/soap+xml o bé text/xml.

La resposta també ens donarà informació de què ha passat. Si rebem un codi 200 és que tot ha anat bé. I si hi ha algun error hem de rebre un codi 500.

Problemes

Una de les crítiques que es fa a SOAP és la mateixa que es fa a XML: malgrat tenir molts avantatges té la **tendència a fer-se molt gran i això fa que SOAP acabi essent més lent** que les altres tecnologies existents com **CORBA o DCOM, que fan servir missatges binaris**, de manera que són molt més curts i ràpids.

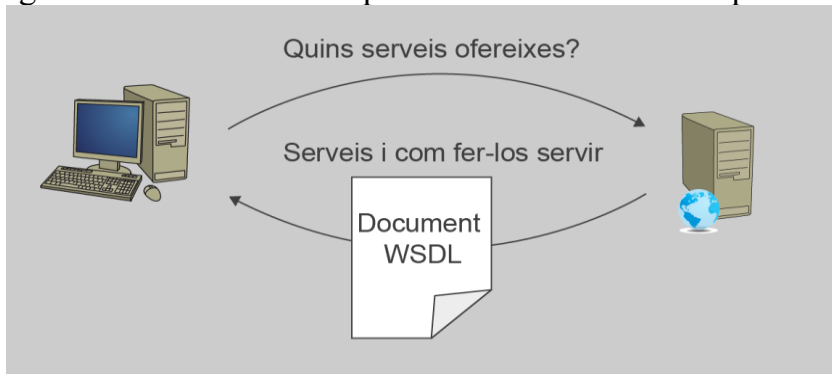
Tot i que aquests problemes són certs, per molta gent els avantatges que aporta -com la facilitat de lectura per als humans, la facilitat de detecció d'errors i el fet que solucioni els problemes de compatibilitat entre plataformes- el fan una opció que s'ha de tenir en compte.

2.3.3 WSDL

El **WSDL** (Web services definition language) és un llenguatge XML per **descriure quins serveis ofereix un servei web i com s'ha de fer per accedir-hi** (figura 2.6).

Normalment l'ús d'un servei web requereix que el client obtingui abans el fitxer WSDL per saber quin servei s'ofereix i com s'ha de fer servir.

Figura 2.6. WSDL serveix per obtenir les característiques tècniques d'un servei web



S'han desenvolupat diverses versions de WSDL: les originals, basades en el número 1 (1.0, 1.1); i la nova versió 2.0⁸, que és incompatible amb les versions anteriors.

El format en que està definit el fitxer WSDL, a pesar de ser XML, està pensat per ser llegit i interpretat per una màquina i no pas per una persona. S'hi defineixen un ampli ventall d'aspectes: des de les adreces mitjançant les quals es pot comunicar amb les diferents funcions del servei web, fins al format en què ha d'estar cada tipus de dades.

El **WSDL no és necessari per fer la comunicació**, de manera que si ja es té informació sobre quins serveis ofereix un servei web i de quina manera s'hi pot accedir es pot desenvolupar un programa que faci servir el servei web sense problemes.

⁸La versió 2.0 és més senzilla que les versions 1.X i ofereix compatibilitat amb un nombre més gran de característiques, però no està totalment implementada per als entorns de desenvolupament.

A pesar d'això obtenir el WSDL abans de comunicar amb un servei web **pot servir per obtenir més informació sobre la manera com funciona el servei**, i per detectar si s'hi han produït canvis.

El fitxer WSDL

El fitxer WSDL és un document XML que té per arrel <definitions> i que fa servir XSD per definir els tipus de dades implicades en el servei. Dins seu conté diferents etiquetes que s'encarreguen de definir quins serveis s'ofereixen i com es pot fer per accedir-hi. Dins de l'element <definitions> hi podem tenir diferents elements segons si el document correspon a la versió 1 o a la 2.

L'element <types> es fa servir per definir amb XSD els tipus de dades que s'empren en tots els serveis oferts. Es defineixen tots els tipus de dades, tant paràmetres d'entrada com de sortida, tipus complexos...

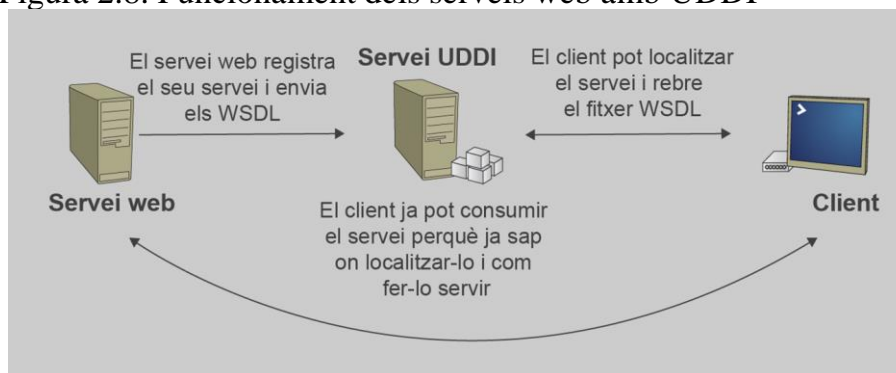
L'element <portType>, o <interface> en la versió 2, defineix les operacions que es poden fer en un servei web i els paràmetres que es fan servir per dur-les a terme.

L'element <binding> defineix el protocol i el format en què es poden fer les operacions. L'element <services>, per la seva part, defineix el lloc físic on hi ha el servei web (adreça URL) mitjançant una col·lecció de punts de connexió <binding>.

2.3.4 UDDI

El protocol **UDDI** (descripció, descoberta i integració universals o universal description, discovery and integration) serveix perquè els creadors de serveis web **puguin registrar-los en una base de dades, i alhora permetre als clients localitzar serveis web que facin allò que volen i proporcionar-los el seu WSDL** (figura 2.8).

Figura 2.8. Funcionament dels serveis web amb UDDI



Des d'un punt de vista d'automatització això és l'ideal. Es pot fer que els programes automàticament localitzin el servei que busquen sense que ni tan sols en tinguin informació prèvia, simplement accedint al registre UDDI.

Per aquest motiu, UDDI va ser creat per donar servei a un registre anomenat UDDI Business Registry o UBR, amb el propòsit de proporcionar una manera senzilla de localitzar serveis web basats en SOAP a Internet. La idea era crear una mena de "pàgines

grogues” que continguessin tots els serveis web de manera que es poguessin localitzar a partir de les funcions que oferien.

Però UBR va fracassar i va tancar el 12 de gener del 2006. Era molt difícil crear categories de serveis que tinguessin prou significat per ser útils; i també s’ha de tenir en compte que UDDI no és necessari per poder fer servir serveis web.

A pesar d’això algunes organitzacions encara fan servir UDDI internament. Però també és cert que altres organitzacions fan servir sistemes diferents per localitzar serveis web: LDAP, bases de dades...

2.3.5 Provar serveis web

Per veure com funcionen els serveis web no hi ha res que vagi tan bé com provar-ne algun. Malauradament el fracàs de crear un sistema de registre ha fet que no sigui fàcil localitzar serveis per fer proves, i moltes de les funcions dels serveis web es fan servir en xarxes locals. Per exemple, molts dels programes de gestió d’empreses ERP permeten interacció mitjançant SOAP.

També hi ha algunes pàgines que contenen enllaços a serveis web públics accessibles per Internet. S’ha de tenir en compte que aquests enllaços no sempre estan totalment actualitzats i que per tant no sempre funcionen.

- Seekda (<http://webservices.seekda.com/browse>)
- xMethods (<http://xmethods.net>)
- WSIndex (http://www.wsindex.org/Web_Services/)

En aquestes pàgines web **s’hi poden trobar serveis web molt diversos**, com obtenir la temperatura de diferents ciutats del món, sistemes de conversió de monedes, informació sobre països, valors de la borsa, traduccions de textos o comprovacions de correu electrònic.

Com que per poder fer servir un servei web el primordial és obtenir el seu WSDL per saber com interactuar-hi, les pàgines solen mostrar un enllaç al WSDL de cadascun dels serveis que tenen registrat.

Eines de prova

Per provar els big Web services sense haver de programar ni una línia tenim diverses alternatives, atès que SOAP i WSDL no deixen de ser llenguatges XML, i molts dels editors XML ofereixen alguna manera de depurar el funcionament dels serveis web.

Els editors XML normalment es limiten a permetre enviar peticions editables i recuperar les respostes que arriben des del servei web en SOAP. A part dels editors XML també hi ha una sèrie de programes que permeten fer tasques molt més elaborades, com ara fer proves de rendiment o de seguretat, comprovar si el servei compleix les regles d’interoperativitat de WS-I, o fer proves automàtiques de rendiment. Alguns d’aquests programes són:

- SoapUI (<http://www.soapui.org/>)
- SOAPSonar (<http://www.crosschecknet.com/>)
- Parasoft SOAtest (<http://www.parasoft.com/>)

Els avantatges que aporten aquests programes són prou importants per tenir-los en compte a l'hora de dissenyar serveis web SOAP professionals.

2.4 Arquitectures basades en REST

Una de les crítiques que es fan als serveis web basats en SOAP és que s'ha de tractar amb massa protocols per aconseguir qualsevol cosa i això fa que siguin massa complicats.

Els serveis basats en REST (representational state transfer) intenten **capturar les característiques que han fet que la web tingués èxit**, i en especial la seva senzillesa: resulta **fàcil d'entendre i de fer servir**. Es parteix de la idea que la web ha revolucionat la manera d'accedir a les dades perquè és simple i fa servir un conjunt petit de tecnologies. Però a més ha passat a ser un sistema distribuït, obert i multiplataforma, i en moltes pàgines web es fan coses tan complexes com en els serveis web sense fer-ho tan complicat com en els sistemes SOAP.

Mentre **els big Web services exposen funcions semblants a les dels llenguatges de programació mitjançant interfícies** més o menys complexes que són **diferents per a cada servei**, els serveis web basats en **REST exposen les dades mitjançant una interfície que sempre és la mateixa**. Aquests serveis intenten aconseguir el mateix èxit que ha tingut la web fent servir les mateixes armes: no es defineixen nous protocols sinó que s'aprofiten els que ja existeixen i tenen èxit. Per què ens hem de complicar la vida? Per què hem de crear nous protocols complexos?

REST no és un protocol sinó un **conjunt de regles** i principis que permeten **desenvolupar serveis web fent servir HTTP**, i basant-se en el fet que la web és plena de recursos que es poden manipular mitjançant les seves representacions.

REST **obliga a fer que els missatges continguin tot el necessari per poder funcionar**, ja que es parteix del fet que no tenen estat, de manera que no poden recordar les peticions anteriors. Això implica que el sistema ha de **tractar cadascuna de les peticions com si fos l'única**, i per tant ha de tenir tot el necessari perquè el servidor pugui fer la tasca. El servidor mai no tindrà en compte les peticions anteriors per tractar l'actual.

Segurament el fet que **REST faci servir tecnologies que ja existien** -la qual cosa permet accedir als recursos d'una manera homogènia i simplifica l'ús, ja que no cal interpretar què fan determinades funcions ni què s'ha de posar en els paràmetres-, ha estat clau en el fet que els serveis basats en REST s'hagin convertit en els preferits per desenvolupar serveis web.

Molts dels antics serveis basats en SOAP, com per exemple les recerques de Google, s'han abandonat en favor d'altres sistemes que sovint estan basats en REST.

2.4.1 Els recursos

L'accés als recursos és el component bàsic en la web, i per tant també ho és de REST. Per a REST, qualsevol cosa que es pugui identificar amb un URI o un URL dins d'Internet **es considera un recurs**, i com que pràcticament tot el que hi ha dins d'Internet es pot identificar mitjançant un URL tenim molts recursos disponibles.

Per a REST tot és un recurs que es pot identificar i que es pot manipular.

Un recurs és un concepte o una idea que es pot associar a un URL o un URI, però a més per a REST els recursos poden tenir diferents representacions segons l'estat en què es trobin.

En sistemes REST les relacions reals s'estableixen mitjançant els enllaços que tinguin les pàgines.

2.4.2 Representacions

Per a REST no hi ha cap problema a fer servir la web per comunicar programes, ja que en realitat la comunicació d'una pàgina web ja és una comunicació entre programes (el servidor i el navegador). L'única cosa que no està pensada per ser usada per programes és el contingut.

El més habitual és que els recursos de la web estiguin pensats per ser consumits per humans mitjançant navegadors, i per aquest motiu la informació s'hi sol representar en HTML. Si l'objectiu és permetre a les màquines fer servir la web només cal que el contingut de les pàgines estigui en un format més amigable per als programes, com per exemple XML. Malgrat fer servir la web, REST no està limitat a fer servir missatges en HTML o XML sinó que pot emprar recursos en qualsevol format. Per tant, en un servei web basat en REST es poden generar des de documents en HTML o PDF fins a imatges en JPG o GIF.

La idea és que no es visualitza el recurs en si sinó només una representació d'aquest recurs. Hi ha recursos que només tindran una sola representació i d'altres que en tindran diverses i per aquest motiu en REST els programes poden demanar quina és la representació que volen obtenir o els servidors poden enviar la que considerin més adient segons un determinat criteri (per exemple, segons el programa que ha demanat el recurs).

Gràcies a les representacions es pot aconseguir que s'uneixi de manera transparent la web dels humans amb la dels programes.

Per exemple, és habitual que si el servidor detecta que s'hi està accedint amb un navegador retorni una representació en HTML, mentre que si és mitjançant un altre programa retorni la representació en XML.

2.4.3 HTTP

REST fa servir HTTP com a sistema de transport (tot i que també podria funcionar amb altres protocols), però també el fa servir com a sistema de missatges. Per aquest motiu s'ha de definir quin és el paper que han de tenir els diferents mètodes HTTP a l'hora de gestionar un recurs.

S'han definit una sèrie d'equivalències entre els mètodes HTTP i la manera de tractar el recurs especificat:

- **POST**. Crear un recurs.
- **GET**. Obtenir el recurs.
- **PUT**. Actualitzar el recurs.
- **DELETE**. Eliminar el recurs.

D'aquesta manera es poden fer les operacions bàsiques simplement enviant una petició HTTP a un URI, triant el mètode HTTP adequat.

En HTTP els mètodes POST i PUT s'han definit per poder enviar informació al servidor i per tant, a més de l'ordre, normalment s'hi envia quina informació conté. Això els fa candidats ideals per inserir o modificar recursos.

2.5 Desenvolupament de serveis web

Els avantatges que poden aportar els serveis web (web services) no han passat desapercebuts i ràpidament els llenguatges de programació han anat incorporant components per desenvolupar-ne. Per exemple, el JDK (Java development kit) de Java incorpora una API per crear serveis web que s'anomena Java Api for XML Web Services (JAX-WS), però ja anteriorment se'n podien crear fent servir Java API for XML-based RPC(JAX-RPC).

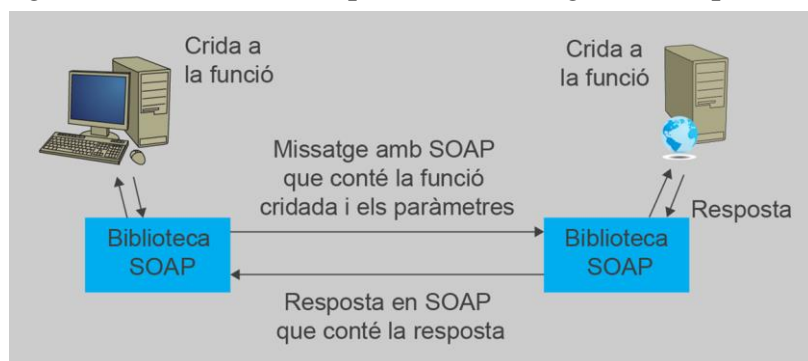
La majoria de les biblioteques o entorns de desenvolupament s'especialitzen en un dels dos grans grups de serveis web, però l'increment de la popularitat dels serveis REST ha fet que molts dels entorns basats en el protocol SOAP també incorporin la possibilitat de funcionar de forma compatible amb REST.

2.5.1 Desenvolupament de serveis basats en SOAP

Les biblioteques SOAP estan pensades per ser transparents amb vista als programadors, i per tant aquests programadors no solen tenir gaires problemes per fer-les servir. És així perquè les llibreries SOAP s'encarreguen de crear i eliminar els missatges SOAP, i al programa només hi arriben les dades que s'han enviat.

Un client demana un servei i internament la biblioteca crea el paquet SOAP per a les dades que ha d'enviar i l'hi envia. Quan la biblioteca rep les dades elimina el paquet SOAP i envia les dades netes al programa (figura 2.21).

Figura 2.21. Les biblioteques SOAP amaguen la implementació als programadors



Per tant, excepte en petits detalls, els programadors es poden despreocupar de l'existència de SOAP i només s'han de preocupar de quins serveis volen oferir.

Tot i que no és absolutament necessari crear un client SOAP, és molt més senzill si es té el WSDL del servei ja que es disposa d'auxiliars que s'encarreguen de preparar els esquelets del programa.

Gairebé sempre es pot obtenir el fitxer WSDL d'un servei web afegint al darrere de l'adreça del servei les lletres `?wsdl`. En alguns entorns aquest fitxer es genera automàticament en el moment en què és demanat i en d'altres és simplement un arxiu.

Els entorns de programació solen oferir auxiliars i eines que simplifiquen al màxim la creació de serveis basats en SOAP, generant automàticament el codi necessari per implementar o fer servir el web i fent que el programador només s'hagi de preocupar de definir què ha de fer el servei web.

Malgrat que els llenguatges més usats per desenvolupar serveis web són sobretot Java i els basats en la plataforma de Microsoft .NET, gairebé tots els llenguatges de programació tenen alguna biblioteca que els permet desenvolupar servidors i clients basats en SOAP. I a més s'ha de tenir en compte que el llenguatge usat és bastant irrellevant, ja que una de les característiques dels serveis web és que no importa en quin llenguatge o en quin sistema s'executin els servidors i els clients que podran operar entre si.

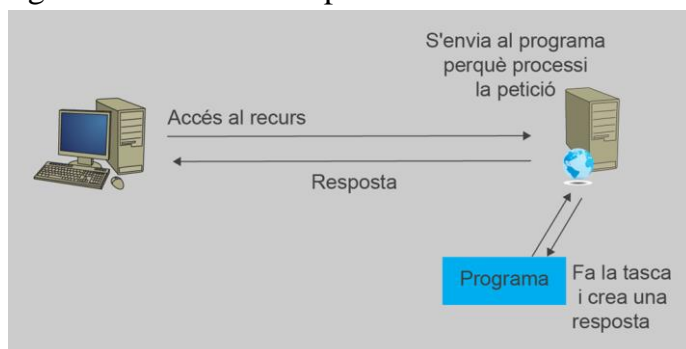
Un dels ajuts més habituals consisteix a fer servir el WSDL del servei per crear automàticament tot el codi que cal per comunicar o generar el servei web.

2.5.2 Desenvolupament de serveis basats en REST

REST està tan integrat en la web que no hi ha gaire diferència entre desenvolupar programes client que funcionin amb REST i desenvolupar programes per connectar i treballar amb servidors web.

Els resultats de fer consultes a serveis REST no es diferencien gens ni mica de les pàgines web excepte en el fet que el resultat obtingut normalment, en comptes de rebre només resultats HTML, està en un llenguatge de marques diferent com XML o JSON (figura 2.23).

Figura 2.23. Desenvolupament REST



Per tant, qualsevol biblioteca que permeti crear clients HTTP serveix per desenvolupar un client de serveis web basats en REST. Es pot fer servir qualsevol de les biblioteques estàndard per desenvolupar clients HTTP, com per exemple HTTPClient de Java, la classe WebClient de .NET o httplib de Python.

Els servidors són una mica més complexos però no gaire més, ja que només cal veure quina adreça es vol visitar per localitzar el recurs i quina instrucció ha fet servir el client per saber què hi vol fer. Aquesta tasca és molt més simple que el treball amb SOAP/WSDL i la majoria dels llenguatges de programació ja disposen d'alguna biblioteca per facilitar el

desenvolupament d'aplicacions REST: JAX-RS, Restlet, Apache CXF, RESTeasy, RESTSharp, openRASTA...

2.5.3 Entorns de desenvolupament

Han aparegut tota una sèrie d'entorns de treball que faciliten encara més la creació i desplegament dels serveis web. Amb uns quants clics es pot crear i desplegar un servei web senzill en pocs segons.

Com passa amb moltes de les tecnologies XML, la majoria d'aquests entorns estan basats **sobretot en dues tecnologies de programació: Java i .NET**. Vegem una llista d'aquests entorns:

<u>Nom</u>	<u>Llenguatge</u>	<u>Serveis</u>
Apache Axis2	Java / C	SOAP, REST
Apache CXF	Java	SOAP, REST
Spring Web Services	Java	SOAP
Metro	Java	SOAP
JAX-WS	Java	SOAP
JAX-RS	Java	REST
.NET Framework	.NET	SOAP, REST
Windows Communication Foundation (WCF)	.NET	SOAP, REST

3. Informàtica en núvol

En l'àmbit de la informàtica s'està posant de moda en els darrers anys el que s'anomena informàtica en núvol (cloud computing). A pesar del que pot semblar no es tracta de cap innovació tecnològica, sinó més aviat de l'evolució de diferents tecnologies.

La informàtica en núvol fa referència a un conjunt de tecnologies que permeten emmagatzemar tots els recursos a Internet. Allà, els proveïdors dels serveis s'organitzen per aprofitar al màxim les seves infraestructures i proveir servei a múltiples usuaris alhora; d'aquesta manera els clients poden aconseguir serveis que s'adaptin perfectament a les seves necessitats sense haver de fer despeses en infraestructura informàtica.

La paraula estrella d'aquestes tecnologies és núvol. Es fa servir aquesta idea per referir-se a totes aquestes tecnologies, ja que el símbol del núvol sempre s'ha fet servir en els diagrames de xarxa per indicar xarxes o simbolitzar les connexions a Internet.

Una de les tasques que la informàtica en núvol simplifica és la distribució de càrregues de treball. Els programes que s'ofereixen per Internet cada vegada són més complexos i requereixen fer tasques que sovint exigeixen distribuir la càrrega de treball entre diferents màquines. Això no és especialment nou ja que fa anys que en els sistemes oberts d'Unix ja existia la idea de processament distribuït. La idea era combinar diferents sistemes a partir de les xarxes locals o esteses per repartir una tasca entre diversos servidors i estacions de treball, i així obtenir resultats semblants als que s'obtidrien en una màquina molt potent i que serien impossibles si es fes servir un sol ordinador.

Tampoc la idea de llogar ordinadors no és especialment nova ja que algunes grans empreses com IBM o HP fa anys oferien serveis de processament i emmagatzematge en bases de

dades per a grans empreses en el que anomenaven computing on demand. Llogaven el temps dels seus ordinadors centrals (mainframes) per fer tasques complexes que difícilment s'haurien pogut fer amb ordinadors personals a empreses que no podien pagar la compra d'una d'aquestes computadores.

La gran innovació de la informàtica en núvol és que parteix de l'evolució d'aquestes tecnologies i d'altres com la virtualització, l'increment de les velocitats d'accés a Internet, o la proliferació de dispositius mòbils, que han desembocat en la seva creació. Amb la informàtica en núvol les organitzacions poden construir la seva infraestructura informàtica d'una manera totalment diferent de com ho feien fins ara. Ja no cal que les empreses tinguin grans instal·lacions informàtiques, ni experts en administració de sistemes, ja que això ho poden obtenir del núvol.

La idea es va portar a la pràctica per primer cop a finals dels anys noranta quan des de la llibreria en línia Amazon es van adonar que tenien una infraestructura informàtica molt potent per poder donar servei als seus clients, però que en realitat l'ús que se'n feia era inferior al 20%. La realitat és que la majoria dels sistemes informàtics no solen anar a ple rendiment.

Intentant aprofitar d'alguna manera aquesta capacitat “sobrant” es va crear el primer servei d'informàtica en núvol, que es va anomenar AmazonWeb Services.

La idea era mirar de treure un rendiment al 80% dels recursos que no s'estaven usant en la seva infraestructura, valent-se de la virtualització. Per exemple, el servei Amazon EC2 (Elastic Compute Cloud) permet llogar els recursos de maquinari en forma de màquina virtual, ampliables, fent servir els recursos que Amazon no estigui usant.

A més, la tecnologia de màquines virtuals és prou dinàmica per permetre que els clients del seu servei puguin anar dimensionant els recursos virtuals que volen fer servir segons les necessitats de cada moment.

3.1 Característiques

La informàtica en núvol ofereix una sèrie de característiques i possibilitats que la fan molt atractiva per a les empreses, ja que permet crear amb poca inversió econòmica infraestructures que s'adapten a les necessitats puntuals de cada organització i a les quals es pot accedir des de qualsevol lloc per Internet.

3.1.1 Ubiquïtat

Quan es parla d'informàtica en núvol es defineix un sistema d'obtenció de serveis o recursos sense necessitat de saber en quin lloc es troben. Al client, doncs, no li cal saber quina és l'arquitectura de xarxa, ni les característiques tècniques del núvol; simplement hi accedeix i en fa servir els serveis que necessita. Això fa que el conjunt pugui ser altament escalable, és a dir, que pugui créixer fàcilment.

Els errors del maquinari del proveïdor en principi no afecten el client, ja que, com que el client no necessita saber on s'estava connectant, en cas de fallada el servei pot ser reubicat en alguna altra màquina de manera totalment transparent. Això fa que la informàtica en núvol sigui bastant tolerant a fallades.

El fet que sigui accessible per Internet també aporta tota una sèrie d'avantatges afegits: es pot accedir al sistema des de qualsevol lloc sempre que es tingui accés a Internet, i com que normalment es controla des d'un navegador web s'hi pot accedir amb ordinadors però també des de qualsevol dispositiu que s'hi pugui connectar.

3.1.2 Infraestructura informàtica

El cloud computing fa que no calgui una gran infraestructura informàtica en l'organització. Des del principi de la informàtica els requeriments de maquinari dels programes no han fet sinó créixer i això ha obligat les empreses a renovar periòdicament el parc de maquinari per poder-se adaptar als nous requeriments.

Fent servir la informàtica en núvol es pot aconseguir crear una infraestructura informàtica completa sense pràcticament haver de tenir un parc informàtic important. Amb un sol ordinador es pot controlar una infraestructura informàtica completa que abans hauria requerit milers d'euros d'inversió.

Això trenca el cicle de renovació de maquinari que s'ha anat seguint tradicionalment per adaptar la infraestructura als nous requeriments del programari. Amb la informàtica en núvol ja no cal fer grans despeses en maquinari ni s'han de renovar els equips: n'hi ha prou amb contractar més recursos. Això permet que les infraestructures en el núvol siguin molt escalables: poden créixer tant com calgui per fer front a les demandes del servei. I com que els clients no necessiten màquines importants per gestionar la seva infraestructura en el núvol el més habitual és fer servir un sistema de gestió que no en consumeixi gaires. L'accés a la gestió dels recursos gairebé sempre es fa mitjançant una aplicació web, de manera que no cal instal·lar cap tipus de programari en l'ordinador del client perquè amb un navegador n'hi ha prou per poder començar a treballar i gestionar els recursos en el núvol.

3.1.3 Adaptació a les necessitats

Els sistemes d'informàtica en núvol, com que es basen en tecnologia de màquines virtuals, són molt més àgils que els serveis tradicionals. Fins a l'entrada en escena de la informàtica en núvol, quan algú volia contractar recursos per Internet definia quins eren els que estava contractant i aquests eren els que se li assignaven, a menys que es canviés el contracte.

La informàtica en núvol trenca amb aquesta manera de funcionar, ja que permet canviar les característiques de la infraestructura contractada en pocs minuts i sense haver de comunicar-ho al proveïdor.

Com que es basa en la virtualització, aconseguir que la màquina que s'hagi contractat tingui més o menys recursos de memòria, espai de disc o CPU no requereix sinó fer uns quants clics. Això aporta un nou avantatge perquè no cal perdre temps comunicant amb el proveïdor per fer canvis en el contracte existent (ja que les condicions canvien), ni cal esperar que tingui a punt el nou pla contractat.

Sovint en molts negocis la demanda sol ser variable fins i tot en un mateix dia, en què segons l'hora hi solen haver més o menys peticions. Amb la informàtica en núvol es pot canviar la capacitat de la infraestructura informàtica de l'empresa perquè reflecteixi aquesta variabilitat i per tant adaptar la infraestructura informàtica a les necessitats de cada moment.

Abans de l'aparició de la informàtica en núvol, intentar fer front als pics de demanda regulars requeria fer inversions en maquinari el funcionament del qual era clar que no s'optimitzaria. Amb el cloud computing s'aconsegueix la capacitat d'ajustar al màxim els costos en infraestructura.

3.1.4 Reducció de costos

Una de les característiques que defineix més bé la manera com s'ha implementat la informàtica en núvol és en la manera com es paga. Com que els proveïdors el que fan és revendre capacitat que altrament no farien servir, han optat per fer que els clients només paguin per l'ús que facin dels serveis. Això ha fet que un bon nombre d'autors, com Nicholas Carr, afirmen que aquestes tecnologies situen la informàtica en un punt molt proper al d'altres serveis com l'electricitat o l'aigua.

Però hi ha una diferència molt gran entre els serveis que ofereix la informàtica en núvol i els altres serveis: la immediatesa. Per fer canvis en el contracte de potència elèctrica cal anar a la companyia proveïdora a modificar el contracte i un cop s'han fet les comprovacions adients es proveeix la nova capacitat. En canvi, en la informàtica en núvol aquest canvi és gestionat directament pel client i s'aconsegueix en pocs minuts.

També s'ha de tenir en compte un dels costos associats a l'ús de sistemes informàtics, que és el consum d'energia elèctrica. Els centres de procés de dades són grans consumidors d'energia elèctrica i els departaments d'informàtica de moltes empreses veuen que bona part del seu pressupost s'ha de destinar a l'energia elèctrica i a mantenir els equips en bones condicions de refrigeració. Per tant, des del punt de vista econòmic, la informàtica en núvol ofereix un estalvi afegit en forma d'un consum d'energia elèctrica inferior.

3.2 Classificació

A pesar que les classificacions solen tenir zones difuses, una de les maneres més senzilles per entendre quines són les diferents possibilitats que ofereix la informàtica en núvol és mitjançant classificacions. El més habitual en aquest àmbit és fer la classificació de dues maneres:

1. Segons qui té la propietat del núvol.
2. Segons el nivell de serveis que s'ofereixen.

3.2.1 Segons la propietat

La primera classificació dels sistemes d'informàtica en núvol es basa en qui és el propietari dels recursos o serveis que s'ofereixen. Si bé la idea general es basa a fer servir recursos que "són" dins del núvol i que per tant són públics, han anat apareixent altres alternatives per donar solucions a totes aquelles persones i organitzacions que no s'acaben de refiar de perdre el control total de les seves infraestructures.

Resistència als serveis en núvol

La sensació de "pèrdua de poder" és un dels factors que ocasiona més resistència a la implantació dels serveis en núvol.

Amb el sistema tradicional una organització podia tenir la certesa de que les dades estaven dins de l'organització, i per tant podia invertir recursos en la seva protecció i control d'accés.

En canvi, al posar-les en el núvol, aquestes dades ja no estan físicament en l'empresa, i

qui s'encarregarà de proporcionar-ne la seguretat serà el proveïdor en núvol. Això fa que tot i els contractes que hi pugui haver hi hagi la sensació de que es perd una part del poder sobre les dades, ja que qui les té realment és el proveïdor.

En certa manera és un problema similar al problema amb què es va trobar Microsoft a l'hora d'intentar que els usuaris de Windows XP no fessin servir el compte d'usuari de l'administrador. La majoria dels usuaris no en va fer cas perquè implicava perdre un suposat

“poder” en la seva màquina. Per aquest motiu les noves versions de Windows han abordat el problema des d'una altra perspectiva: el sistema només executa els programes amb privilegis de l'administrador quan li és absolutament necessari.

Basant-nos en qui té la propietat de les infraestructures del núvol podem definir tres grans grups de núvols:

- Privats.
- Públics.
- Híbrids.

Públics

Quan algú pensa en la informàtica en núvol el primer que li ve al cap són els núvols públics. Es consideren núvols públics tots aquells núvols que són administrats per un proveïdor i no pel client que en fa ús.

Els núvols públics ofereixen diversos avantatges, entre els quals destaquem que no requereixen una gran inversió en infraestructura per part del client ja que el client simplement defineix quina és la infraestructura que li fa falta i al cap de poca estona ja la pot començar a fer servir. Això fa que qualsevol usuari, sense haver de proveir cap mena d'espai físic per instal·lar el maquinari i sense haver comprat els equips informàtics ni haver-hi d'aplicar cap mena de manteniment, pot aconseguir una infraestructura informàtica que abans poques empreses es podien permetre.

A més, com que la infraestructura és en el CPD del proveïdor també és aquest proveïdor el qui s'ha d'encarregar del manteniment i de les actualitzacions de maquinari, de contractar personal tècnic...

El més normal és que els núvols públics es comparteixin entre diferents clients dins de la infraestructura del proveïdor, de manera que cadascun dels clients obtingui només els recursos que necessita en cada moment.

L'exemple clàssic són els serveis que ofereix l'empresa Amazon amb el que anomena Amazon Web Services (AWS). Aquest producte ofereix múltiples serveis que es poden contractar i que van des del simple espai d'emmagatzematge fins a màquines virtuals.

Comunitaris

La idea dels núvols comunitaris rau a compartir la infraestructura no amb tot el món sinó només amb un grup d'organitzacions d'una comunitat específica amb uns objectius comuns. En general el que es fa és compartir una infraestructura d'aplicacions que, en cas que fos gestionada per cada organització, seria massa cara, ineficient o complexa.

L'administració del núvol pot ser tan privada com pública però els costos i els beneficis es reparteixen entre els membres de la comunitat. Normalment se sol cobrar per l'ús que en faci cadascuna de les organitzacions.

La cooperació entre competidors és més habitual del que pot semblar en les empreses i per això la informàtica en núvol els ofereix tota una sèrie de possibilitats de creació d'infraestructures que els serien molt cares d'implementar separatament.

Un exemple d'aquest tipus de núvols és el format per les empreses Ford, Chrysler i General Motors, anomenat USCAR, que està pensat per a la recerca de materials per fabricar vehicles.

Privats

Moltes empreses encara recelen de posar les seves dades i infraestructura crítica al núvol sota control d'un tercer. Per aquest motiu alguns dels problemes que tenen els núvols públics s'han intentat solucionar fent servir la mateixa tecnologia per crear núvols privats.

Amb aquests núvols s'aconsegueix aprofitar millor els recursos de què es disposa en la xarxa local i s'aconsegueixen part dels avantatges de la informàtica en núvol, sense perdre mai el control de l'aplicació ni les seves dades, sense dependre de tercers i mantenint el control total sobre els recursos.

El terme privat implica que només el client fa servir el núvol i que no es comparteix aquesta infraestructura amb cap empresa externa.

Això comporta que els sistemes han de ser administrats pel client com en un sistema tradicional i que per tant s'ha de renovar el maquinari i fa falta invertir en infraestructures i manteniment.

En general els preus baixos dels núvols públics són deguts al fet que el que fan és vendre les capacitats de computació, memòria i disc que els sobren de les seves pròpies infraestructures. En general les màquines del sistema no treballen intensament sinó que, tot i que en moments puntuals exprimeixen els recursos computacionals, normalment no ho fan.

Per tant, amb un núvol privat s'aconsegueix fer que els sistemes facin un ús més eficient dels recursos de què es disposa. Crear un núvol privat en la infraestructura de l'empresa fa que es puguin redirigir algunes de les tasques dels ordinadors més ocupats a d'altres que no treballen intensament per optimitzar el rendiment i l'ús del sistema.

Amés, la creació d'un núvol privat deixa la porta oberta a la possibilitat que, en cas que calgui, es puguin reduir costos fent el mateix que fan els proveïdors públics: revendre els recursos sobrants a altres empreses.

Els núvols privats permeten optimitzar els recursos propis però suposen la pèrdua de dos grans avantatges d'aquesta tecnologia: la baixa inversió i la possibilitat d'adaptació del sistema informàtic a les necessitats del moment. En un sistema privat els recursos són els que són, i si en un moment puntual es requereixen més recursos dels que es disposa, no es té la capacitat d'adaptar-se ràpidament als nous requeriments; de la mateixa manera, si les

previsions han estat massa optimistes i la infraestructura és massa gran, tampoc no s'hi pot fer res perquè la inversió ja està feta.

La llista de programari que permet crear núvols privats no para de créixer, tant en l'àmbit del codi propietari com en el del codi obert: VMware vSphere, Citrix OpenCloud, OpenNebula, OpenStack, Eucalyptus, Nimbus... Sovint aquest programari també es pot fer servir per crear núvols públics o bé interactuar-hi d'alguna manera.

Híbrids

Els núvols híbrids van sorgir per intentar obtenir els avantatges dels núvols privats i dels públics. Es basen a disposar una part de la infraestructura en núvols públics i una altra en privats, de manera que es tingui una part de la informàtica fora de l'empresa i una altra a dins.

Per aprofitar els avantatges cal que els núvols privats i públics es puguin "entendre", i per això és essencial que disposin d'alguna forma d'interacció entre si.

L'objectiu és que en el moment en què la infraestructura privada no pugui fer front a la càrrega de treball es puguin contractar automàticament recursos en el núvol públic.

Per exemple, la implementació de codi obert Eucalyptus permet implementar un núvol privat que és compatible amb Amazon EC2, de manera que el núvol privat es pot comunicar amb un de públic que sigui en la infraestructura d'Amazon.

Els núvols híbrids fins ara s'han mostrat ideals per a diverses tasques com les còpies de seguretat remotes o la implementació de plans de contingència, però tothom espera que la seva importància creixerà amb el pas dels anys.

3.2.2 Segons els nivells de servei ofert

És important tenir present que no tots els serveis en el núvol ofereixen les mateixes capacitats. En aquest àmbit s'ofereix des de maquinari (normalment en forma de màquines virtuals) fins a serveis de programari més o menys complexos.

Per tant, és important definir algun tipus de classificació dels serveis oferts en el núvol. La classificació més acceptada sol dividir els serveis en tres grans grups:

- SaaS (programari com a servei o software as a service).
- PaaS (plataforma com a servei o platform as a service).
- IaaS (infraestructura com a servei o infrastructure as a service).

La diferència entre els serveis està en el nivell d'abstracció respecte al maquinari i sobretot en la dependència del programari que ofereixen. En IaaS disposem d'una màquina que podem personalitzar i en la qual instal·lem el nostre propi programari mentre que en SaaS tenim un programa ja desenvolupat per fer-lo servir.

Aquesta classificació deixa clar que abans que algú triï un sistema o un altre ha de determinar quin és el sistema que s'ajusta més als seus objectius.

IaaS

Els IaaS són el nivell més baix d'informàtica en núvol. El que fan els proveïdors és permetre que una part de la seva potència de càlcul i del seu emmagatzematge sigui oferta als clients en forma d'instàncies de màquines virtuals.

Els sistemes classificats com a IaaS el que ofereixen és la possibilitat de crear la infraestructura informàtica de l'organització en el núvol.

Per a l'usuari del servei, a part que no veu físicament el servidor, no hi ha cap més diferència entre tenir un servidor en les seves instal·lacions o accedir-hi mitjançant el núvol. El control total de la màquina implica que és el client qui s'encarrega de definir quin sistema operatiu i programes hi instal·la, i qui s'encarrega de controlar i gestionar el sistema.

Per tant, això porta associades una sèrie d'obligacions ja que el proveïdor cedeix completament el control de la màquina virtual i és l'usuari qui es fa responsable de les operacions. Si la màquina arriba al límit de capacitat és l'usuari el responsable de modificar-la per adaptar-se a les noves necessitats.

La diferència entre obtenir un servidor de núvol o instal·lar-lo en la nostra infraestructura està en els costos. S'aconsegueix tenir un sistema que es pot controlar de la mateixa manera que si fos físicament a l'empresa però:

- No cal fer cap inversió en maquinari ni pensar quan s'ha de renovar o expandir.
- No s'ha d'invertir en les infraestructures de xarxa necessàries per connectarlo. I difícilment una empresa es podrà permetre tenir una infraestructura de xarxa de la qualitat de les d'Amazon o Google.
- No cal tenir grans coneixements d'administració de xarxes ja que és el proveïdor qui gestionarà aquest àmbit.

Fins a l'aparició de la informàtica en núvol, quan algú volia externalitzar algun dels seus serveis els contractava a un proveïdor d'allotjament (hosting), el qual normalment oferia entorns preinstal·lats amb limitacions pel que fa a sistemes operatius, espai de disc i taxa de transferència de dades. El client es veia obligat a adaptar-se al "paquet" que oferia el proveïdor que més bé s'adaptés a les seves necessitats del moment. Aquest sistema tenia diverses limitacions:

- Canviar les característiques del sistema normalment requeria un procés de negociació amb el proveïdor i era molt complicat aconseguir canviar el servei per a moments puntuals en què la demanda pogués créixer.
- El control del sistema era relativament reduït ja que la gestió estava en mans del proveïdor i no se'n solia cedir el control.
- Com que l'entorn ofert era fix els preus també solien ser fixos.

Amb IaaS s'aconsegueix que qualsevol pugui crear la seva pròpia infraestructura de maquinari:

- Escollir quines són les característiques concretes que ha de tenir el sistema i modificar-les tant a l'alça com a la baixa en pocs minuts.
- Instal·lar el sistema operatiu i el programari que es vulgui en cada moment.
- Gestionar totalment el sistema operatiu incloent-hi els mecanismes de seguretat.
- Triar entre una oferta de preus en relació amb l'ús i amb la capacitat contractada, de manera que s'obtenen preus variables.

El proveïdor de IaaS que té més quota de mercat avui en dia és sense cap mena de dubte Amazon amb el seu producte EC2 (Elastic Compute Cloud), però d'ençà la popularització de la informàtica en núvol n'han aparegut molts d'altres, com per exemple Rackspace o GoGrid.

PaaS

Els serveis del núvol que ofereixen PaaS el que fan és oferir un entorn de desenvolupament en què un usuari pot desenvolupar les seves pròpies aplicacions, i un entorn de desplegament per poder desplegar les aplicacions desenvolupades.

En l'entorn s'hi inclou tot el que cal a un programador per crear prototipus, i també desenvolupar, provar, documentar, analitzar i posar en funcionament les aplicacions.

Normalment aquests sistemes ofereixen algun tipus d'API (interfície de programació d'aplicacions o application programming interface) que simplifica notablement el desenvolupament de les aplicacions en la infraestructura a pesar que limiten els llenguatges de programació que s'hi poden fer servir. Alhora aquest avantatge a vegades és vist com un inconvenient perquè fa que canviar de proveïdor sigui molt més complicat, ja que l'API sol ser específica de cada proveïdor. Això fa per exemple que sigui complicat canviar un programa emprat amb l'API de Google per adaptar-lo a la infraestructura de Microsoft Azure.

En general en tots hi sol haver:

- Un servei de base de dades.
- Un espai d'emmagatzematge.
- Algun sistema de desenvolupament col·laboratiu per permetre a diferents desenvolupadors treballar alhora.

En general el PaaS ofereix abstracció dels detalls tècnics de l'entorn en què s'executa l'aplicació, i permet així que el desenvolupador només s'hagi de concentrar en els detalls concrets de l'aplicació que vol desenvolupar.

El contracte sol incloure la garantia que el sistema és escalable (no caldrà canviar l'aplicació quan creixi) i també la disponibilitat del servei. També sol estipular que el proveïdor s'ha d'encarregar de fer còpies de seguretat perquè l'usuari no se n'hagi de preocupar.

A més, no cal que el client faci cap despesa en la compra de nou maquinari ni programari, i tampoc no tindrà despeses en llicències de programari, i podrà treballar des de qualsevol lloc i en qualsevol dispositiu que tingui accés a Internet i un navegador web.

Entre els proveïdors de PaaS més importants hi ha Google, amb el Google App Engine; Microsoft, amb Windows Azure; o Salesforce, amb Force.com.

SaaS

El SaaS (programari com a servei) és el servei més conegut de la informàtica en núvol perquè és el que fan servir més els usuaris finals. Per fer servir SaaS no cal tenir cap mena de coneixements d'administració ni cal ser desenvolupador; només cal saber fer servir el programa.

El SaaS fa referència a aplicacions instal·lades pel proveïdor en el seu sistema i que estan preparades perquè les usin els clients en qualsevol moment.

La idea és que en comptes de comprar una còpia d'un determinat programari -per exemple, un editor de textos com Microsoft Word- el que es fa és pagar un lloguer -a un preu molt més baix- per tenir dret a usar-lo durant un determinat període. No cal instal·lar cap tipus de

programa en el sistema del client sinó que s'hi accedeix mitjançant un navegador, i per tant es pot editar des de qualsevol lloc o des de qualsevol dispositiu que tingui accés a Internet.

El que caracteritza més aquest servei és que el client no ha d'instal·lar cap tipus de programari en el seu sistema, sinó que simplement amb un navegador i una connexió a Internet pot aconseguir treballar amb el programari com si el tingués instal·lat en la seva màquina.

El SaaS permet que el client es despreocupi de tots els aspectes relacionats amb la tecnologia: no li cal instal·lar el programa, ni comprovar si han aparegut actualitzacions de programari o si el programa funciona correctament. De les actualitzacions, el suport i la disponibilitat del programari se n'encarrega el proveïdor.

Com que el pagament s'estableix segons l'ús que se'n fa no requereix els costos inicials associats al programa. Les despeses inicials solen ser molt baixes en comparació de les que es requereixen en cas de voler muntar la infraestructura en la instal·lació pròpia, ja que no cal que es facin despeses en maquinari, ni en llicències de programari, ni cal ni tenir personal informàtic, perquè qualsevol amb una formació mínima pot fer servir les aplicacions.

Aquest darrer aspecte és especialment important en entorns empresarials ja que els programes per a empreses sempre han tingut la tendència a ser complexos, cars i normalment no solen ser senzills d'instal·lar. No és estrany que calgui un equip d'experts per instal·lar, configurar i provar el programari abans de poder-lo posar en producció, i sempre cal fer un seguiment del producte per mantenir-lo actualitzat, solucionar-ne els possibles errors...

En els darrers anys hi hagut programes de tota mena que han començat a oferir els seus serveis en el núvol i ja s'hi poden trobar aplicacions de tot tipus: sistemes de còpies de seguretat, programes d'ofimàtica, ERP...

Amb una visió més general, pràcticament totes les aplicacions web que ofereixin un servei es poden considerar SaaS. Els serveis de correu electrònic en el Web com Gmail o Hotmail, o els paquets d'ofimàtica en línia com Google Docs o Microsoft Office 360 es poden considerar SaaS.

3.3 Resistències a l'adopció de la informàtica en núvol

Els sistemes en el núvol ofereixen tota una sèrie d'avantatges en la gestió eficaç dels recursos, en l'estalvi en inversió en equipament informàtic i en les possibilitats d'escalabilitat de la infraestructura, que superen notablement el sistema tradicional de creació d'infraestructures en local. Aleshores, si els sistemes en el núvol ofereixen tants avantatges, per què no els estan adoptant massivament les empreses?

A part de la resistència dels departaments d'informàtica, que solen veure com un competidor els serveis en el núvol, els motius més importants solen respondre a tres grans aspectes:

- Seguretat i disponibilitat de les dades.
- Captivitat en un proveïdor.
- Aspectes legals de l'emmagatzematge de dades.

3.3.1 Privacitat i seguretat de les dades

Les empreses han destinat grans quantitats de diners a la compra, instal·lació i renovació dels sistemes informàtics, que s'han convertit en un aspecte fonamental de les organitzacions, i les dades que s'emmagatzemen en aquests sistemes són l'actiu principal de les companyies.

Per aquest motiu les organitzacions solen ser reticents a desar dades que poden ser vitals per a la seva supervivència en un sistema que pot ser a l'altra banda del món.

Mentre la informació resideix dins de l'empresa ningú es no preocupa, perquè se sap on és i sempre és disponible; però en el moment en què les dades crítiques s'emmagatzemen fora de l'empresa comencen les pors.

Entre les pors més importants hi ha les següents:

- Es pot accedir a les dades sempre que es vulgui?
- Pot ser que algun dia el proveïdor tanqui? En aquest cas, com es poden recuperar les dades?
- Com es pot garantir que les dades no són accessibles per a un tercer?

Es pot accedir a les dades sempre que es vulgui?

Els sistemes en el núvol posen la informació de les organitzacions que els contracten en mans dels diferents proveïdors que hi estan implicats. Això representa que es depèn no solament del proveïdor dels serveis en el núvol sinó també del proveïdor de connexions a Internet.

Si una empresa té tot el programari en el núvol, una fallada en el sistema de connexió a Internet pot provocar que aquesta empresa quedi paralitzada. Tot i que les comunicacions han millorat, la majoria de les companyies de comunicacions no poden garantir un determinat grau de qualitat en el servei i això pot ser un problema per a determinades empreses o en determinats llocs.

L'altre possible punt de fallada està en els proveïdors de serveis en el núvol.

Malgrat que solen garantir disponibilitats del seu sistema superiors al 90%, com que no tenen el programari i les dades en una infraestructura local, tot el sistema informàtic de l'empresa pot quedar inoperant si sorgeix un determinat problema fins que el proveïdor no soluciona aquest problema.

A més, si es dona la circumstància que es produeix un desastre, no es podrà fer servir cap dels sistemes tradicionals de recuperació de dades ja que no hi ha cap lloc per a aplicar-los, i es depèn de les capacitats tècniques del proveïdor, i també de les còpies de seguretat que aquest proveïdor hagi fet.

Pot ser que algun dia el proveïdor tanqui? En aquest cas, com es poden recuperar les dades?

Es tracta d'un problema que les empreses ja afronten habitualment. Què passaria, per exemple, si tanca una entitat bancària en què l'empresa diposita els seus diners?

És gairebé tan difícil que tanquin alguns dels grans proveïdors de serveis en el núvol -com Google, Amazon o Microsoft- com que tanqui el banc amb què s'està treballant.

Com en qualsevol altre servei, contractar en el núvol requereix que s'investigui prèviament si el proveïdor té prou estabilitat i capacitat per garantir la continuïtat futura del negoci.

Com es pot garantir que les dades no són accessibles per a un tercer?

Per a molts experts en seguretat la informàtica en núvol té una vulnerabilitat inherent que es deriva del fet que, com que els servidors són accessibles des de qualsevol lloc, també poden ser atacats des de qualsevol lloc.

A més, alguns dels problemes de seguretat habituals en els sistemes quan aquests sistemes són en local esdevenen més importants quan el programa passa al núvol.

Que un usuari faci servir una contrasenya dèbil com 123456 és un problema de seguretat en un sistema local, però es converteix en una porta oberta al públic si el sistema és obert a la Xarxa.

3.3.2 Captivitat en el proveïdor

Un dels problemes que té actualment la informàtica en núvol és que no sempre tots els núvols són compatibles entre si. En aquest moment no existeix un estàndard d'interoperativitat entre núvols i per tant cada proveïdor fa les coses a la seva manera.

La falta d'un estàndard d'interoperativitat que segueixin tots els proveïdors fa que si tenim un núvol en un determinat proveïdor i volem canviar-nos a un altre no sempre sigui senzill, ja que aquest pas requereix:

- Configurar la xarxa i els mecanismes de seguretat en el nou proveïdor, adaptant-se a les seves eines.
- Tornar a crear l'aplicació en el nou proveïdor.
- Posar en funcionament l'aplicació en el nou proveïdor i comprovar que funciona igual que en l'anterior.
- Moure les dades del proveïdor original al nou amb els mecanismes de seguretat adequats.

Les complicacions del procés solen provocar que sovint, tot i que es consideri que seria positiu, el canvi no es dugui a terme i per tant l'empresa quedi a mercè d'un proveïdor concret.

És probable que en un futur proper la interoperativitat entre núvols acabi existint; ja hi ha veus que reclamen la creació d'estàndards en el núvol que garanteixin la interoperativitat.

3.3.3 Aspectes Legals

Un aspecte que s'ha de tenir en compte són els problemes de compliment de la legislació existent en cada país. Internet és una xarxa internacional sense fronteres i per tant les dades que emmagatzemem al núvol poden anar des del país en què s'han generat fins a qualsevol altre punt del món.

La ubicació física on s'emmagatzemen les dades és un factor determinant des d'un punt de vista legal, ja que l'emmagatzematge de dades que continguin informació de caràcter personal és regulat per la Llei orgànica de protecció de dades de caràcter personal (LOPD). La LOPD estableix que no es poden fer transferències internacionals de dades a països que no tinguin una legislació internacional semblant a la que hi ha a l'Estat espanyol. I per tant no es consideren transferències internacionals de dades l'enviament a qualsevol país de la Unió Europea o un grup de països que han signat un conveni com l'Argentina o Suïssa. S'ha de tenir en compte, per tant, que cal tenir clar en quin lloc s'emmagatzemaran les dades per no tenir problemes legals.

La LOPD també té altres aspectes que s'han de tenir en compte a l'hora de posar les dades en el núvol:

- Obliga a tenir un responsable de les dades que és qui s'ha d'encarregar d'establir tota una sèrie de mecanismes de seguretat destinats a garantir la integritat i la confidencialitat de les dades que s'emmagatzemen. Com pot saber el responsable de seguretat que això s'està fent si les dades són a l'altra punta del món?
- Posar les dades a disposició del proveïdor de serveis en el núvol es pot considerar tractament de dades per un tercer i per tant s'han de complir tota una sèrie de normes i mecanismes que ha de dur a terme el proveïdor.

Tots aquests problemes legals es poden esquivar definint clàusules en els contractes amb els proveïdors en el núvol que s'adaptin a la legislació espanyola. Això requereix que les organitzacions abans de contractar un servei en el núvol es vegin obligades a repassar el contracte de serveis per veure si realment aquest contracte s'adapta a la legislació local o no.